



TRABALHO DE CONCLUSÃO DE CURSO

**Aplicação de Aprendizado de Máquina na
Detecção de COVID-19 e Outras Doenças
Utilizando Dados de Smartwatches**

Luiz Felipe Folha Tavares

Brasília, Maio de 2022



UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE CONCLUSÃO DE CURSO

Aplicação de Aprendizado de Máquina na Detecção de COVID-19 e Outras Doenças Utilizando Dados de Smartwatches

Luiz Felipe Folha Tavares

*Trabalho de conclusão de curso submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro Eletricista*

Banca Examinadora

Prof. Eduardo Peixoto Fernandes da Silva, _____
ENE/UnB
Orientador

Prof. Daniel Guerreiro e Silva, ENE/UnB _____
Examinador Interno

Prof. Francisco Assis de Oliveira Nascimento, _____
ENE/UnB
Examinador Interno

FICHA CATALOGRÁFICA

TAVARES, LUIZ FELIPE FOLHA

Aplicação de Aprendizado de Máquina na Detecção de COVID-19 e Outras Doenças Utilizando Dados de Smartwatches [Distrito Federal] 2022.

xvi, 59 p., 210 x 297 mm (ENE/FT/UnB, Engenheiro, Engenharia Elétrica, 2022).

Trabalho de conclusão de curso - Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Aprendizado de máquina

2. Smartwatches

3. Detecção

4. COVID-19

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

TAVARES, L. F. F. (2022). *Aplicação de Aprendizado de Máquina na Detecção de COVID-19 e Outras Doenças Utilizando Dados de Smartwatches*. Trabalho de conclusão de curso, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 59 p.

CESSÃO DE DIREITOS

AUTOR 1: Luiz Felipe Folha Tavares

AUTOR 2:

TÍTULO: Aplicação de Aprendizado de Máquina na Detecção de COVID-19 e Outras Doenças Utilizando Dados de Smartwatches.

GRAU: Engenheiro Eletricista ANO: 2022

É concedida à Universidade de Brasília permissão para reproduzir cópias desto Trabalho de conclusão de curso e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Os autores reservam outros direitos de publicação e nenhuma parte desse Trabalho de conclusão de curso pode ser reproduzida sem autorização por escrito dos autores.

Luiz Felipe Folha Tavares

Dept. de Engenharia Elétrica (ENE) - FT

Universidade de Brasilia

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

Agradecimentos

Agradeço antes de tudo à minha mãe, Luiza Folha, por todo amor, apoio, dedicação e incentivos prestados durante esses 23 anos de vida. Agradeço também às forças que regem o universo pela dádiva que é ter nascido em uma família que sempre me amou e me incentivou a ser o melhor de mim a cada dia, por ter a mãe, irmãs, avós, tios, primos, amigos, namorada e cachorros que tenho e tanto amo, assim como ter o privilégio de poder partilhar esse mundo e época com todos, como diria Carl Sagan. Presto minha imensa gratidão ao meu amigo e irmão do peito, Matheus Virgílio, que me deu apoio e que dedicou muitas horas do seu tempo para me ensinar o conteúdo que foi base da realização deste trabalho. Agradeço ao meu orientador, Eduardo Peixoto, por toda a paciência e ensinamentos que me foi passado. Agradeço à minha namorada Giovana pelo carinho, paciência e apoio que sempre me deu. Agradeço aos meus irmãos de vida, Fábio e Máx, pelo enorme companheirismo e aventuras vividas durante todo o tempo de graduação. Agradeço aos meus companheiros de curso Thiago, Igor Rubens, Matheus N, Felipe S, Felipe R, Paulo, Marlon, Victor G, Thaís, Gabriel P e Marcos por todo apoio e anos de sofrimento e diversão que passamos juntos. Agradeço também aos meus amigos Abner, Matheus A, João Rebas, Fabiano, e todos os outros que de alguma forma foram e são essenciais pra mim.

Luiz Felipe Folha Tavares

SUMÁRIO

1 INTRODUÇÃO	2
1.1 MOTIVAÇÃO	2
1.2 OBJETIVO.....	3
1.3 ORGANIZAÇÃO DO TRABALHO	3
2 TRABALHOS RELACIONADOS	4
2.1 <i>Pre-symptomatic detection of COVID-19 from smartwatch data</i>	4
2.2 <i>Real-time alerting system for COVID-19 and other stress events using wearable data</i>	6
2.3 <i>Deep learning-based detection of COVID-19 using wearables data</i>	10
2.4 <i>COVID-19 detection from Xray and CT scans using transfer learning</i> ...	11
2.5 CONCLUSÕES ACERCA DO CAPÍTULO.....	14
3 FUNDAMENTAÇÃO TEÓRICA	15
3.1 MEIOS CONVENCIONAIS PARA A DETECCÃO DE COVID-19.....	15
3.1.1 TESTE MOLECULAR RT-PCR.....	15
3.1.2 TESTES SOROLÓGICOS	16
3.2 SENsoRES E DADOS DE SMARTWATCHES APLICADOS NA SAÚDE	19
3.3 APRENDIZADO DE MÁQUINA.....	21
3.3.1 APRENDIZADO SUPERVISIONADO, NÃO SUPERVISIONADO E POR REFORÇO	22
3.3.2 CLASSIFICAÇÃO E REGRESSÃO EM APRENDIZADO DE MÁQUINA	22
3.3.3 MÉTRICAS EM APRENDIZADO DE MÁQUINA	22
3.3.4 DECISION TREES, RANDOM FOREST E ISOLATION FOREST	25
3.3.5 REDES NEURAIS E APRENDIZADO PROFUNDO	29
3.4 CONCLUSÕES ACERCA DO CAPÍTULO.....	32
4 METODOLOGIA	33
4.1 DELIMITAÇÃO DO ESCOPO	33
4.2 OBTENÇÃO DA BASE DE DADOS	33
4.2.1 FILTRAGEM DOS DADOS	34
4.3 TREINAMENTO DOS MODELOS	36
4.3.1 <i>Decision Trees E Random Forest</i>	36
4.3.2 <i>Isolation Forest</i>	37
4.3.3 <i>Autoencoder Neural Network</i>	38
4.4 CONCLUSÕES ACERCA DO CAPÍTULO.....	39

5	RESULTADOS.....	41
5.1	UTILIZANDO <i>Decision Trees E Random Forest</i>	41
5.2	UTILIZANDO <i>Isolation Forest</i>	46
5.3	UTILIZANDO <i>Autoencoder Neural Network</i>	49
5.4	ANÁLISE E DISCUSSÃO DOS RESULTADOS	52
5.5	CONCLUSÕES DO CAPÍTULO.....	53
6	CONCLUSÃO.....	54
6.1	TRABALHOS FUTUROS	55
REFERÊNCIAS BIBLIOGRÁFICAS.....		56

LISTA DE FIGURAS

2.1	Associação da frequência cardíaca com a doença COVID-19	5
2.2	Máquina de estados finitos	6
2.3	Exemplos de alertas para participantes - 1	8
2.4	Exemplos de alertas para participantes - 2	9
2.5	Efeitos da vacinação COVID-19 na RHR média durante a noite	9
2.6	Etapas do autoencoder LSTM	10
2.7	Sumário de resultados da abordagem utilizando o LAAD	11
2.8	Exemplos do processo de data augmentation	12
2.9	InceptionV3 ajustado para problemas de 2 e 3 classes	13
2.10	DenseNet ajustado para problemas de 2 e 3 classes.....	13
2.11	Modelo DenseNet ajustado testando imagens de raios-X.....	14
2.12	Modelo DenseNet ajustado testando imagens de tomografia computadorizada.....	14
3.1	Fluxograma esquemático de métodos de detecção molecular para a COVID-19	16
3.2	Fluxograma esquemático da utilização da metodologia ELISA	17
3.3	Fluxograma esquemático do FLA	18
3.4	Comparação entre métodos moleculares e sorológicos	18
3.5	Variação estimada do tempo de detecção para diferentes testes.....	19
3.6	Formas de energia e suas respectivas informações associadas	20
3.7	Representação esquemática de um sistema típico de medição de sinais fisiológicos	20
3.8	Curva ROC.....	24
3.9	Underfitting, good fitting e overfitting.....	25
3.10	Exemplo de árvore gerado pelo método <i>Decision Trees</i>	26
3.11	Exemplo de resultado de classificação com <i>Decision Trees</i>	27
3.12	Diagrama do algoritmo <i>Random Forest</i>	28
3.13	Diagrama do algoritmo <i>Isolation Forest</i>	28
3.14	Estrutura de um nó de uma rede neural	29
3.15	Estrutura de um nó de uma rede neural	30
3.16	Diversos tipos de rede neural - parte 1	30
3.17	Diversos tipos de rede neural - parte 2	31
4.1	Diagrama de alto nível para a filtragem.....	34
4.2	Componentes da sazonalidade em um sinal de RHR	35
4.3	Arranjo do dataframe obtido.....	35
4.4	Diagrama de alto nível para o Decision Trees e Random Forest.....	36
4.5	Diagrama de alto nível para o Isolation Forest	37
4.6	Diagrama de alto nível para o Autoencoder.....	39

5.1	Efeito do hiperparâmetro max_depth para o Decision Trees	42
5.2	Efeito do hiperparâmetro min_samples_leaf para o Decision Trees	42
5.3	Matriz de confusão para o Decision Trees	43
5.4	Efeito do hiperparâmetro n_estimators para o Random Forest.....	44
5.5	Efeito do hiperparâmetro min_samples_leaf para o Random Forest.....	44
5.6	Matriz de confusão para o Random Forest (min_samples_leaf = 15)	45
5.7	Resultados do Isolation Forest - 1	47
5.8	Resultados do Isolation Forest - 2	48
5.9	Perda de validação e de treinamento.....	50
5.10	Curva de acurácia para o Autoencoder	50
5.11	Resultado da reconstrução de um RHR normal - 1.....	51
5.12	Resultado da reconstrução de um RHR normal - 2.....	51
5.13	Resultado da reconstrução de um RHR anormal - 1	52
5.14	Resultado da reconstrução de um RHR anormal - 2	52

LISTA DE TABELAS

3.1	Matriz de confusão	23
5.1	Métricas obtidas para Decison Trees e Random Forest	45
5.2	Percentual dos testes diagnósticos para os modelos.....	46
5.3	Resumo dos resultados obtidos por meio do Isolation Forest.....	49

LISTA DE ACRÔNIMOS

BPM	<i>Batimentos Por Minuto</i>
COVID	<i>Corona Virus Disease (Doença Causada Pelo Coronavírus)</i>
DNA	<i>Ácido Desoxirribonucleico</i>
FSM	<i>Finite-State Machine (Máquina de Estados Finitos)</i>
HROS	<i>Heart Rate Over Steps (Frequência Cardíaca Calculada Sob a Quantidade de Passos)</i>
LSTM	<i>Long Short-Term Memory (Memória de Longo-Curto Prazo)</i>
NaN	<i>Not a Number (Valor Numérico Indefinido)</i>
RHR	<i>Resting Heart Rate (Frequência Cardíaca Em Repouso)</i>
RNA	<i>Acido Ribonucleico</i>
ROC	<i>Receiver Operating Characteristic (Caracterista de Operação do Receptor)</i>
RT-PCR	<i>Real-Time Reverse Transcriptase Polymerase Chain Reaction (Reação em Cadeia da Polimerase Transcriptase Reversa em Tempo Real)</i>
SARS-CoV-2	<i>Severe Acute Respiratory Syndrome Coronavirus 2 (Síndrome Respiratória Aguda Grave Coronavírus 2)</i>

RESUMO

Os *smartwatches*, que estão cada vez mais populares, têm uma série de sensores que podem capturar importantes sinais fisiológicos de forma não invasiva.

Dessa forma, esses dados podem ser utilizados para monitorar a saúde dos seus usuários, surgindo assim a possibilidade de utilizar modelos com o propósito de prever o surgimento de doenças antes mesmo do aparecimento de sintomas . Isso é especialmente importante para doenças altamente infecciosas, como a COVID-19, onde uma detecção precoce pode interromper cadeias de transmissão e diminuir a propagação da doença.

Assim, este trabalho propõe não só discutir acerca de estudos relacionados à detecção de COVID-19 e outras doenças utilizando métodos computacionais e dados obtidos via *smartwatches*, como também demonstrar a utilização de técnicas de *machine learning* em conjunto com tais dados, criando, dessa forma, modelos capazes de detectar doenças, como por exemplo COVID-19, gripe, pneumonia, entre outras.

ABSTRACT

Smartwatches, which are increasingly popular, have a number of sensors capable of non-invasively capture important physiological signals.

Thus, this recorded data can be used to monitor the health of its users, creating the possibility of using predicting models to detect diseases even before the symptoms appearance . This is especially important for highly infectious diseases such as COVID-19, which an early detection can interrupt a series of transmissions and prevent the spread of the disease.

This work proposes not only to discuss studies related to the detection of COVID-19 and other diseases using computational methods and data obtained via smartwatches, but also to demonstrate the use of machine learning techniques in conjunction with such data, creating a diseases detector model for COVID-19, flu, pneumonia, among others.

1 INTRODUÇÃO

Vivemos em um mundo extremamente conectado, com diversos tipos de dados sendo gerados e coletados a todo momento, não sendo diferente para as informações fornecidas pelo corpo humano. Com o aumento da popularização e da acessibilidade, os *smartwatches* (relógios inteligentes) vêm se tornando uma excelente alternativa para o monitoramento constante dos sinais fisiológicos e da saúde das pessoas, pois na maioria das vezes possuem sensores para acompanhar sinais emitidos de diferentes formas pelo corpo, tais como: batimentos cardíacos, pressão sanguínea, quantidade de passos, qualidade do sono, entre outros.

Devido ao cenário pandêmico que o mundo vivenciou no ano de 2020, surgiu ainda mais a necessidade do desenvolvimento de ferramentas e técnicas que auxiliem no diagnóstico prévio de doenças, principalmente as que possuem um alto nível de propagação, assim, os *smartwatches* podem ser bons aliados para essa causa.

1.1 MOTIVAÇÃO

A motivação deste trabalho se dá por conta do cenário e dos prejuízos (tanto em termos de vidas como de maneira econômica) causados pela pandemia do novo corona vírus. Dessa forma, se faz essencial a sinergia entre as áreas da saúde e de tecnologia, de forma que seja possível minimizar os impactos causados pela propagação de doenças.

Com a utilização dos dados fornecidos de forma frequente pelo corpo, captados via *smartwatches*, seria possível elaborar um sistema de detecção de diversas doenças. Tal sistema poderia enviar alertas para seus usuários, passando informações acerca de alterações de certos sinais vitais, indicando assim a presença de anormalidades em sua saúde. Ainda com a utilização desse sistema, surgiria a possibilidade de estudar o efeito de diversos fenômenos cotidianos no organismo de forma não invasiva, como o comportamento dos sinais fisiológicos em cada estação do ano, ou até mesmo os efeitos relacionados à ingestão de certos alimentos ou remédios. Seria possível também estudar como os sinais vitais da população se alteram durante épocas de surtos de algumas doenças, como a dengue no cenário brasileiro, podendo assim auxiliar o Estado na tomada de decisões mais eficazes no contexto da saúde pública, assim como obter indicadores constantemente atualizados a respeito da saúde populacional.

É interessante destacar que o sistema criado não necessariamente precisaria fornecer alertas conclusivos, mas que possam indicar ao usuário a necessidade de realizar uma consulta médica ou de fazer um *check-up*.

1.2 OBJETIVO

O presente trabalho visa empregar técnicas de aprendizado de máquina (ou *machine learning*), juntamente com as informações que podem ser obtidas através dos sensores de *smartwatches* (atualmente dados referentes à frequência cardíaca e quantidade de passos), para criar um sistema de alerta quando na detecção de alguma alteração dos sinais fisiológicos que estão sendo captados. Tal sistema pode ser essencial na detecção precoce de doenças, uma vez que os alertas serviriam como indicação para o usuário procurar ajuda médica especializada, realizando a testagem e recebendo o diagnóstico correto, podendo assim se cuidar antes mesmo dos sintomas se agravarem.

1.3 ORGANIZAÇÃO DO TRABALHO

O capítulo 2 faz uma revisão bibliográfica acerca dos estudos que utilizaram dados de *smartwatches* na detecção de COVID-19 ou outras doenças, assim como a utilização de técnicas de aprendizado de máquina para o mesmo propósito.

O capítulo 3 fornece a base teórica para o desenvolvimento deste trabalho, provendo informações sobre os tipos e funcionamentos de testes para COVID-19, além de informações acerca de aprendizado de máquina e outros conceitos relacionados.

Por sua vez, o capítulo 4 provê informações sobre os métodos empregados, explicando sobre a criação e definição de parâmetros dos modelos utilizados, enquanto que o capítulo 5 apresenta os resultados obtidos e levanta o questionamento em relação aos mesmos.

Por último, o capítulo 6 apresenta as conclusões do trabalho, justificando as limitações e indicando as possibilidades para desenvolvimento de trabalhos futuros na área.

2 TRABALHOS RELACIONADOS

O objetivo deste capítulo é comentar acerca de estudos que serviram como motivação e base para o desenvolvimento desse trabalho. Em tais estudos, foram empregadas diversas técnicas computacionais para o auxílio na detecção de COVID-19 e outras doenças.

2.1 PRE-SYMPOMATIC DETECTION OF COVID-19 FROM SMARTWATCH DATA

O primeiro estudo que será discutido foi publicado na revista *Nature Biomedical Engineering*, no qual foram coletados entre os meses de fevereiro e julho de 2020, dados de 5262 indivíduos. Toda a coleta de dados foi realizada via *smartwatches*, onde eram medidos sinais referentes à quantidade de passos e de frequência cardíaca [1]. De acordo com o estudo, alterações nas medidas da frequência cardíaca de um paciente em repouso e da quantidade de passos, ao longo de um dado período de tempo, podem indicar com antecedência a presença de alguma perturbação na saúde.

Os dados dos indivíduos foram coletados seguindo o protocolo 55577, tendo aprovação da Universidade de Stanford. Uma vez que a maioria dos participantes utilizavam a pulseira Fitbit para realizar as medições, a pesquisa procedeu utilizando os dados de 3325 pessoas que utilizavam essa pulseira. Dentro dessa amostra, 114 informaram a contaminação com COVID-19 e 47 pessoas informaram estar com outras infecções respiratórias.

Após a coleta dos dados, os mesmos passaram por um pré-processamento, no qual os valores de frequência cardíaca inferiores a 30 batimentos por minuto e superiores a 200 foram removidos, assim como valores duplicados de frequência cardíaca, quantidade de passos e de sono. Ainda na etapa de pré-processamento, os dados referentes a data e hora foram padronizados para um mesmo fuso horário. Foram obtidos dados estatísticos como média e mediana da frequência cardíaca, foram obtidos também valores para a frequência cardíaca em repouso (RHR), quantidade de passos no dia e informações sobre os estágios do sono.

No estudo, foram utilizadas duas abordagens para a detecção de anomalias, que poderiam vir a ser um indicativo de COVID-19 ou alguma outra doença respiratória. Tais abordagens foram denominadas como *RHR-Diff offline anomaly detection* e *HROS-AD offline anomaly detection*.

Na primeira metodologia, após obtidos os dados referentes ao RHR de cada individuo, os valores foram padronizados em uma resolução de 1 hora de duração, tendo como referência a média diária em uma janela deslizante de 28 dias. A identificação do tempo de anomalia baseou-se em um varreduras de classificação. Através de um nível de significância de 0,05 utilizando tempos longos de detecção e descartando detecções inferiores a 24 horas de duração, ocorria o envio de alerta.

Por sua vez, na *HROS-AD offline anomaly detection*, os dados de frequência cardíaca e de passos foram combinados, utilizando uma média móvel de 400 horas, em seguida, a resolução de duração foi alterada para 1 hora. A detecção de anomalias foi implementada utilizando o envelopamento elíptico, no qual determinados pontos que estavam acentuadamente distantes de outros pontos (fora do envelope) eram denominados anomalias ou *outliers*, enquanto que os pontos que ficavam próximos dos demais eram vistos como normais.

A figura 2.1 mostra os resultados obtidos através do estudo, fornecendo o tipo de abordagem utilizada em cada detecção. Analisando a figura, é possível perceber que o método *RHR-Diff* foi capaz de detectar COVID-19 antes do aparecimento de sintomas em vários casos, como por exemplo para os indivíduos AJWW3IY, AOYM4KG, AKXN5ZZ e AS2MVDL. O método *RHR-Diff* também conseguiu detectar outras doenças, como nos casos dos indivíduos A0VFT1N, A4E0D03 e AFJ1YC. Em diversos casos, ambos os métodos empregados na pesquisa (*RHR-Diff* e *HROS-AD*) conseguiram detectar COVID-19, todavia, o método *RHR-Diff* demonstrou maior robustez na detecção precoce da doença.

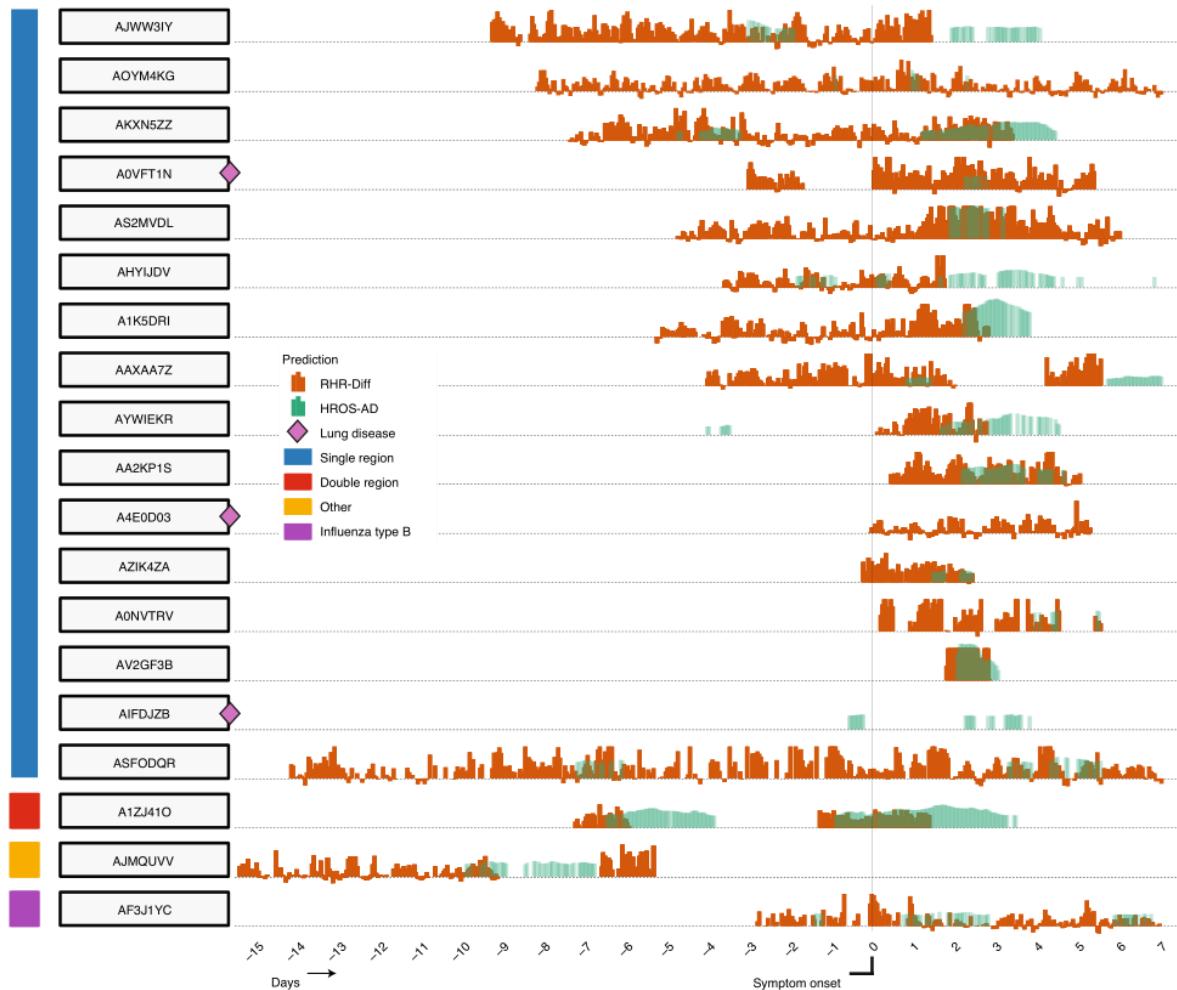


Figura 2.1: Resumo dos dados coletados de 32 participantes do estudo que relataram um diagnóstico confirmado de COVID-19 com início dos sintomas e/ou data do teste. Adaptado do estudo *Pre-symptomatic detection of COVID-19 from smartwatch data* (1).

2.2 REAL-TIME ALERTING SYSTEM FOR COVID-19 AND OTHER STRESS EVENTS USING WEARABLE DATA

Nesse estudo, sob o protocolo 57022, foram coletado dados de 3318 indivíduos com idade entre 18 e 80 anos. Através do aplicativo MyPHD, foi possível transferir informações de *smartwatches* FitBit, Apple Watch e Garmin para uma plataforma de pesquisa da Universidade de Stanford, no qual uma equipe realizava as análises [2]. Desses 3318 participantes, 2117 foram capazes de utilizar corretamente o aplicativo, reportando o estado de saúde e enviando os dados corretamente.

Como demonstrado no estudo anterior [1], os dados dos batimentos cardíacos dos indivíduos em repouso são melhores para ser processados, pois os dados de batimento ao longo de todo tempo podem possuir alterações devido à práticas esportivas ou quaisquer outras ações do cotidiano. Na metodologia denominada *NightSignal*, os dados dos indivíduos em repouso também foram utilizados, todavia, foram considerados somente os dados no período em que os participantes estavam dormindo (entre 00:00 h e 07:00 h).

O monitoramento em tempo real utiliza uma máquina de estados finitos (FSM) para disparar os sinais de alerta. A máquina de estados possui seis estados possíveis, rotulados por cores e símbolos para a mudança de estado. A mudança de estados ocorre uma vez por dia, tendo como base a média da RHR atual durante a noite, somada com o nível de desvio da linha de base calculada anteriormente. Um alerta amarelo é gerado se a RHR média durante a noite for maior que 4 batimentos por minuto (bpm) em relação à linha de base calculada, caso isso ocorra por duas noites consecutivas, é gerado um alerta vermelho. A figura demonstra a máquina de estados utilizada:

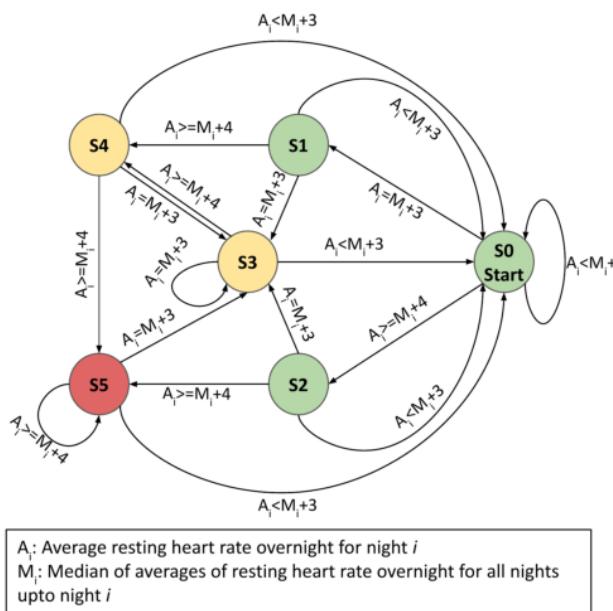


Figura 2.2: Máquina de estados finitos utilizada no estudo *Real-time alerting system for COVID-19 and other stress events using wearable data* (2)

Por sua vez, na abordagem denominada RHRAD, o primeiro mês dos dados (744 horas de leitura) é considerado como linha de base, em seguida é utilizada uma janela deslizante de 1h a cada 1h para detectar as anomalias. A partir disso podem ser enviados sinais de alerta classificados como amarelo ou vermelho, caso as anomalias ocorram em um intervalo entre 1h e 5h, são enviados alertas amarelos, acima de 5h são enviados alertas vermelhos.

Ainda nesse estudo, foi implementado um algoritmo de *Isolation Forest* [2], que trata-se de um modelo não supervisionado para detecção de anomalias, baseado em árvores de decisão. O algoritmo se baseia no quanto distante um determinado valor está dos demais para conseguir detectar os dados anômalos.

Como resultado, a pesquisa não só demonstra os efeitos da COVID-19 nos sinais de RHR noturno, como também os efeitos da utilização de diferentes vacinas de COVID-19, alterações causadas pelo consumo de bebidas alcoólicas, estresses do cotidiano, entre outros. As figuras 2.3 e 2.4 demonstram os resultados obtidos, enquanto a figura 2.5 demonstra o efeito de diferentes vacinas para COVID-19 no sinal de RHR.

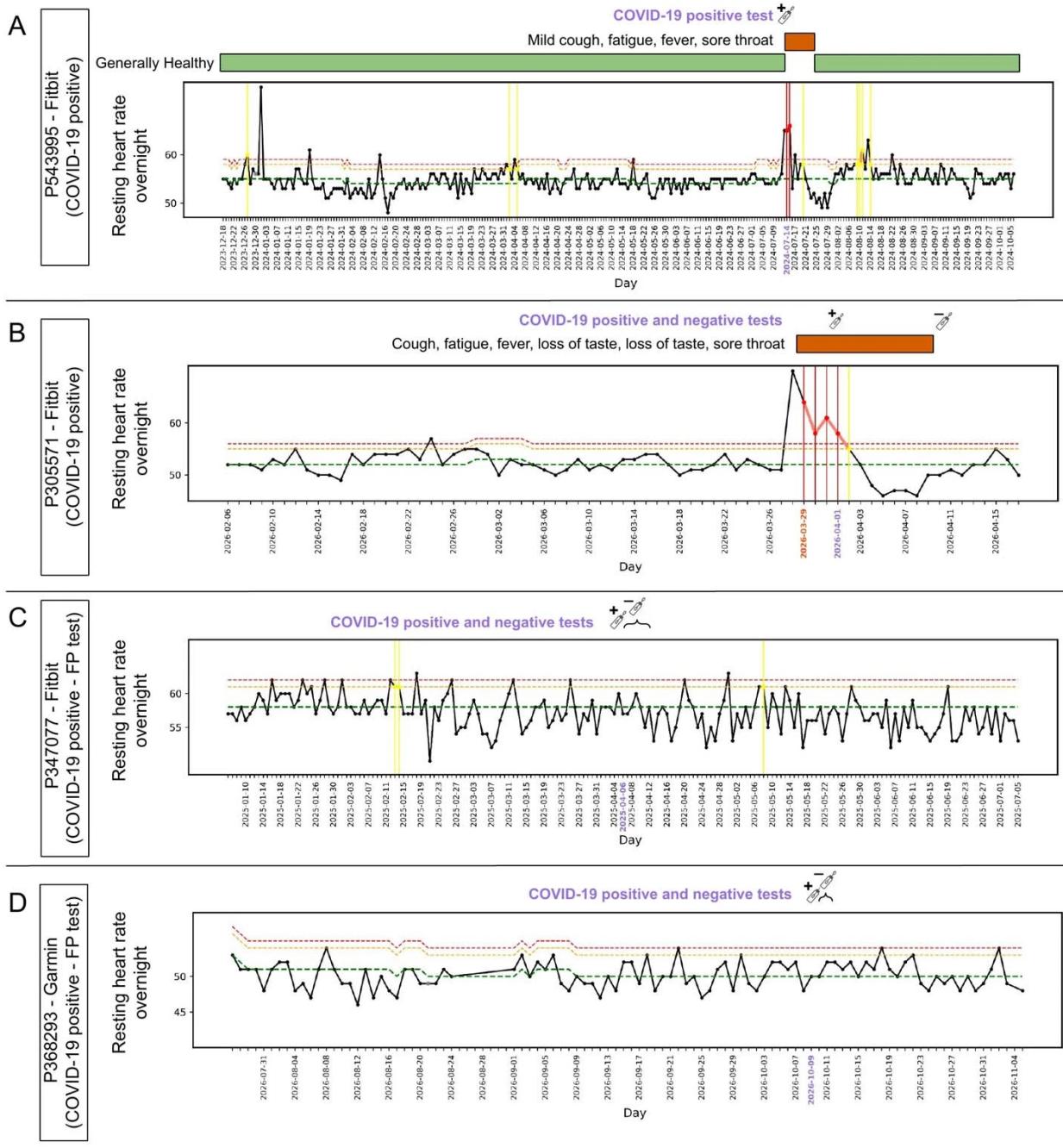


Figura 2.3: Exemplos de alertas para participantes COVID-19 positivos, COVID-19 negativos e não testados. Retirado do estudo *Real-time alerting system for COVID-19 and other stress events using wearable data* [2].

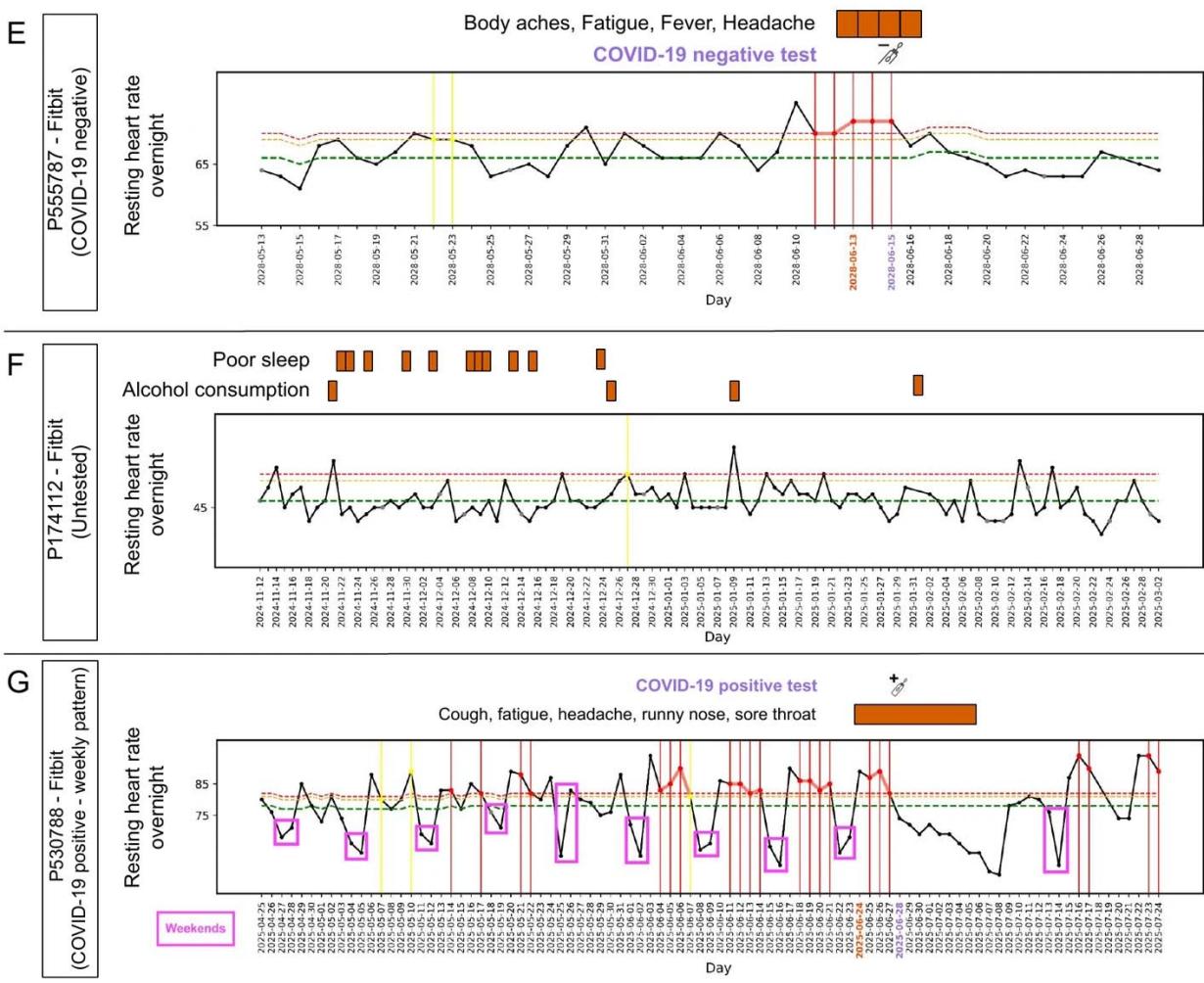


Figura 2.4: Exemplos de alertas para participantes COVID-19 positivos, COVID-19 negativos e não testados. Retirado do estudo *Real-time alerting system for COVID-19 and other stress events using wearable data* [2].

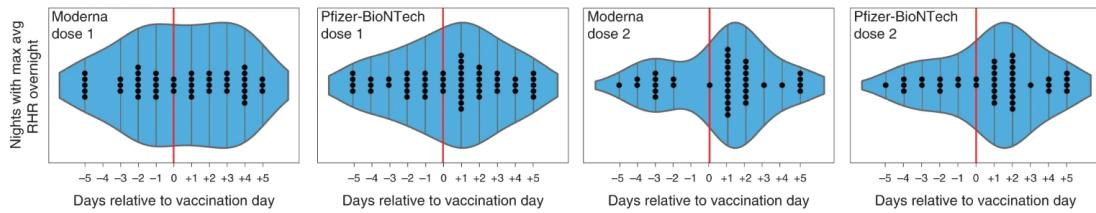


Figura 2.5: Efeitos da vacinação COVID-19 na RHR média durante a noite. Retirado do estudo *Real-time alerting system for COVID-19 and other stress events using wearable data* [2].

2.3 DEEP LEARNING-BASED DETECTION OF COVID-19 USING WEARABLES DATA

Nesse trabalho, foi construído um *framework* denominado LAAD que utiliza aprendizado profundo (ou *deep learning*) para a detecção de COVID-19 [3], no qual, foram selecionados dados de 25 indivíduos que testaram positivo para o novo coronavírus , 11 com alguma outra doença respiratória e 70 de pessoas saudáveis.

Para cada pessoa, os dados de frequência cardíaca foram alinhados com os de passos, baseando-se na data exata de ambos os dados, o resultado obtido no agrupamento foi então re amostrado com resolução de 1 minuto. Em seguida, a frequência cardíaca em repouso foi calculada quando a quantidade de passos estava zerada por 12 minutos a partir de um certo ponto, e para suavizar os dados, foi aplicada uma média móvel de 400 horas.

Os dados foram rotulados como infeciosos, não-infeciosos e em período de recuperação, posteriormente passaram pela fase de *data augmentation* (aumento de dados) [4], onde foram aplicadas algumas técnicas de aumento de dados em séries temporais, tais como: multiplicações por escalar, rotações, permutações, deformação temporal, etc.

Foi utilizado um *autoencoder* LSTM (Long short-term memory) [5], que contém um codificador LSTM que aprende as representações vetoriais de comprimento fixo dos dados de entrada (RHR) e um decodificador LSTM para reconstruir a série temporal do RHR a partir do estado oculto atual e do valor previsto. Um certo valor de erro de reconstrução foi definido como limiar, e a partir disso qualquer erro maior que o valor estabelecido indicava a presença de uma anomalia. A figura 2.6 demonstra as etapas do processo descrito.

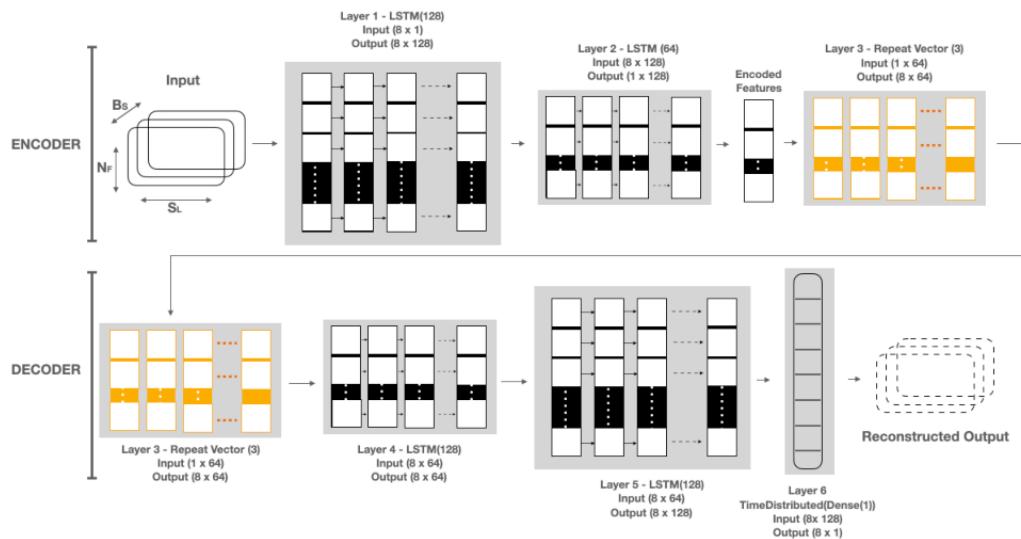


Figura 2.6: Etapas do *autoencoder* LSTM. Adaptado do estudo *Deep learning-based detection of COVID-19 using wearables data* [3].

Através do *framework* LAAD, foram detectadas alterações no sinal de RHR de 14 indivíduos antes do aparecimento de sintomas de COVID-19, em 9 indivíduos durante o período em que existiam sintomas, falhando somente em 2 casos em que os dados eram de pessoas com resultado positivo de COVID-19, como resumido na figura 2.7

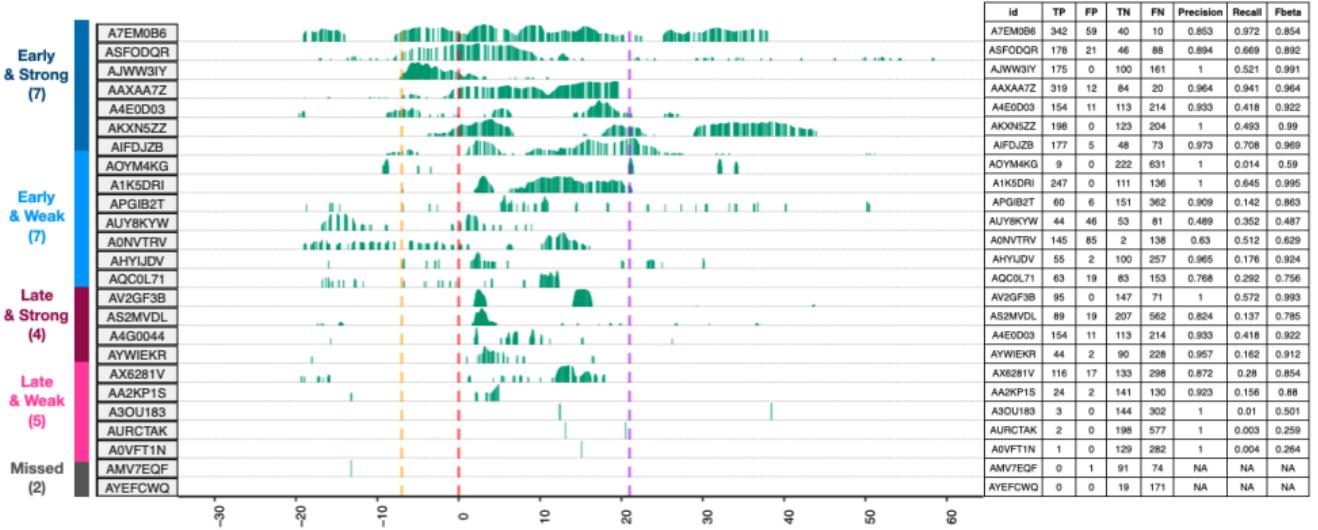


Figura 2.7: Sumário de resultados da abordagem utilizando o LAAD. Adaptado do estudo *Deep learning-based detection of COVID-19 using wearables data* [3].

2.4 COVID-19 DETECTION FROM XRAY AND CT SCANS USING TRANSFER LEARNING

Nesse estudo [6], a detecção de COVID-19 foi feita utilizando imagens de raios-x e tomografias computadorizadas da região do tórax. A base de dados utilizada continha 1130 imagens de raio-x, sendo 430 de casos positivos para COVID-19, 326 de pneumonia e 374 de pessoas saudáveis. A base de dados continha ainda 2482 imagens de tomografia computadorizada, no qual 1252 eram de pacientes com COVID-19 e 1230 de pacientes não infectados.

Na etapa de pré-processamento, as imagens foram normalizadas com resolução de 224x224 pixels. Em seguida, com o objetivo de aumentar o tamanho da base de dados, as imagens passaram por uma etapa de *data augmentation*, onde foram rotacionadas, invertidas horizontalmente, ampliadas e deslocadas, processo demonstrado na figura 2.8

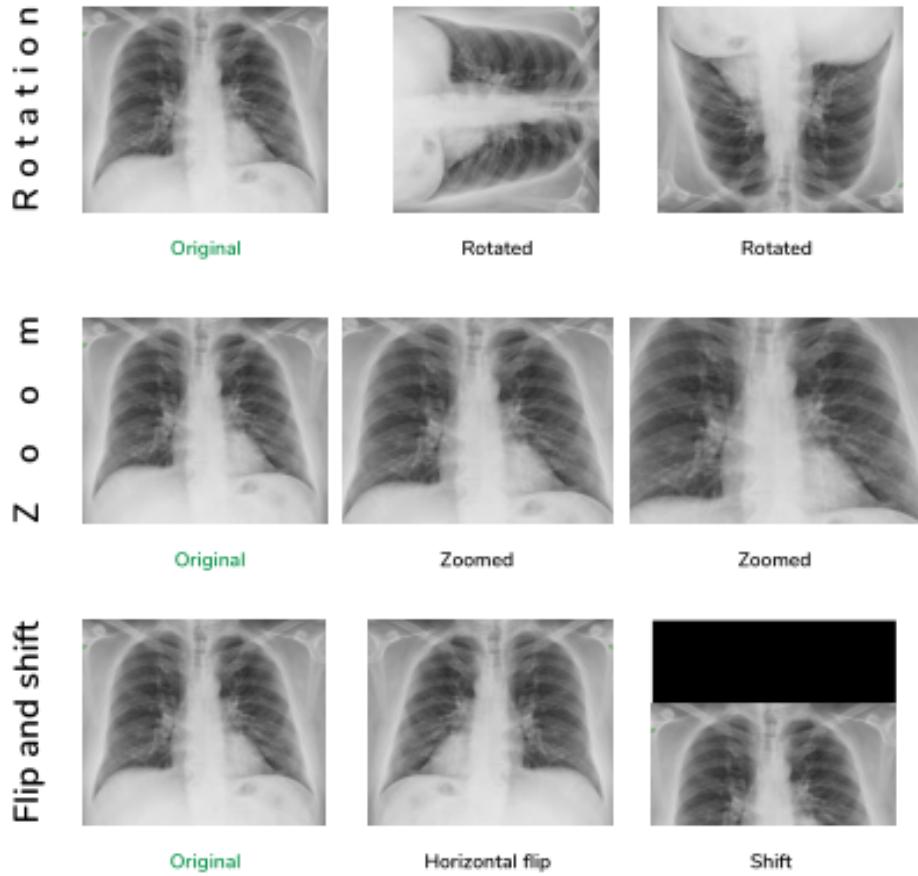


Figura 2.8: Exemplos do processo de *data augmentation*. Retirado do estudo *COVID-19 detection from Xray and CT scans using transfer learning* [6].

No estágio de treinamento, foram utilizados os modelos *DenseNet* e *InceptionV3* previamente treinados através extensa da base de dados *ImageNet*, em seguida os pesos foram congelados e ajustados para a tarefa de detecção de COVID-19, substituindo a ultima camada completamente conectada por uma nova e adicionando uma função *softmax* na saída. Os dados foram divididos 80% para treinamento e 20% para testar os modelos. As figuras 2.9 e 2.10 demonstram o funcionamento dos modelos *InceptionV3* e *DenseNet*, respectivamente.

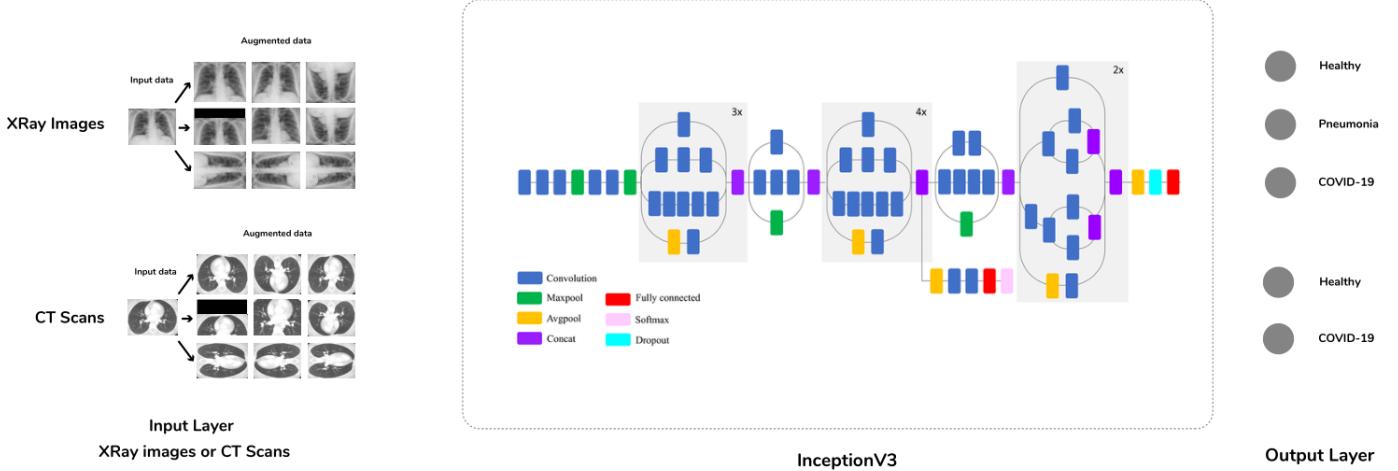


Figura 2.9: *InceptionV3* ajustado para problemas de 2 e 3 classes. Retirado do estudo *COVID-19 detection from Xray and CT scans using transfer learning* [6].

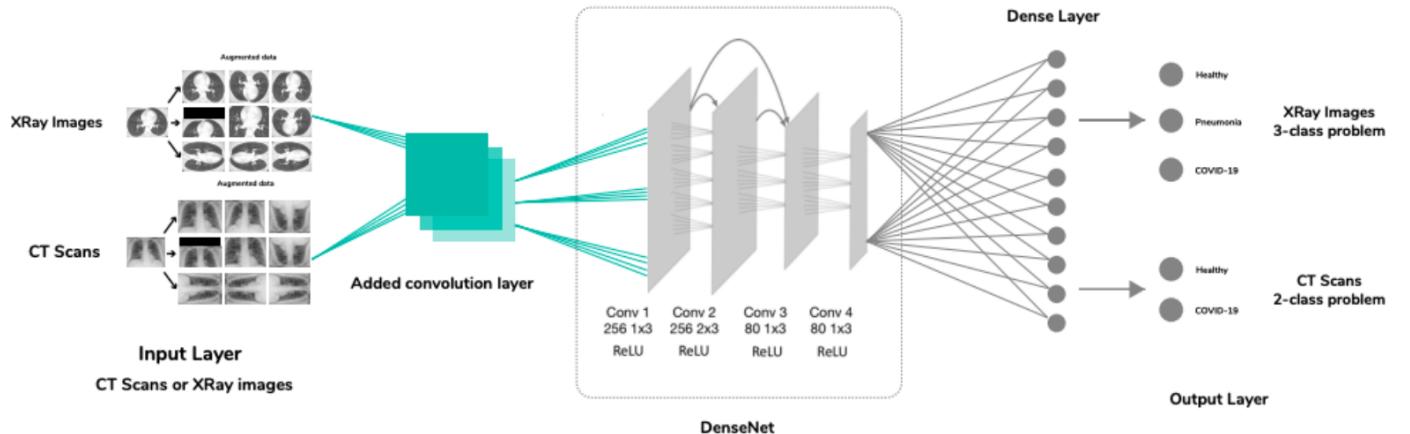


Figura 2.10: *DenseNet* ajustado para problemas de 2 e 3 classes. Retirado do estudo *COVID-19 detection from Xray and CT scans using transfer learning* [6].

Como resultado, para o problema de classificação utilizando as imagens de raio-x, foi obtida uma acurácia de 92,35% com o modelo InceptionV3, 63,56% com o DenseNet e 85% para o New-DenseNet. Por sua vez, utilizando as imagens de tomografia computadorizada, a acurácia encontrada foi de 84,51% para o InceptionV3, 60% para o DenseNet e 95,98% para o New-DenseNet. É possível observar que ao ser adicionada uma camada de convolução ao modelo DenseNet, a performance do mesmo aumenta consideravelmente. As figuras 2.11 e 2.12 mostram os resultados obtidos utilizando o modelo *DenseNet* tendo como entrada imagens de raio-x e de tomografia computadorizada.

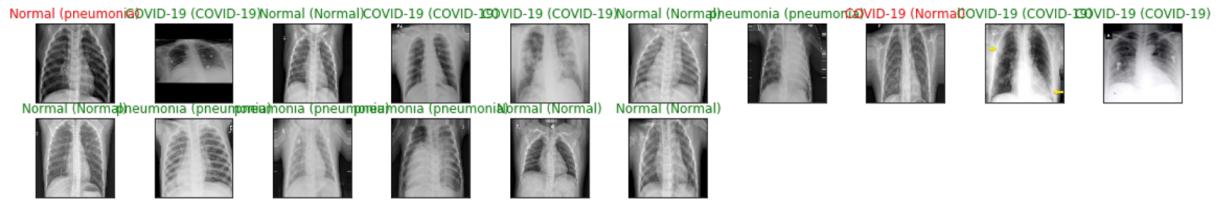


Figura 2.11: Modelo DenseNet ajustado testando imagens de raios-X. Retirado do estudo *COVID-19 detection from Xray and CT scans using transfer learning* [6].

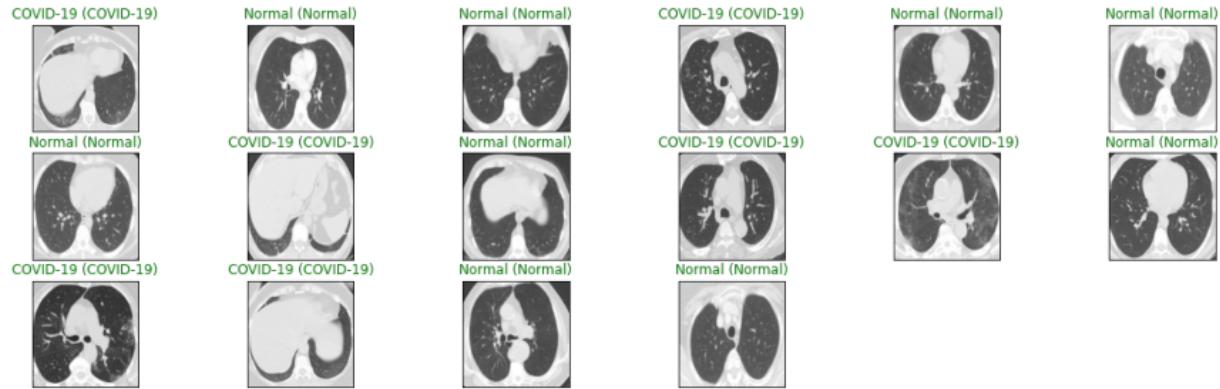


Figura 2.12: Modelo DenseNet ajustado testando imagens de tomografia computadorizada. Retirado do estudo *COVID-19 detection from Xray and CT scans using transfer learning* [6].

2.5 CONCLUSÕES ACERCA DO CAPÍTULO

O capítulo forneceu uma visão geral das pesquisas que estão ocorrendo em todo o mundo na detecção de COVID-19 e outras doenças utilizando metodologias não invasivas, como a análise e o processamento de dados de *smartwatches*, ou através de imagens de tomografia computadorizada e de raio-x. Através da leitura dos diversos trabalhos citados ao longo do capítulo, foi possível perceber a importância de métricas de qualidade na coleta dos dados, assim como todo o cuidado que é necessário ao lidar com a saúde e as informações obtidas dos participantes das pesquisas.

3 FUNDAMENTAÇÃO TEÓRICA

Esse capítulo abordará a teoria por trás das formas convencionais de detecção da COVID-19, além de fornecer uma base teórica a respeito das metodologias utilizadas durante o estudo, adentrando nos conceitos de aprendizado de máquina e aprendizado profundo.

3.1 MEIOS CONVENCIONAIS PARA A DETECÇÃO DE COVID-19

É de fundamental importância a detecção da COVID-19 em pessoas infectadas, uma vez que trata-se de uma doença com alto nível de contágio e com formas de tratamento ainda sendo estudadas [7]. Existem diferentes abordagens para a detecção eficaz da doença [8], alguns serão abordados a seguir, levando em consideração as formas de utilização, de funcionamento e eficácia.

3.1.1 Teste molecular RT-PCR

Os testes moleculares RT-PCR (*real-time Reverse Transcriptase Polymerase Chain Reaction*, ou em português: Reação em Cadeia da Polimerase Transcriptase Reversa em tempo real) são considerados padrão-ouro na detecção de COVID-19, podendo detectar a doença em pacientes sintomáticos ou assintomáticos, desde o primeiro até o décimo dia da presença do vírus, dando o resultado em até 72 horas [9] [10]. A coleta do material é realizada introduzindo um *swab* (cotonete) nas vias nasais, obtendo células presentes na nasofaringe. Em seguida, o material é processado em um laboratório, onde será aplicado o teste RT-PCR na amostra [11].

O RT-PCR possui duas etapas básicas: transcrição reversa e a reação em cadeia da polimerase. Na primeira etapa, com o auxílio da enzima transcriptase reversa, o RNA do vírus é transformado em DNA complementar (cDNA). Por sua vez, na segunda etapa, ocorre uma cadeia de ciclos que faz com que os *primers* (sequências curtas de DNA que complementam o DNA alvo) se conectem ao DNA alvo, seguindo da cópia deste a partir da ligação com o *primer*. Através da fluorescência, é possível observar a ampliação do DNA alvo [8] [12] [13] [10]. A figura 3.1 demonstra o processo de realização do teste RT-PCR através de um fluxograma:

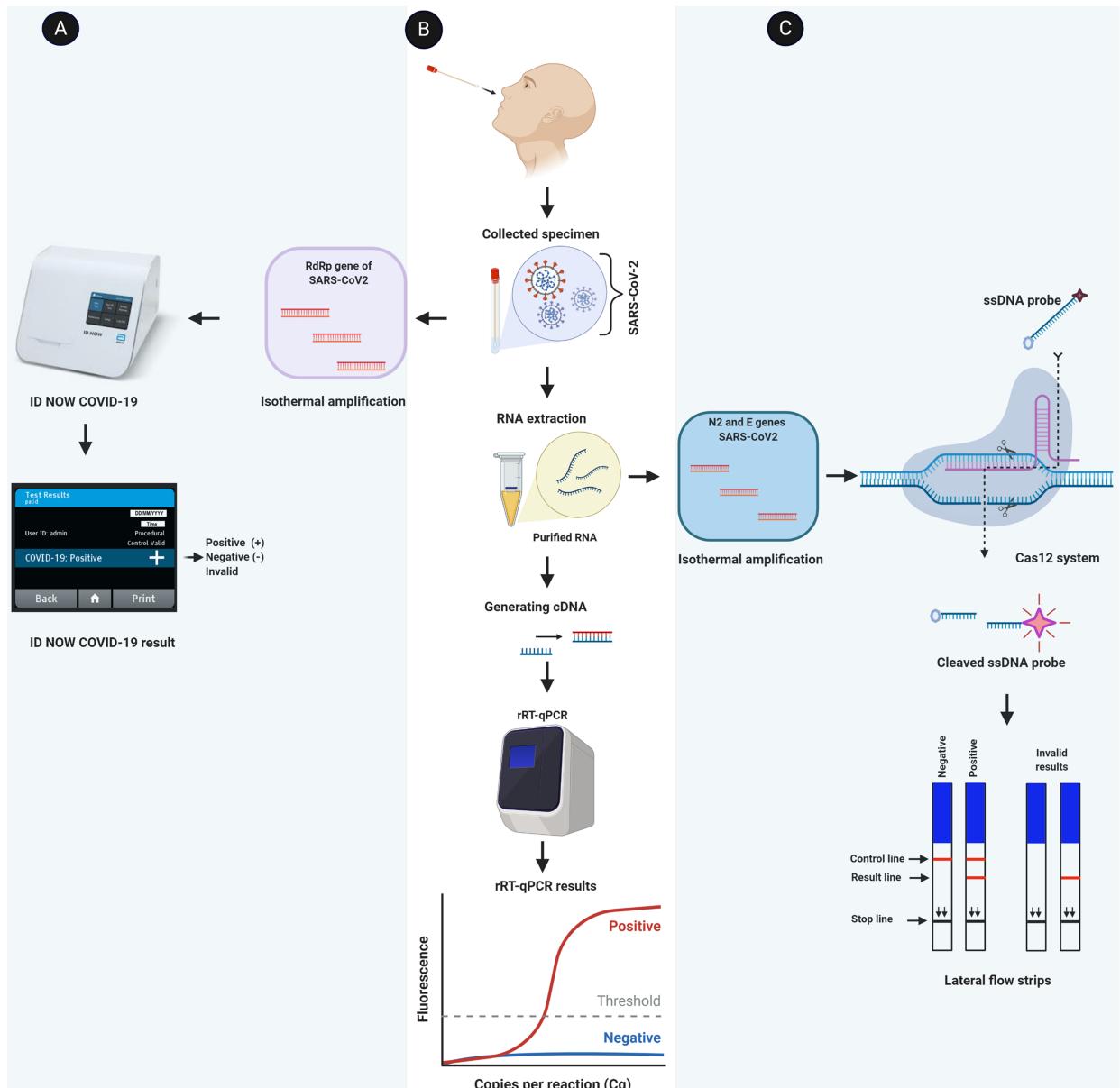


Figura 3.1: Fluxograma esquemático de métodos de detecção molecular para a COVID-19. Retirado do estudo *COVID-19: molecular and serological detection methods* [8].

3.1.2 Testes Sorológicos

Os testes sorológicos utilizam a parte plasmática do sangue, assim, a coleta do material necessário para a realização do exame é feita através da retirada de sangue do paciente [14]. Os testes sorológicos têm como objetivo determinar os抗ígenos que referem-se a diversos microrganismos, assim como identificar também os anticorpos, também conhecidos como imunoglobulinas (Ig), que o organismo produz [15]. No contexto da COVID-19, servem para indicar se uma pessoa foi ou não exposta ao SARS-CoV-2, medindo a resposta imune do indivíduo à infecção [16].

Existem vários tipos de imunoglobulinas no nosso organismo, todavia, para a detecção de COVID-19 têm-se como referência as imunoglobulinas IgA, IgG, IgM e totais podendo ser detectadas por testes convencionais ou testes rápidos imunocromatográficos [17].

Os testes sorológicos convencionais inspecionam os anticorpos IgG e IgM por meio de quimioluminescência, IgG e IgA por meio de ensaio imunoabsorvente ligado a enzima (ELISA) e abordagens que detectam os anticorpos totais ou IgG através de eletroquimioluminescência, enquanto os testes rápidos de antígeno (TR-Ag), ou ensaios de fluxo lateral (FLA), utilizam a imunocromatografia [8] [17] [18] [19]. A figura 3.2 demonstra o fluxograma da metodologia ELISA e a figura 3.3 demonstra o processo dos ensaios de fluxo lateral.

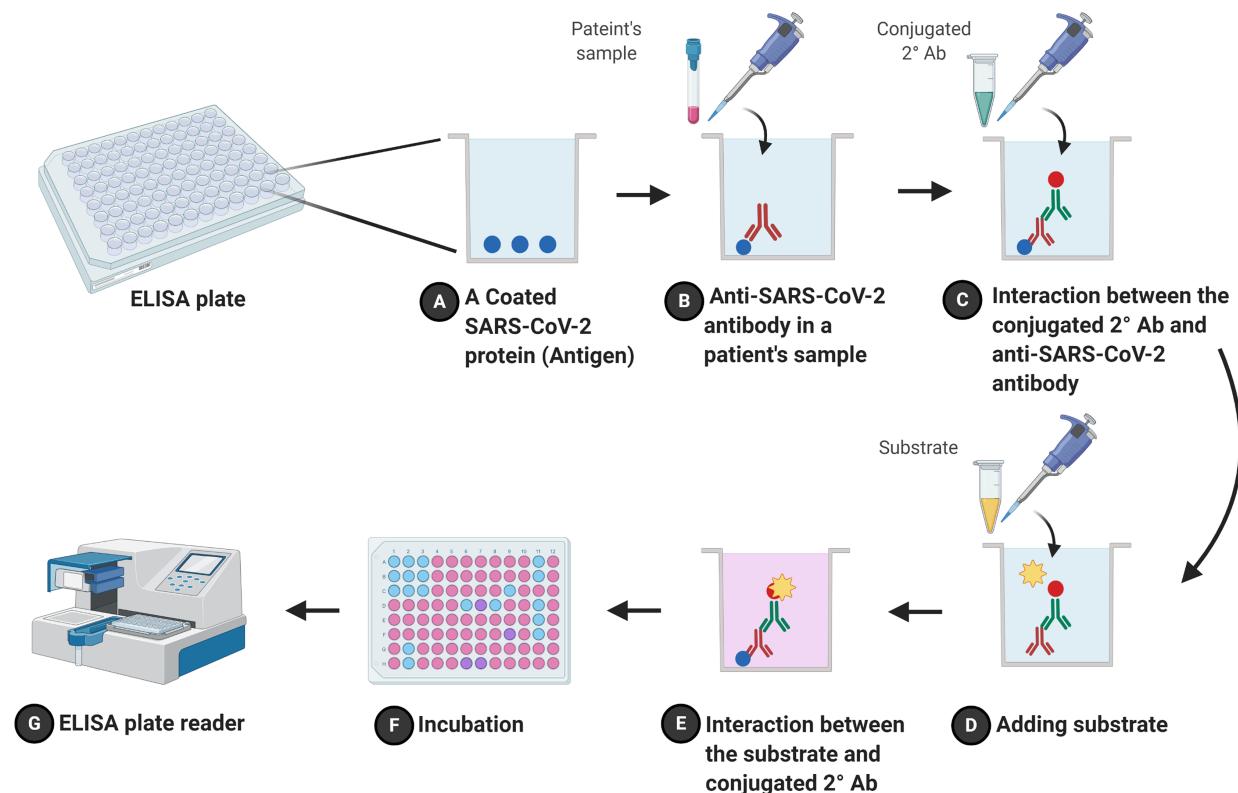


Figura 3.2: Fluxograma esquemático da utilização da metodologia ELISA. Retirado do estudo *COVID-19: molecular and serological detection methods* [8].

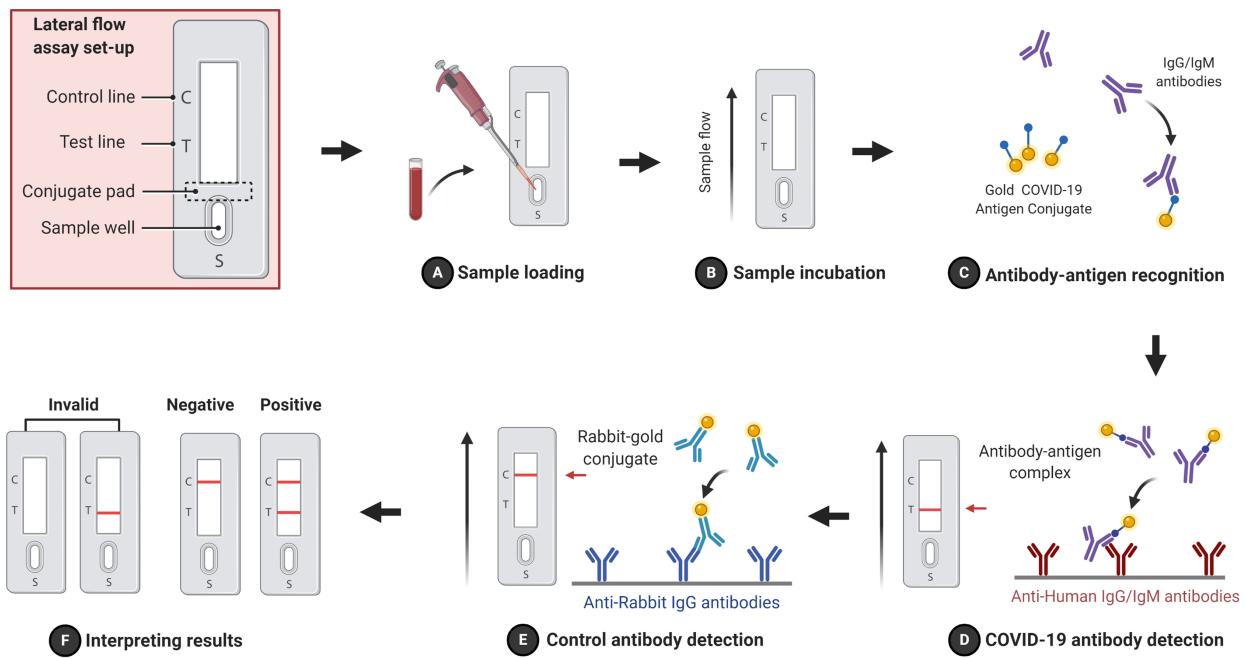


Figura 3.3: Fluxograma esquemático dos ensaios de fluxo lateral. Retirado do estudo *COVID-19: molecular and serological detection methods* [8].

As comparações entre as técnicas de detecção de COVID-19, levando em consideração a acurácia dos métodos, tempo de duração do exame, custos e outros aspectos, podem ser conferidos na figura 3.4, enquanto que a figura 3.5 demonstra o intervalo de detecção dos tipos de testes. Pode-se observar que os testes moleculares são mais acurados que os testes sorológicos, além de serem capazes de detectar a doença com mais precisão a partir dos primeiros sintomas.

	Molecular methods			Serological methods	
Technique based	rRT-PCR	Isothermal amplification	CRISPR-Cas12	LFA	ELISA
Sample	RNA	RNA	RNA	Ag or Ab	Ag or Ab
Accuracy	High	High/Moderate	High	Low	Moderate
Time*	Hours	Minutes	Minutes	Minutes	Hours
Professional skills need	Yes	Yes	YES/No	No	Yes
POCT	No	Yes/No	Yes/No	Yes	No
Availability	Limited	Limited	Limited	Available	Available
Cost	Very high	High	Average	Low	Average
High throughput	Yes	No	No	No	Yes

Note:

* Time that is required for running the test without preparation time.

Figura 3.4: Comparação entre métodos moleculares e sorológicos para detecção da COVID-19. Retirado do estudo *COVID-19: molecular and serological detection methods* [8].

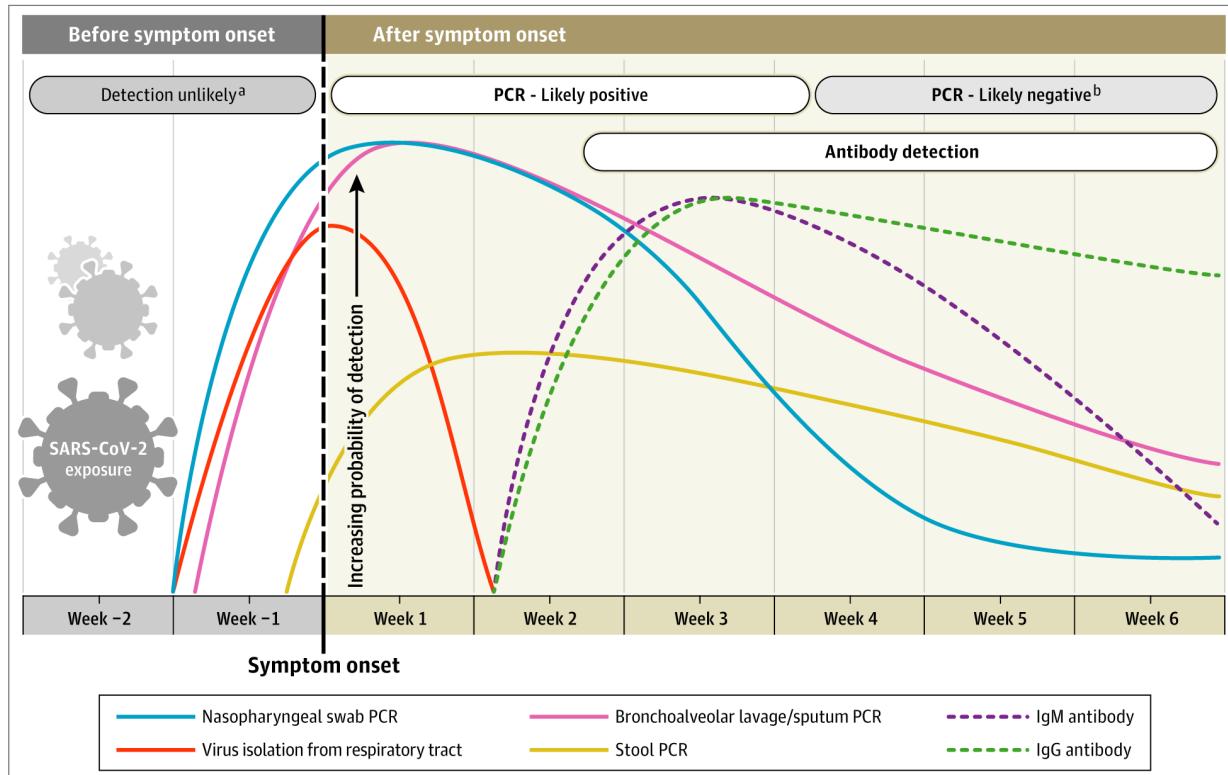


Figura 3.5: Variação estimada ao longo do tempo em testes de diagnóstico para detecção de infecção por SARS-CoV-2 em relação ao início dos sintomas. Retirado do estudo *Interpreting Diagnostic Tests for SARS-CoV-2* [16].

3.2 SENsoRES E DADOS DE SMARTWATCHES APlicados na SAÚDE

Smartwatches e outros dispositivos vestíveis podem ser bastante úteis no monitoramento de diversos fatores da saúde humana, pois podem realizar constantemente a leitura dos sinais fisiológicos gerados pelo corpo [20]. O funcionamento básico da leitura de tais sinais por sensores utiliza como base um transdutor, no qual sinais ou estímulos são convertidos em sinais elétricos. No contexto de biosinais, podem ser utilizados a conversão de energia mecânica, térmica, química, entre outros [21]. A figura 3.6 demonstra as formas de energia geradas pelo corpo humano e suas respectivas informações associadas, por sua vez, a figura 3.7 demonstra o processo básico para aquisição de sinais fisiológicos.

Energy	Variables (Specific Fluctuation)	Common Measurements
Chemical	Chemical activity and/or concentration	Blood ion, O ₂ , CO ₂ , pH, hormonal concentrations, and other chemistry
Mechanical	Position Force, torque, or pressure	Muscle movement, cardiovascular pressures, muscle contractility, valve, and other cardiac sounds
Electrical	Voltage (potential energy of charge carriers) Current (charge carrier flow)	EEG, ECG, EMG, EOG, ERG, EGG, and GSR
Thermal	Temperature	Body temperature and thermography

Figura 3.6: Formas de energia e suas respectivas informações associadas. Retirado do livro *Biosignal and Medical Image Processing* [21].

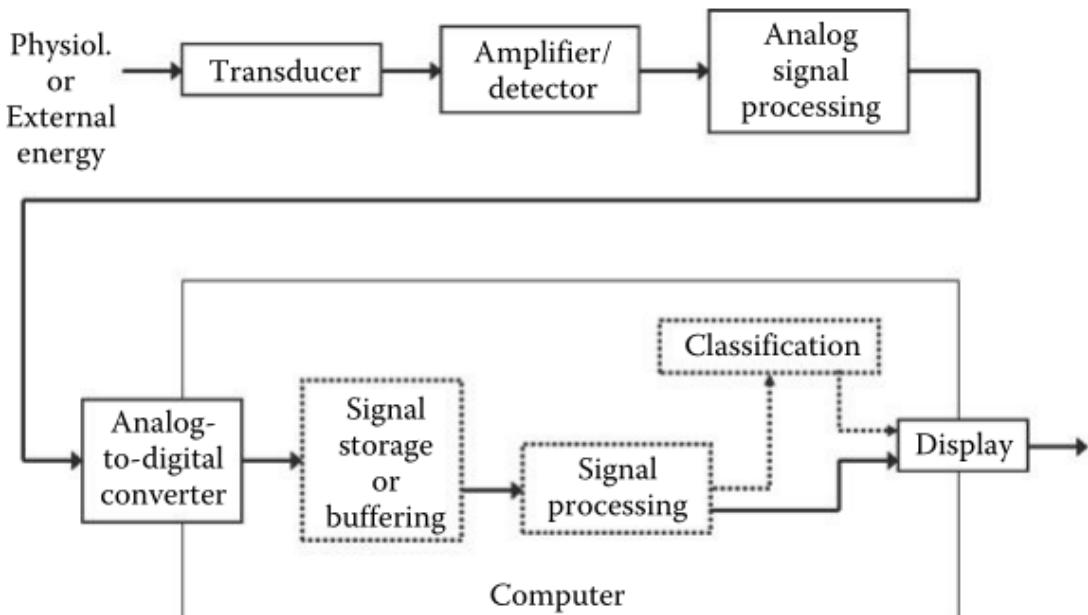


Figura 3.7: Representação esquemática de um sistema típico de medição de sinais fisiológicos. Retirado do livro *Biosignal and Medical Image Processing* [21].

Têm-se observado que ao longo dos anos o desenvolvimento de *smartwatches* e outros dispositivos vestíveis está focando nos seguintes aspectos: *design* amigável ao usuário; possibilidade da interação do usuário com esses dispositivos (incentivando a proatividade do usuário com sua própria saúde); tratamento personalizado (onde a informação oferecida pode variar de acordo com os interesses do usuário) [22]. Existem diversos modelos de *smartwatches* disponíveis no mercado, com especificações que atendem as necessidades e gostos de muitos usuários. Uma análise mais criteriosa a respeito desses dispositivos podem ser conferidas no trabalho de conclusão de curso em Engenharia de Redes da Universidade de Brasília dos alunos *CARDOZO H. M* e *VILELA T.S* [23].

Uma vez que os *smartwatches* realizam a leitura contínua dos dados dos usuários (como batimentos cardíacos e pressão sanguínea por exemplo), o processamento desses dados podem fornecer informações valiosas acerca do estado de saúde de seus utilizadores, sendo possível detectar doenças antes mesmo do aparecimento de sintomas [24]. Nas circunstâncias da COVID-19, a coleta de dados através de *smartwatches* é extremamente útil para o diagnóstico precoce da doença, possibilitando o isolamento eficaz de pessoas infectadas e dessa forma evitando a propagação do vírus, além de fornecer um melhor acompanhamento da evolução da doença e as alterações que ela pode causar nos sinais fisiológicos [1] [2] [3].

3.3 APRENDIZADO DE MÁQUINA

Diferentemente da programação clássica, que utiliza um conjunto de regras bem definidas para resolver problemas, o campo do aprendizado de máquina (ou *machine learning*) baseia-se em treinar o sistema com um certo conjunto de dados para que então sejam encontrados padrões e estruturas estatísticas que possam auxiliar na resolução ou automação das questões levantadas [25]. Em algoritmos de aprendizado de máquina existem três pontos chaves: dados de entrada, exemplos esperados para a saída e alguma forma de medir o quanto a saída encontrada pelo algoritmo se assemelha com a saída esperada [25] [26].

A seguir serão abordados mais conceitos relacionadas com aprendizado de máquina, tais como: formas de aprendizado, tipos de algoritmos e métodos de aprendizado profundo (aprendizado profundo).

3.3.1 Aprendizado Supervisionado, Não Supervisionado e Por Reforço

Na criação de um modelo em aprendizado de máquina, existem três paradigmas que podem ser utilizadas para treinamento, sendo eles: o aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço. No primeiro caso, os dados são previamente rotulados (fornecendo entradas e suas respectivas saídas associadas), assim, o modelo é capaz de aprender com base nessas informações. No aprendizado não supervisionado, o modelo encontra relações a partir dos dados informados que não foram previamente rotulados, separando, dessa forma, as informações em grupos. Por sua vez, no aprendizado por reforço, o modelo aprende a partir da interação com o ambiente em que está inserido, por meio de um processo de recompensas para erros e acertos, sendo necessário explorar um conjunto de ações e optar de forma progressiva pelas que parecerem melhores [25] [27] [28].

3.3.2 Classificação e Regressão em Aprendizado de Máquina

Os problemas de regressão e classificação fazem parte das principais vertentes do aprendizado supervisionado. Nos problemas de regressão, a partir de um conjunto de dados rotulados, o algoritmo deve retornar um valor numérico como saída. Prever o preço de ações, preços de imóveis ou qual será a demanda de algum produto são alguns exemplos que se enquadram em problemas de regressão . Casos em que o resultado deve ser uma categoria ou classe tratam-se de problemas de classificação. Alguns exemplos são: classificar se um paciente está ou não com uma determinada doença, assim como dizer se em uma foto está presente um cão ou um gato [25] [28] [29].

3.3.3 Métricas em aprendizado de máquina

Métricas são fundamentais para mensurar o desempenho dos modelos de aprendizado de máquina, podendo assim indicar o quanto próximos estão da realidade [30]. A seguir serão fornecidos mais detalhes sobre as principais métricas utilizadas.

3.3.3.1 Matriz de Confusão

Através da matriz de confusão é possível analisar a performance de um modelo utilizado para problemas de classificação. Na matriz, são indicados os valores em números absolutos ou de forma percentual dos resultados para a quantidade de verdadeiro-positivos, falso-positivos, verdadeiro-negativos e falso-negativos [31]. A tabela 3.1 demonstra o funcionamento de uma matriz de confusão:

		PREDITO	
		POSITIVO	NEGATIVO
REAL	POSITIVO	VP (VERDADEIRO-POSITIVO)	FN (FALSO-NEGATIVO)
	NEGATIVO	FP (FALSO-POSITIVO)	VN (VERDADEIRO-NEGATIVO)

Tabela 3.1: Funcionamento de uma matriz de confusão

3.3.3.2 Acurácia, Precisão, Revocação, *F1 Score* e curva ROC

A acurácia serve para indicar quantos exemplos foram classificados de maneira correta [31], sua fórmula é definida pela eq. 3.1.

$$Acuracia = \frac{VP + VN}{VP + VN + FP + FN} \quad (3.1)$$

Por sua vez, a precisão é definida pela eq. 3.2, onde pode ser observado que há um foco maior nos erros por falso-positivos [31].

$$Precisao = \frac{VP}{VP + FP} \quad (3.2)$$

A revocação (ou *recall*) foca nos aspectos relacionados aos erros por falso-negativos [31], sendo definida pela eq. 3.3.

$$Recall = \frac{VP}{VP + FN} \quad (3.3)$$

Já a métrica *F1 Score* engloba a precisão e a revocação, sendo definida pela média harmônica de ambas [31]. A eq. 3.4 define a equação para a métrica F1.

$$F1 = \frac{2 * Precisao * Recall}{Precisao + Recall} \quad (3.4)$$

Por último, a curva ROC (do inglês *Receiver Operating Characteristic*), demonstra o desempenho de um modelo em classificar duas classes distintas, tendo como base as medidas entre as taxas de verdadeiro-positivo e de -falso positivo [31], definidas pelas equações 3.5 e 3.6.

$$TVP = \frac{VP}{VP + FN} \quad (3.5)$$

$$TFP = \frac{FP}{FP + VN} \quad (3.6)$$

A figura 3.8 mostra como é criado um gráfico para a curva ROC, sendo que o valor da área abaixo da curva pode ser utilizado como métrica.

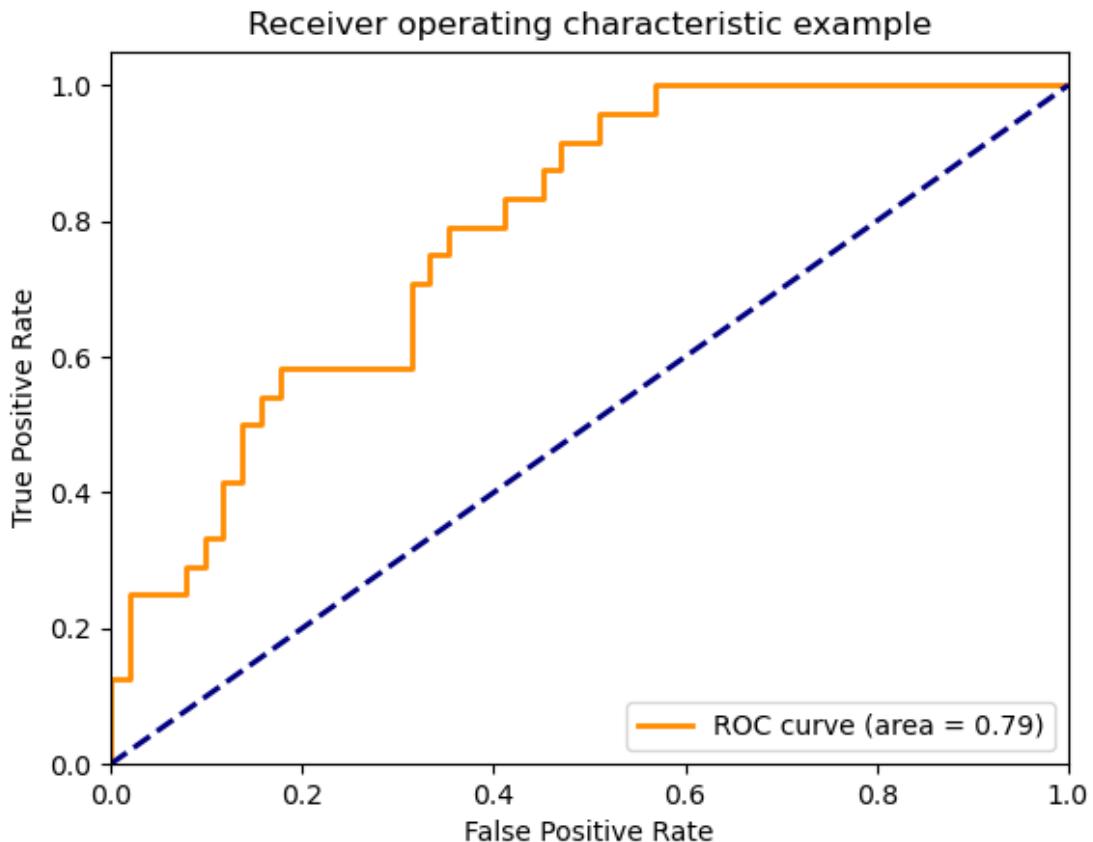


Figura 3.8: Curva ROC. Retirado da documentação do *Scikit-learn* [32]

3.3.3.3 Funções de Perda: Treinamento e Validação

O valor da perda de treinamento ao longo do tempo serve para indicar o desempenho do modelo ao se ajustar com os dados utilizados para treinamento, enquanto que a perda de validação avalia o desempenho do modelo ao se ajustar com novos dados [33].

A partir da comparação das perdas de treinamento e validação, é possível analisar: se o modelo está conseguindo generalizar de forma correta para novos dados; se o modelo não foi capaz de encontrar corretamente as relações entre os dados de treinamento (*underfitting*); ou se o modelo conseguiu aprender de forma excelente com os dados de treinamento mas não tem um desempenho bom com os dados de teste (*overfitting*) [34]. A figura 3.9 representa os três cenários citados:

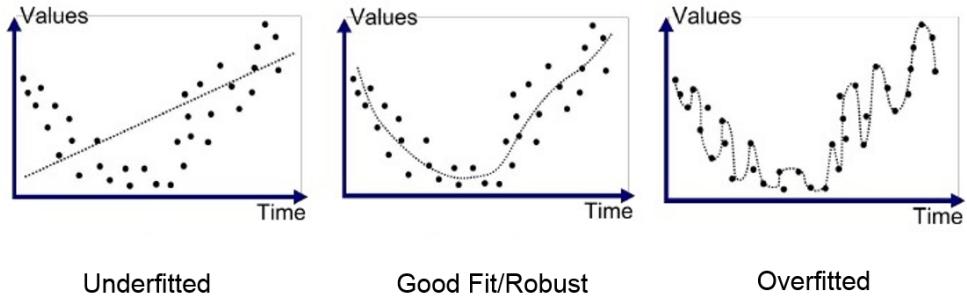


Figura 3.9: *Underfitting*, *good fitting* e *overfitting*. Retirado de um artigo da Associação Brasileira de Ciências de Dados [35].

3.3.4 Decision Trees, Random Forest e Isolation Forest

Decision Trees podem ser consideradas um dos métodos supervisionados mais simples de aprendizado de máquina, podendo ser aplicadas em problemas de regressão ou classificação [36]. Seu funcionamento consiste em utilizar, repetidas vezes, estruturas de decisão arranjadas em forma de árvore para encontrar os resultados esperados [37] [38]. As estruturas de uma árvore de decisão são [32]:

- Nó raiz - nó inicial ou nó pai em uma árvore. A partir desse nó ocorrerão as divisões na árvore;
- Nós de decisão - nós subsequentes ao nó raiz;
- Nós folha - nós que não possuem divisão;
- Sub-árvores - subdivisões de uma árvore.

A definição se um nó será raiz ou um nó de decisão se baseia em alguns critérios para mensurar a incerteza ou impureza em um conjunto de dados, o princípio da entropia é um deles. Todavia, é essencial também medir o quanto a incerteza se altera em cada nó, dessa forma utiliza-se o conceito de ganho de informação. As equações 3.7 e 3.8 representam o cálculo básico para a entropia e ganho de informação utilizadas em problemas de classificação, onde, para o cálculo da entropia, S é o subconjunto de dados utilizados no treinamento, P_+ é a probabilidade da classe ser positiva, e P_- é a probabilidade da classe ser negativa. Por sua vez, no cálculo do ganho de informação, Y se refere ao nó raiz e X aos demais nós que podem estar presentes na árvore [36]. Mais detalhes a respeito da formulação matemática do método podem ser conferidas na documentação do modelo implementado na biblioteca do *Scikit-learn* [32][39].

$$H(S) = P_+ \log(P_+) - P_- \log(P_-) \quad (3.7)$$

$$GI = H(Y) - H\left(\frac{Y}{X}\right) \quad (3.8)$$

O exemplo das figuras 3.10 e 3.11, retirados da documentação do *Scikit-learn* [32], demonstram a utilização de *Decision Trees* para a classificação de flores, baseando-se em características como espessura de pétalas, entre outros.

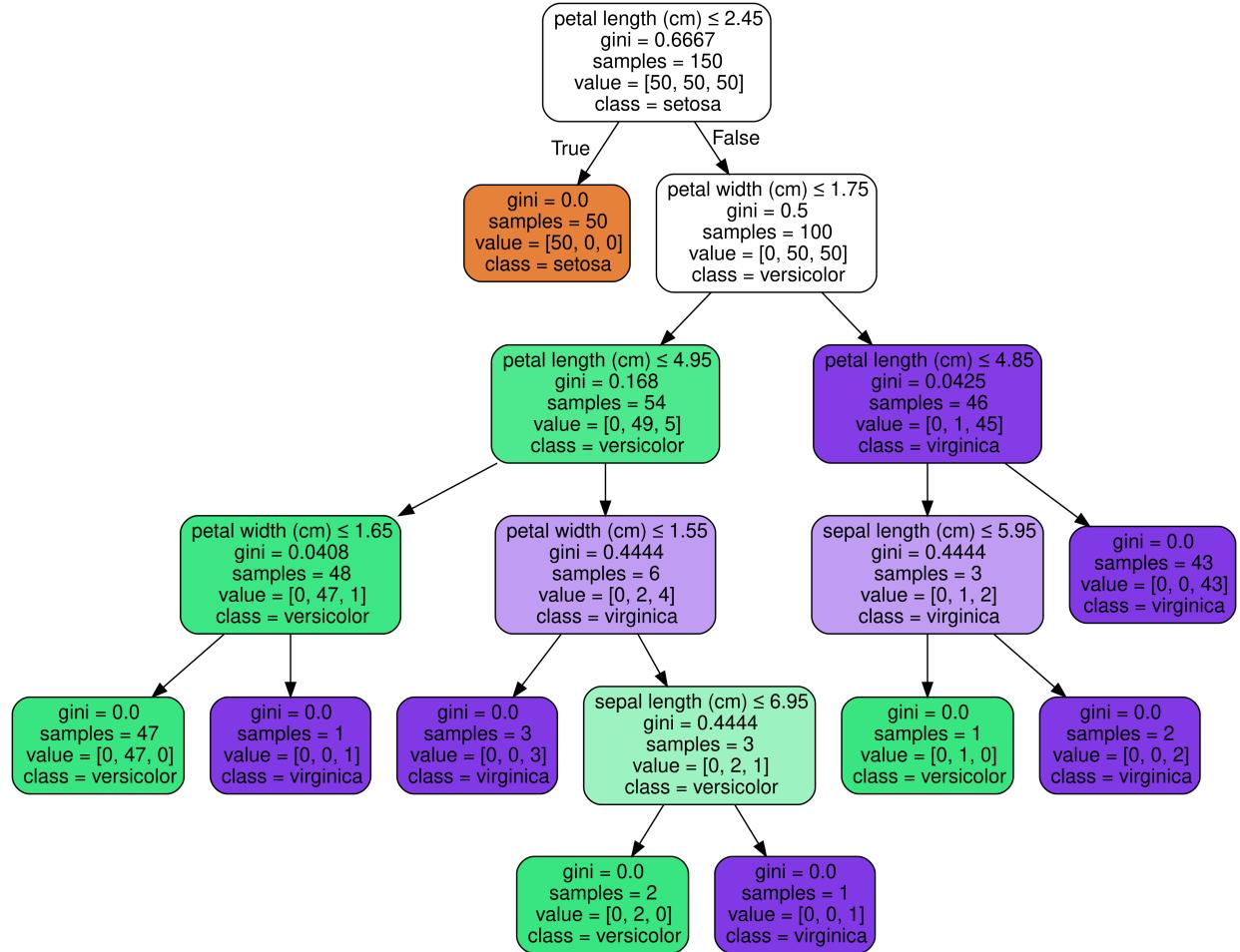


Figura 3.10: Exemplo de árvore gerado pelo método *Decision Trees*. Retirado de um exemplo da documentação do *Scikit-learn* [32]

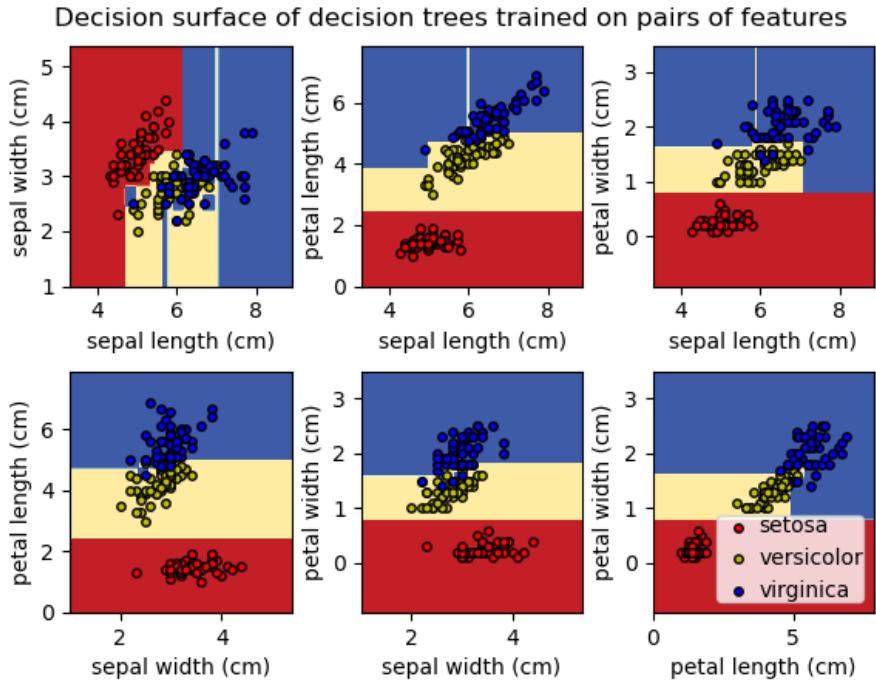


Figura 3.11: Exemplo de resultado de classificação com *Decision Trees*. Retirado de um exemplo da documentação do *Scikit-learn* [32]

O algoritmo de *Random Forest* pode ser visto como uma evolução de *Decision Trees*, sendo também como um método supervisionado para a regressão e classificação. O funcionamento do *Random Forest* vem da técnica de agrupamento conhecida como *bagging (bootstrap aggregation)*, em que são criados novos subconjunto de dados, selecionando de forma aleatória os dados iniciais com reposição, assim, são criados vários *datasets* diferentes. Com os variados *datasets* criados, são treinados vários modelos de árvores de decisão, em seguida, através de uma votação majoritária a partir dos resultados de cada árvore, é encontrado o resultado final [40] [41] [42]. A figura 3.12 descreve esse processo:

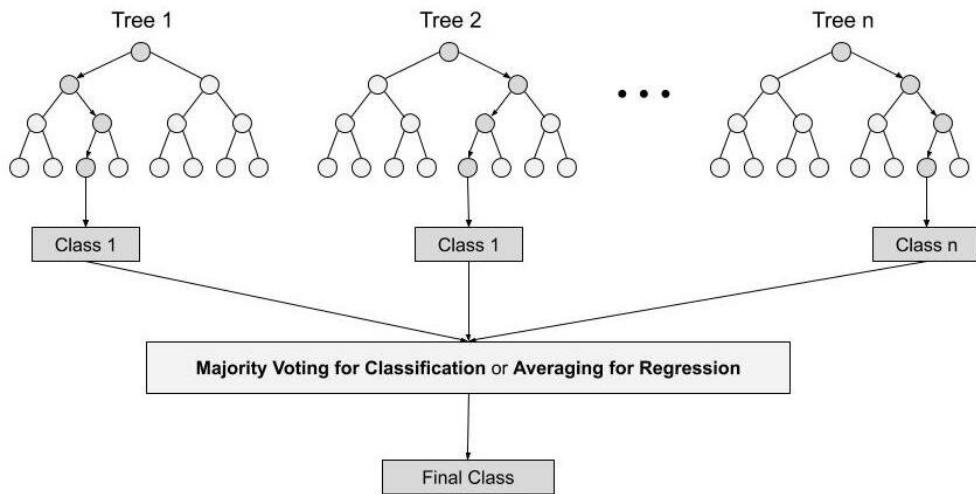


Figura 3.12: Diagrama do algoritmo *Random Forest*. Retirado de um portal de informações para *data science* [42].

Por sua vez, o *Isolation Forest* é outro algoritmo que também utiliza, assim como o *Random Forest*, árvores de decisão na sua construção. É um algoritmo não supervisionado, com o foco em detectar anomalias e *outliers* [43].

Inicialmente são criadas várias árvores de decisão (divididas de maneira aleatória) com o objetivo de isolar as observações em seus nós folhas, de forma que cada folha contenha apenas uma observação a respeito do conjunto de dados. O comprimento do caminho entre o nó raiz e os nós folhas é utilizado como medida de normalidade, sendo que quanto menor for essa medida, mais próxima a observação vai estar de ser um valor anômalo [43] [32]. A figura 3.13 exemplifica esse desenvolvimento:

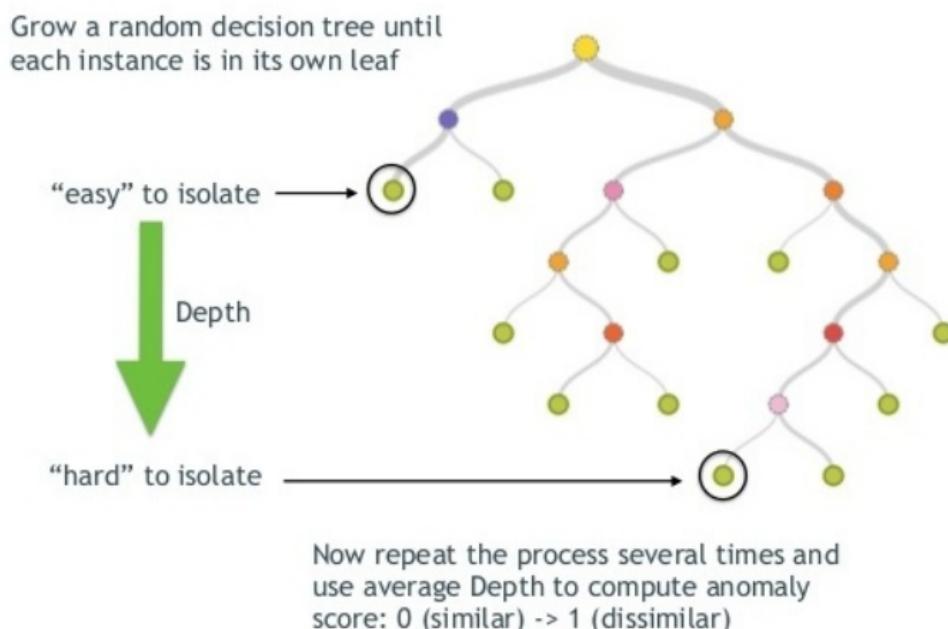


Figura 3.13: Diagrama do algoritmo *Isolation Forest*. Retirado do artigo intitulado *Anomalous Ozone Measurements Detection Using Unsupervised Machine Learning Methods* [44].

3.3.5 Redes Neurais e Aprendizado Profundo

Aprendizado profundo (ou *deep learning*) é uma sub-área extremamente extensa do ramo de aprendizado de máquina, possuindo sua base no funcionamento do cérebro humano, todavia, antes de adentrar nos conceitos de aprendizado profundo, é necessário entender o funcionamento de uma rede neural [45].

A metodologia empregada na construção de uma rede neural tem fundamento nas conexões entre os neurônios dentro do nosso cérebro, onde cada nó da rede pode ser visto como um neurônio do ponto de vista biológico. O nó é o ponto em que ocorrem os cálculos matemáticos e computacionais, no qual as entradas são ajustadas a partir de pesos, que amplificam ou atenuam esses sinais. Em seguida, esses dados ponderados são somados e passam por uma função de ativação, no qual é decidido se a informação processada é relevante ou não, sendo adicionada uma certa não-linearidade na saída [46] [25] [45]. A figura 3.14 demonstra esse processo e o equacionamento obtido:

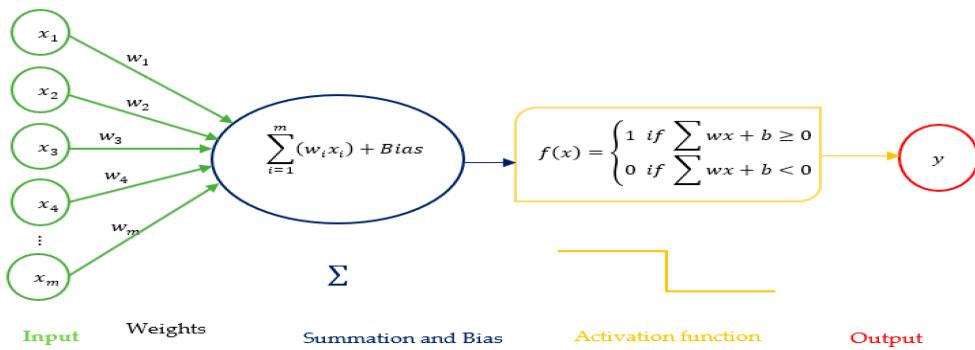


Figura 3.14: Estrutura de um nó de uma rede neural. Retirado de um artigo que demonstra a abordagem de redes neurais em assuntos financeiros [47].

Uma rede neural possui no mínimo três camadas, onde a primeira camada é conhecida como camada de entrada, a última como camada de saída e todas as outras camadas entre a de entrada e saída são chamadas de camadas ocultas. Uma rede neural superficial possui apenas uma camada oculta, por sua vez, uma rede neural profunda (utilizada em aprendizado profundo) possui de duas até milhares de camadas ocultas. Vale ressaltar que existem diversos tipos de redes neurais, tendo suas características desenvolvidas de acordo com o tipo de problema a ser resolvido [48] [45]. A figura 3.15 demonstra a estrutura de uma rede neural simples e de uma rede neural profunda (aprendizado profundo), enquanto as figuras 3.16 e 3.17 exemplificam os diversos tipos de redes neurais existentes.

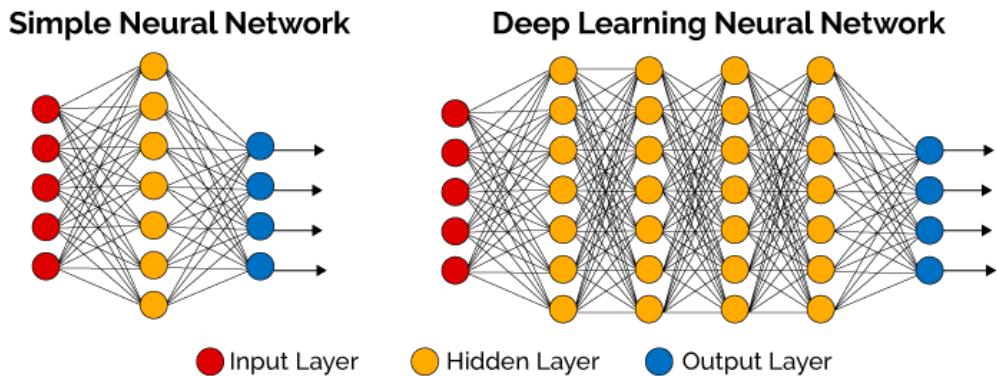


Figura 3.15: Estrutura de um nó de uma rede neural. Retirado de um livro aberto e online para ensino de aprendizado profundo [48].

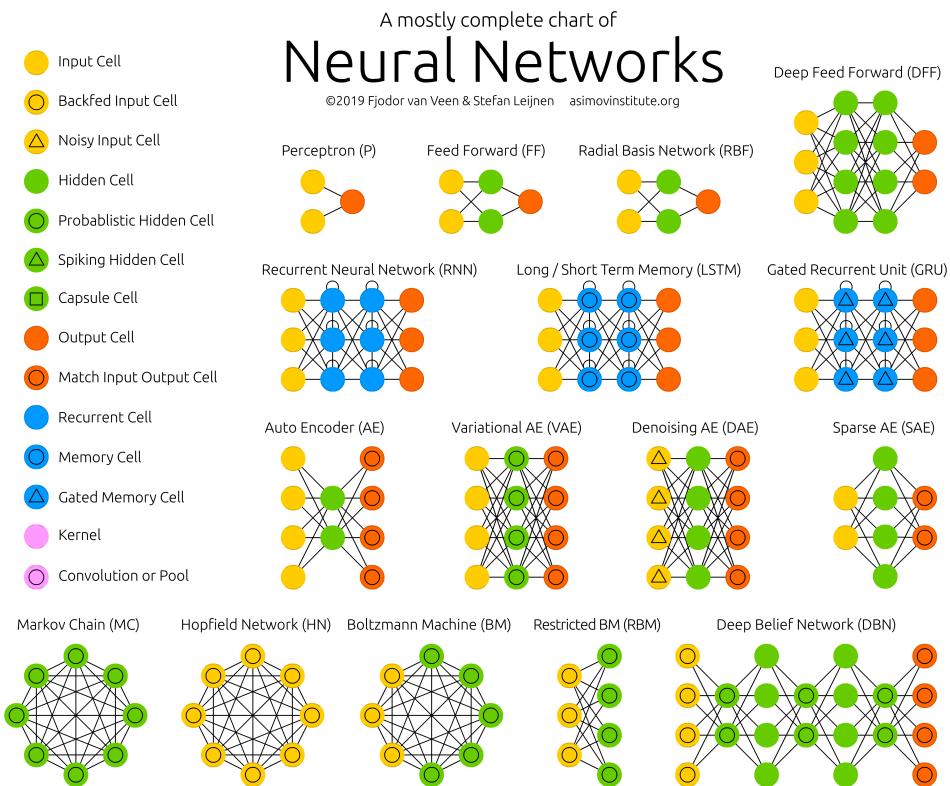


Figura 3.16: Diversos tipos de rede neural - parte 1. Retirado de um guia desenvolvido pelo *Asimov Institute* que descreve diversos tipos de rede neural [49].

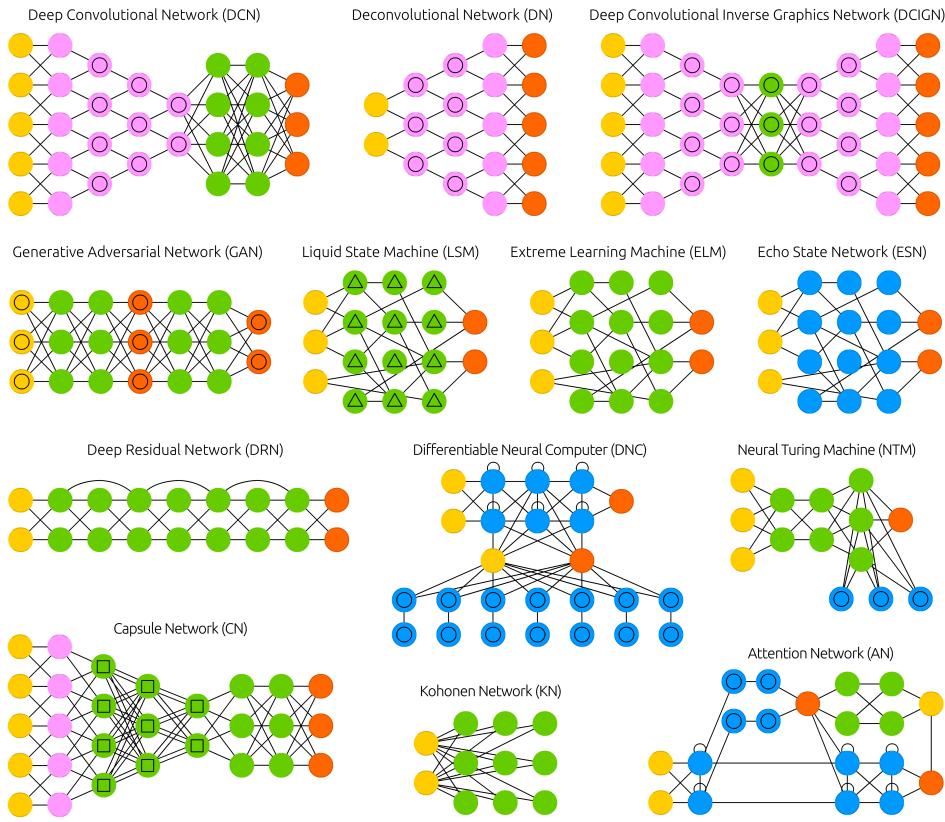


Figura 3.17: Diversos tipos de rede neural - parte 2. Retirado de um guia desenvolvido pelo *Asimove Institute* que descreve diversos tipos de rede neural [49].

Técnicas de aprendizado profundo são extremamente úteis em realizar certos processos que somente os humanos eram capazes de fazer até algum tempo atrás, como por exemplo reconhecer imagens e fala, detectar doenças, identificar padrões de comportamentos de usuários de redes sociais e sites de compras, entre outras inúmeras aplicações [50].

Podem ser utilizadas várias abordagens para a criação de modelos de aprendizado profundo, seja por meio de processos de aprendizado supervisionado, não supervisionado ou por reforço, dependendo também do problema a ser resolvido, podendo se tratar de um problema de regressão ou classificação. Graças ao enorme avanço da tecnologia nos últimos tempos, existem excelentes bibliotecas de código aberto, muito bem documentadas, que facilitam na criação de modelos de aprendizado profundo, além de disponibilizarem tutoriais para diversas aplicações [51]. Podem ser citadas aqui as bibliotecas *TensorFlow* (desenvolvida pela *Google*) [52] e *PyTorch* (desenvolvida pela equipe do *Facebook*) [53].

Para mais informações a respeito da história, base matemática, e criação de modelos de aprendizado profundo, é recomendada a leitura dos livros: *Neural Networks and Deep Learning* [46], *How smart machines think* [54] e *Deep learning with Python* [25].

3.4 CONCLUSÕES ACERCA DO CAPÍTULO

Esse capítulo proporcionou o embasamento a respeito dos tipos de teste de Covid-19 utilizados mundialmente, uma vez que tais testes servem como alicerces para comparação com os métodos computacionais utilizados na detecção da infecção por SARS-CoV-2. As informações contidas nesse capítulo também fornecem informações básicas sobre os métodos de aprendizado de máquina e aprendizado profundo que foram utilizados na realização desse trabalho.

4 METODOLOGIA

Nesse capítulo serão descritos os métodos utilizados durante o andamento do trabalho, tais como a obtenção e tratamento da base de dados, treinamento dos modelos e a descrição das abordagens adotadas. Os códigos utilizados no desenvolvimento deste trabalho estão disponíveis na forma de *jupyter notebooks* em um repositório do GitHub [55].

4.1 DELIMITAÇÃO DO ESCOPO

Os objetivos centrais do trabalho podem ser divididos em dois. O primeiro objetivo consiste em gerar alertas no momento em que forem detectadas anomalias nos dados que estão sendo processados, dando indícios de que o usuário pode estar doente, antes mesmo de ter sintomas e fazer o teste para COVID-19 ou outras doenças. Por sua vez, o segundo objetivo é dizer se, a partir dos dados coletados, o usuário teve COVID-19 ou alguma outra doença no período em que as informações estavam sendo obtidas. Vale ressaltar que esse processamento não ocorre em tempo real, uma vez que os dados já foram obtidos em um outro estudo utilizado como base para esse [1]. É importante enfatizar também que para o desenvolvimento do trabalho está sendo utilizada a linguagem de programação *Python*, em conjunto com outras bibliotecas, pois é uma linguagem de alto nível e muito bem documentada [56].

4.2 OBTENÇÃO DA BASE DE DADOS

A base utilizada foi fornecida pela revista científica *Nature*, obtida através de um estudo denominado *Pre-symptomatic detection of COVID-19 from smartwatch data* [1]. A base de dados original contém dados de 120 pacientes, relacionados à frequência cardíaca, quantidade de passos, e qualidade de sono, coletados durante um período de tempo que variou para cada usuário, podendo ser entre 1 a 4 meses. Dos 120 pacientes, 47 foram identificados com COVID-19 ou alguma outra doença respiratória. Para esse trabalho, os dados de qualidade de sono não foram utilizados, considerando apenas as amostras de 113 pessoas, sendo 46 delas identificadas com COVID-19 ou outra doença respiratória. Essa filtragem inicial é necessária uma vez que alguns dos dados da base de dados original estão corrompidos ou apresentam algum problema na etapa de filtragem e criação do *dataset*, que serão discutidas a seguir.

4.2.1 Filtragem dos dados

A etapa de filtragem dos dados possui três processos básicos: obtenção do RHR a partir dos dados de frequência cardíaca e de passos, correção sazonal e pré-processamento. A figura 4.1 mostra o diagrama de alto nível dessa etapa.

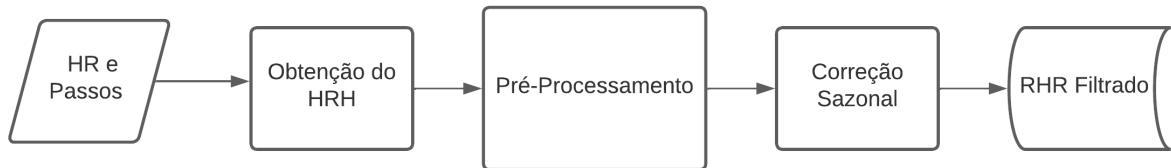


Figura 4.1: Diagrama de alto nível para a filtragem dos dados e obtenção do RHR utilizado nas etapas seguintes.

A primeira etapa do processo consiste em obter os sinal iniciais de RHR, para isso os dados de frequência cardíaca foram convertidos em dois *dataframes* separados, utilizando a biblioteca Pandas [57], em seguida os *dataframes* são unidos em um terceiro *dataframe*, tendo como base a data e horário de leitura dos dados, e por seguiante, as linhas em que os dados de passos são maior que zero no último *dataframe* a ser criado, são removidas. Se a quantidade de passos estiver zerada em determinado período, indica que o paciente estava sob repouso [1].

Na segunda etapa, de pré-processamento, os valores ausentes (*NaN values*) são deletados do sinais de RHR obtidos na etapa anterior, em seguida são suavizados através de uma média móvel de 400 períodos, e por último são reamostrados em um intervalo de 12 horas.

Na terceira e última etapa de filtragem, ocorre a correção sazonal dos sinais de RHR que passaram pela etapa de pré-processamento. A correção sazonal é necessária para remover qualquer viés devido à periodicidade na obtenção dos dados [58][23]. A figura 4.2 obtida com o auxílio da biblioteca *scikit-learn*, demonstra as componentes da sazonalidade em um sinal de RHR de um determinado paciente.

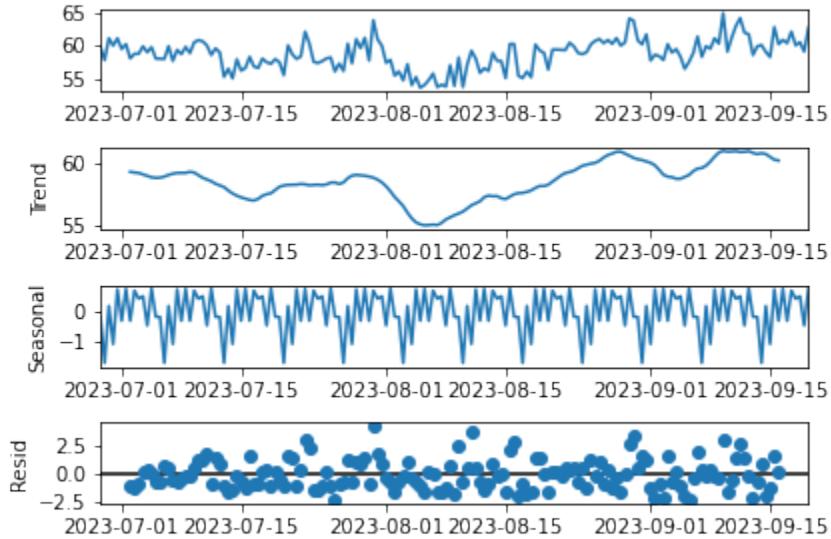


Figura 4.2: Componentes da sazonalidade em um sinal de RHR.

Após a obtenção dos sinais de RHR filtrados, os mesmos são utilizados no modelo *Isolation Forest* de forma individual. Para os demais modelos (*Decision Trees*, *Random Forest* e *Auto-encoder*), todos os sinais são concatenados em apenas um *dataframe*, tendo os valores ausentes preenchidos com a mediana de cada sinal individual de RHR, passando também os rótulos 0 ou 1 na última coluna. O valor 0 no *dataframe* indica que o paciente permaneceu saudável no período de aquisição dos dados (não teve COVID-19 ou alguma outra doença respiratória), enquanto que o valor 1 indica que o paciente testou positivo para COVID-19 ou alguma outra doença respiratória. A figura 4.3 mostra o arranjo do *dataframe* obtido, no qual cada *feature* representa a média da frequência cardíaca obtida no período de reamostragem dos dados (a cada 12 horas). É válido esclarecer que alguns trechos de códigos, utilizados para as etapas de filtragem e pré-processamento, foram adaptados dos códigos que foram fornecidos pelo estudo que serviu como base para a realização desse trabalho [1].

	0	1	2	3	4	5	6	7	8	9	...	238	239	240
1	A06L7KF	68.952997	66.533750	66.955455	70.242966	67.118095	65.252242	62.757705	63.828825	61.209766	...	64.446450	64.446450	0
2	A0822M0	69.663125	69.772143	69.560663	69.110682	69.379625	69.630179	69.593750	69.750385	69.187344	...	66.023472	66.023472	0
3	A0KX894	75.059714	74.688333	73.722344	73.987500	74.179659	74.210665	75.404000	75.470735	75.103750	...	69.456114	69.456114	1
4	A0L9BM2	75.728333	75.727063	75.780263	75.055469	75.196172	75.180847	74.849000	73.944594	73.946354	...	74.518000	74.518000	0
5	A0NVTRV	81.401250	81.590408	80.735000	81.565429	83.601250	85.440395	85.076250	84.739107	84.937551	...	78.353409	78.353409	1
...
109	AYWIEKR	71.641667	71.998750	72.726792	73.001471	72.732138	73.207542	73.454009	73.497187	73.355000	...	72.869604	72.869604	1
110	AZ2RYW7	74.464821	74.588974	73.761094	73.680000	73.553777	73.577368	73.089542	72.891477	73.490784	...	73.772422	73.772422	0
111	AZ35PI5	76.989706	77.800179	76.737350	74.224537	72.903022	73.892089	71.835729	70.377500	70.803529	...	71.013131	71.013131	0
112	AZIK4ZA	58.874189	59.155714	58.407000	58.321196	58.451429	58.280329	58.084375	57.955303	58.122833	...	59.026786	59.026786	1
113	AZKZ0AI	79.839167	80.435600	80.730000	81.257292	81.333500	81.421919	81.679375	81.559844	81.571875	...	79.300272	79.300272	0

Figura 4.3: Arranjo do *dataframe* obtido a partir dos sinais individuais de RHR.

4.3 TREINAMENTO DOS MODELOS

Nessa seção serão abordados os tópicos referentes à metodologia utilizada no treinamento dos modelos para detecção de COVID-19 e outras doenças respiratórias, assim como a configuração dos parâmetros de cada modelo.

Os modelos que utilizam *Decision Trees*, *Random Forest* e *Isolation Forest* foram implementados através da biblioteca *scikit-learn* [32], por sua vez, o modelo que utiliza *Autoencoder Neural Network* foi implementado através da biblioteca *TensorFlow* [52]. Destaca-se também que para o algoritmo *Isolation Forest*, os sinais de RHR servem como entradas individuais, com o modelo identificando separadamente as anomalias nos RHRs de cada paciente. Por sua vez, para os demais modelos, é utilizado o *dataframe* gerado (exemplificado na figura 4.3), pois é necessário utilizar dados rotulados uma vez que tratam-se de modelos supervisionados.

As definições dos hiperparâmetros utilizados pelos modelos da biblioteca *scikit-learn* foram retiradas da documentação da própria biblioteca, sendo aqueles que não são citados nesta seção, foram definidos da forma padrão da documentação [32]. Vale deixar claro também que os valores de vários hiperparâmetros foram obtidos de forma empírica, sendo necessário testar inúmeras vezes os modelos para tal obtenção.

4.3.1 *Decision Trees* e *Random Forest*

Os modelos que utilizam *Decision Trees* e *Random Forest*, têm como objetivo classificar se um determinado paciente teve ou não COVID-19 durante o período de leitura dos dados. Utilizam como *dataset* o *dataframe* rotulado que foi criado com os dados de RHR de todos os pacientes, sendo 75% dos dados utilizados para treinamento dos modelos e 25% para teste. A figura 4.4 demonstra o funcionamento dos modelos por meio de um diagrama de alto nível, sendo que o resultado 0 significa que o indivíduo não adquiriu COVID-19 ou alguma outra doença respiratória durante a aquisição dos dados, em contrapartida, o resultado 1 significa o contrário.

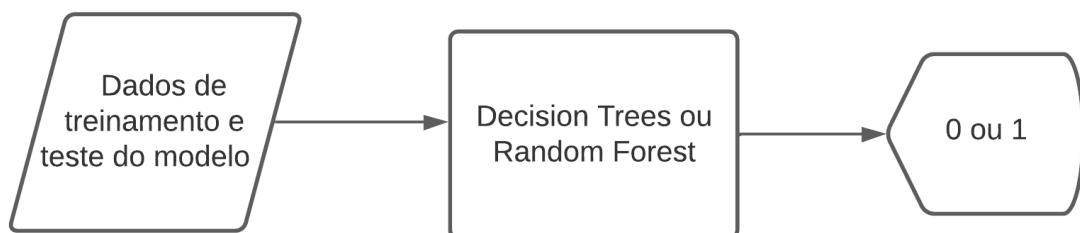


Figura 4.4: Diagrama de alto nível para o *Decision Trees* e *Random Forest*.

Para o treinamento do modelo que utiliza *Decision Trees*, os hiperparâmetros foram definidos da forma a seguir:

- $\max_depth = 6$ - profundidade máxima da árvore;
- $\min_samples_leaf = 25$ - valor mínimo de amostras necessárias para estar em um nó folha;
- $\random_state = 0$ - definido dessa forma, faz com que a pseudo-aleatoriedade não exista na seleção de *features*, possibilitando uma melhor comparação quando os demais parâmetros são modificados para fins de teste.

Por sua vez, para o treinamento do modelo que utiliza *Random Forest*, os hiperparâmetros foram definidos da seguinte maneira:

- $n_estimators = 10$ - quantidade de árvores de decisão que o modelo utiliza;
- $\min_samples_leaf = 15$ - mesma definição do modelo *Decision Trees*;
- $\random_state = 0$ - mesmo propósito do modelo *Decision Trees*.

4.3.2 Isolation Forest

O modelo que utiliza *Isolation Forest* funciona detectando anomalias nos dados, sem a necessidade de estarem rotulados. O modelo tem como entrada o RHR filtrado, a partir disso, é obtido como saída a pontuação de anomalia dos dados de entrada. Quanto menor é o valor da pontuação de determinado dado, mais ele se aproxima de um dado anômalo, dessa forma, dados cuja pontuação de anomalia é negativa, são rotulados como anômalos ou *outliers*. A figura 4.5 demonstra o processo através de um diagrama de alto nível:

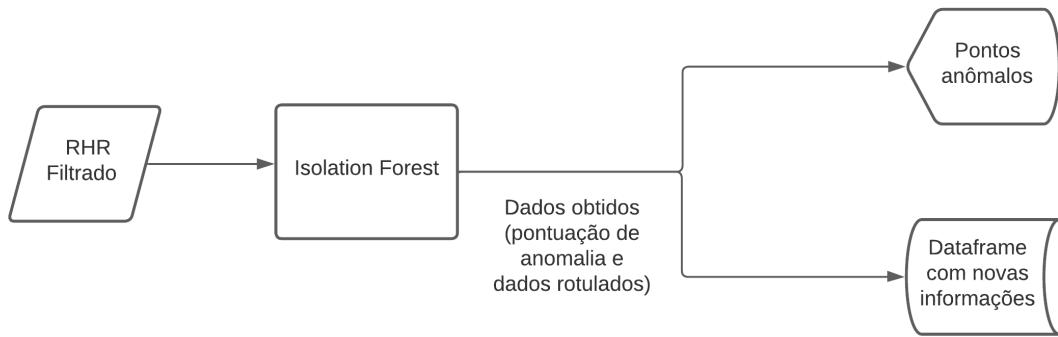


Figura 4.5: Diagrama de alto nível para a utilização do *Isolation Forest*.

Para o treinamento do modelo, foram utilizados os seguintes hiperparâmetros como base, sendo que os mesmos foram alterados diversas vezes durante os testes:

- $n_estimators = 100$ - número de árvores de decisão (ou estimadores de base) utilizado;
- $max_samples = \text{'auto'}$ - quantidade de amostras obtidas para treinar cada estimador de base;
- $contamination = 0.12$ - quantidade de contaminação do conjunto de dados, usado para definir o limite nas pontuações das amostras;
- $random_state = 42$ - controla a pseudo-aleatoriedade da seleção de *features* e os valores de divisão para cada etapa de ramificação.

4.3.3 Autoencoder Neural Network

O modelo que utiliza o *Autoencoder Neural Network* tem como objetivo detectar anomalias nos dados. A utilização do mesmo foi baseada em um tutorial do *TensorFlow*, cujo objetivo era de detectar batimentos ectópicos através da rede neural [52], no qual o modelo aprendia com os dados de entrada o que eram batimentos normais e batimentos ectópicos, em seguida, através do modelo treinado, eram comparados os sinais originais de entrada com os sinais reconstruídos, quando a reconstrução falhava, poderia ser um indício da presença de um batimento ectópico.

No presente trabalho, o *autoencoder* está sendo utilizado para detectar COVID-19 ou outras doenças respiratórias, aprendendo como são os sinais de RHR de indivíduos que não contraíram COVID-19. Dessa forma, na reconstrução do RHR pelo modelo, quando o valor do erro de reconstrução é alto, pode ser um indicativo da presença de anomalia. Os dados foram divididos em 75% para treinamento e 25% para teste do modelo.

Para a criação do modelo, foram utilizadas três camadas ocultas totalmente conectadas para o codificador (*encoder*), e três para o decodificador (*decoder*). O esquemático da figura 4.6 representa o modelo criado, sendo *units* referente à dimensionalidade do vetor de cada camada e *activation* à função de ativação utilizada.

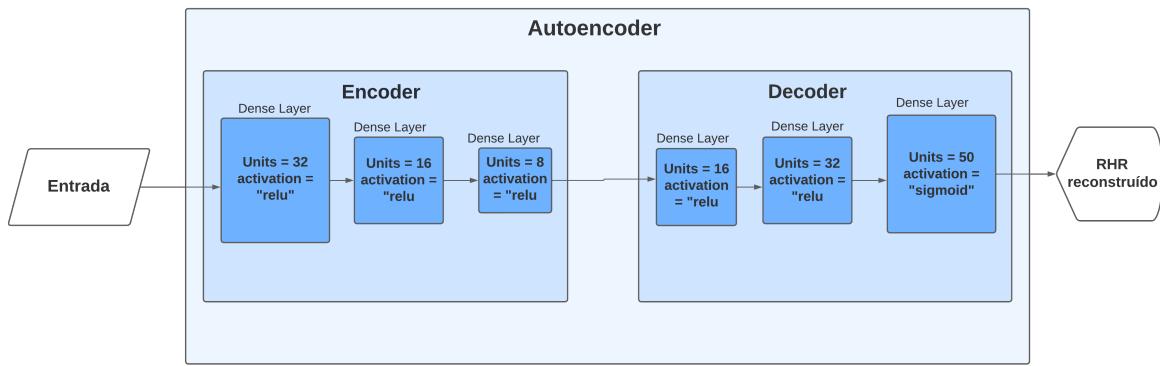


Figura 4.6: Diagrama de alto nível para o *Autoencoder Neural Network*.

Para o treinamento do modelo, os hiperparâmetros foram definidos da seguinte forma :

- *optimizer* = 'adam' - altera os atributos do modelo a fim de reduzir as perdas;
- *loss* = 'mae' - função de perda, calcula a quantidade que um modelo deve procurar minimizar durante o treinamento (mesma função utilizada no exemplo do *TensorFlow*);
- *epochs* = 30000 - quantidade de ciclos que o algoritmo percorre o *dataset*;
- *batch_size* = 4 - número de amostras de treinamento utilizadas em cada iteração ;
- *shuffle* = *True* - embaralha os dados com o objetivo de evitar relações entre os mesmos.

Para mais informações acerca do *Autoencoder Neural Netwrk* é recomendada a leitura do livro *Deep Learning*, dos autores Ian Goodfellow, Yoshua Bengio e Aaron Courville [59], por sua vez, mais informações acerca dos hiperparâmetros e configuração do modelo, podem ser obtidas na documentação do mesmo [52] [60].

4.4 CONCLUSÕES ACERCA DO CAPÍTULO

A leitura deste capítulo proporcionou o entendimento a respeito da obtenção e tratamento dos dados utilizados ao longo do trabalho, as informações e objetivos por trás das abordagens empregadas, assim como as definições dos hiperparâmetros utilizados na criação e treinamento de cada modelo.

A utilização da biblioteca *pandas* facilita na etapa de pré-processamento dos dados e criação dos *datasets*, assim como as bibliotecas *scikit-learn* e *TensorFlow* simplificam bastante na criação e treinamento dos modelos, sendo de fácil usabilidade e com muita informação disponível na internet para consulta.

5 RESULTADOS

Nesse capítulo serão discutidos os resultados obtidos em cada abordagem, sendo que cada uma possui um objetivo distinto da outra. Os métodos que utilizam *Decision Trees* e *Random Forest* têm como objetivo apenas informar se determinado indivíduo teve ou não COVID-19 ou alguma outra doença respiratória durante o período no qual os dados foram coletados. A abordagem que utiliza *Isolation Forest* é capaz de detectar anomalias nos dados de cada indivíduo, podendo informar se o mesmo encontra-se com COVID-19 ou outra doença antes mesmo de ter sintomas. Por sua vez, o método que utiliza o *Autoencoder* tem como foco avaliar a utilização de redes neurais na detecção de COVID-19 ou outras doenças, por meio de detecção de anomalias nos dados, porém difere da metodologia empregada no *Isolation Forest* devido à necessidade de treinamento prévio com dados rotulados. Mais detalhes serão dados nas seções a seguir.

5.1 UTILIZANDO DECISION TREES E RANDOM FOREST

Inicialmente, os valores dos hiperparâmetros que apresentavam resultado razoável de precisão para o *Decision Trees* foram encontrados empiricamente, sendo eles max_depth igual a 6 e min_samples_leaf definido como 22. Estabelecendo o hiperparâmetro $\text{min_samples_leaf} = 22$ e fazendo o hiperparâmetro max_depth variar, foi obtido o gráfico da figura 5.1. Através da análise do gráfico é possível perceber que $\text{max_depth} = 6$ é uma boa estimativa. Em seguida, fixando $\text{max_depth} = 6$ e fazendo min_samples_leaf variar, foi gerado o gráfico da figura 5.2. Dessa forma, para criar o modelo mais confiável possível em relação ao objetivo pretendido, os melhores valores para os hiperparâmetros foram definidos como $\text{max_depth} = 6$ e $\text{min_samples_leaf} = 25$.

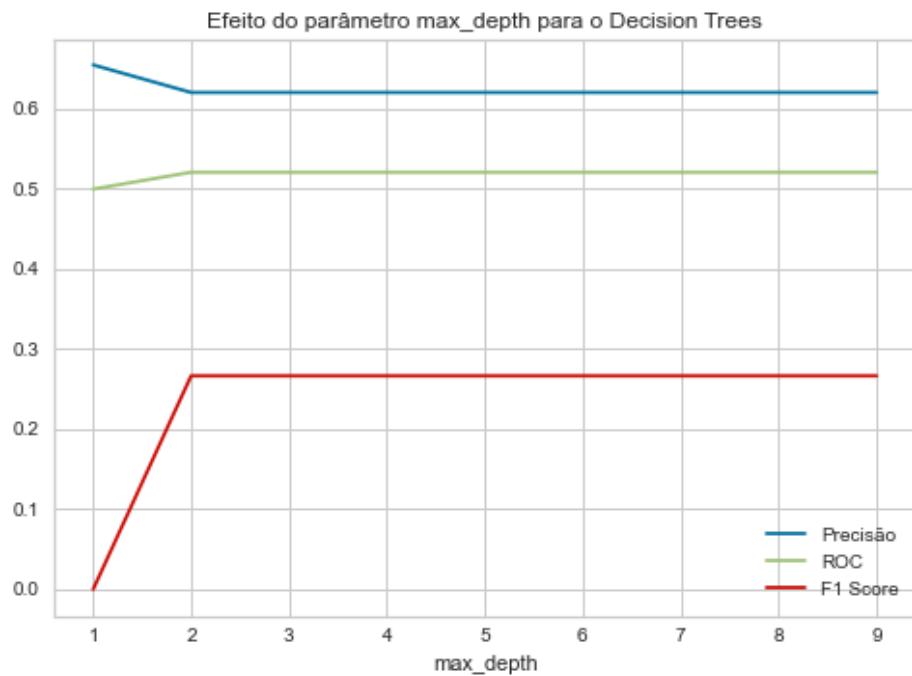


Figura 5.1: Efeito do parâmetro `max_depth` para o *Decision Trees*

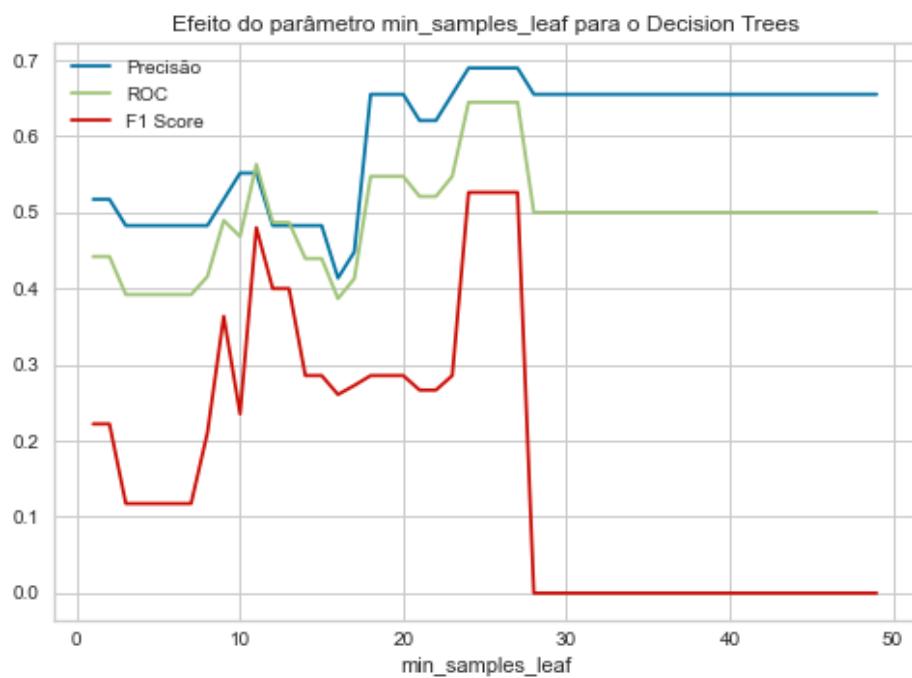


Figura 5.2: Efeito do parâmetro `min_samples_leaf` para o *Decision Trees*

Com os hiperparâmetros do modelo para o *Decision Trees* definidos, foi obtida a matriz de confusão demonstrada na figura 5.3. Na matriz, o campo descrito como "Verdadeiro"significa a classificação como positivo para COVID-19 ou alguma outra doença respiratória, por sua vez o campo denominado "Falso"significa que o resultado é negativo. Analisando a matriz de confusão é possível perceber que para os casos de verdadeiro-positivos (quando o indivíduo realmente contraiu COVID-19 e o modelo acertou a predição) é obtido um percentual razoável de 79%, todavia, o percentual para falso-positivos apresenta o mesmo valor que verdadeiro-negativos (50%), isto é, existe a mesma probabilidade de um indivíduo que na realidade não contraiu a doença ter um resultado positivo, assim como ser classificado corretamente como negativo. Ainda em relação ao modelo, a probabilidade de se obter falso-negativos é de 21%, que para o objetivo do trabalho não é um resultado ruim, pois no cenário de detecção de doenças facilmente transmissíveis, é mais vantajoso ter uma probabilidade maior de ter resultados falso-positivos do que falso-negativos.

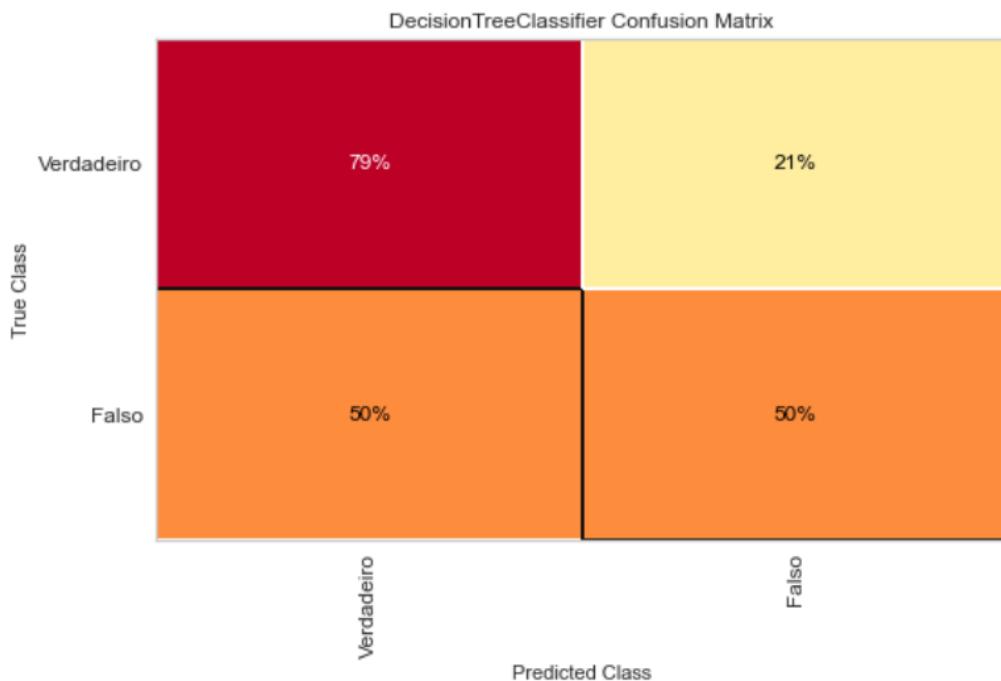


Figura 5.3: Matriz de confusão para o *Decision Trees* quando $\text{max_depth} = 6$ e $\text{min_samples_leaf} = 25$

O mesmo procedimento foi adotado para o modelo que utiliza *Random Forest*, com a diferença que ao invés de utilizar o hiperparâmetro max_depth , foi utilizado o hiperparâmetro $n_estimators$. Fixando min_samples_leaf com valor 15 e fazendo o $n_estimators$ variar, foi obtido o gráfico da figura 5.4, em seguida adotando $n_estimators = 10$ e fazendo min_samples_leaf variar, foi obtido o gráfico mostrado na figura 5.5. Analisando os gráficos, é possível concluir que $n_estimators = 10$ e $\text{min_samples_leaf} = 15$ são ótimas escolhas para o modelo de *Random Forest*.

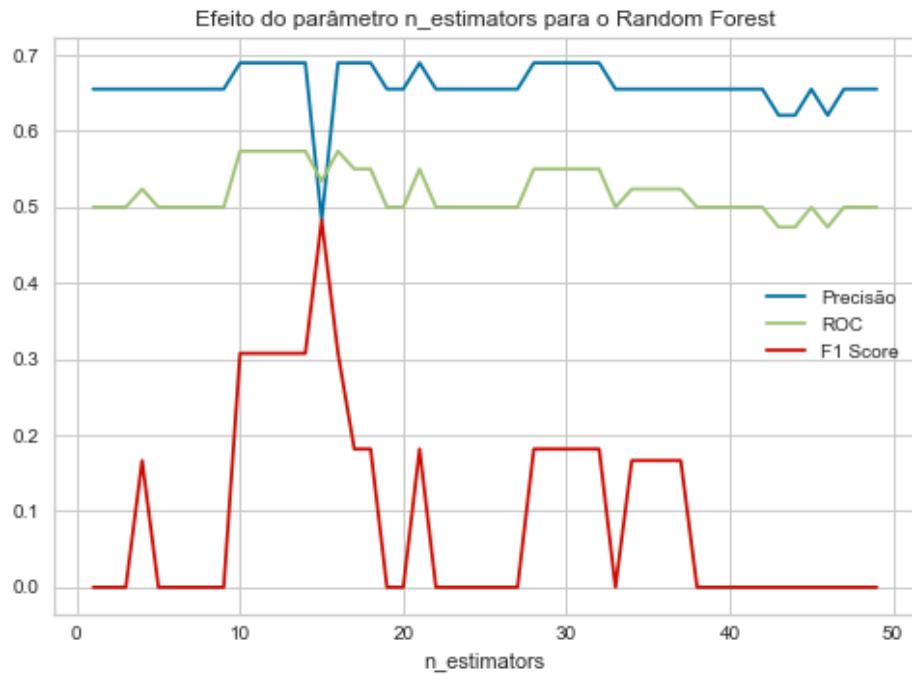


Figura 5.4: Efeito do parâmetro *n_estimators* para o Random Forest

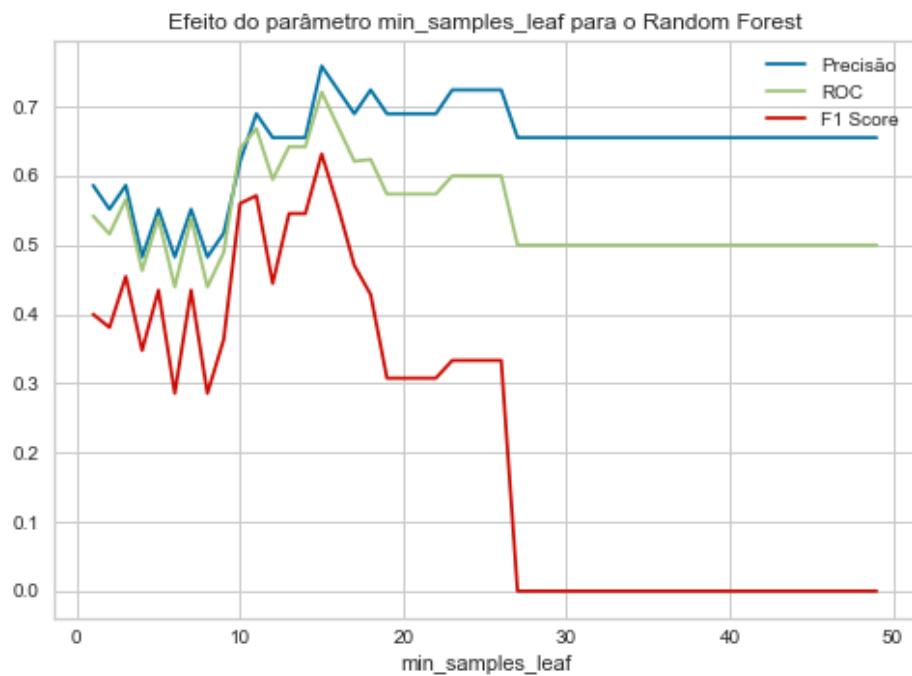


Figura 5.5: Efeito do parâmetro *min_samples_leaf* para o Random Forest

Após definir os valores ideais para os hiperparâmetros do *Random Forest*, foi gerada a matriz de confusão presente na figura 5.6. Observando a matriz de confusão, percebe-se que a mesma apresenta resultados superiores aos que foram obtidos para o modelo *Decision Trees*. A probabilidade de detectar COVID-19 quando realmente o indivíduo está doente é de 84%, enquanto que as probabilidades de serem obtidos resultados falso-positivos, falso-negativos e verdadeiro-negativos são de 40%, 16% e 60%, respectivamente.

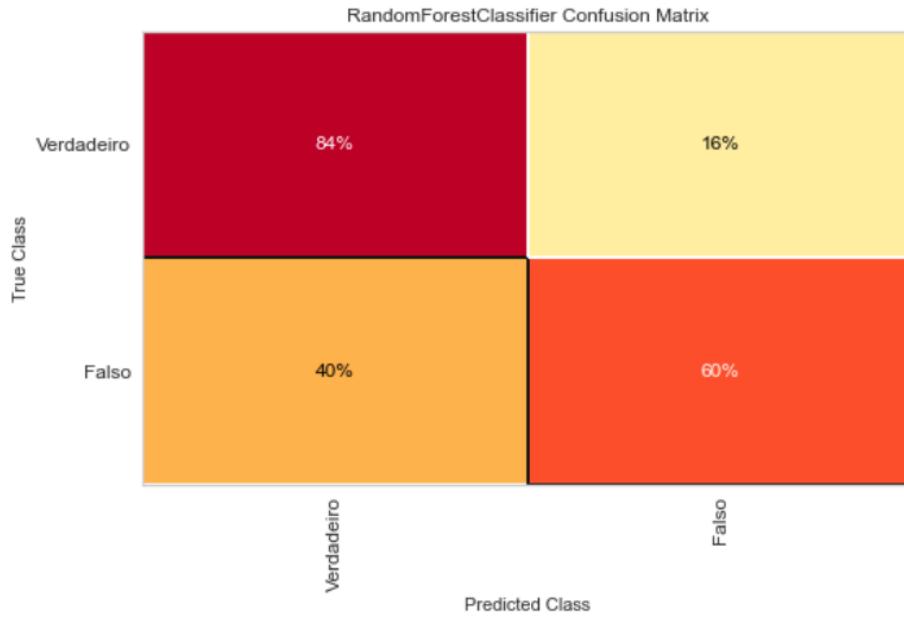


Figura 5.6: Matriz de confusão para o *Random Forest* ($\text{min_samples_leaf} = 15$)

As tabelas 5.1 e 5.2 resumem os resultados obtidos para os modelos que utilizam *Decision Trees* e *Random Forest* quando têm seus hiperparâmetros ajustados da melhor forma possível. O modelo *Random Forest* demonstrou mais robustez, uma vez que obteve valores mais altos de precisão, *F1 Score* e ROC, possuindo uma maior taxa de acerto nas previsões, como observado nas matrizes de confusão de ambos os modelos.

Métricas	Método	
	Decision Trees	Random Forest
Precisão	0,690	0,759
F1 Score	0,526	0,632
ROC	0,644	0,721

Tabela 5.1: Métricas obtidas para *Decision Trees* e *Random Forest* com os valores ideias de hiperparâmetros ajustados.

Método		
Testes Diagnósticos	Decision Trees	Random Forest
Verdadeiro-positivo	79%	84%
Falso-positivo	50%	40%
Verdadeiro-negativo	50%	60%
Falso-negativo	21%	16%

Tabela 5.2: Percentual dos testes diagnósticos para *Decision Trees* e *Random Forest* com os valores ideias de hiperparâmetros ajustados.

É válido destacar que existe uma limitação de utilização para os algorítimos, dada a forma que a base de dados fornecida foi gerada. Torna-se muito difícil a rotulação para trechos dos dados de RHR, que poderia possibilitar a detecção de doenças em diferentes períodos de leitura dos dados, e não apenas indicar que determinado usuário adoeceu ou não durante toda época em que os dados foram coletados.

5.2 UTILIZANDO *ISOLATION FOREST*

O modelo que utiliza *Isolation Forest* não só conseguiu detectar COVID-19 ou outras doenças respiratórias no período de aparecimento de sintomas, como também alguns dias antes em alguns casos. As datas de aparecimento de sintomas, assim como as datas de diagnóstico, foram fornecidas através do estudo no qual esse trabalho se baseia, juntamente com a base de dados que foi utilizada [1]. É valido esclarecer que os valores das datas não coincidem com a realidade, uma vez que tal medida foi necessária para proteger as identidades das pessoas que participaram do estudo original.

As figuras 5.7 e 5.8 mostram os resultados obtidos através do *Isolation Forest* para alguns indivíduos. A presença de pontos isolados marcados nos gráficos pode ser desconsiderada, enquanto que trechos que apresentam uma densidade maior de pontos devem ser levados em consideração, uma vez que representam as anomalias que de fato podem indicar a presença de COVID-19 ou alguma outra doença.

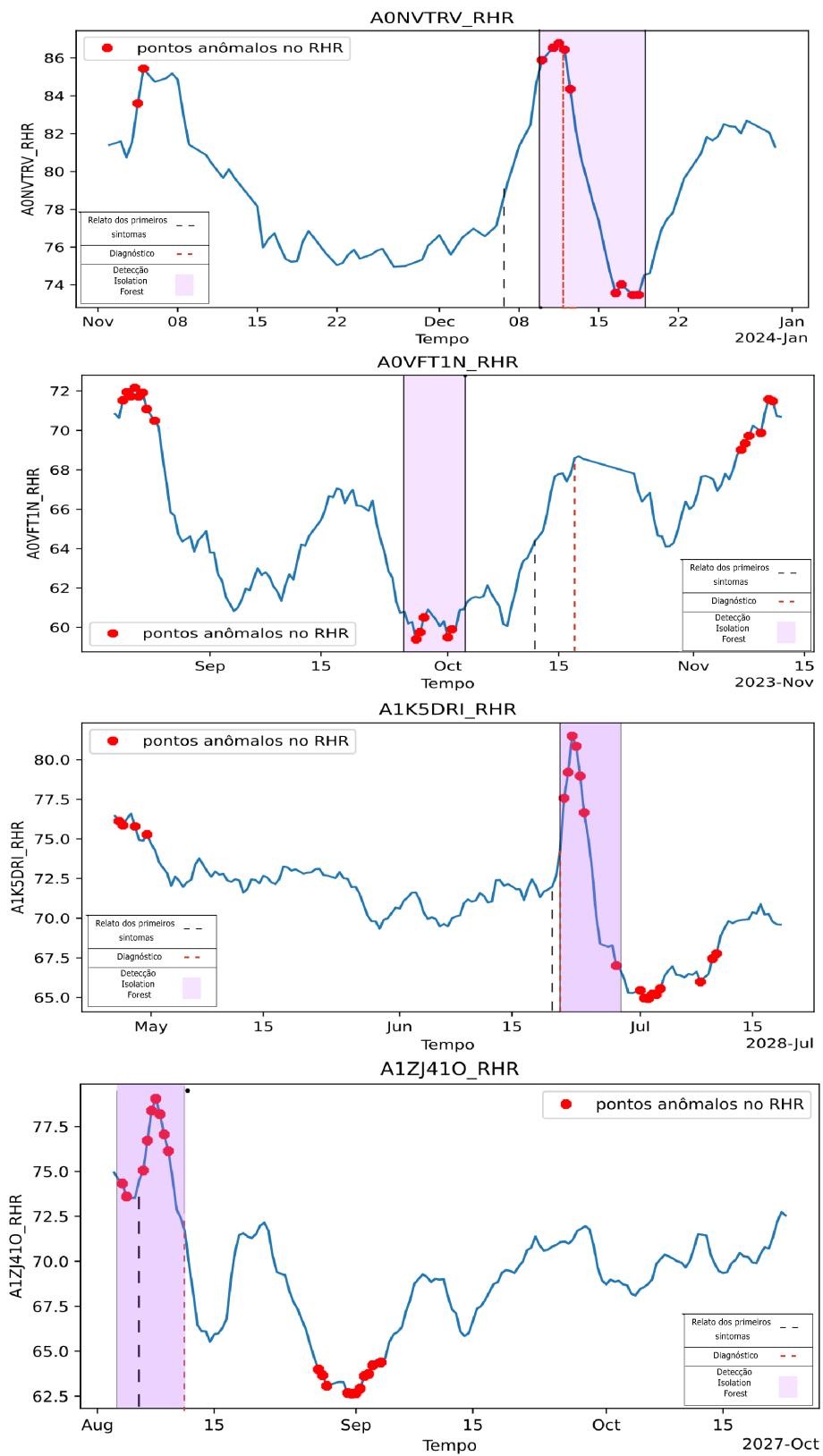


Figura 5.7: Resultados do *Isolation Forest*: pontos anômalos identificados no RHR, período de detecção, data de inicio de sintomas e data do diagnóstico.

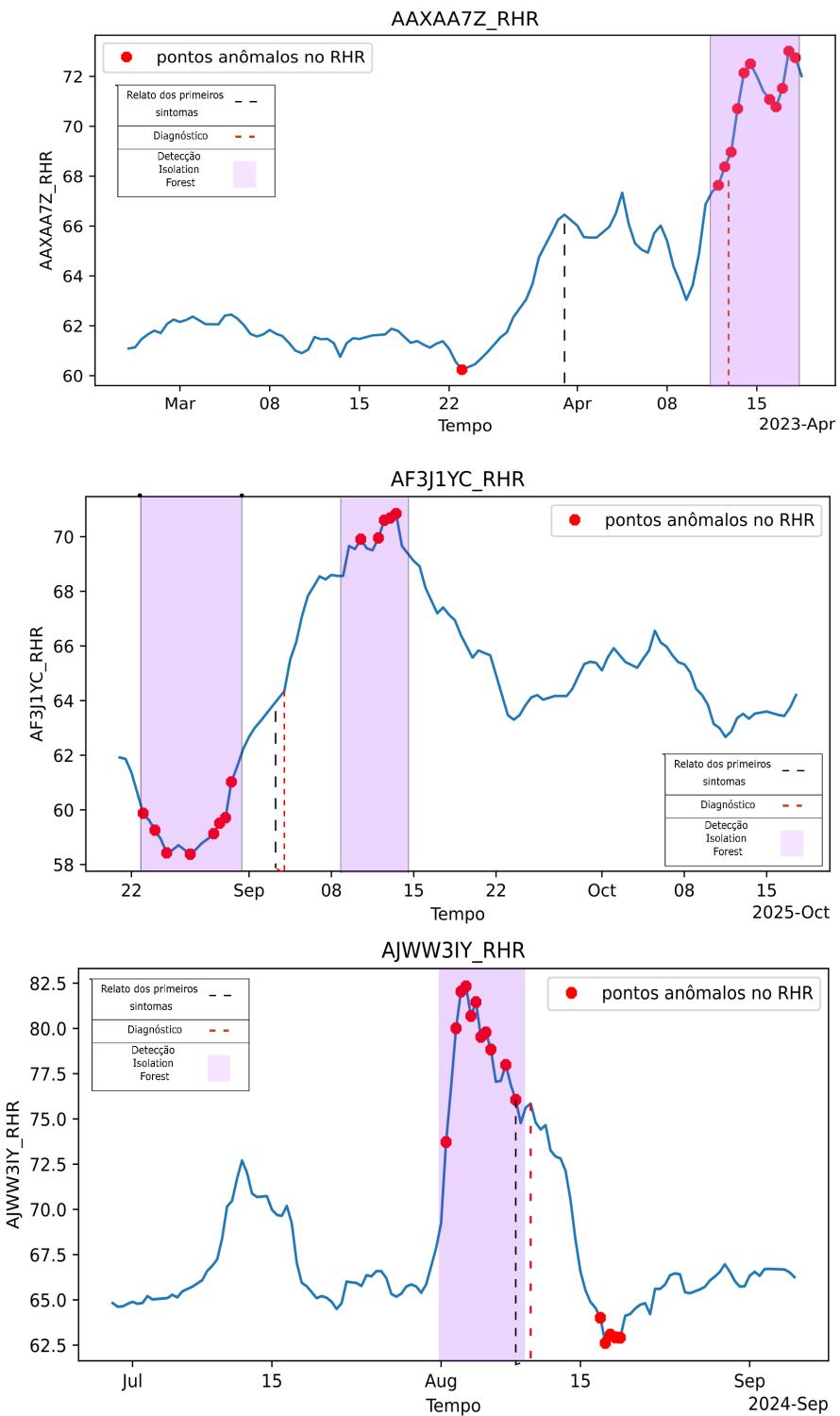


Figura 5.8: Resultados do *Isolation Forest*: pontos anômalos identificados no RHR, período de detecção, data de inicio de sintomas e data do diagnóstico.

A tabela 5.3 apresenta o resumo dos resultados obtidos através do método. Como demonstrado na tabela, o modelo conseguiu detectar outras doenças respiratórias além de COVID-19, assim como foi capaz de identificar doenças antes dos relatos de sintomas iniciais dos indivíduos. É importante esclarecer que outros trechos nos sinais de RHR (demonstrados nas figuras 5.7, 5.8 que possuem alta densidade de pontos anômalos e não foram identificados no estudo original, podem indicar a presença de outras doenças ou algum fenômeno que alterou os batimentos cardíacos dos indivíduos naquele período de tempo.

Identificador	Período de detecção: Isolation Forest	Relato dos primeiros sintomas	Data de diagnóstico	Diagnóstico
A0NVTRV	10/12/2023 - 18/12/2023	06/12/2023	11/12/2023	COVID-19
A0VFT1N	27/09/2023 - 01/10/2023	13/10/2023	16/10/2023	COVID-19
A1K5DRI	21/06/2028 - 28/06/2028	20/06/2028	21/06/2028	COVID-19
A1ZJ41O	04/08/2027 - 09/08/2027	06/08/2027	10/08/2027	COVID-19
AAXAA7Z	12/04/2023 - 18/04/2023	30/03/2023	13/04/2023	COVID-19
AF3J1YC	23/08/2025 - 13/09/2025	02/09/2025	03/09/2025	Influenza B
AJWW3IY	01/08/2024 - 19/08/2024	09/08/2024	10/08/2024	COVID-19

Tabela 5.3: Resumo dos resultados obtidos por meio do Isolation Forest

5.3 UTILIZANDO AUTOENCODER NEURAL NETWORK

Tendo como base os hiperparâmetros descritos para o *Autoencoder* no capítulo 4, após o modelo ser treinado, foram obtidos os resultados demonstrados na figura 5.9 para as curvas de perda de validação e de treinamento. Observando a figura, é possível perceber que ocorreu *underfitting* para o modelo, pois a curva de perda de validação permanece acima da curva de perda de treinamento durante todo o tempo de treinamento. Esse resultado já era esperado, pois a quantidade de dados utilizada para treinamento era bastante limitada. Já na figura 5.10, é demonstrado o acréscimo da acurácia a medida que vão aumentando as épocas durante o treinamento, estabelecendo-se em torno de 64%.

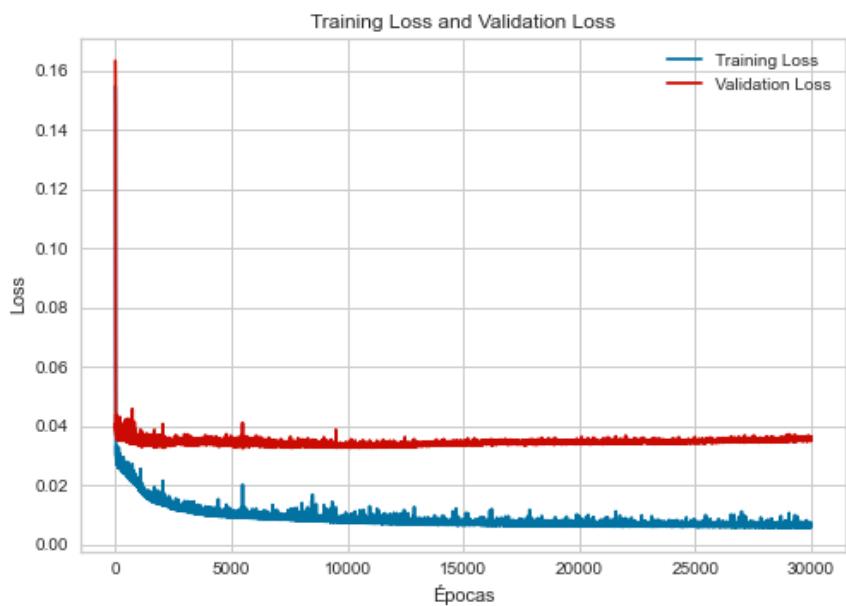


Figura 5.9: Perda de validação e de treinamento para o *Autoencoder Neural Network*

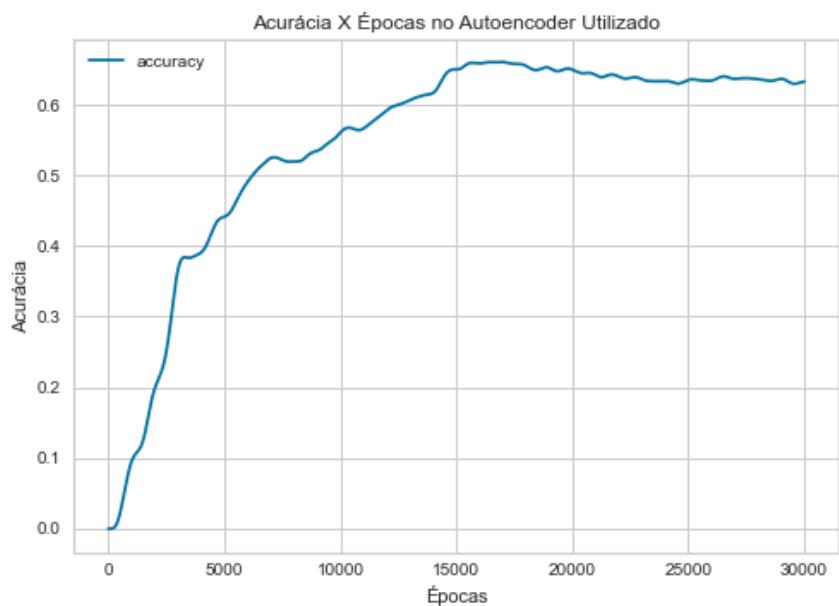


Figura 5.10: Curva de acurácia para o *Autoencoder*

As figuras 5.11 e 5.12 demonstram o processo de reconstrução de um RHR de um indivíduo que não teve COVID-19 ou alguma outra doença respiratória durante a aquisição dos dados. Analisando as figuras, é possível observar que os sinais reconstruídos se assemelham de certa forma com os sinais originais. As figuras 5.13 e 5.14 demonstram o processo de reconstrução de sinais em que os indivíduos foram diagnosticados com COVID-19 ou outra doença respiratória. Observa-se que os sinais reconstruídos pelos modelos claramente se diferenciam dos sinais originais.

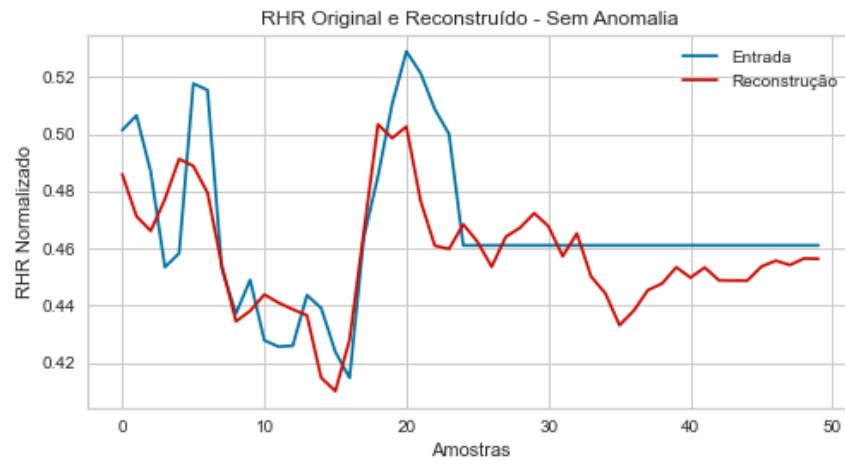


Figura 5.11: Resultado da reconstrução de um RHR normal pelo *Autoencoder*.

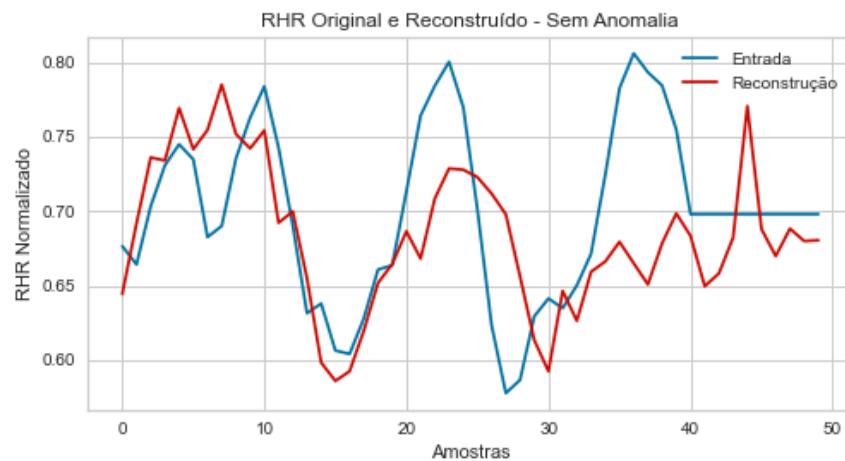


Figura 5.12: Resultado da reconstrução de um RHR normal pelo *Autoencoder*

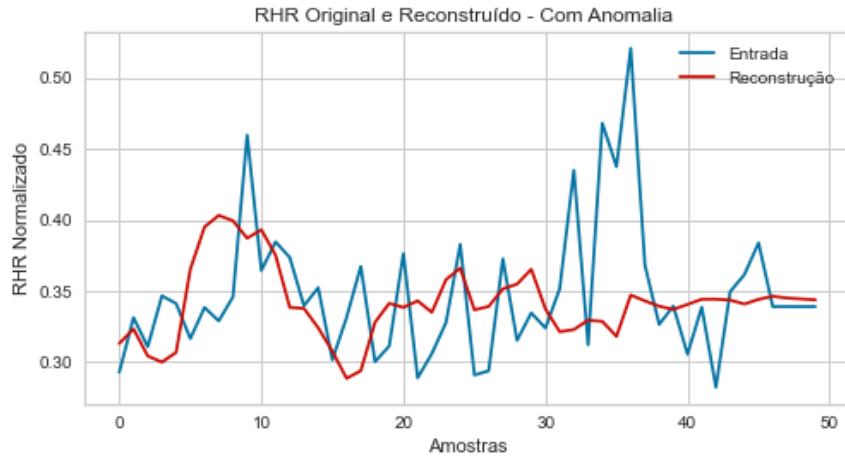


Figura 5.13: Resultado da reconstrução de um RHR anormal pelo *Autoencoder*

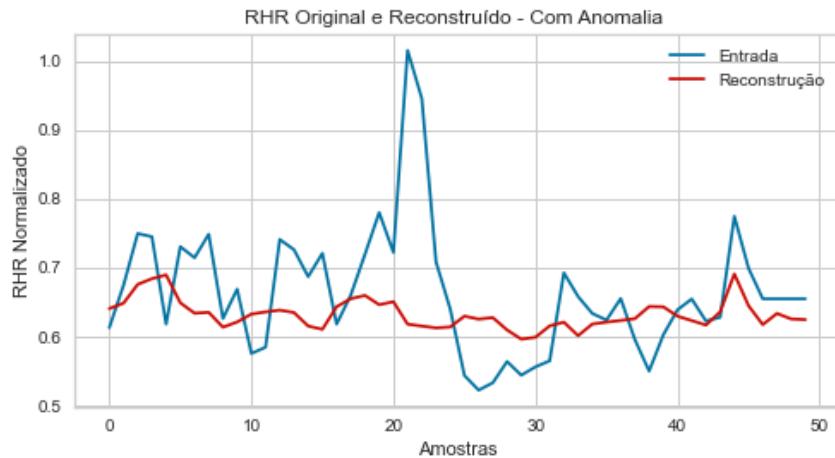


Figura 5.14: Resultado da reconstrução de um RHR anormal pelo *Autoencoder*

5.4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Os modelos utilizados para classificação, *Decision Trees* e *Random Forest*, se mostraram promissores na classificação para COVID-19 ou alguma outra doença respiratória dada a limitação de dados, sendo que o modelo de *Random Forest* obteve melhores resultados em relação ao *Decision Trees*.

Para a detecção de anomalias, o modelo de *Isolation Forest* demonstrou-se bastante eficaz, podendo detectar a presença de COVID-19 e outras doenças respiratórias antes mesmo do aparecimento de sintomas em alguns casos. A vantagem dessa abordagem é que não é necessário uma base de dados extensa para treinar o modelo, pois os sinais entram de forma individual no algoritmo.

Por sua vez, o *Autoencoder Neural Network* utilizado para detectar anomalias não obteve bons resultados, ocorrendo *underfitting* no modelo. Isso é justificado devido à baixa quantidade de dados disponível para treinar o *Autoencoder*.

5.5 CONCLUSÕES DO CAPÍTULO

Neste capítulo houve a demonstração dos resultados obtidos, assim como a discussão acerca dos mesmos. As técnicas de aprendizado de máquina e aprendizado profundo se mostraram promissoras na detecção de COVID-19 ou outras doenças respiratórias, podendo ser excelentes ferramentas no combate à propagação de infecções, podendo dessa forma minimizar ou evitar os impactos causados nas vidas das pessoas, assim como evitar os futuros cenários pandêmicos.

6 CONCLUSÃO

Através da utilização de técnicas de aprendizado de máquina, em conjunto com os dados fornecidos via *smartwatches*, foi possível obter resultados animadores, apesar da limitação causada pela baixa quantidade de dados disponíveis.

Os modelos *Decision Trees* e *Random Forest*, utilizados para classificação, possuem um percentual razoável para classificar casos verdadeiro-positivos, contudo, o percentual obtido para falso-positivos é ainda bastante alto. Ainda referente aos modelos de *Decision Trees* e *Random Forest*, é valido destacar que foi possível utilizá-los somente para classificar se certo usuário teve ou não COVID-19 ou alguma outra doença durante todo o período de leitura dos dados, uma vez que a rotulação dos trechos de RHR era inviável tendo em vista a configuração da base de dados original fornecida.

Por sua vez, o modelo *Isolation Forest*, aplicado na detecção de anomalias, demonstrou ser uma ótima abordagem, principalmente por não ser necessário ter em mãos um *dataset* extenso. O modelo foi capaz de detectar a presença de COVID-19 ou outras doenças em diversos casos antes mesmo do aparecimento de sintomas.

Já o modelo baseado no *Autoencoder Neural Network* não alcançou bons valores para ser considerado um modelo acurado, efeito que já era esperado devido à quantidade de dados disponível para treinamento. Todavia, como demonstrado, o modelo foi capaz de reconstruir com certa semelhanças sinais de RHR saudáveis, enquanto que falhou na reconstrução de sinais de RHR não saudáveis, indicando a presença de anomalias em tais sinais.

Uma observação válida se deve ao fato da base de dados utilizada ter sido gerada ainda no ano de 2020, momento em que o cenário era diferente do atual, pois a população ainda não estava com o ciclo de vacinas completo, além não haver ainda a circulação de outras variantes da COVID-19. Tendo isso em mente, observa-se que há a necessidade de utilizar dados atualizados para a criação dos modelos, assim elas serão mais condizentes com o cenário atual.

6.1 TRABALHOS FUTUROS

Em trabalhos futuros, fica como sugestão a utilização dos métodos de aprendizado de máquina em conjunto com uma nova base de dados mais recente e completa que está disponível, criada a partir do estudo *Real-time alerting system for COVID-19 and other stress events using wearable data* [2]. É sugerido também a utilização de sinais referentes à oxigenação e pressão sanguínea, que os *smartwatches* mais modernos possuem a capacidade de coletar. Além disso, é possível utilizar métodos supervisionados em conjunto com não-supervisionados, de modo que os modelos se tornem mais precisos. Um exemplo do que pode ser feito é utilizar o método *Isolation Forest* para obter trechos de RHR rotulados, utilizando-os em seguida para treinar modelos de *Decision Trees* e *Random Forest*, desviando da limitação causada pela forma em que a base de dados utilizada no estudo foi gerada.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 MISHRA, T.; WANG, M.; METWALLY, A. A.; BOGU, G. K.; BROOKS, A. W.; BAHMANI, A.; ALAVI, A.; CELLI, A.; HIGGS, E.; DAGAN-ROSENFELD, O. et al. Pre-symptomatic detection of covid-19 from smartwatch data. *Nature biomedical engineering*, Nature Publishing Group, v. 4, n. 12, p. 1208–1220, 2020.
- 2 ALAVI, A.; BOGU, G. K.; WANG, M.; RANGAN, E. S.; BROOKS, A. W.; WANG, Q.; HIGGS, E.; CELLI, A.; MISHRA, T.; METWALLY, A. A.; CHA, K.; KNOWLES, P.; ALAVI, A. A.; BHASIN, R.; PANCHAMUKHI, S.; CELIS, D.; ADITYA, T.; HONKALA, A.; ROLNIK, B.; HUNTING, E.; DAGAN-ROSENFELD, O.; CHAUHAN, A.; LI, J. W.; BEJIKIAN, C.; KRISHNAN, V.; MCGUIRE, L.; LI, X.; BAHMANI, A.; SNYDER, M. P. Real-time alerting system for COVID-19 and other stress events using wearable data. *Nature Medicine*, v. 28, n. 1, p. 175–184, jan. 2022. ISSN 1546-170X. Disponível em: <<https://doi.org/10.1038/s41591-021-01593-2>>.
- 3 BOGU, G. K.; SNYDER, M. P. Deep learning-based detection of covid-19 using wearables data. *medRxiv*, Cold Spring Harbor Laboratory Press, 2021. Disponível em: <<https://www.medrxiv.org/content/early/2021/01/09/2021.01.08.21249474>>.
- 4 SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of big data*, Springer, v. 6, n. 1, p. 1–48, 2019.
- 5 GENSLER, A.; HENZE, J.; SICK, B.; RAABE, N. Deep learning for solar power forecasting—an approach using autoencoder and lstm neural networks. In: IEEE. *2016 IEEE international conference on systems, man, and cybernetics (SMC)*. [S.I.], 2016. p. 002858–002865.
- 6 BERRIMI, M.; HAMDI, S.; CHERIF, R. Y.; MOUSSAOUI, A.; OUSSALAH, M.; CHABANE, M. Covid-19 detection from xray and ct scans using transfer learning. In: *2021 International Conference of Women in Data Science at Taif University (WiDSTAif)*. [S.I.: s.n.], 2021. p. 1–6.
- 7 HADI, A. G.; KADHOM, M.; HAIRUNISA, N.; YOUSIF, E.; MOHAMMED, S. A. A review on covid-19: origin, spread, symptoms, treatment, and prevention. *Biointerface Res. Appl. Chem*, v. 10, n. 6, p. 7234–7242, 2020.
- 8 DHAMAD, A. E.; RHIDA, M. A. A. Covid-19: molecular and serological detection methods. *PeerJ*, PeerJ Inc., v. 8, p. e10180, 2020.
- 9 MALAVÉ, M. *Testes para a Covid-19: como são e quando devem ser feitos*. 2020. <https://portal.fiocruz.br/noticia/testes-para-covid-19-como-sao-e-quando-devem-ser-feitos>.
- 10 UNIBRASIL. *Teste de detecção do SARS-CoV-2: como funciona?* 2020. <https://www.unibrasil.com.br/teste-de-deteccao-do-sars-cov-2-como-funciona/>.
- 11 GERAIS, N. de Ações e Pesquisa em Apoio Diagnóstico da Faculdade de Medicina – NUPAD Universidade Federal de M. *DETECÇÃO MOLECULAR DO VÍRUS SARS-COV-2*. 2020. <https://www.nupad.medicina.ufmg.br/doencas-infeciosas/instrucoes-coleta-covid-19/>.
- 12 WALLER, J. V.; KAUR, P.; TUCKER, A.; LIN, K. K.; DIAZ, M. J.; HENRY, T. S.; HOPE, M. Diagnostic tools for coronavirus disease (covid-19): comparing ct and rt-pcr viral nucleic acid testing. *American Journal of Roentgenology*, American Roentgen Ray Society, v. 215, n. 4, p. 834–838, 2020.

- 13 CORMAN, V. M.; LANDT, O.; KAISER, M.; MOLENKAMP, R.; MEIJER, A.; CHU, D. K.; BLEICKER, T.; BRÜNINK, S.; SCHNEIDER, J.; SCHMIDT, M. L. et al. Detection of 2019 novel coronavirus (2019-ncov) by real-time rt-pcr. *Eurosurveillance*, European Centre for Disease Prevention and Control, v. 25, n. 3, p. 2000045, 2020.
- 14 HEYMANN, D. Serology testing in the covid-19 pandemic response. *Lancet Infect Dis*, 2020.
- 15 ZHOU, Y.; WU, Y.; DING, L.; HUANG, X.; XIONG, Y. Point-of-care covid-19 diagnostics powered by lateral flow assay. *TrAC Trends in Analytical Chemistry*, Elsevier, v. 145, p. 116452, 2021.
- 16 SETHURAMAN, N.; JEREMIAH, S. S.; RYO, A. Interpreting Diagnostic Tests for SARS-CoV-2. *JAMA*, v. 323, n. 22, p. 2249–2251, 06 2020. ISSN 0098-7484. Disponível em: <<https://doi.org/10.1001/jama.2020.8259>>.
- 17 DIAS, V. d. C.; CARNEIRO, M.; MICHELIN, L.; VIDAL, C. d. L.; COSTA, L.; FERREIRA, C. d. S.; ROSSETO-WELTER, E.; LINS, R. S.; KFOURI, R.; COSTA, S. F. et al. Testes sorológicos para covid-19: Interpretação e aplicações práticas. *J Infect Control [Internet]*, p. 1–41, 2020.
- 18 ISHII, T.; SASAKI, M.; YAMADA, K.; KATO, D.; OSUKA, H.; AOKI, K.; MORITA, T.; ISHII, Y.; TATEDA, K. Immunochromatography and chemiluminescent enzyme immunoassay for covid-19 diagnosis. *Journal of Infection and Chemotherapy*, Elsevier, v. 27, n. 6, p. 915–918, 2021.
- 19 SCHIVE, K. *How does the COVID-19 antigen test work?* 2020. <https://medical.mit.edu/covid-19-updates/2020/05/how-does-covid-19-antigen-test-work>.
- 20 SESHDARI, D. R.; DAVIES, E. V.; HARLOW, E. R.; HSU, J. J.; KNIGHTON, S. C.; WALKER, T. A.; VOOS, J. E.; DRUMMOND, C. K. Wearable sensors for covid-19: a call to action to harness our digital infrastructure for remote patient monitoring and virtual assessments. *Frontiers in Digital Health*, Frontiers, p. 8, 2020.
- 21 SEMMLOW, B. G. J. L. *Biosignal and Medical Image Processing*. 3. ed. [S.I.]: CRC Press, 2014. ISBN 978-1-4665-6737-5.
- 22 BHATT, C.; DEY, N.; ASHOUR, A. *Internet of Things and Big Data Technologies for Next Generation Healthcare*. Springer International Publishing, 2017. (Studies in Big Data). ISBN 9783319497365. Disponível em: <<https://books.google.com.br/books?id=dt7TDQAAQBAJ>>.
- 23 M, V. T. S. C. H. Detecção de doenças usando smartwatches. *Trabalho de conclusão de curso, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, UnB*, p. 60, 2021.
- 24 LI, X.; DUNN, J.; SALINS, D.; ZHOU, G.; ZHOU, W.; ROSE, S. M. S.-F.; PERELMAN, D.; COLBERT, E.; RUNGE, R.; REGO, S. et al. Digital health: tracking physiomes and activity using wearable biosensors reveals useful health-related information. *PLoS biology*, Public Library of Science San Francisco, CA USA, v. 15, n. 1, p. e2001402, 2017.
- 25 CHOLLET, F. *Deep learning with Python*. [S.I.]: Simon and Schuster, 2021.
- 26 WANG, X.; LEI, Z.; ZHANG, X.; ZHOU, B.; PENG, J. Machine learning basics. *Deep learning*, p. 98–164, 2016.
- 27 MAHESH, B. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, v. 9, p. 381–386, 2020.
- 28 SUTTON, R. S.; BARTO, A. G. *Reinforcement learning: An introduction*. [S.I.]: MIT press, 2018.
- 29 LOH, W.-Y. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, Wiley Online Library, v. 1, n. 1, p. 14–23, 2011.

- 30 CARVALHO, D. V.; PEREIRA, E. M.; CARDOSO, J. S. Machine learning interpretability: A survey on methods and metrics. *Electronics*, Multidisciplinary Digital Publishing Institute, v. 8, n. 8, p. 832, 2019.
- 31 BISHOP, C. M.; NASRABADI, N. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006. v. 4.
- 32 PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.
- 33 HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H.; FRIEDMAN, J. H. *The elements of statistical learning: data mining, inference, and prediction*. [S.l.]: Springer, 2009. v. 2.
- 34 ZHOU, J.; GANDOMI, A. H.; CHEN, F.; HOLZINGER, A. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, Multidisciplinary Digital Publishing Institute, v. 10, n. 5, p. 593, 2021.
- 35 BRANCO, H. *Overfitting e underfitting em Machine Learning*. 2022. Disponível em: <<https://abracd.org/overfitting-e-underfitting-em-machine-learning/>>.
- 36 SAINI, A. *Decision Tree Algorithm – A Complete Guide*. 2021. <https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/>.
- 37 QUINLAN, J. R. Learning decision tree classifiers. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 28, n. 1, p. 71–72, 1996.
- 38 MYLES, A. J.; FEUDALE, R. N.; LIU, Y.; WOODY, N. A.; BROWN, S. D. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, Wiley Online Library, v. 18, n. 6, p. 275–285, 2004.
- 39 BUITINCK, L.; LOUPPE, G.; BLONDEL, M.; PEDREGOSA, F.; MUELLER, A.; GRISEL, O.; NICULAE, V.; PRETTENHOFER, P.; GRAMFORT, A.; GROBLER, J.; LAYTON, R.; VANDERPLAS, J.; JOLY, A.; HOLT, B.; VAROQUAUX, G. API design for machine learning software: experiences from the scikit-learn project. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. [S.l.: s.n.], 2013. p. 108–122.
- 40 BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.
- 41 EDUCATION, I. C. *What is Random Forest?* 2020. <Https://www.ibm.com/cloud/learn/random-forest>.
- 42 R, S. E. *Understanding Random Forest*. 2021. <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>.
- 43 LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. In: IEEE. *2008 eighth ieee international conference on data mining*. [S.l.], 2008. p. 413–422.
- 44 CHAUHAN, A.; VAMSI, P. R. Anomalous ozone measurements detection using unsupervised machine learning methods. In: IEEE. *2019 International Conference on Signal Processing and Communication (ICSC)*. [S.l.], 2019. p. 69–74.
- 45 CHOI, R. Y.; COYNER, A. S.; KALPATHY-CRAMER, J.; CHIANG, M. F.; CAMPBELL, J. P. Introduction to machine learning, neural networks, and deep learning. *Translational Vision Science & Technology*, The Association for Research in Vision and Ophthalmology, v. 9, n. 2, p. 14–14, 2020.

- 46 AGGARWAL, C. C. et al. *Neural networks and deep learning*. [S.l.]: Springer, 2018. v. 10. 978–3 p.
- 47 GBONGLI, K.; XU, Y.; AMEDJONEKOU, K. M. Extended technology acceptance model to predict mobile-based money acceptance and sustainability: A multi-analytical structural equation modeling and neural network approach. *Sustainability*, Multidisciplinary Digital Publishing Institute, v. 11, n. 13, p. 3639, 2019.
- 48 ACADEMY, D. S. *Deep Learning Book*. 2022. Disponível em: <<https://www.deeplearningbook.com.br/>>.
- 49 LEIJNEN, S.; VEEN, F. v. The neural network zoo. In: *Multidisciplinary Digital Publishing Institute Proceedings*. [S.l.: s.n.], 2020. v. 47, n. 1, p. 9.
- 50 DENG, L.; YU, D. Deep learning: methods and applications. *Foundations and trends in signal processing*, Now Publishers Inc. Hanover, MA, USA, v. 7, n. 3–4, p. 197–387, 2014.
- 51 CHIRODEA, M. C.; NOVAC, O. C.; NOVAC, C. M.; BIZON, N.; OPROESCU, M.; GORDAN, C. E. Comparison of tensorflow and pytorch in convolutional neural network-based applications. In: IEEE. *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. [S.l.], 2021. p. 1–6.
- 52 ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G. S.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; GOODFELLOW, I.; HARP, A.; IRVING, G.; ISARD, M.; JIA, Y.; JOZEFOWICZ, R.; KAISER, L.; KUDLUR, M.; LEVENBERG, J.; MANÉ, D.; MONGA, R.; MOORE, S.; MURRAY, D.; OLAH, C.; SCHUSTER, M.; SHLENS, J.; STEINER, B.; SUTSKEVER, I.; TALWAR, K.; TUCKER, P.; VANHOUCKE, V.; VASUDEVAN, V.; VIÉGAS, F.; VINYALS, O.; WARDEN, P.; WATTENBERG, M.; WICKE, M.; YU, Y.; ZHENG, X. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: <<https://www.tensorflow.org/>>.
- 53 PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; DESMAISON, A.; KOPF, A.; YANG, E.; DEVITO, Z.; RAISON, M.; TEJANI, A.; CHILAMKURTHY, S.; STEINER, B.; FANG, L.; BAI, J.; CHINTALA, S. Pytorch: An imperative style, high-performance deep learning library. In: WALLACH, H.; LAROCHELLE, H.; BEYGELZIMER, A.; ALCHÉ-BUC, F. d'; FOX, E.; GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019. p. 8024–8035. Disponível em: <<http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>>.
- 54 GERRISH, S. *How smart machines think*. [S.l.]: MIT Press, 2018.
- 55 TAVARES, L. F. F. *CovidDetection*. 2022. Disponível em: <<https://github.com/lfelipefolha/CovidDetection>>.
- 56 ROSSUM, G. V.; DRAKE, F. L. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.
- 57 TEAM, T. pandas development. *pandas-dev/pandas: Pandas*. Zenodo, 2020. Disponível em: <<https://doi.org/10.5281/zenodo.3509134>>.
- 58 GHYSELS, E.; OSBORN, D. R.; RODRIGUES, P. M. Forecasting seasonal time series. *Handbook of economic forecasting*, Elsevier, v. 1, p. 659–711, 2006.
- 59 GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.
- 60 CHOLLET, F. et al. *Keras*. GitHub, 2015. Disponível em: <<https://github.com/fchollet/keras>>.