Análise de classificadores otimizando hiperparâmetros e reduzindo a dimensionalidade aplicados na detecção de apneia do sono

Lucas de Souza Rodrigues
Dept. de Eng. Elétrica (PPGEE, UFPR)
Curitiba, PR, Brasil
lucasdesouzarodrigues@hotmail.com

Leandro dos Santos Coelho
Dept. de Eng. Elétrica (PPGEE, UFPR)
Pós-Graduação em Eng. de Produção e Sistemas (PPGEPS, PUCPR)
Curitiba, PR, Brasil
lscoelho2009@gmail.com

Viviana Cocco Mariani
Dept. de Eng. Elétrica (PPGEE, UFPR)
Pós-Graduação em Eng. Mecânica (PPGEM, PUCPR)
Curitiba, PR, Brasil
viviana.mariani@pucpr.br

***Abstract*—A apneia do sono é um distúrbio do sono caracterizado por episódios de dificuldade ou obstrução da respiração durante o sono. Cerca de 4% dos homens adultos e 2% das mulheres adultas no mundo sofrem com este tipo de distúrbio, podendo afetar populações mais jovens, totalizando cerca de 200 milhões de pessoas. Diversos trabalhos tem focado em criar formas automáticas para detecção da apneia do sono, destacando-se o uso de aprendizado de máquinas e o eletrocardiograma (ECG). Estes trabalhos visam utilizar um tipo específico de modelo e configurando seus hiperparâmetros empiricamente na maioria das vezes. Neste aspecto, este artigo visa o projeto de um ambiente computacional que possibilite o desenvolvimento, comparação, otimização de hiperparâmetros e explicabilidade de diferentes abordagens de classificação de aprendizado de máquina. A base pública da PhysioNet de ECGs com apneia do sono foi utilizada neste artigo para construção de um total de 11 modelos. Optuna foi empregado para busca de hiperparâmetros ideais e uma análise dos valores SHAP para explicabilidade. O melhor modelo encontrado obteve um resultado de 77,23% de acuracidade, 81,02% de sensibilidade e 82,65% de precisão. O ambiente computacional mostrou-se eficaz e possível de se utilizar em mais modelos.**

***Index Terms*—apneia do sono, eletrocardiograma, classificação, aprendizado de máquina, otimização de hiperparâmetro, explicabilidade**

I. Introdução

A apneia do sono é um distúrbio do sono caracterizado por episódios de dificuldade ou obstrução da respiração durante o sono, podendo acarretar microdespertares noturnos. Cerca de 4% dos homens adultos e 2% das mulheres adultas no mundo sofrem com este tipo de distúrbio, totalizando cerca de 200 milhões de pessoas. A apneia do sono é um fator de risco em doenças cardiovasculares (CVD), além de diminuir a qualidade de vida dos pacientes, causando diferentes sintomas como cansaço, menor capacidade cognitiva durante o dia, perca de foco, sonolência diurna e acidentes. [1]–[3].

O padrão ouro de diagnóstico é o exame de polissonografia. Este exame consiste em coletar diversos sinais biomédicos em um ambiente clínico específico. Além do paciente precisar dormir fora do seu local habitual, o exame precisa ser acompanhado por diferentes especialistas da área da saúde. Devido a estas dificuldades, muitos dos pacientes acometidos com apneia do sono não são diagnosticados corretamente e, portanto, tem sua vida afetada sem a busca de tratamentos adequados [3]–[6].

De forma a auxiliar no diagnóstico, tem-se buscado formas de identificar a apneia do sono em exames médicos de mais fácil coleta e menos dependente de especialistas. Neste contexto, a aplicação de técnicas de modelagem de inteligência artificial (IA) no ECG tem se mostrado uma abordagem promissora, tanto pelos resultados práticos obtidos, como pela possibilidade de automação do processo [7]–[9]. Entre os desafios no uso desta abordagem estão a seleção do modelo adequado, a otimização dos hiperparâmetros e a explicabilidade dos resultados, pois em muitos casos, os modelos construídos se comportam como caixas-pretas, sendo difícil interpretação do como o modelo chegou em um determinado resultado [7], [10], [11].

Este trabalho tem como objetivo ajudar na construção de um ambiente computacional que auxilie no desenvolvimento, otimização e comparação de modelos de IA na detecção de apneia do sono em ECG de um único canal. Dentro deste ambiente computacional, também são exploradas técnicas que auxiliem na interpretabilidade dos modelos construídos. Aplicamos este ambiente computacional na base pública da PhysioNet e na construção de um total de 11 diferentes modelos de IA [12].

As mais variadas estratégias de modelagem foram experimentadas na classificação da apneia do sono, tanto utilizando modelos de aprendizado de máquina clássico (ML, do inglês *machine learning*) e de aprendizado profundo (DL, do inglês *deep learning*). Em geral, a maioria dos trabalhos utilizam bases públicas para construção dos modelos. As principais bases públicas são: a PhysioNet (utilizada neste trabalho), a UCDDB da Universidade College de Dublin e a MIT-BIH polissonografia do laboratório do sono do hospital Beth Israel de Boston [13]. O processamento do ECG e a extração e calculo das características também variou nos trabalhos.

Bahrami e Forouzanfar7 utilizaram a base PhysioNet e

compararam um total de 33 de ML e DL. Para isto, os autores extraíram as séries R-R do ECG, calcularam características para os modelos ML e para os modelos DL treinaram diretamente com as séries R-R. O melhor modelo de DL foi o híbrido de DL ZFNet-BiLSTM a melhor acurácia com 88,13% e o melhor modelo de ML foi híbrido de voto majoritário com 79,39%. Yang et al. 14 utilizaram as bases PhysioNet e UCDDB, extraíram as séries R-R e treinaram um modelo de DL do tipo de rede de grupo residual de espremer e excitar unidimensional, que apresentou uma acurácia de 90,30%. Faust et al. 6 utilizaram um modelo de DL do tipo rede neural de memória de longo prazo com recorrência (LSTM, do inglês *long short-term memory network*) nas série R-R extraídas do ECG da base da PhysioNet e obtiveram uma acurácia de 81,30%. Bozkurt et al. 15 trabalharam com uma base própria, coletando o ECG de pacientes para o estudo. Em seguida, extraíram as séries R-R, calcularam diversas características e treinaram oito tipos de modelos ML, obtendo uma acuracidade de 85,71% com um *ensemble* de árvores de decisão (DT, do inglês *decision trees*), máquinas de vetores de suporte (SVM, do inglês *support vector machines*) e modelos do tipo k vizinhos próximos (KNN, do inglês *k-nearest neighbours*). Srinivasulu et al. 16 utilizaram a base MIT-BIH, extraindo as séries R-R e calculando características a partir delas, para treinar 4 tipos diferentes de ML, com a maior acurácia obtida com um *ensemble-bagged* de árvores de decisão de 89,60%. Padovano et al.17 treinaram uma SVM e um modelo do tipo KNN, com características calculadas das séries R-R a partir dos dados de ECG da base da PhysioNet, obtendo 81,40% de acurácia com a melhor SVM.

O restante do artigo está organizado da seguinte forma: na seção II é mostrado a metodologia do trabalho e as bases e algoritmos utilizados; na seção III os resultados são apresentados; a seção IV discute e compara os resultados obtidos e a seção V conclui o artigo, detalhando trabalhos futuros e continuações.

II. Materiais e Métodos

O trabalho foi dividido em etapas: leitura e pré-processamento dos dados, extração das séries R-R e das características, redução de dimensionalidade, busca dos hiperparâmetros, treinamento e teste dos modelos, comparação dos resultados e análise de explicabilidade. O macro fluxo pode ser visualizado na 1. Todo o projeto foi desenvolvido em Python. [1]

A. Base de Dados

A base de dados utilizada foi a Apnea-ECG Database 1.0.0 da PhysioNet. A base consiste em 70 registros do ECG de canal único, gravados a 100 Hz de 32 indivíduos, 25 homens e 7 mulheres. Os registros variam entre 401 a 534 minutos e foram analisados por especialistas que anotaram em cada minuto se o paciente estava em apneia ou não. Se no início de cada minuto o paciente estava em apneia do sono, aquele minuto foi marcado com o símbolo A, caso contrário com símbolo N. Os registros foram divididos em quatro grupos: A, B, C e X. O grupo corresponde A pacientes com índices de apneia do sono elevados (mais de 5 horas registradas de apneia), o grupo B são pacientes com ao menos 1 hora de apneia registrada, o grupo C é o grupo controle, com nenhuma hora completa de apneia registrada. O grupo X é o grupo originalmente designado para teste dos modelos [7], [12], [14], [18]. A metodologia de processamento foi baseada nos trabalhos de 7], [19]–[21. Neste trabalho, utilizou-se a separação de dados propostas pela PhysioNet. Os dados foram segmentados em janelas de 1 minuto com seu respectivo rótulo. Um filtro passa média proposto por [22] foi aplicado a cada janela. Em seguida, foram extraídas as séries R-R de cada janela utilizando o algoritmo de Hamilton, conforme descrito em [23].

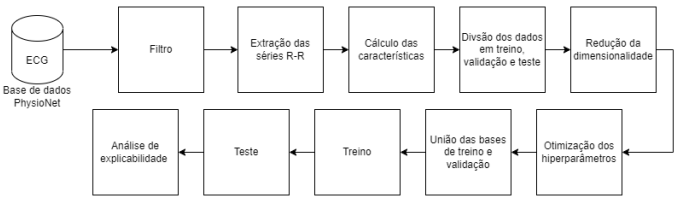

Fig. 1. Fluxo de processamento dos dados

B. Extração das Características

Para cada janela de série R-R de 1 minuto de duração foram calculadas 23 diferentes características que foram utilizadas no treinamento dos modelos de ML. Estas características pode ser divididas em características do domínio do tempo e da frequência [7], [24]–[26]:

1) *Características do domínio do tempo*: são características que calculam os aspectos estatísticos da série R-R e dos batimentos cardíacos, foram utilizadas: média do intervalo R-R (média) em ms; mediana dos intervalos R-R em ms (mediana); diferença entre o maior e o menor intervalo R-R em ms (amplitude); desvio padrão da janela inteira (SDNN); desvio padrão de dois picos R-R consecutivos (SDSD); número de intervalos R-R que diferem em mais de 50 ms em toda janela (NN50); porcentagem de NN50 em toda série (pNN50); número de intervalos R-R que diferem em mais de 20 ms (NN20); porcentagem de NN20 em todo intervalo (pNN20); raiz quadrada da média dos quadrados das diferenças dos intervalos R-R (RMSSD); coeficiente de variação da diferença dos intervalos R-R (CVSD); coeficiente de variação dos intervalos R-R (CVNNI); frequência cardíaca máxima do intervalo em BPM (MAXHR); frequência cardíaca mínima do intervalo em BPM (MINHR); média das frequências cardíacas do intervalo em BPM (média HR) e o desvio padrão da frequência cardíaca em BPM (STDHR).
2) *Características do domínio da frequência*:A variabilidade cardíaca possui três componentes predominantes de

[1]O código está disponível em https://github.com/lucas-sr/Mestrado

frequência, sendo as frequências muito baixas que vão de 0,003 a 0,04 Hz, as baixas frequências que vão 0,04 Hz até 0,15 Hz e as frequências altas de 0,15 até 0,4 Hz. Foram extraídas as potências destas três componentes, indicadas como VLF, LF e HF respectivamente. Além destas características, também foram calculadas: a razão entre as altas e baixas frequências (LF/HF); potência total (potência total); porcentagem de potência de baixa frequência na potência total (LFnu) e porcentagem de potências de alta frequência na potência total (HFnu).

As janelas que apresentaram erros durante o processo de cálculo foram removidas. Desta forma, um total de 34174 séries R-R foram extraídas.

C. Redução de Dimensionalidade

Para a redução de dimensionalidade foi empregada a técnica de análise das componentes principais (PCA, do inglês *principal components analysis*). Inicialmente foi aplicado em toda a base de dados sem limitação de componentes. Então foram analisados os resultados e computado quanto de variabilidade cada componente explicava até se obter 99% de explicabilidade. Então o PCA foi recalculado com este limite de componentes. Foram realizados experimentos com e sem a aplicação do PCA.

D. Divisão dos Dados

Os dados foram divididos em três grupos: treino, validação e teste, seguindo a proposta da PhysioNet, isto é, mantendo o grupo X para teste. Primeiro, foi selecionado uma amostra contendo 80% dos dados dos grupos A, B, C para treinamento, os 20% restantes foram para base de validação. A tabela I resume esta divisão.

Tabela I
SEPARAÇÃO DOS DADOS EM TREINO, VALIDAÇÃO E TESTE

Base	Grupos	Percentual dos Grupos	Total de Dados
Treino	A, B e C	80%	13587
Validação	A, B e C	20%	3397
Teste	X	100%	17191

E. Sintonia de Hiperparâmetros

A sintonia dos hiperparâmetros dos modelos de classificação pode se tornar um dos grandes obstáculos no treino de modelos em inteligência artificial, em termos de tempo computacional e complexidade de testes [27]. Nos experimentos deste trabalho, a otimização dos foi realizada através da ferramenta Optuna. Esta ferramenta consiste em uma busca otimizada de hiperparâmetros através de valores pré-definidos de cada hiperparâmetro e um critério de otimização (função objetivo). O principio de busca do Optuna se baseia na combinação das estratégias de amostragem e poda (*pruning*) com um filtro bayesiano. De forma simplificada, o Optuna busca os hiperparâmetros onde os melhores resultados estão sendo gerados, ”podando” iterações que não parecem promissoras [28].

Os modelos passaram por um processo de 100 iterações em busca dos melhores hiperparâmetros utilizando a acuracidade como métrica de otimização.

Em cada iteração foram utilizadas as bases de treinamento e validação. Os modelos foram treinados na base de treinamento com os hiperparâmetros selecionados para aquela interação e têm sua acuracidade medida na base de validação. A próxima seção detalha os espaços de busca utilizados.

F. Modelos de Classificação

Foram analisados 11 modelos de ML: Floresta Aleatória (RF, do inglês *random forest*), SVM, classificador bayesiano (Nayves-Bayes), Redes Neurais Artificiais do tipo multicamadas de perceptrons (MLP, do inglês *Multi-Layer Perceptron*), regressão logística (LR, do inglês *Linear Regression*), aglomeração de classificadores (*Bagging*), *Light Gradient Boosting* (LightGBM), *Extreme Gradient Boosting* (xgBoost), *Adaptive Boosting* (AdaBoost), *Ensemble* por voto majoritário (VM), DT.

1) Extreme Gradient Boosting (xgBoost): Neste estudo os hiperparâmetros testados para o xgBoost foram: taxa de aprendizado com amplitude de busca entre 0,1 e 2,0; a quantidade de estimadores, que variou entre 1 e 1000; a fração do tamanho das amostras utilizadas, com valores entre 0,1 e 1,0 e a profundidade máxima de cada estimador (número máximo de nós em uma árvore), cujo os valores foram selecionados entre 1 e 10.

2) Floresta Aleatória: Para o modelo RF foram avaliados o número de estimadores, 10 a 1000, o critério de medida da divisão do nó, que foi selecionado entre os valores ”índice de Gini” e ”entropia”; profundidade máxima da árvore, cujos valores variaram de 2 a 32, mas seguindo as potências de 2 (4, 8, 16) e a quantidade mínima de amostras necessárias antes de realizar uma divisão do nó, que variaram entre 2 e 10.

3) Máquina de Vetores de Suporte: No modelo SVM, o *kernel* foi escolhido entre os tipos ”linear”, ”polinomial”, ”RBF” (do inglês *radial basis function*) e ”sigmoide”, enquanto o hiperparâmetro C teve valores indo de 0,1 até 2,0, com um passo de 0,1.

4) Naive-Bayes: Para o classificador Naive-Bayes, não foi feito ajuste de hiperparâmetros, pois o modelo calcula todas as variáveis com base apenas nos dados utilizados.

5) Adaptive boosting : Outro tipo de modelo do tipo *ensemble* utilizado foi o AdaBoost. Neste modelo os hiperparâmetros foram o algoritmo empregado, cujos valores possíveis eram ”SAMME” e ”SAMME.R”, que são estimadores utilizados para classificação, sendo o ”SAMME.R” a versão com regularização do algoritmo SAMME [29], [30]. A taxa de aprendizado foi configurada com valores entre 0,1 e 2, com incrementações de 0,1 e o número de estimadores foi escolhido entre 1 e 1000.

6) Ensemble Voto Majoritário: Para o modelo VM, foram considerados *ensembles* de árvores de decisões (DT) e máquinas de vetores de suporte (SVM). A quantidade de estimadores foi escolhida com valores entre 2 e 4. Para os estimadores DT apenas o hiperparâmetro de profundidade

máxima foi otimizado, com objetivo de simplificar a execução. Este hiperparâmetro variou de 1 a 10. No caso dos estimadores de SVM, os hiperparâmetros C e gama foram estudados. Ambos tiveram valores entre 0.00001 e 100000. Além dos estimadores, foi criado um espaço de busca para os pesos de cada modelo variando entre 0 e 1,0.

7) Rede Neural do tipo Multicamadas de Perceptrons: No modelo MLP o espaço de busca consistiu no total de camadas de 1 a 4, a quantidade de perceptrons variando entre 1 e 100, os valores da função ativação variando entre "identidade" (sem função de ativação, retornando o próprio valor), "logística", "tangente hiperbólica" (tanh) e "retificação linear" (ReLu, do inglês *Rectified Linear Unit*); o parâmetro alpha de 0,0001 até 0,001, com passos de 0,0005 e a taxa de aprendizado de 0,0001 até 0,1, com passos de 0,005.

8) Regressão Logística: O modelo LR possui três hiperparâmetros principais: penalidade, C e peso da penalidade L1 (quando a mesma é aplicada). Para estes hiperparâmetros, os espaços de busca foram penalidade com "L1", "L2" e "*elastic net*"; C com valores entre 0,1 e 10 e o peso da penalidade "L1" com valores 0.01 até 0,99.

9) Light Gradient Boosting: Para o modelo LightGBM, os hiperparâmetros são quantidade de folhas máximas para os estimadores, com valores entre 2 e 256, taxa de aprendizado entre 0,01 e 0,1; profundidade máxima para os estimadores, cujos valores foram selecionados entre 2 e 16; quantidade mínima do somatório de pesos necessário em uma folha (as vezes referida como *child*, do inglês criança), com valores entre 1 e 100; tamanho da razão da amostragem para instância de treinamento, com valores entre 0,1 e 1; tamanho da razão de amostras de colunas quando utilizados em DT, que variou de 0.1 e 1; os reguladores de peso alfa e lambda, ambos tendo valores escolhidos entre 1×10^{-9} e 10 e quantidade de estimadores escolhida entre 50 e 500.

10) Agregação de Classificadores): Na agregação de classificadores, os classificadores consistiram na seguinte lista no espaço de busca: "SVM", "árvore de decisão" (DT), "regressão logística" (LR) e "k-próximos vizinhos" (KNN, do inglês *k-nearest neighbours*). Os demais hiperparâmetros foram: quantidade total de estimadores, de 10 até 1000, selecionando valores em potência de 10, a razão do número máximo de amostras retiradas da base de treinamento, que variou de 0,2 até 0,8 e se esta subamostragem foi com reposição ou não (no caso, uma lista com verdadeiro, indicando com reposição e falso, sem reposição).

11) Árvores de Decisão: Na modelagem por DT, os hiperparâmetros do espaço de busca foram o critério para medir a qualidade da divisão dos nós, cujos valores poderiam ser "índice de Gini" e "entropia", a estratégia para divisão do nó, se "melhor" ou "aleatória", em que a divisão do nó será de forma aleatória ou utilizará a melhor divisão possível [31], a profundidade máxima da árvore, que variou de 2 até 64, seguindo potências de 2 e a quantidade mínima de amostras necessárias para dividir um nó interno, cujos valores foram escolhidos entre 2 e 10.

Após a busca dos hiperparâmetros, os modelos foram treinados novamente, desta vez utilizando a combinação da base de treinamento com a de validação, com os hiperparâmetros selecionados pelo Optuna. Em seguida os modelos foram testados com a base de teste, tendo sua acuracidade, sensitividade, sensibilidade e F-1 *score* medidos.

III. Resultados

Neste estudo foram conduzidos dois experimentos com todos modelos treinados com o fluxo descrito, mudando a técnica de redução de dimensão empregada. O tempo total de execução sem redução de dimensão foi de 732 minutos, enquanto com aplicação do PCA foi de 710 minutos. Os resultados estão dispostos nas tabelas III e III. A tabela III apresenta os hiperparâmetros selecionados para cada modelo durante a otimização e o tempo decorrido na otimização. A tabela III apresenta os resultados de métricas de desempenho obtidas para os modelos de aprendizado de máquina.

IV. Discussão

Alguns modelo apresentaram hiperparâmetros com valores similares para os experimentos sem redução e com redução de PCA, como os modelos xgBoost, SVM e DT. Os modelos VM e MLP tiveram sua arquitetura modificada em cada caso, tendo mais camadas e estimadores com o PCA.

A redução de dimensionalidade diminuiu em aproximadamente 30 (min) o tempo total de execução. Os modelos xgBoost, RF, AdaBoost, LR e *bagging* tiveram o tempo de otimização menor com uso do PCA. A arquitetura mais complexa selecionada para os modelos VM e MLP impactou em um maior tempo de otimização. Em ambos os experimentos, os modelos do tipo agregadores, VM e *bagging*, foram os que tiveram maior tempo na otimização.

A acurácia dos modelos foi, de forma geral, ligeiramente pior quando aplicado o PCA. A maior acurácia foi obtida pelo AdaBoost no experimento I (AdaBoost-I) e pelo *bagging* no experimento II (Bagging-II), ambas de 77,23%.

Ainda observando os resultados dos melhores modelos, é possível mensurar o impacto de cada um dos hiperparâmetros no desemepnho e no tempo gasto durante o treino do modelo [28]. A figura 2 mostra para ambos os modelos a importância de cada hiperparâmetro no resultando, enquanto a 3 mostra o impacto no tempo de otimização. Como esperado para modelo *bagging*, o tipo de estimador utilizado foi o que mais impactou nos resultados e no tempo de processamento. Já para o modelo AdaBoost, enquanto a taxa de aprendizado foi o hiperparâmetro que mais impactou no resultado, o tempo de otimização foi praticamente dominado pela quantidade de estimadores.

Os valores SHAP (do inglês *shapley additive explanations*) auxiliam na explicabilidade do resultado de um modelo, mostrando o quanto a presença ou ausência de uma característica contribuiu para o resultado. Mais detalhes podem ser obtidos em [32]. Neste trabalho foram calculados valores SHAP para o AdaBoost-I com uma amostra aleatória de 200 dados da base de teste. A média dos valores absolutos SHAP é

Tabela II
HIPERPARÂMETROS SELECIONADOS E TEMPO DE OTIMIZAÇÃO DE CADA MODELO EM CADA EXPERIMENTO

Modelo	Hiperparâmetro	Valor (sem redução)	Valor (com PCA)	Tempo (min-sem redução)	Tempo (min-com PCA)
xgBoost	Taxa de aprendizado	0,1	0,2	102,20	43,03
	N° de estimadores	968	948		
	Fração das amostras	0,5636495421	0,3640537916		
	Produndidade máxima	8	10		
RF	N° de estimadores	739	167	50,33	23,58
	Critério	Entropia	Gini		
	Produndidade máxima	30	29		
	Quantidade mínima de amostras	7	3		
SVM	C	2	1,9	12,78	13,06
	Kernel	RBF	RBF		
AdaBoost	Algoritmo	SAMME.R	SAMME.R	30,03	23,45
	Taxa de aprendizado	0,2	0,1		
	N° de estimadores	693	994		
VM	N° de estimadores	3	4	363,05	464,92
	Produndidade máxima (DT 1)	9	10		
	C (SVM 1)	93756,31004	45083,02637		
	Gama (SVM 1)	48879,60707	10512,42835		
	Produndidade máxima (DT 2)	7	6		
	C (SVM 2)	7896,597223	80724,09656		
	Gama (SVM 2)	7873,042218	99211,74773		
	Produndidade máxima (DT 3)	6	6		
	C (SVM 3)	51232,20802	31342,87679		
	Gama (SVM 3)	44921,14289	7615,321521		
	Produndidade máxima (DT 4)	N/A	8		
	C (SVM 4)	N/A	45578,86932		
	Gama (SVM 4)	N/A	76410,59746		
	Peso 1	0,3849179657	0,5952377575		
	Peso 2	0,8526480215	0,8082523159		
	Peso 3	0,9104036239	0,8285626392		
	Peso 4	0,7707194459	0,3521565459		
	Peso 5	0,6798722113	0,6694154497		
	Peso 6	0,1406540305	0,1727339096		
	Peso 7	N/A	0,226812838		
	Peso 8	N/A	0,3015213248		
MLP	Número de camadas	2	4	6,53	8,05
	N° de neurônios na camada 1	30	32		
	N° de neurônios na camada 2	1	46		
	N° de neurônios na camada 3	N/A	87		
	N° de neurônios na camada 4	N/A	64		
	Função de ativação	ReLu	ReLu		
	Alfa	0,0001	0,0001		
	Taxa de aprendizado	0,0101	0,0501		
LR	Penalidade	l2	l2	1,33	0,14
	C	2,257354716	0,6674282539		
	Peso da penalidade L1	0,5517660896	0,7420867606		
LightGBM	N° de folhas	117	182	0,77	1,20
	Taxa de aprendizado	0,02561994674	0,07392790535		
	Profundidade máxima	10	10		
	Mínimo de dados em folha (child)	65	31		
	Tamanho da amostra	0,4888985799	0,2955943978		
	Tamanho da amostra	0,606875972	0,9738689274		
	Alfa	5,30621652	4,78386346		
	Lambda	9,527773716	9,378478357		
	N° de estimadores	397	481		
Bagging	Estimador	DT	SVM	108,67	77,47
	N° de estimadores	472	14		
	Amostras retiradas da base de treinamento	0,2813649597	0,7808223034		
	Reposição	Verdadeiro	Verdadeiro		
DT	Critério	Gini	Entropia	0,26	0,22
	Estratpegia de divisão	Melhor	Melhor		
	Profundidade máxima	7	7		
	Mínimo de amostras para divisão	3	4		

Tabela III
RESULTADOS DAS MÉTRICAS DE DESEMPENHO OBTIDAS PARA OS MODELOS DE CLASSIFICAÇÃO

Experimento	Modelo	Acurácia	Sensibilidade	Precisão	F-1 Score
I (nenhuma redução)	XGBoost	74,25%	78,64%	80,31%	79,47%
	RF	76,17%	80,09%	81,97%	81,02%
	SVM	76,94%	79,93%	83,88%	81,86%
	Naive-Bayes	65,16%	**82,18%**	55,97%	66,59%
	AdaBoost	**77,23%**	81,03%	82,65%	81,83%
	VM	75,66%	78,03%	84,58%	81,17%
	MLP	75,63%	82,00%	77,81%	79,85%
	LR	74,38%	76,22%	85,31%	80,51%
	LightGBM	76,29%	80,43%	81,64%	81,03%
	Bagging	76,09%	80,04%	81,88%	80,95%
	DT	73,81%	79,74%	77,48%	78,59%
II (com redução PCA)	XGBoost	69,94%	75,69%	75,92%	75,81%
	RF	75,38%	78,74%	82,62%	80,63%
	SVM	76,98%	79,90%	84,03%	81,92%
	Naive-Bayes	69,84%	75,00%	77,07%	76,02%
	AdaBoost	76,76%	79,28%	84,67%	81,89%
	VM	72,00%	73,00%	87,08%	79,42%
	MLP	75,54%	80,79%	79,47%	80,12%
	LR	73,96%	75,75%	85,35%	80,27%
	LightGBM	74,25%	77,87%	81,73%	79,75%
	Bagging	**77,23%**	79,94%	84,51%	82,16%
	DT	71,99%	75,45%	81,29%	78,26%

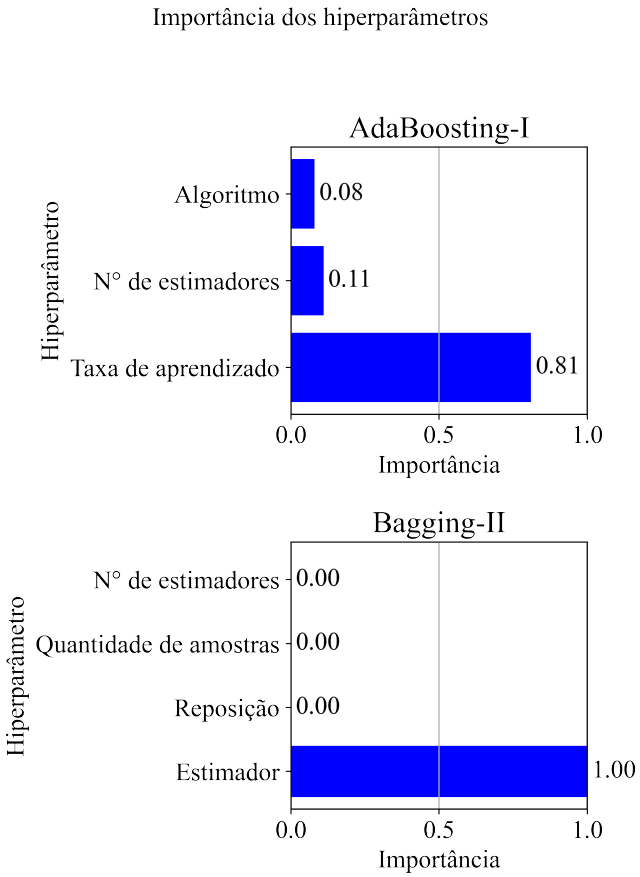

Fig. 2. Importância de cada hiperparâmetro nos resultados de treinamento

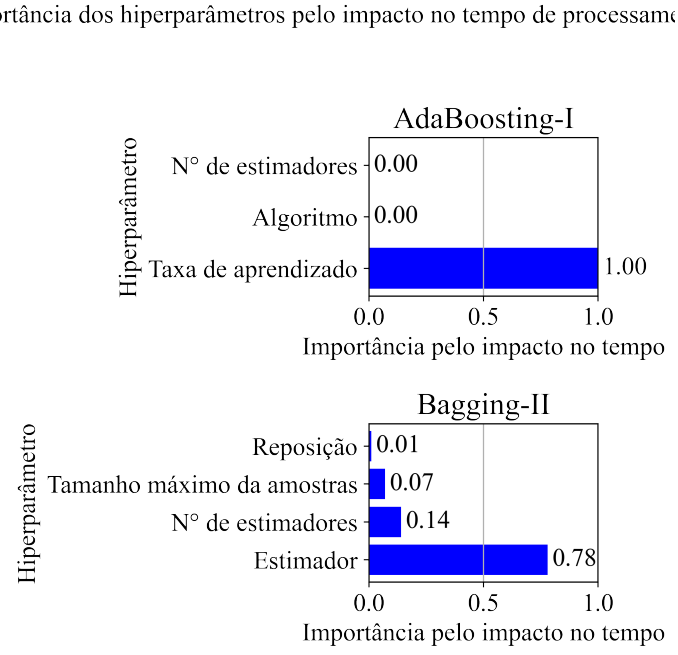

Fig. 3. Impacto no tempo de otimização em cada modelo

mostrada na figura 4. A partir desta amostra, é possível notar que a característica potência total foi a que mais contribuiu para o resultado do modelo, seguida por NN20 e CVNNi, do domínio do tempo, e por LFnu, do domínio da frequência. Não foram calculados os valores SHAP para Bagging-II, devido a limitações computacionais.

Quando comparado aos trabalhos similares, os modelos aqui construídos tiveram um desempenho, de forma geral, pior que os modelos de DL. Quando comparado a outros modelos de

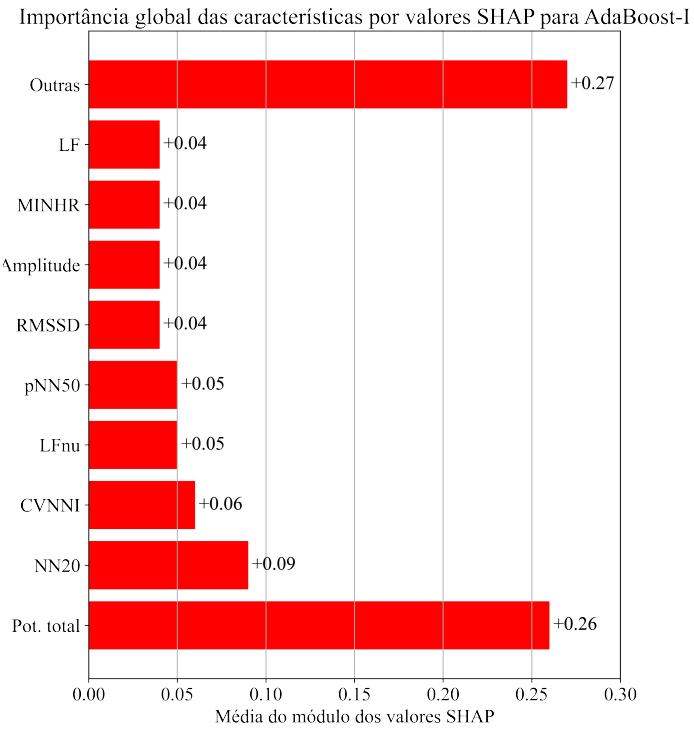

Fig. 4. Valores SHAP para AdaBoost-I

ML, os resultados apresentaram uma acurácia semelhante. As principais diferenças nas metodologias foram na quantidade de dados empregados para treinamento e na forma de seleção de hiperparâmetros. A tabela IV apresenta o comparativo de metodologias, enquanto a tabela V mostra o comparativo de resultados deste com outros trabalhos.

Em termos de explicabilidade, Bahrami e Forouzanfar [7] buscaram avaliar o impacto das características através do uso de uma DT, medindo o valor do peso de cada característica. Encontraram a HF e VLF como as principais características para detectar a apneia do sono. Padovano et al. [17] utilizou da seleção de características para listar as que mais impactaram na classificação, também encontraram as características do domínio da frequência como as com maior poder de classificação, com destaque para HF extraídas com periodograma Lomb-Scargle. Estes achados estão em sincronia com a característica que mais contribuiu para o resultado do modelo AdaBoost-I, potência total.

V. Conclusão e trabalhos futuros

Este trabalho teve como objetivo analisar o desempenho de diferentes modelo de ML aplicados na classificação da apneia do sono. Neste aspecto, este trabalho contribui para classificação da apneia do sono com uso de ECG ao adicionar dois novos fatores na discussão dos resultados: explicabilidade com uso de valores SHAP e uso de Optuna na otimização dos hiperparâmetros. Outra característica deste trabalho foi apontar o tempo total de execução, de cada modelo e de cada hiperparâmetro. Isso possibilitou novas análises e ponderações sobre custo-benefício do emprego de um determinado modelo na detecção da apneia do sono. O próximo passo é aplicar o ambiente computacional desenvolvido em modelos DL e a execução de testes de hipóteses, a fim de estabelecer um intervalo de confiança dos modelos. Alguns modelos candidatos são CNN, LSTM, AlexNet e modelos híbridos de ML com DL. O ambiente também pode ser testado com outras técnicas de redução de dimensionalidade como UMAP (do inglês *uniform manifold approximation and projection*) e t-SNE (do inglês *t-distributed stochastic neighbor embedding*). O aumento do tamanho da base de treinamento também pode produzir melhores resultados, estudo dos valores SHAP também pode ser feito em trabalhos futuros, analisando quais características mais contribuíram em todos os modelos e criando um *rank* das mais importantes.

Referências

[1] S. Javaheri, F. Barbe, F. Campos-Rodriguez, J. A. Dempsey, R. Khayat, S. Javaheri, A. Malhotra, M. A. Martinez-Garcia, R. Mehra, A. I. Pack, V. Y. Polotsky, S. Redline, and V. K. Somers, "Sleep apnea: Types, mechanisms, and clinical cardiovascular consequences," pp. 841–858, 2 2017.

[2] A. V. Benjafield, N. T. Ayas, P. R. Eastwood, R. Heinzer, M. S. Ip, M. J. Morrell, C. M. Nunez, S. R. Patel, T. Penzel, J. L. D. Pépin, P. E. Peppard, S. Sinha, S. Tufik, K. Valentine, and A. Malhotra, "Estimation of the global prevalence and burden of obstructive sleep apnoea: a literature-based analysis," *The Lancet Respiratory Medicine*, vol. 7, pp. 687–698, 8 2019.

[3] D. Ayonara, M. H. L. D. Silva, C. C. D. Vasconcelos, R. D. O. Costa, R. Ribeiro, and S. Costa, "Set-out," pp. 1621–1626, 2014.

[4] T. Kasai and T. D. Bradley, "Obstructive sleep apnea and heart failure: Pathophysiologic and therapeutic implications," pp. 119–127, 1 2011.

[5] S. C. Veasey and I. M. Rosen, "Obstructive sleep apnea in adults," *New England Journal of Medicine*, vol. 380, pp. 1442–1449, 4 2019.

[6] O. Faust, R. Barika, A. Shenfield, E. J. Ciaccio, and U. R. Acharya, "Accurate detection of sleep apnea with long short-term memory network based on rr interval signals," *Knowledge-Based Systems*, vol. 212, 1 2021.

[7] M. Bahrami and M. Forouzanfar, "Sleep apnea detection from single-lead ecg: A comprehensive analysis of machine learning and deep learning algorithms," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, 2022.

[8] S. K. Saini and R. Gupta, "Artificial intelligence methods for analysis of electrocardiogram signals for cardiac abnormalities: state-of-the-art and future challenges," *Artificial Intelligence Review*, vol. 55, pp. 1519–1565, 2 2022.

[9] N. Salari, A. Hosseinian-Far, M. Mohammadi, H. Ghasemi, H. Khazaie, A. Daneshkhah, and A. Ahmadi, "Detection of sleep apnea using machine learning algorithms based on ecg signals: A comprehensive systematic review," 1 2022.

[10] N. Singh and R. H. Talwekar, ""comparison of machine learning and deep learning classifier to detect sleep apnea using single-channel ecg and hrv: A systematic literature review," vol. 2273. Institute of Physics, 2022.

[11] A. Anand, T. Kadian, M. K. Shetty, and A. Gupta, "Explainable ai decision model for ecg data of cardiac disorders," *Biomedical Signal Processing and Control*, vol. 75, p. 103584, 2022.

[12] T. Penzel, G. B. Moody, R. G. Mark, A. L. Goldberger, and J. H. Peter, "The apnea-ecg database," in *Computers in Cardiology 2000. Vol. 27 (Cat. 00CH37163)*. IEEE, 2000, pp. 255–258.

[13] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

Tabela IV
COMPARATIVO DAS CONFIGURAÇÕES COM A LITERATURA

Autores	Modelo	Dados para treino	Seleção de hiperparâmetros	N° de características
Srinivasulu et al. [16]	Bagging de árvores de decisão	90%	Não discutido	6
Yang et al. [14]	CNN 1D-SERestGNet	50%	Empiracamente	1 (série R-R)
Padovano et al. [17]	SVM	-	Não discutido	15
Faust et al. [6]	LSTM	-	Fixados	1 (série R-R)
Bozkurt et al. [15]	Agregador de modelos	-	Não discutido	10
Bahrami e Forouzanfar [7]	Híbrido ZFNet-BiLSTM	80%	Fixados	2 (série e amplitude R-R)
	Agregador com voto majoritário	80%	Empiracamente	27*
Este trabalho	AdaBoost-I	50%	*Framework Optuna*	23
	Bagging-II	50%	*Framework Optuna*	10

Nota: Redes neurais convolucionais (CNN do inglês *convolutional neural network*). Redes neurais de memória de longo prazo (LSTM, do inglês *long short-term memory*) . *Foi utilizado PCA para redução de dimensionalidade, mas não foi informado o total de componentes utilizadas.

Tabela V
COMPARATIVOS DOS RESULTADOS COM DADOS DA LITERATURA

Autores	Modelo	Acurácia	Sensibilidade	Precisão
Srinivasulu et al. [16]	Bagging de árvores de decisão	89,60%	95,40%	66,10%
Yang et al. [14]	CNN 1D-SERestGNet	90,30%	91,90%	87,60%
Padovano et al. [17]	SVM	81,40%	82,00%	74,00%
Faust et al. [6]	LSTM	81,30%	59,90%	91,80%
Bozkurt et al. [15]	Agregador de modelos	87,10%	90,00%	85,00%
Bahrami e Forouzanfar [7]	Híbrido ZFNet-BiLSTM	88,10%	81,50%	92,30%
	Agregador com voto majoritário	79,40%	68,70%	85,60%
Este trabalho	AdaBoost-I	77,20%	81,00%	82,70%
	Bagging-II	77,20%	79,90%	84,50%

[14] Q. Yang, L. Zou, K. Wei, and G. Liu, "Obstructive sleep apnea detection from single-lead electrocardiogram signals using one-dimensional squeeze-and-excitation residual group network," *Computers in Biology and Medicine*, vol. 140, 1 2022.

[15] F. Bozkurt, M. K. Uçar, C. Bilgin, and A. Zengin, "Sleep–wake stage detection with single channel ecg and hybrid machine learning model in patients with obstructive sleep apnea," *Physical and Engineering Sciences in Medicine*, vol. 44, pp. 63–77, 3 2021.

[16] A. Srinivasulu, S. Mohan, T. Harika, P. Srujana, and Y. Revathi, "Apnea event detection using machine learning technique for the clinical diagnosis of sleep apnea syndrome." Institute of Electrical and Electronics Engineers Inc., 5 2021, pp. 490–493.

[17] D. Padovano, A. Martinez-Rodrigo, J. M. Pastor, J. J. Rieta, and R. Alcaraz, "An experimental review on obstructive sleep apnea detection based on heart rate variability and machine learning techniques." Institute of Electrical and Electronics Engineers Inc., 10 2020.

[18] H. Almutairi, G. M. Hassan, and A. Datta, "Classification of obstructive sleep apnoea from single-lead ecg signals using convolutional neural and long short term memory networks," *Biomedical Signal Processing and Control*, vol. 69, 8 2021.

[19] K. Li, W. Pan, Y. Li, Q. Jiang, and G. Liu, "A method to detect sleep apnea based on deep neural network and hidden markov model using single-lead ecg signal," *Neurocomputing*, vol. 294, pp. 94–101, 6 2018.

[20] P. D. Chazal, C. Heneghan, E. Sheridan, R. Reilly, P. Nolan, and M. O'Malley, "Automated processing of the single-lead electrocardiogram for the detection of obstructive sleep apnoea," *IEEE Transactions on Biomedical Engineering*, vol. 50, pp. 686–696, 6 2003.

[21] K. Feng, H. Qin, S. Wu, W. Pan, and G. Liu, "A sleep apnea detection method based on unsupervised feature learning and single-lead electrocardiogram," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021.

[22] L. Chen, X. Zhang, and C. Song, "An automatic screening approach for obstructive sleep apnea diagnosis based on single-lead electrocardiogram," *IEEE transactions on automation science and engineering*, vol. 12, no. 1, pp. 106–115, 2014.

[23] P. Hamilton, "Open source ecg analysis," in *Computers in cardiology*. IEEE, 2002, pp. 101–104.

[24] T. F. o. t. E. S. o. C. t. N. A. S. o. P. Electrophysiology, "Heart rate variability: standards of measurement, physiological interpretation, and clinical use," *Circulation*, vol. 93, no. 5, pp. 1043–1065, 1996.

[25] G. Clifford, "Signal processing methods for heart rate variability," Ph.D. dissertation, Oxford University, UK, 2002.

[26] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *American journal of physiology-heart and circulatory physiology*, 2000.

[27] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A.-L. Boulesteix *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 13, no. 2, p. e1484, 2023.

[28] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[29] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.

[30] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[31] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[32] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf