UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

RENAN BORTOLUZZI DA SILVA

Detecção de Apneia do Sono Utilizando *Machine Learning* Baseado em Modelos Estatísticos

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Engenharia da Computação

Orientador: Prof. Dr. Leandro Krug Wives
Coorientador: MSc. Oscar Ortegon

Porto Alegre
2020

“Se a educação sozinha não transforma a sociedade,
sem ela tampouco a sociedade muda.”
— PAULO FREIRE

AGRADECIMENTOS

Agradeço primeiramente a minha família, especialmente minha mãe, que me oportunizou estudar em uma universidade tão conceituada, sem vocês isso não seria possível.

Agradeço a meu orientador Leandro Wives, e ao coorientador Oscar pela paciência e boa vontade para me ajudar a fazer um trabalho relevante. Deixo também um agradecimento a todos os professores do curso que de alguma forma contribuíram para a minha formação.

Agradeço à Clínica do Sono, na pessoa do Dr. Denis Martinez, pela parceria no desenvolvimento deste trabalho, fornecendo os exames e disponibilizando os técnicos para que nos orientassem sobre o procedimento de diagnóstico realizado atualmente. Sem esta parceria, este trabalho não teria isso possível.

Quero agradecer ao meu colega João Gubert com quem desde o início deste trabalho pude discutir os problemas e pensar soluções relacionadas ao trabalho. Um agradecimento especial a todos os amigos que conheci durante estes anos de estudos e que sempre me serviram de apoio nas horas que mais precisei.

Por fim, agradeço a minha grande amiga Luiza Butzge, que me acompanha desde o primeiro dia de aula nesta universidade e teve um papel fundamental de apoio psicológico e afetivo durante todo o curso e durante este trabalho.

RESUMO

Apneia Obstrutiva do Sono é uma doença comum causada pela obstrução das vias aéreas superiores (fossas nasais, faringe e laringe) durante o sono, e afeta aproximadamente 4% da população mundial. O exame de polissonografia é o método mais comum usado para para diagnosticar essa doença, e ele fornece vários sinais fisiológicos. Neste trabalho são investigadas técnicas de *machine learning* baseadas em modelos matemáticos para realizar a detecção automática da Apneia Obstrutiva do Sono. Mais especificamente, utiliza-se *Support Vector Machines* e *Ada Boost*, e é realizada uma avaliação do seu desempenho na detecção de apneias. Avaliando os modelos, obteve-se que a execução com época de 15 segundos utilizando o algoritmo *Ada Boost* com *cross-validation* e dados balanceados foi a que apresentou melhores resultados, obtendo acurácia de 65%, precisão de 74% e *recall* de 60%. No geral, os modelos utilizando o método *train-test* apresentam resultados de acurácia superiores quando comparados aos resultados obtidos com o método *cross-validation*. No entanto, levando em conta outras métricas como precisão e *F1 score*, percebe-se que os resultados obtidos com método *train-test* são consideravelmente inferiores aos resultados obtidos com o método de *cross-validation*.

Palavras-chave: Distúrbios do sono. Polissonografia. Seleção de *características*. Particionamento de sinais. Detecção de apneia do sono. Aprendizado de máquina.

Obstructive Sleep Apnea Detection using
Machine Learning based on statistical models

ABSTRACT

Obstructive Sleep Apnea is a prevalent sleep disease caused by upper airway closure (nasal cavities, pharynx, and larynx) during sleeping affects approximately 4% of the worldwide population. The polysomnography exam is the most common method to diagnose obstructive sleep apnea as it provides lots of physiological signals. In this work, we investigate machine learning based on mathematical methods to detect obstructive sleep apnea automatically. Thus, we propose using Support Vector Machine and Ada Boost with decision tree algorithms to test and compare obstructive sleep apnea detection results. After evaluating the models, it was observed that the better combination was 15 seconds slices, Ada Boost, cross-validation, and balanced data, achieving an accuracy of 65%, 74% of precision, and 60% of recall. However, in general, the models using the train-test method showed superior accuracy compared to cross-validation. Nevertheless, the other metrics are considerably lower in this case. Considering precision and F1 score, it is clear that the results obtained with the train-test method are considerably lower than the results obtained with the cross-validation method.

Keywords: Sleep disorder, Polysomnography, Feature Selection, Signal partitioning, Sleep apnea detection, *Machine learning*.

LISTA DE ABREVIATURAS E SIGLAS

ACS	Apneia Central do Sono
AMS	Apneia Mista do Sono
AOS	Apneia Obstrutiva do Sono
IAH	Índice de apneia-hipopneia
ECG	Eletrocardiograma
PSG	Polissonografia
HMM	Hidden Markov Models (Modelos Ocultos de Markov)
RQ	*Research Question* (Questão de Pesquisa)
SVM	*Support Vector Machine*
SpO2	Saturação de oxigênio
ML	*Machine Learning* (Aprendizado de Máquina)
DL	Deep Learning
TQWQ	*Tunable-Q wavelet transform*
NIG	Normal Inverse Gaussian
EDF	European Data Format

LISTA DE FIGURAS

Figura 2.1 Aplicação de ML na Saúde 16
Figura 2.2 Árvore de decisão aprendida para dados de tênis 17
Figura 2.3 Fluxograma de treinamento do algoritmo do tipo ensemble 18
Figura 2.4 Tabela de contingência binária 19

Figura 3.1 Amostra de segmentos de sinal de ECG normais e apneicos 23

Figura 4.1 Etapas de desenvolvimento do trabalho 24
Figura 4.2 Informações do exame de polissonografia 25
Figura 4.3 Sinal da cinta torácica original e após tratamento 27
Figura 4.4 Dados de diagnóstico desbalanceados 28
Figura 4.5 Dados de diagnóstico após aplicação de *random under-sampling* 29

Figura 5.1 Conjunto de treino e teste para *cross-validation* 32

Figura 6.1 Acurácia com *cross-validation* e dados balanceados 37
Figura 6.2 Acurácia com *cross-validation* e dados desbalanceados 37
Figura 6.3 Comparação do *F1 Score* com *cross-validation* 38
Figura 6.4 Acurácia com *train-test* e dados desbalanceados 39
Figura 6.5 Acurácia do método SVM com dados desbalanceados 39
Figura 6.6 Acurácia método SVM com dados desbalanceados 40
Figura 6.7 *F1 Score* do modelo SVM com *cross-validation* 41
Figura 6.8 Acurácia do modelo *Ada Boost* com *cross-validation* 41
Figura 6.9 *F1 score* do modelo *Ada Boost* com *cross-validation* 42

LISTA DE TABELAS

Tabela 3.1 Trabalhos Relacionados 21

Tabela 5.1 Avaliação com *Train-Test* - 2s 33
Tabela 5.2 Avaliação com *Cross-Validation* - 2s 33
Tabela 5.3 Avaliação com *Train-Test* 34
Tabela 5.4 Avaliação com *Cross-Validation* 34
Tabela 5.5 Avaliação com *Train-Test* 34
Tabela 5.6 Avaliação com *Cross-Validation* 35
Tabela 5.7 Avaliação com *Train-Test* 35
Tabela 5.8 Avaliação com *Cross-Validation* 35
Tabela 5.9 Avaliação com *Train-Test* 36
Tabela 5.10 Avaliação com *Cross-Validation* 36

Tabela 6.1 Matriz de confusão SVM com dados desbalanceados 40

SUMÁRIO

1 INTRODUÇÃO **11**
2 MARCO CONCEITUAL **13**
2.1 Apneia Obstrutiva do Sono **13**
2.2 Exame de Polissonografia **14**
2.3 *Machine Learning* **15**
2.3.1 *Support Vector Machine* 15
2.3.2 *Decision Tree* 16
2.3.3 AdaBoost 17
2.3.4 Avaliação de modelos 18
3 TRABALHOS RELACIONADOS **20**
4 METODOLOGIA **24**
4.1 Aquisição dos dados **24**
4.2 Armazenamento e anonimização de dados **24**
4.3 Limpeza dos sinais e eventos **25**
4.3.1 Pré-processamento de sinais 25
4.3.2 Pré-processamento do diagnóstico 27
4.3.3 Dados Desbalanceados 28
4.4 Particionamento dos sinais **29**
4.5 Extração de *features* **30**
5 EXPERIMENTOS **31**
5.1 Época de 2s **32**
5.2 Época de 5s **33**
5.3 Época de 10s **34**
5.4 Época de 15s **35**
5.5 Época de 30s **36**
6 ANÁLISE E DISCUSSÃO DOS RESULTADOS **37**
6.1 Comparação de modelos **37**
7 CONCLUSÃO **43**
REFERÊNCIAS **44**

1 INTRODUÇÃO

A Apneia Obstrutiva do Sono (AOS) é um distúrbio crônico grave no qual há falta de fluxo de ar no nariz e na boca por pelo menos 10 segundos (Gutta; Cheng, 2016). Isso geralmente é acompanhado por uma redução na saturação de oxigênio no sangue e leva ao despertar do sono para respirar. A AOS é importante fator de risco de implicações para a saúde, como aumento de doenças cardiovasculares, morte súbita, depressão e dificuldades de aprendizagem (Pombo; Garcia; Bousson, 2017).

A gravidade da AOS é medida pelo número de ocorrências durante uma hora de sono. Segundo Pombo, Garcia e Bousson (2017), a doença pode ser classificada como Apneia Obstrutiva do Sono (AOS), Apneia Central do Sono (ACS), Apneia Mista do Sono (AMS) ou Hipopneia.

A ferramenta mais comum para diagnosticar a apneia do sono é o exame de polissonografia (PSG). Esse método, embora comum, possui algumas desvantagens. É invasivo, necessita de um período prolongado para coletar os dados, em geral uma ou duas noites de sono, e é uma experiência desconfortável para o paciente, exigindo vários fios e eletrodos conectados a ele durante a gravação do sinal (Song et al., 2016). No entanto, como fornece aproximadamente 16 sinais fisiológicos, incluindo fluxo de ar respiratório, ronco, saturação periférica de oxigênio (SpO2), eletrocardiografia (ECG), cintas abdominais e torácicos, ainda é muito importante nos dias de hoje.

De acordo com Mahmud et al. (2018) a necessidade de soluções de saúde e esforços contínuos para compreender as bases biológicas das patologias impulsionou uma extensa pesquisa em ciências biológicas nos últimos dois séculos. Com base nisso, diversos métodos computacionais foram propostos para detectar ou prever automaticamente a apneia do sono, como a classificação baseada em limiares e o *machine learning* (*ML*). A classificação baseada em limiares opera usando limites diferentes, sendo um método muito dependente da seleção de valores apropriados. Métodos de ML fornecem a capacidade de lidar com um grande e complexo conjunto de dados eletrônicos e fornecem resultados precisos e confiáveis (Pombo; Garcia; Bousson, 2017).

De acordo com Pombo, Garcia e Bousson (2017), métodos baseados em limiares são adequados para detectar discrepâncias e anormalidades no processamento do sinal. No entanto, um método de calibração eficiente ainda é necessário. Como o número de sinais e dados é significativo, o diagnóstico manual da apneia do sono torna-se difícil e pode levar a erros humanos. Muitos dos métodos propostos para detectar automatica-

mente a apneia do sono são baseados no sinal de ECG e SpO2, de acordo com Gutta e Cheng (2016).

Em parceria com uma clínica especializada em distúrbios do sono de Porto Alegre [1], foi realizado um estudo de campo, onde foi possível acompanhar o dia-a-dia dos técnicos que fazem o diagnóstico. Neste acompanhamento observou-se que os técnicos utilizam os sinais de fluxo aéreo, cinta abdominal e torácica e saturação de oxigênio para as análises. Constatou-se também os problemas decorrentes do diagnóstico manual, que são a demora no resultado, maior propensão a erros e demasiado esforço manual. Assim, a clínica forneceu dados de exames de polissonografia para que fosse realizada uma pesquisa detalhada a fim de propor uma solução automatizada para o diagnóstico a apneia obstrutiva do sono.

Assim, o objetivo neste trabalho consiste em verificar se os sinais acima mencionados contém informações necessárias para detectar a AOS usando *machine learning* baseado em modelos estatísticos. Serão analisados alguns modelos estatísticos, como *Support Vecto Machine (SVM)* e árvore de decisão, para verificar se essas informações combinadas fornecem bons resultados.

O grande desafio na detecção manual de AOS é o tempo para análise dos dados e a probabilidade de erros humanos. Portanto, implementar métodos computacionais para detectar/prever apneia ajudará os profissionais a encontrar resultados mais precisos, o que pode reduzir o tempo para entregar o diagnóstico, baratear os custos e, assim, permitir que mais pessoas tenham acesso ao diagnóstico de tratamento da AOS.

Para encontrar um método mais acurado e os sinais fisiológicos que melhor expressam a AOS, foram propostas as seguintes perguntas de pesquisa (RQs):

RQ1: Quais são os métodos computacionais mais precisos para detectar a AOS automaticamente?

RQ2: Quais sinais poderiam ser usados para diagnosticar com precisão a AOS?

RQ3: É possível obter bons resultados usando *machine learning* baseado em um modelo estatístico?

Nas próximas seções serão apresentados mais detalhes sobre apneia obstrutiva do sono, métodos de aprendizado de máquina, uma visão geral dos trabalhos relacionados, como funciona o diagnóstico manual e, por fim, será proposto um modelo que apresentou melhor desempenho quanto à métricas de avaliação que foram analisadas.

[1] Endereço web da clínica: http://www.sono.com.br/

2 MARCO CONCEITUAL

Este capítulo descreve os conceitos abordados neste trabalho, fundamentais para a compreensão da pesquisa realizada.

2.1 Apneia Obstrutiva do Sono

A Apneia Obstrutiva do Sono (AOS), um estado de distúrbio do sono, é definida como apneia repetitiva, hipóxia crônica, dessaturação de oxigênio e hipercapnia. É caracterizada por cursos recorrentes de colapso total ou parcial das vias aéreas superiores durante o sono, é um estado de distúrbio do sono que se tornou um problema de saúde pública significativo ao longo do tempo (LIU et al., 2020).

Durante um evento de AOS, o fluxo de ar cessa por um tempo, usualmente 20-40 segundos. Um evento de apneia é, geralmente, acompanhado de redução do oxigênio no sangue, e ao final do episódio, a respiração pode ser rápida por um período para absorver mais oxigênio. De acordo com Ng et al. (2007), os pacientes com AOS podem enfrentar muitos problemas de saúde graves, incluindo problemas cardiovasculares, sonolência excessiva durante o dia, fadiga, depressão, irritabilidade, dificuldades de aprendizado e memória. Quase 4% dos homens de meia idade e 2% das mulheres de meia idade são afetados pela AOS (Hassan, 2016). Ainda, Liu et al. (2020) mencionam que a obesidade, idade e sexo parecem ser três dos fatores de risco mais importantes associados com AOS.

Para Song et al. (2016), a gravidade da AOS é geralmente medida pelo número de eventos de apneia e hipopneia por hora durante o sono; esse parâmetro é conhecido como índice de apneia-hipopneia (IAH). Tradicionalmente, o diagnóstico da AOS é realizado por médicos especialistas, com base na observação visual do sinal da polissonografia. A polissonografia é a maneira mais comum de diagnosticar a AOS (Ng et al., 2007), necessitando que o paciente durma durante a noite em um laboratório especial, onde diversos eletrodos fisiológicos estão conectados às suas pernas, torso, mãos, cabeça e rosto.

A apneia do sono tem tratamento, porém cerca de 90% dos pacientes não são diagnosticados e, portanto, não são tratados, segundo Xie e Minn (2012).

2.2 Exame de Polissonografia

A apneia do sono é, geralmente, diagnosticada através de um exame do sono chamado polissonografia (PSG). Este exame envolve monitoramento noturno dos sinais fisiológicos do paciente, como eletrocardiograma (ECG), eletroencefalograma (EEG), eletro-oculograma (EOG), eletromiograma (EMG), fluxo de ar, sinal de saturação periférica de oxigênio (SpO2) do oxímetro de pulso, sinais de movimento torácico e abdominal, etc., em um laboratório do sono no hospital (Gutta et al., 2018). Uma informação importante que pode ser obtida com este exame, é o intervalo RR, que segundo Almazaydeh, Elleithy e Faezipour (2012), é o intervalo de tempo de uma onda R para a próxima onda R. Também pode ser definido como variações cíclicas na duração de um batimento cardíaco.

O exame de polissonografia tem um alto custo, consome bastante tempo, em geral 8 horas de sono em uma clínica, mais o tempo de preparo para o exame, e é inconveniente para o paciente, que precisa dormir com vários eletrodos fisiológicos ligados ao seu corpo. Além disso, segundo Gutta et al. (2018), a disponibilidade limitada de laboratórios e especialistas em sono leva a maiores tempos de espera para diagnóstico e tratamento da apneia do sono.

Para a aquisição dos sinais eletroencefalográficos, utilizou-se o sistema de poligrafia digital POLIWIN da empresa EMSA, que funciona através de um microcomputador ligado a um equipamento de amplificação analógica e que permite a aquisição de até 32 canais, de acordo com Tafner (1999). As informações dos sinais fisiológicos gerados a partir do PSG geralmente são salvas em um arquivo no formato EDF que é aberto e visualizado no software do Poliwin, que é o software empregado na clínica na qual está sendo aplicado este trabalho. Nesta ferramenta é possível analisar o exame de polissonografia e visualizar todos os dados fisiológicos do paciente coletados durante o exame.

Nesse aplicativo, os técnicos fazem a análise do sinais em intervalos de aproximadamente quinze segundos e anotam onde ocorrem eventos de apneia, baseados no seu próprio conhecimento sobre o tema. Em geral, uma noite de sono no laboratório tem 8 horas, então, conforme a equação 2.1 onde a variável “x” que indica o número de horas de sono do exame, o técnico deve fazer a análise de, no mínimo, 1920 intervalos de quinze segundos.

$$f(x) = \frac{(x * 60 * 60)}{15} \tag{2.1}$$

2.3 *Machine Learning*

Machine learning (ML), ou Aprendizado de Máquina, é uma área dentro da grande área de inteligência artificial e é um tópico de computação muito popular e importante. O ML mostra resultados promissores e aplicações úteis (Pombo; Garcia; Bousson, 2017) quando usado na área médica. Esta é uma técnica em que computadores e algoritmos reconhecem padrões e aprendem recursos importantes dos dados de entrada, ajudando a prever resultados com base no que foi aprendido. O ML é dividido em três categorias principais: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço.

Em um modelo de aprendizado supervisionado, um conjunto de dados de entradas e saídas é fornecido e a máquina aprende como eles estão relacionados. Nesse modelo, o algoritmo pode prever a saída de um novo dado com base no que foi aprendido. O método de aprendizado não supervisionado cria grupos entre os objetos em um conjunto de dados, identificando sua similaridade, usando-os para classificar as incógnitas. O aprendizado por reforço permite que um sistema aprenda com as experiências que obtém ao interagir com seu ambiente, em um processo contínuo.

Um grande número de artigos que estão sendo publicados na literatura de engenharia biomédica descreve o uso de técnicas de ML para desenvolver classificadores para detecção ou diagnóstico de doenças, de acordo com Foster, Koprowski e Skufca (2014).

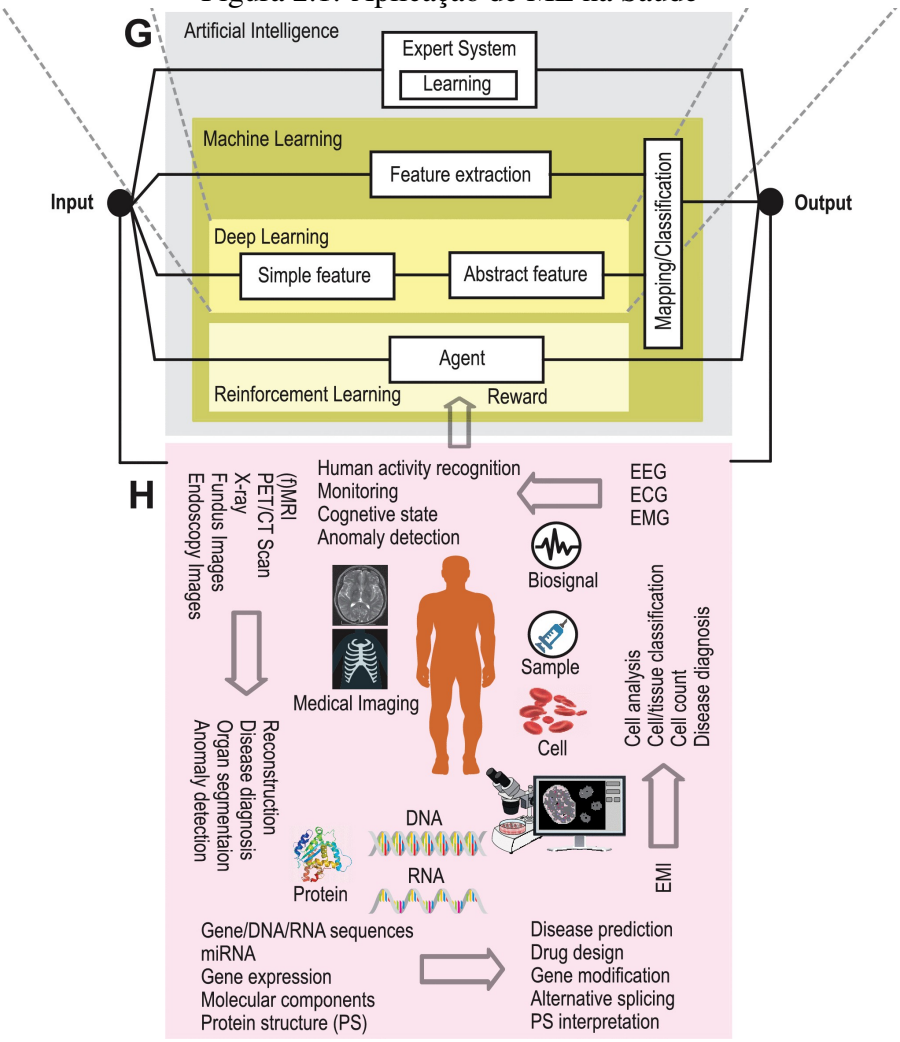
A Figura 2.1 mostra uma visão esquemática de como as técnicas de ML são usadas em aplicações médicas. O ML extrai *features* de dados principalmente por meio de modelagem estatística, fornecendo uma saída preditiva quando aplicado a dados desconhecidos. O *Deep Learning* (DL) é uma subdivisão do ML e extrai mais *features* abstratas de um conjunto maior de dados de treinamento de maneira hierárquica, semelhante ao princípio de funcionamento do nosso cérebro.

Neste trabalho utilizou-se, para os experimentos, algoritmos de aprendizado supervisionado. Na seção a seguir, serão apresentados mais detalhes sobre os algoritmos adotados

2.3.1 *Support Vector Machine*

Support Vector Machine (SVM) é um algoritmo de aprendizado supervisionado. De acordo com Braun, Weidner e Hinz (2011) o SVM foi projetado para resolver proble-

Figura 2.1: Aplicação de ML na Saúde



Fonte: Adaptado de Mahmud et al. (2018)

mas de classificação de grandes margens como uma implementação da teoria de aprendizagem estatística. Esse modelo busca prever a classe a qual o conjunto de dados de entrada pertence, podendo assim ser considerado um modelo de classificação linear binário.

Para Lu, Meng e Cao (2010), o objetivo do SVM é encontrar um hiperplano ideal para separar a classe positiva e a classe negativa com a maior margem. No entanto, com um grande número de amostras de treinamento, o SVM é muito demorado e a memória pode explodir, reduzindo o número de amostras de treinamento é significativo.

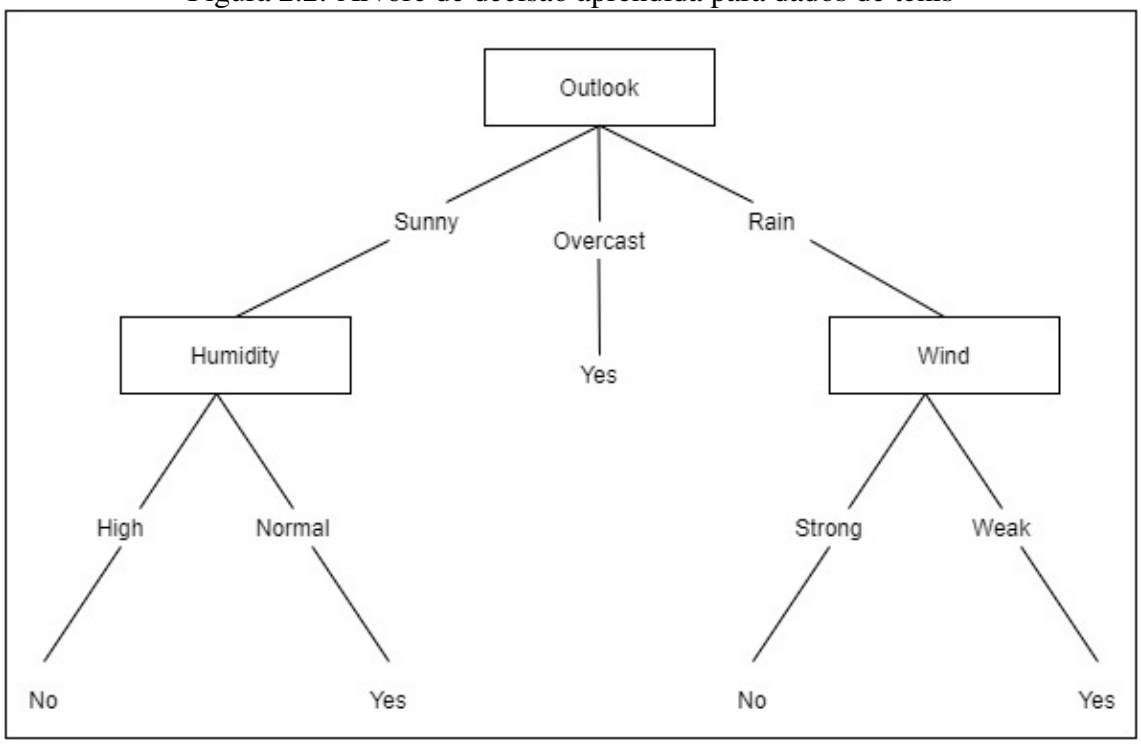
2.3.2 *Decision Tree*

De acordo com Yang (2019), árvores de decisão têm sido amplamente reconhecidas como uma metodologia de mineração de dados e aprendizado de máquina que recebe

um conjunto de valores de atributos como entrada e gera uma decisão booleana como saída. Este algoritmo funciona semelhante a uma árvore binária, onde iniciando por um nodo raiz segue uma sequência de dados até que seja alcançado um nodo folha, definindo um resultado booleano.

A Figura 2.2 mostra um exemplo de árvore de decisão, a qual inicia testando o atributo raiz (outlook, neste caso) e classifica para o ramo apropriado, seguindo este procedimento até chegar ao nó folha, o qual contém a saída predita. Existem dois tipos principais de árvores de decisão: de classificação e e regressão. A árvore de classificação é utilizada quando a variável *target* é categórica. A árvore de regressão é utilizada em casos onde a variável *target* pode assumir valores contínuos, como números reais.

Figura 2.2: Árvore de decisão aprendida para dados de tênis



Fonte: Adaptado de Gavankar e Sawarkar (2017)

2.3.3 AdaBoost

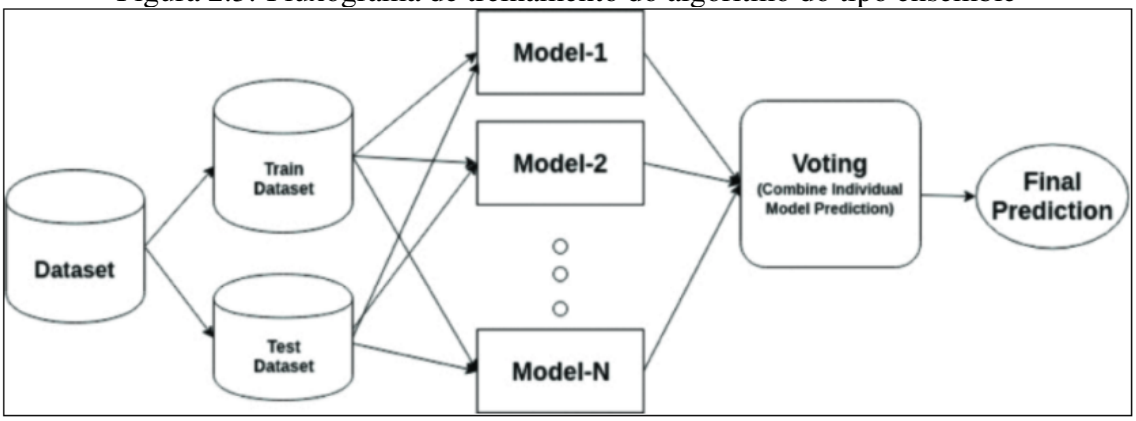
De acordo com An e Kim (2010) classificador *emsemble* consiste em vários classificadores individuais. *Bagging* e *boosting* são métodos bem-sucedidos para gerar classificadores de membros do classificador de *ensemble* porque os algoritmos geram um conjunto de classificadores de membros diversos. Ainda de acordo com o autor, o algo-

ritmo AdaBoost é um dos algoritmos famosos para fazer um classificador de conjunto selecionando classificadores de membros fracos.

O algoritmo AdaBoost gera um classificador forte combinado com vários classificadores fracos. AdaBoost encontra uma combinação de classificador fraco com o ajuste de peso por meio do processo de repetição, e o conjunto de dados de treinamento original não é alterado. Uma das razões pelas quais o algoritmo AdaBoost realiza bons resultados é a diversidade entre classificadores fracos. De forma geral, o AdaBoost gera um classificador forte com classificadores fracos.

A Figura 2.3 representa o fluxograma de treinamento de algoritmos do tipo *ensemble*. O conjunto de dados é dividido entre treino e teste, e são executados quantos modelos forem parametrizados no algoritmo. Ao final da execução de cada modelo, é feita a combinação dos resultados a fim de encontrar a predição final.

Figura 2.3: Fluxograma de treinamento do algoritmo do tipo ensemble



Fonte: Adaptado de Yue e Yang (2020)

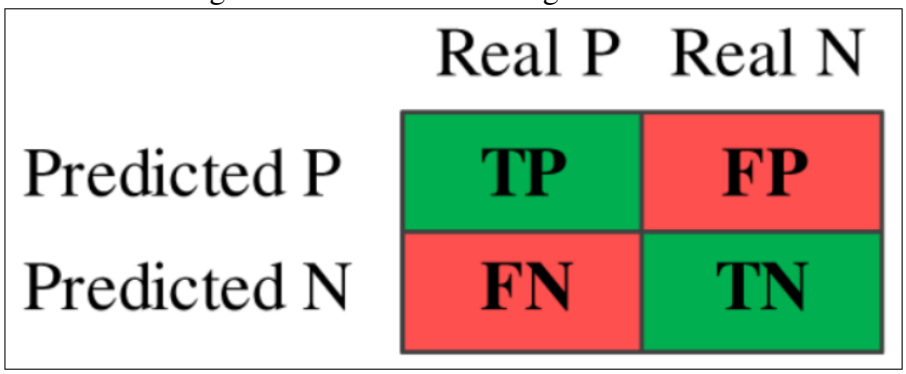
2.3.4 Avaliação de modelos

A Figura 2.3.4 apresenta a tabela de contingência, também conhecida como matriz de confusão. As células em verde representam as predições corretas, já as células em rosa representam as predições incorretas. Considerando as classes citadas, os autores Gharib e Bondavalli (2019) afirmam que várias medidas para avaliar o desempenho de algoritmos de ML têm sido usadas na literatura. Por exemplo, recall e precisão se concentram principalmente no número de predições positivas corretas, ou seja, eles têm pouca ênfase em previsões incorretas.

De acordo com Wardhani et al. (2019), acurácia, *F1 Score* e *g-mean* foram mé-

tricas amplamente usadas para medir o desempenho do classificador no aprendizado de máquina. Para Gharib e Bondavalli (2019), a métrica de *recall* indica a proporção de casos de verdadeiro positivo (TP) que são corretamente preditos positivos, conforme Equação 2.2. A métrica *precision* denota a proporção de casos positivos previstos corretamente em relação a todos os exemplos classificados como positivos, como pode-se observar na Equação 2.3. Por fim, a métrica *F1 score*, tem como objetivo combinar medidas de *precision* e *recall* em uma única medida de "eficácia" de pesquisa.

Figura 2.4: Tabela de contingência binária



Fonte: Adaptado de Gharib e Bondavalli (2019)

$$\textit{Recall} = \frac{TP}{TP + FN} \tag{2.2}$$

$$\textit{Precision} = \frac{TP}{TP + FP} \tag{2.3}$$

$$\textit{F1 score} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.4}$$

3 TRABALHOS RELACIONADOS

Nesta seção, são descritos os métodos de pesquisa utilizados para encontrar artigos relacionados à detecção de AOS usando modelos matemáticos/probabilísticos. Também descreveremos os métodos e resultados dos trabalhos mais relevantes que encontramos sobre o este assunto.

A busca baseou-se no Google Scholar e nas seguintes bibliotecas digitais: IEEE, Pubmed ACM e Science Direct. Como o tema detecção da apneia do sono com modelos matemáticos é muito grande, foi necessário restringir o escopo dos artigos pesquisados. O texto que utilizamos para a pesquisa foi "Detecção de apneia" e "modelo matemático" e também a expressão em inglês "*Apnea Detection*" e "*mathematical models*", e com elas foram encontrados em torno de quinze artigos.

Para cada artigo foi realizada a leitura de seu resumo para verificar se o mesmo contemplava o assunto da pesquisa. Lendo o resumo e os resultados de cada um dos artigos, os mesmos foram classificados pelos critérios de "ano de publicação", "número de citações" e "acurácia".

A Tabela 3.1 contém um resumo dos artigos mais relevantes que foram encontrados nas pesquisas. A coluna "Citação" mostra o número de citações baseadas no Google Scholar, a coluna "Método" representa o método de ML utilizado no artigo e, a última, representa a "acurácia" do método, isto é, a sua taxa de acerto na classificação de novas instâncias. Cabe salientar que os trabalhos relacionados estudados trouxeram como métrica de avaliação apenas a acurácia, não apresentando outros indicadores.

Tabela 3.1: Trabalhos Relacionados

Autor(es)	*Ano*	*Citações*	*Método*	*Acurácia*
Song et al.	2016	45	Hidden Markov model (HMM)	97.1%
Chung et al.	2014	103	STOP-Bang questionnaire	87%
Gutta e Cheng	2016	2	Nonlinear least square optimization	-
Delibasoglu, Avci e Akbas	2011	2	Wavelet decompositions	96.6%
Pombo, Garcia e Bousson	2017	2	Systematic Review	-
Mendez et al.	2007	76	K-nearest neighbor (KNN)	85%
Hassan	2016	55	Normal inverse Gaussian (NIG)	85%
Mansour et al.	2002	32	Polynomial function	$\sim$99%
Ng et al.	2007	38	Bispectral Analysis	77%
Gutta et al.	2018	2	Vector-valued Gaussian	85%
Almazaydeh, Elleithy e Faezipour	2012	19	Support Vector Machine	96.5%
Sadr e de Chazal	2016	4	Extreme Learning Machine Classifier	76.4%
Gutiérrez-Tobal et al.	2016	30	Ada Boost	86.5%

Fonte: O Autor

A maior parte dos trabalhos relacionados é baseada na extração de *features* do sinal de ECG como intervalo RR, amplitude, duração. Esse sinal é muito utilizado na detecção de AOS, pois fornece evidências fisiológicas da presença de AOS. Outro sinal importante é a saturação periférica de oxigênio (SpO2), que está relacionada ao sinal de respiração. Durante um episódio de apneia, o valor de SpO2 cai abaixo de um certo nível (Gutta; Cheng, 2016).

Song et al. (2016) propuseram uma abordagem usando HMM aplicados ao sinal de ECG. Os dados usados em seu trabalho foram retirados do banco de dados PhysioNet, que é uma biblioteca pública de fisiologia baseada na web.

Por outro lado, Hassan (2016) propôs um método para detecção de AOS usando parâmetros Gaussianos inversos normais (NIG) e reforço adaptativo com um único canal, sinal de ECG. No método proposto por Hassan, o sinal de ECG foi decomposto na base de 1 minuto, conforme mostrado na 3.1. Para Hassan, a motivação do uso do TQWT (*Tunable-Q wavelet transform*) no método proposto decorre de várias vantagens do TQWT e seu amplo uso para a análise de vários sinais fisiológicos. Hassan (2016) menciona ainda que a triagem manual, devido à sua natureza demorada, está fadada ao fracasso em tais situações. Com isso, fica claro que é necessário um método automatizado para diagnosticar AOS.

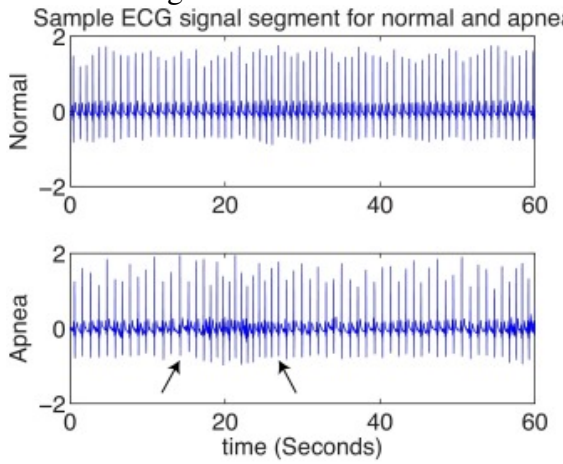
Mendez et al. (2007) propuseram um trabalho usando o Modelo Autorregressivo. O estudo usou o banco de dados do site Physionet, um banco de dados PSG público. O método derivou sinais como área R, intervalos RR, que foram extraídos de um conjunto de recursos que foram usados para o algoritmo de reconhecimento de padrões. Ainda, de acordo com Mendez et al. (2007), os intervalos RR mostram oscilações características (braditaquicardia) durante um evento de apneia, esse padrão produz uma frequência muito baixa no espectro do sinal que poderia ajudar a encontrar uma apneia.

Outro sinal importante do PSG é o sinal do ronco. Ng et al. (2007) propuseram um método para detectar AOS baseado em sinais de ronco. Eles usaram um sistema de aquisição robusto para adquirir sons de ronco, uma vez que o disco poderia estar contaminado com sons de fundo e interferências eletromagnéticas. A análise bi-espectral foi realizada com base no método direto que usa o Algoritmo da Transformada Rápida de Fourier para reduzir o tempo de cálculo para estimar o bi-espectro. O sinal do ronco foi classificado em dois tipos, apneico ou benigno, com base nos resultados clínicos reais.

Por fim, o modelo proposto por Almazaydeh, Elleithy e Faezipour (2012) concentra-se em um algoritmo de classificação automatizado que processa períodos de curta duração

dos dados do eletrocardiograma (ECG). A técnica de classificação utilizada foi o SVM que foi treinado com dados com e sem apneia. Esta técnica mostrou-se muito efetiva, alcançando até 96,5% de acurácia. Esse foi o modelo que motivou este trabalho.

Figura 3.1: Amostra de segmentos de sinal de ECG normais e apneicos

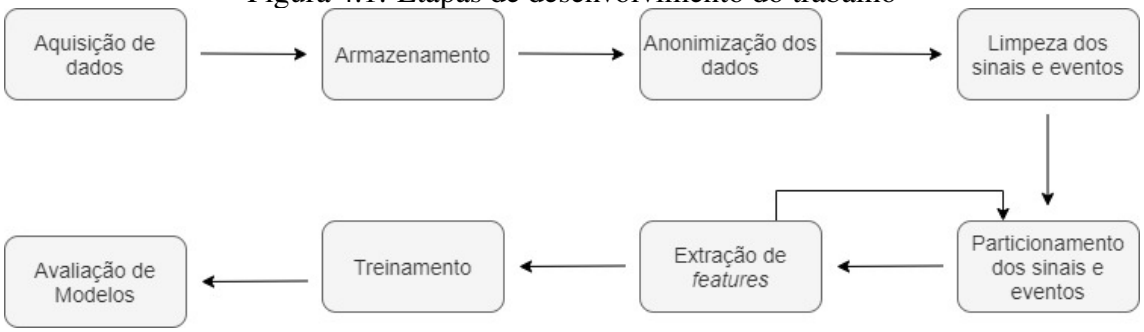


Fonte: Hassan (2016)

4 METODOLOGIA

Nesta seção serão descritos os materiais e a metodologia que foram usados para realizar os experimentos. Na Figura 4.1 é mostrado um diagrama esquemático das etapas deste trabalho, desde a obtenção do conjunto de dados até a etapa de treinamento do algoritmo.

Figura 4.1: Etapas de desenvolvimento do trabalho



Fonte: O Autor

4.1 Aquisição dos dados

Nos trabalhos que foram usados como referência, observou-se que a maioria deles utiliza dados do repositório digital *PhysioNet*, que é uma plataforma que fornece dados publicamente. Para o nosso trabalho, utilizamos dados reais de exames realizados na Clínica do Sono, localizada na cidade de Porto Alegre, que nos forneceu os dados. Ao total foram processados 51 exames, 35 de pessoas do sexo masculino e 16 do sexo feminino. Optou-se por utilizar os dados fornecidos pela clínica pelo fato de que estes foram disponibilizados como *raw data*, o que neste caso foi importante para fazer uma análise completa dos sinais, visto que os sinais disponibilizados pelo *PhysioNet* já haviam sido filtrados e tratados.

4.2 Armazenamento e anonimização de dados

Após aquisição dos dados junto à clinica, os mesmos foram armazenados em repositórios online privados, de modo a garantir a segurança e integridade, visto que são dados sigilosos de pacientes. Para preservar a identidade dos pacientes, foi feita uma etapa de anonimização, onde aplicamos funções próprias da linguagem Python, disponíveis na bi-

blioteca “mne” [1], para anonimizar os dados antes de prosseguir com o pré-processamento.

4.3 Limpeza dos sinais e eventos

Com os dados armazenados foi iniciada a etapa de limpeza dos sinais e eventos. Os dados que recebemos da clínica, vem no formato de arquivo EDF [2] e não possuem nenhum tipo de tratamento, recebemos exatamente os dados extraídos dos eletrodos anexados aos pacientes durante o exame. Pelo fato dos sinais serem obtidos através de eletrodos podem haver ruídos devido a superfície de contato com o corpo do paciente, suor e movimentos bruscos podem causar esses ruídos.

Os arquivos em formato EDF contêm os sinais coletados, informações relativas ao paciente, que foram anonimizadas, e frequência dos sinais, como pode ser observado na Figura 4.2. Essa informação é necessária para a etapa de limpeza e particionamento dos sinais. Pode-se perceber que houve uma suave alteração dos sinais.

Figura 4.2: Informações do exame de polissonografia

```
<Info | 9 non-empty values
 bads: []
 ch_names: OC1, OC2, EEG FR, EEG OCI, EEG A1, EEG A2, Quei, Card, Tib_E, ...
 chs: 20 EEG
 custom_ref_applied: False
 description: Anonymized using a time shift to preserve age at acquisition
 experimenter: mne_anonymize
 highpass: 0.0 Hz
 lowpass: 50.0 Hz
 meas_date: 2000-01-01 00:00:00 UTC
 nchan: 20
 projs: []
 sfreq: 100.0 Hz
>
```

Fonte: O Autor

4.3.1 Pré-processamento de sinais

Nesta seção será descrito como foi feito o pré-processamento dos sinais, visto que os mesmos foram recebidos sem nenhum tratamento prévio.

Para o treinamento dos modelos, foram selecionados os mesmos sinais que são utilizados na clínica para o diagnóstico manual, que são: fluxo aéreo, cinta torácica, cinta

[1] Biblioteca *open-source* para explorar, visualizar e analisar dados neurofisiológicos humanos. Mais detalhes em: <https://mne.tools/stable/index.html>

[2] Formato de arquivo simples e flexível para troca e armazenamento de sinais biológicos e físicos. Mais detalhes em: <https://www.edfplus.info/>

abdominal, saturação de oxigênio (SpO2) e frequência cardíaca. O fluxo aéreo é medido com sensores nasais e tem relação direta com o sinal de SpO2. Segundo Gutta e Cheng (2016) o sinal SpO2 está relacionado ao sinal de respiração. Por exemplo, durante os episódios de apneia do sono, a falta de fluxo de ar por um determinado período diminui a quantidade de oxigênio disponível nos pulmões e o valor de SpO2 cai abaixo de um determinado nível. As cintas abdominal e torácica são utilizadas para medir o esforço respiratório, que durante um evento de apneia pode cair significativamente.

Para iniciar, fez-se necessário realizar uma limpeza nos sinais, pois no início dos exames há geralmente uma etapa de calibragem dos equipamentos, e os sinais desse intervalo de tempo não são úteis pois estão distorcidos. Durante o exame também podem aparecer ruídos causados por algum movimento brusco do paciente.

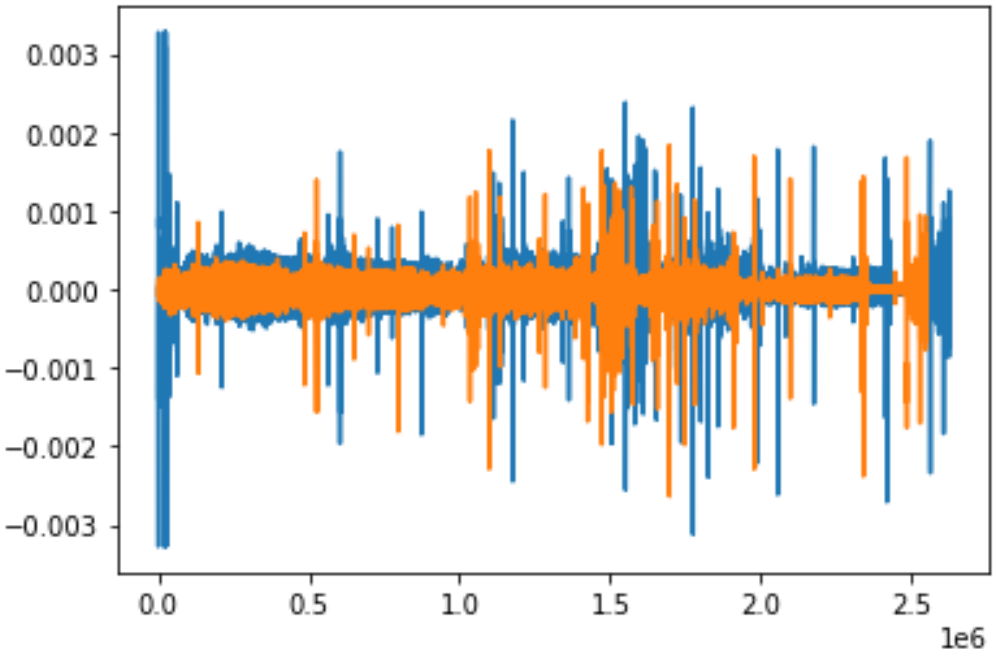
O arquivo de eventos é onde encontram-se todas as anotações feitas pelos técnicos que fazem o diagnóstico da AOS, e também eventos que são anotados automaticamente pelo *software* Poliwin, aqui incluem-se os eventos de início e fim do sono, estagiamento do sono, entre outros. Esses eventos contém informações do tempo de início, duração, horário, entre outros.

Visando aplicar a primeira etapa de limpeza dos sinais, todos os valores antes do evento de início do sono e após ao evento de final do sono foram excluídos da análise. Os dados antes do evento de início do sono, geralmente, são dados de calibragem do equipamento. E os dados após o evento de fim do sono são dados que quando o paciente já está despertando e que podem conter muito ruído devido a movimentos que o paciente faz.

Com os sinais limpos, foi aplicado o filtro de Savitzky–Golay, também conhecido como filtro de suavização de mínimos quadrados, o qual, segundo Uddin et al. (2016), é usado para suavizar funções do modo intrínseco de ruído dominante. Isso é obtido, com o processo de convolução, ajustando subconjuntos sucessivos de pontos de dados adjacentes com um polinômio de baixo grau pelo método dos mínimos quadrados lineares.

Na Figura 4.3 pode-se observar o sinal original e o resultado após o tratamento. Nesse caso, usou-se como exemplo o sinal da cinta torácica. Na cor azul encontra-se o sinal original, e, na cor laranja, está o sinal após a aplicação dos filtros de pré-processamento.

Figura 4.3: Sinal da cinta torácica original e após tratamento



Fonte: O Autor

4.3.2 Pré-processamento do diagnóstico

Neste trabalho todas os tipos de apneia foram classificados apenas como uma classe positiva.

Os dados do diagnóstico também foram pré-processados. Foram disponibilizados arquivos no formato XML com os eventos de apneia anotados pelos técnicos da clínica. Para obter os dados de diagnóstico, foram filtrados apenas eventos que indicassem a ocorrência de apneia, e, desses eventos, foram lidas as informações de tempo de início e duração. Os eventos que indicam apneia podem ser classificados em:

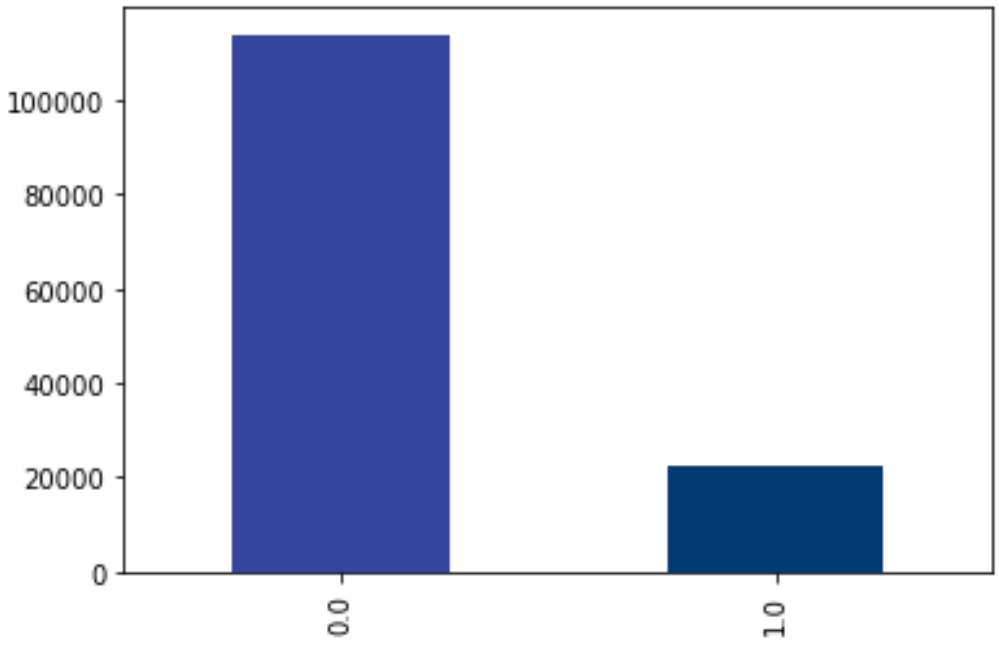
- rera: despertar relacionado a esforço respiratório
- apnobst: apneia obstrutiva
- hipopn: hipopneia
- apcent: apneia central

Após feita a seleção dos eventos, realizou-se a remoção dos sinais antes do evento de início do sono e após o evento de fim do sono, e aplicou-se o filtro de Savitzky–Golay.

4.3.3 Dados Desbalanceados

Um conjunto de dados é classificado como desbalanceado quando há uma desproporção entre o número de dados de uma classe em relação a outras. Segundo Moturu, Johnson e Liu (2010), dados desequilibrados são comumente observados em muitas aplicações, como detecção de fraude de cartão crédito, detecção de intrusão de rede, gerenciamento de risco de seguro, classificação de texto e diagnóstico médico. Ainda segundo Moturu, Johnson e Liu (2010) a maioria dos algoritmos de classificação assume que a distribuição de classes é uniforme. A Figura 4.4 mostra o desbalanço que temos entre a classe 0, intervalo sem apneia, e a classe 1, intervalo com apneia. Nesta aplicação, a proporção de entre as classes era de aproximadamente 5,14 elementos da classe 0 (sem apneia) para cada elemento da classe 1 (com de apneia).

Figura 4.4: Dados de diagnóstico desbalanceados



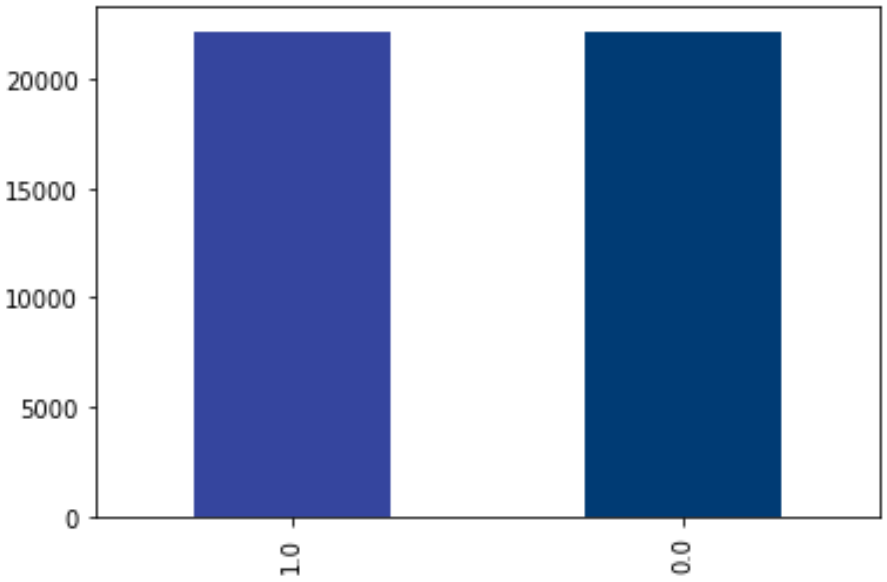
Fonte: O Autor

Para os autores Tripathi, Batra e Pandey (2019), se treinarmos nosso classificador ou modelo com dados desequilibrados, observamos a previsão parcial da classe majoritária, o que causa acurácia enganosa. De acordo com Moturu, Johnson e Liu (2010), as duas soluções mais comuns para esse problema incluem amostragem não aleatória *under-sampling* ou *over-sampling*. Para tratar o problema do desbalanceamento das classes, usou-se a técnica do *under-sampling*. Essa técnica elimina de forma aleatória entradas da classe com maior número de ocorrências. Na Figura 4.5 temos um exemplo da técnica

under-sampling onde diminuímos a amostra da classe de maior número para que ela fique com o mesmo número de amostras da classe menor.

Foi escolhida a técnica *under-sampling* pois o *over-sampling*, cria, de forma randômica, instâncias adicionais para a classe menor com base nos valores existentes. De acordo com Drummond e Holte (2003), o custo computacional da utilização da técnica *over-sampling* não se justifica, pois o desempenho obtido é, na melhor das hipóteses, o mesmo que o da técnica *under-sampling*.

Figura 4.5: Dados de diagnóstico após aplicação de *random under-sampling*



Fonte: O Autor

4.4 Particionamento dos sinais

Com os dados tratados e com os ruídos removidos, iniciamos a etapa de extração das *features*. Para a extração o sinal foi particionado em 5 épocas distintas:

- época de 2s;
- época de 5s.
- época de 10s;
- época de 15s;
- época de 30s;

As épocas de 2 e 5 segundos, foram criadas pois as mesmas são usadas para análise pelo médico especialista da clínica. Já as demais foram propostas pelos autores Al-

mazaydeh, Elleithy e Faezipour (2012), que que explicam que a apneia é uma pausa na respiração por um intervalo de, no mínimo, 10 segundos de duração. Após a definição das épocas, o sinal foi particionado da seguinte forma: a frequência dos sinais é 100Hz, logo, a cada 10ms há um ponto. Para um intervalo de 10s, usamos, então, 1000 pontos.

4.5 Extração de *features*

As *features* são definidas como características ou atributo do conjunto de dados. As *features* selecionadas neste trabalho foram escolhidas de forma diferente dos trabalhos relacionados, visto que a proposta deste trabalho era testar outros sinais e, também, testar diferentes *features*. De cada época foram extraídas as seguintes *features*:

- Média;
- Mediana;
- Desvio Padrão.

Com os sinais particionados e a *features* extraídas de cada época, obtivemos o modelo de dados para ser aplicado nos algoritmos de *machine learning*. Foram realizados treinamentos com distintas épocas, como listado anteriormente, e os resultados serão apresentados a seguir, no próximo capítulo.

5 EXPERIMENTOS

Neste capítulo serão descritos os experimentos realizados para avaliar os modelos a fim de chegar em um com as melhores métricas para solucionar o problema proposto. Com os dados particionados em épocas, dividimos os mesmos em variáveis dependentes e independentes. As variáveis dependentes representam os sinais fisiológicos captados no laboratório. As variáveis independentes são representadas pelas anotações de eventos de apneia realizada pelos técnicos do laboratório.

Para os experimentos foram selecionados 2 algoritmos: SVM e Ada Boost com árvore de decisão. As escolhas dos algoritmos se deu com base nos trabalhos relacionados estudados, e que indicaram boa acurácia com esses algoritmos. Cada modelo foi executado com os seguintes conjuntos de dados:

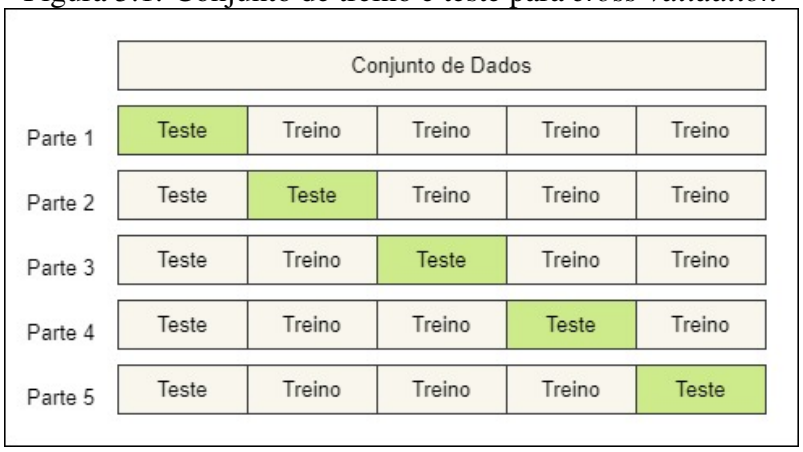
- Amostra 1: dados desbalanceados;
- Amostra 2: dados com *under-sampling*.

Inicialmente foram realizados experimentos usando o método *train_test* ou *hold-out*. De acordo com Yadav e Shukla (2016), nesse método os dados são divididos em duas partes não sobrepostas e essas duas partes são usadas para treinamento e teste, respectivamente. Neste trabalho utilizou-se a proporção 80% de dados para treino e 20% dos dados para testes. A seguir, pode-se observar os resultados obtidos utilizando esse método.

Como validação complementar ao método acima, outros experimentos foram realizados utilizando o método de *cross-validation* com *stratified* K*-Fold*. De acordo com Yadav e Shukla (2016), no K*-Fold cross-validation*, os dados são divididos em k partes iguais e o treinamento é feito com $k - 1$ partes, e uma parte é deixada de fora para teste, como pode ser observado na Figura 5.1. O processo é repetido k vezes enquanto muda a parte de teste uma a uma. Esse método aumenta consideravelmente o tempo de treinamento do modelo.

Neste trabalho, foi utilizado o método k*-Fold* estratificado com 5 *splits*. Esse método é semelhante ao k*-fold*, mas garante que cada classe será representada com a mesma distribuição em todas as k partes. De acordo com Yadav e Shukla (2016), a estratificação é o processo de reorganizar os dados de forma que cada *fold* seja um bom representante do todo.

Figura 5.1: Conjunto de treino e teste para *cross-validation*



Fonte: O Autor

Nos trabalhos pesquisados, a grande maioria utilizou como métrica de avaliação apenas a acurácia. Neste trabalho, as métricas analisadas foram:

- Acurácia;
- Precisão;
- *Recall*;
- F1 *Score*.

Foram analisadas outras métricas, além da acurácia, pelo fato de que trabalhamos com classes desbalanceadas e utilizar somente a acurácia como métrica de comparação pode apresentar resultados não realistas.

5.1 Época de 2s

A Tabela 5.2 mostra o resultado das avaliações feitas com época de 2s utilizando *cross-validation*. Comparando-a com a Tabela 5.1, que utilizou o método *train-test*, observa-se que houve uma queda considerável da acurácia, ao passo que o *F1-Score* é maior nos casos em que utiliza-se dados balanceados com o método *under-sampling*.

A acurácia maior no método *train-test* pode ser justificada pelo fato de que esse modelo executa o algoritmo apenas uma vez, com um conjunto de dados fixo. Ao passo que no *cross-validation* o modelo é treinado e testado k vezes e então, a acurácia média é considerada como a acurácia do modelo. Outra justificativa para o resultado superior é que pelo fato do método *train-test* ser executado uma única vez, pode ocorrer de o modelo ter selecionado, randomicamente, um conjunto de dados considerado bom, que

tenha amostras significativas das duas classes. Isso explicou o motivo, de neste modelo, os resultados de cada execução poderem apresentar resultados muito diferentes entre si. Yadav e Shukla (2016) explicam que ,para grandes conjuntos de dados, geralmente é sugerido usar o método *train-test* para reduzir o tempo que leva durante o treinamento do modelo.

Tabela 5.1: Avaliação com *Train-Test* - 2s

Modelo	Acurácia	*Recall*	*Precision*	*F1 Score*
SVM com dados desbalanceados	76%	20%	21%	20%
SVM com *Under-Sampling*	53%	**63%**	18%	28%
ADA Boost com dados desbalanceados	**83%**	25%	45%	32%
Ada Boost com *Under-Sampling*	65%	53%	**71%**	**59%**

Fonte: O Autor

Tabela 5.2: Avaliação com *Cross-Validation* - 2s

Modelo	Acurácia	*Recall*	*Precision*	*F1 Score*
SVM com dados desbalanceados	50%	14%	14%	13%
SVM com *Under-Sampling*	48%	47%	48%	47%
Ada Boost com dados desbalanceados	59%	22%	37%	31%
Ada Boost com *Under-Sampling*	**66%**	**60%**	**71%**	**63%**

Fonte: O Autor

5.2 Época de 5s

A Tabela 5.3 mostra a avaliação das métricas do método *train-test* com época de 5 segundos. Comparando-os com a Tabela 5.4 observa-se que a acuária é maior no modelo *train-test*, no entanto o *F1 Score* é menor. Outra constatação é que o *recall* é maior quando se utiliza dados balanceados com a técnica *under-sampling*. A acurácia maior no modelo com dados desbalanceados justifica-se pelo fato de que, como a proporção entre classes é aproximadamente 5:1, o número de previsões corretas é maior, porém apenas para uma das classes de dados, a que possui quantidade expressiva de dados, que é a classe relacionado com não ter apneia.

Tabela 5.3: Avaliação com *Train-Test*

Modelo	Acurácia	*Recall*	*Precision*	*F1 Score*
SVM com dados desbalanceados	75%	21%	22%	21%
SVM com *Under-Sampling*	41%	51%	13%	21%
Ada Boost com dados desbalanceados	**84%**	16%	**49%**	**35%**
Ada Boost com *Under-Sampling*	64%	**62%**	25%	**35%**

Fonte: O Autor

Tabela 5.4: Avaliação com *Cross-Validation*

Modelo	Acurácia	*Recall*	*Precision*	*F1 Score*
SVM com dados desbalanceados	54%	20%	21%	21%
SVM com *Under-Sampling*	58%	26%	64%	37%
Ada Boost com dados desbalanceados	55%	18%	23%	21%
Ada Boost com *Under-Sampling*	**64%**	**56%**	**68%**	**59%**

Fonte: O Autor

5.3 Época de 10s

A Tabela 5.5 mostra o resultado das avaliação feitas com época de 10s com o modelo *train-test*. Pode-se perceber que os métodos que usam dados desbalanceados possuem menor taxa de *recall*, *precision* e, por consequência, F1 *score*, no entanto a acurácia é maior. Isto porque a métrica de acurácia presume que as classes são uniformes, e como as classes estão desbalanceadas, os resultados positivos da classe de maior proporção se destacam em relação aos resultados da classe de menor proporção.

Tabela 5.5: Avaliação com *Train-Test*

Modelo	Acurácia	*Recall*	*Precision*	*F1 Score*
SVM com dados desbalanceados	73%	24%	26%	25%
SVM com *Under-Sampling*	63%	47%	68%	55%
Ada Boost com dados desbalanceados	**81%**	25%	53%	34%
Ada Boost com *Under-Sampling*	63%	**52%**	**69%**	**58%**

Fonte: O Autor

Tabela 5.6: Avaliação com *Cross-Validation*

Modelo	Acurácia	*Recall*	*Precision*	*F1 Score*
SVM com dados desbalanceados	50%	18%	17%	17%
SVM com *Under-Sampling*	45%	45%	47%	45%
Ada Boost com dados desbalanceados	61%	28%	38%	36%
Ada Boost com *Under-Sampling*	**67%**	**58%**	**73%**	**63%**

Fonte: O Autor

5.4 Época de 15s

Na Tabela 5.7 observa-se de forma clara que a acurácia é maior quando os modelos são executados com dados desbalanceados, no entanto os valores de precisão, *recall* e *F1 Score* são menores. O conjunto de dados com época de 15 segundos pode distorcer a detecção de AOS pois como um episódio de apneia apresenta no mínimo 10 segundos de duração, este conjunto de dados pode conter mais de um evento no intervalo, porém o mesmo será contado apenas uma vez.

Tabela 5.7: Avaliação com *Train-Test*

Modelo	Acurácia	*Recall*	*Precision*	*F1 Score*
SVM com dados desbalanceados	73%	27%	31%	29%
SVM com *Under-Sampling*	63%	47%	**70%**	52%
Ada Boost com dados desbalanceados	**80%**	32%	54%	40%
Ada Boost com *Under-Sampling*	64%	**51%**	68%	**57%**

Fonte: O Autor

Tabela 5.8: Avaliação com *Cross-Validation*

Modelo	Acurácia	*Recall*	*Precision*	*F1 Score*
SVM com dados desbalanceados	55%	17%	28%	21%
SVM com *Under-Sampling*	42%	43%	43%	42%
Ada Boost com dados desbalanceados	63%	33%	42%	40%
Ada Boost com *Under-Sampling*	**69%**	**60%**	**74%**	**65%**

Fonte: O Autor

5.5 Época de 30s

Por fim, analisando a época de 30 segundos, observa-se que a acurácia foi maior quando utilizado o método *train-test* com dados desbalanceados. Este método também obteve o maior *F1 Score*.

A análise desta época pode ser tendenciosa, pois como é uma época grande, pode haver mais de um evento de apneia dentro de um mesmo intervalo, mas que está sendo considerado apenas como um evento, uma vez que um evento de AOS tem, no mínimo, 10 segundos de duração.

Observa-se também, que a acurácia é maior nos casos utilizando o método *train-test* e, no geral, o *F1 Score* também foi maior. Analisando a acurácia e *F1 Score* esta época foi a que o obteve o segundo melhor resultado.

Tabela 5.9: Avaliação com *Train-Test*

Modelo	Acurácia	*Recall*	*Precision*	*F1 Score*
SVM com dados desbalanceados	62%	19%	**96%**	32%
SVM com *Under-Sampling*	62%	41%	67%	49%
Ada Boost com dados desbalanceados	**73%**	**47%**	92%	**62%**
Ada Boost com *Under-Sampling*	62%	46%	64%	51%

Fonte: O Autor

Tabela 5.10: Avaliação com *Cross-Validation*

Modelo	Acurácia	*Recall*	*Precision*	*F1 Score*
SVM com dados desbalanceados	51%	35%	34%	34%
SVM com *Under-Sampling*	45%	25%	44%	30%
Ada Boost com dados desbalanceados	53%	38%	29%	29%
Ada Boost com *Under-Sampling*	**64%**	**50%**	**67%**	**57%**

Fonte: O Autor

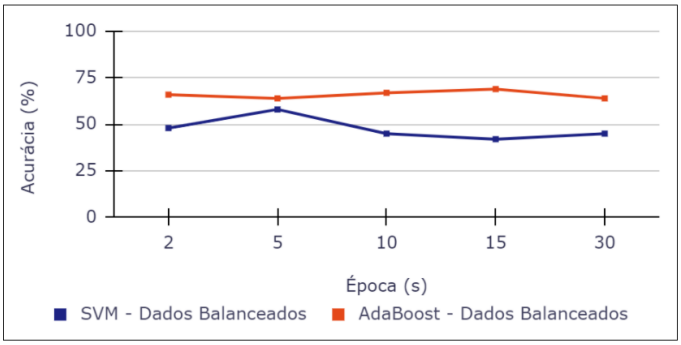
6 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Neste capítulo serão realizadas comparações entre os modelos propostos com o objetivo de encontrar o melhor modelo e os melhores parâmetros para a detecção de AOS.

6.1 Comparação de modelos

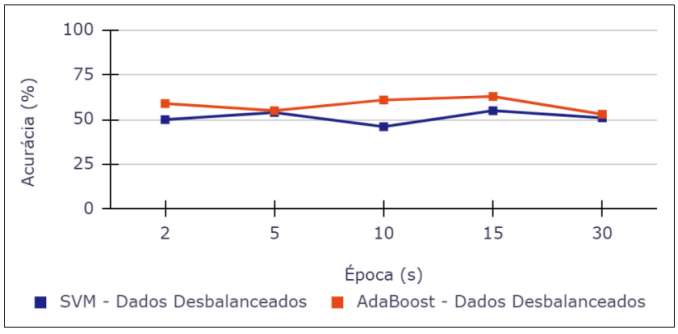
A Figura 6.1 exibe a comparação dos algoritmos SVM e *Ada Boost* utilizando o método *cross-validation* com dados balanceados com a técnica *under-sampling*. Observa-se que a acurácia do algoritmo *ADA Boost* é superior ao do SVM. Pode-se perceber também que executando os mesmos algoritmos, porém com dados desbalanceados, a acurácia decai em ambos os casos, como é observado na Figura 6.2. Outra constatação é que a época de 15 segundos foi a que obteve melhor acurácia nos dois casos.

Figura 6.1: Acurácia com *cross-validation* e dados balanceados



Fonte: O Autor

Figura 6.2: Acurácia com *cross-validation* e dados desbalanceados

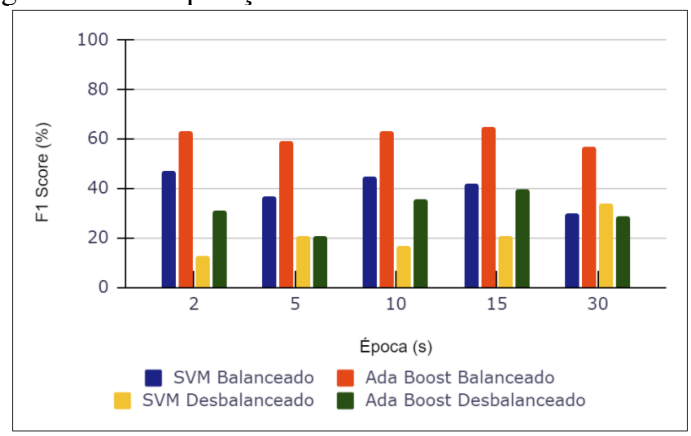


Fonte: O Autor

A Figura 6.3 mostra a comparação da métrica *F1 Score* entre os modelos executados com *cross-validation*. Observa-se que o valor desta métrica para a época de 15

segundos é o maior. Com esta análise temos um primeiro resultado que indica que a época de 15 segundos, quando utilizada com dados balanceados e com o algoritmo *Ada Boost* e com *cross-validation*, tem um desempenho superior se comparado aos demais modelos, tanto analisando a acurácia, observada na Figura 6.1, quanto o *F1 score*.

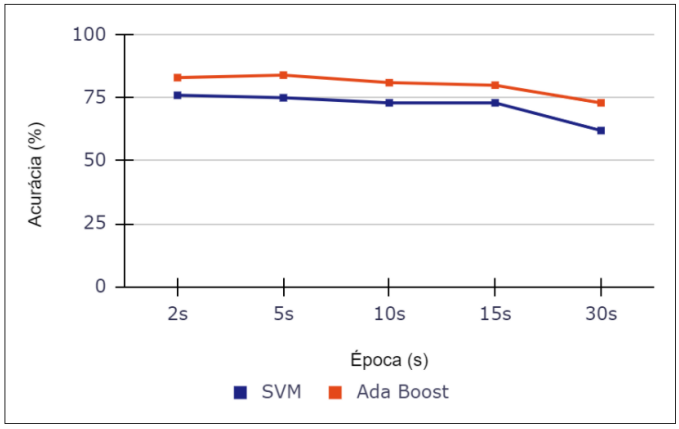
Figura 6.3: Comparação do *F1 Score* com *cross-validation*



Fonte: O Autor

Na Figura 6.4 observa-se os resultados para testes executados com o modelo *train-test* e dados desbalanceados. Nesta análise percebe-se que o valor da acurácia é consideravelmente superior aos analisados utilizando o modelo *cross-validation*, representado na Figura 6.2. Uma justificativa plausível para este fato é que no modelo *train-test* o treinamento é realizado apenas uma vez, assim os dados são escolhidos aleatoriamente uma única vez, o algoritmo é treinado e apresenta resultados de acurácia melhores. Outra informação importante, é que a acurácia do algoritmo Ada Boost é levemente melhor do que a do algoritmo SVM, uma vez que o Ada Boost, no geral, potencializa a performance dos algoritmos.

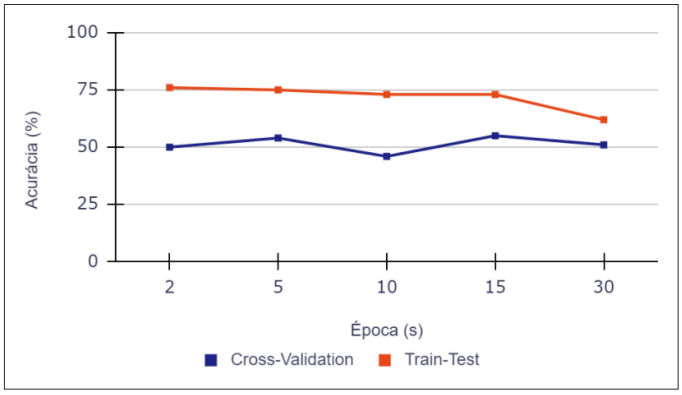
Figura 6.4: Acurácia com *train-test* e dados desbalanceados



Fonte: O Autor

A Figura 6.5 faz a comparação da acurácia do algoritmo SVM com dados desbalanceados nos métodos *cross-validation* e *train-test*. Observa-se que o método *train-test* possui melhor acurácia que o modelo *cross-validation*. No entanto como este modelo foi executado utilizando-se dados desbalanceados, pode haver um interpretação equivocada, visto que o número de instâncias sem apneia é muito superior a classe com apneia, e assim a classe sem apneia é predita certamente muitas vezes, fato este que acaba elevando a acurácia do modelo.

Figura 6.5: Acurácia do método SVM com dados desbalanceados



Fonte: O Autor

Na Tabela 6.1 pode-se observar a matriz de confusão que demonstra este caso. A classe sem apneia é significativamente maior que a classe com apneia e isso faz com que o número de verdadeiros positivos seja alto, e o número de verdadeiros negativos baixo, o que tem impacto na precisão, *recall*, e, consequentemente, no *F1 score*.

A Figura 6.6 apresenta a evolução da acurácia em cada época utilizando o algoritmo SVM e dados debalanceados, e confirma que com o método *train-test* a acurácia é

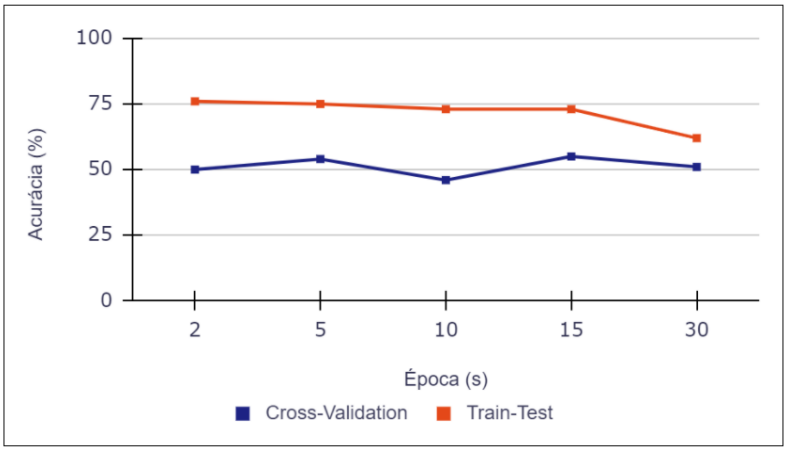
Tabela 6.1: Matriz de confusão SVM com dados desbalanceados

Real	Previsto	
	0 - Sem apneia	1 - Com apneia
0 - Sem apneia	2647	127
1 - Com apneia	47	21

Fonte: O Autor

maior e vai decaindo a medida que as épocas ficam maiores, visto que o volume de dados decresce. No modelo *cross-validation* é calculada a média harmônica da acurácia em cada execução, então observa-se que não segue um padrão, e que os valores são inferiores aos observados no método *train-test*.

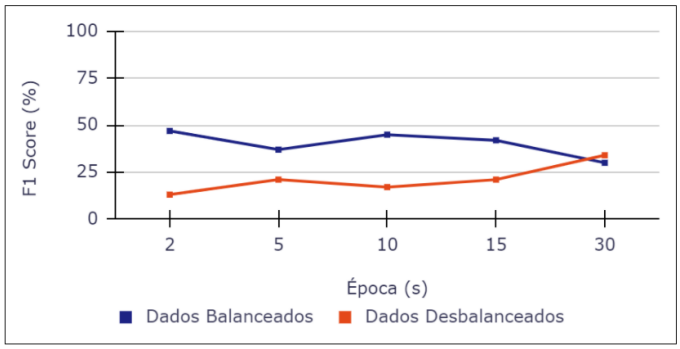
Figura 6.6: Acurácia método SVM com dados desbalanceados



Fonte: O Autor

A Figura 6.7 apresenta a comparação da métrica *F1 score* para o algoritmo SVM. Pode-se observar que quando utilizados dados balanceados, a métrica apresenta um resultado superior em relação a quando utilizado com dados desbalanceados. No entanto, este valor ainda é considerado baixo para esta aplicação. O modelo executado com época de 30 segundos com dados desbalanceados, apresentou um resultado sutilmente maior comparado ao modelo com dados balanceados, isso pode ocorrer pelo fato que o conjunto de dados desta época é menor, e, assim, pode-se obter um resultado mais acurado, mas que pode não ser confiável.

Figura 6.7: *F1 Score* do modelo SVM com *cross-validation*



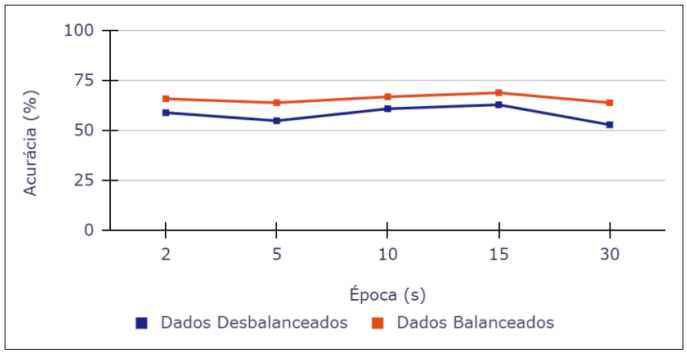
pi

Fonte: O Autor

A Figura 6.8 demonstra o teste executado com o algoritmo Ada Boost com *cross-validation*. É possível inferir que com dados balanceados o modelo apresenta acurácia superior quando comparado ao modelo com dados desbalanceados. Neste modelo, observa-se que a época de 15 segundos, utilizando dados balanceados, é a que apresenta melhor acurácia e também o melhor valor de *F1 score*, como pode ser observado na Figura 6.9.

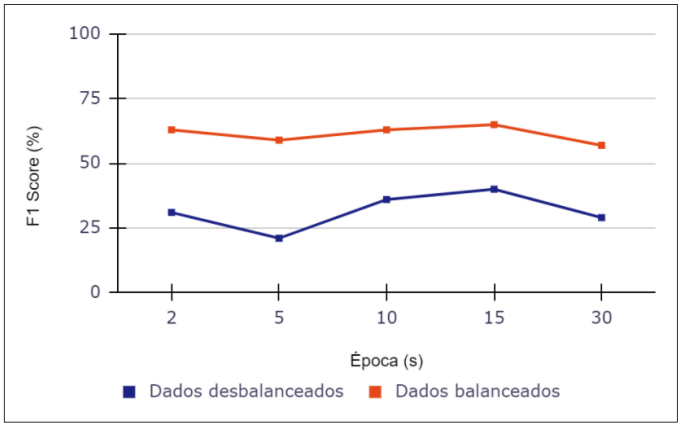
Observando as curvas dos valores obtidos com dados balanceados e com dados desbalanceados, na Figura 6.9, vemos que no último caso há uma variação maior dos valores. Isso é justificado pelo fato de que em cada teste executado com dados desbalanceados a chance de termos um conjunto de treino desbalanceado é significativa. Ao passo que o conjunto de dados balanceados não apresenta este problema.

Figura 6.8: Acurácia do modelo *Ada Boost* com *cross-validation*



Fonte: O Autor

Figura 6.9: *F1 score* do modelo *Ada Boost* com *cross-validation*



Fonte: O Autor

Após análise e interpretação dos resultados acima, pode-se inferir que a época de 15 segundos quando executada o algoritmo Ada Boost com *cross-validation* e com conjunto dados balanceados, foi a que apresentou métricas com melhores resultados onde obtivemos 65 %acurácia, 60% de *recall*, 74% de precisão, e 65% de *F1 score*. Este foi o melhor resultado obtido nos experimentos. Embora os resultados foram abaixo do ideal, pode-se perceber que a época de 15 segundos é ideal para realizar outros experimentos a fim de buscar métricas melhores.

7 CONCLUSÃO

A detecção manual da AOS é um processo que consome tempo e está suscetível a erros humanos. Dessa forma, a detecção automática surge como uma alternativa para diminuir o tempo de diagnóstico e para fornecer um diagnóstico mais confiável. Neste trabalho, o objetivo foi identificar se os sinais utilizados atualmente para a detecção manual, possuem informações suficientes e relevantes para a detecção automática utilizando *machine learning* com modelos estatísticos. Os modelos foram avaliados por meio das métricas de acurácia e *F1 score*.

O desenvolvimento deste trabalho fornece materiais e evidências para aprofundar as pesquisas na detecção de AOS utilizando *machine learning* baseado em modelos estatísticos. Nesta pesquisa buscamos propor modelos que utilizam variados sinais fisiológicos em algoritmos de treinamento, além do eletrocardiograma.

Com a execução e avaliação dos modelos, percebe-se que o método de extração de *features* utilizado não foi a mais adequada, visto que os resultados obtidos foram significativamente abaixo dos resultados encontrados nos trabalhos relacionados, que, em sua ampla maioria, utiliza apenas um sinal fisiológico para a análise. O principal desafio do trabalho foi definir as *features* e os sinais que seriam utilizados. Embora os trabalhos relacionados apresentem as *features* utilizadas, no modelo que propomos, utilizando vários sinais, as mesmas não se aplicaram. Vários experimentos foram realizados com o objetivo de obter o melhor modelo, que apresente as melhores métricas.

Os resultados deste trabalho mostram que a utilização de um conjunto de dados balanceados, com a técnica *under-sampling*, fornece melhores métricas, portanto são mais adequados para esta aplicação. Conclui-se também, que a escolha correta dos dados e das *features* é de extrema importância e decisivo para ter um modelo adequado ou não. Nesta pesquisa, verificamos e validamos, por meio de experimentos, *features* que não contém informação adequada para realizar a detecção automática da AOS com os algoritmos utilizados.

Este trabalho também é importante para a área da saúde, visto que hoje a detecção da AOS ainda é um exame com custo financeiro elevado e isso faz com que poucas pessoas tenham acesso ao diagnóstico e tratamento, especialmente no serviço público de saúde. Iniciativas e pesquisas na área da saúde são uma importante forma de fornecer resultados mais precisos e rápidos e oportunizar que mais pessoas possam ter uma qualidade de vida melhor por meio do tratamento da AOS.

REFERÊNCIAS

Almazaydeh, L.; Elleithy, K.; Faezipour, M. Obstructive sleep apnea detection using svm-based classification of ecg signal features. **34th Annual International Conference of the IEEE EMBS**, 2012.

An, T.; Kim, M. A new diverse adaboost classifier. In: **2010 International Conference on Artificial Intelligence and Computational Intelligence**. [S.l.: s.n.], 2010. v. 1, p. 359–363.

Braun, A. C.; Weidner, U.; Hinz, S. Support vector machines, import vector machines and relevance vector machines for hyperspectral classification — a comparison. In: **2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)**. [S.l.: s.n.], 2011. p. 1–4.

Chung, F. et al. Alternative scoring models of stop-bang questionnaire improve specificity to detect undiagnosed obstructive sleep apnea. **Journal of Clinic Sleep Medicine**, 2014.

Delibasoglu, I.; Avci, C.; Akbas, A. Ecg based sleep apnea detection using wavelet analysis of instantaneous heart rates. p. 5, 2011.

DRUMMOND, C.; HOLTE, R. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats oversampling. **Proceedings of the ICML'03 Workshop on Learning from Imbalanced Datasets**, 01 2003.

FOSTER, K. R.; KOPROWSKI, R.; SKUFCA, J. D. Machine learning, medical diagnosis, and biomedical engineering research - commentary. **J.D. BioMed Eng OnLine**, 2014.

Gavankar, S. S.; Sawarkar, S. D. Eager decision tree. In: **2017 2nd International Conference for Convergence in Technology (I2CT)**. [S.l.: s.n.], 2017. p. 837–840.

Gharib, M.; Bondavalli, A. On the evaluation measures for machine learning algorithms for safety-critical systems. In: **2019 15th European Dependable Computing Conference (EDCC)**. [S.l.: s.n.], 2019. p. 141–144.

Gutiérrez-Tobal, G. C. et al. Utility of adaboost to detect sleep apnea-hypopnea syndrome from single-channel airflow. **IEEE Transactions on Biomedical Engineering**, 2016.

Gutta, S.; Cheng, Q. Modeling of oxygen saturation and respiration for sleep apnea detection. **50th Asilomar Conference on Signals, Systems and Computers**, 2016.

Gutta, S. et al. Cardiorespiratory model-based data-driven approach for sleep apnea detection. **IEEE Journal of Biomedical and Health Informatics**, 2018.

Hassan, A. R. Computer-aided obstructive sleep apnea detection using normal inverse gaussian parameters and adaptive boosting. **Biomedical Signal Processing and Control**, 2016.

LIU, X. et al. The relationship between inflammation and neurocognitive dysfunction in obstructive sleep apnea syndrome. **Journal of Neuroinflammation volume 17**, 2020.

Lu, S.; Meng, J.; Cao, G. Support vector machine based on a new reduced samples method. In: **2010 International Conference on Machine Learning and Cybernetics**. [S.l.: s.n.], 2010. v. 3, p. 1510–1514.

Mahmud, M. et al. Applications of deep learning and reinforcement learning to biological data. **IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS**, 2018.

Mansour, K. F. et al. A mathematical model to detect inspiratory flow limitation during sleep. **American Journal of Physiology**, 2002.

Mendez, M. O. et al. Detection of sleep apnea from surface ecg based on features extracted by an autoregressive model. **29th Annual International Conference of the IEEE EMBS**, 2007.

MOTURU, S.; JOHNSON, W.; LIU, H. Predictive risk modelling for forecasting high-cost patients: A real-world application using medicaid data. **Int. J. Biomedical Engineering and Technology Int. J. Biomedical Engineering and Technology**, v. 3, p. 114–132, 01 2010.

Ng, A. K. et al. Bispectral analysis of snore signals for obstructive sleep apnea detection. **2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society**, 2007.

Pombo, N.; Garcia, N.; Bousson, K. Classification techniques on computerized systems to predict and/or to detect apnea. **Computer Methods and Programs in Biomedicine**, 2017.

Sadr, N.; de Chazal, P. A fast approximation method for principal component analysis applied to ecg derived respiration for osa detection. **2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**, 2016.

Song, C. et al. An obstructive sleep apnea detection approach using a discriminative hidden markov model from ecg signals. **IEEE Transactions on Biomedical Engineering**, 2016.

Tafner, M. A. Estagiamento automático do sono usando uma rede neural artificial. **IV Congresso Brasileiro de Redes Neurais**, 1999.

Tripathi, S.; Batra, S.; Pandey, S. Unbiased mortality prediction for unbalanced data using machine learning. In: **2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON)**. [S.l.: s.n.], 2019. p. 1–5.

UDDIN, M. B. et al. A new machine learning approach to select adaptive imfs of emd. **2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)**, 2016.

Wardhani, N. W. S. et al. Cross-validation metrics for evaluating classification performance on imbalanced data. In: **2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)**. [S.l.: s.n.], 2019. p. 14–18.

Xie, B.; Minn, H. Real-time sleep apnea detection by classifier combination. **IEEE Transactions on Information Technology in Biomedicine**, 2012.

Yadav, S.; Shukla, S. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In: **2016 IEEE 6th International Conference on Advanced Computing (IACC)**. [S.l.: s.n.], 2016. p. 78–83.

Yang, F. An extended idea about decision trees. **2019 International Conference on Computational Science and Computational Intelligence (CSCI)**, 2019.

Yue, Y.; Yang, Y. Improved ada boost classifier for sports scene detection in videos: from data extraction to image understanding. In: **2020 International Conference on Inventive Computation Technologies (ICICT)**. [S.l.: s.n.], 2020. p. 1–4.