

Christian Wendt dos Santos

**Modelagem Computacional Dirigida por  
Dados para Diagnosticar Falhas de  
Rolamentos e Cavitação em Bombas  
Centrífugas**

Cascavel-PR

2024

Christian Wendt dos Santos

# **Modelagem Computacional Dirigida por Dados para Diagnosticar Falhas de Rolamentos e Cavitação em Bombas Centrífugas**

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós-Graduação em Ciência da Computação (PPGComp) da Universidade Estadual do Oeste do Paraná – Unioeste, campus de Cascavel.

Universidade Estadual do Oeste do Paraná – Unioeste

– Cascavel.

Centro de Ciências Exatas e Tecnológicas – CCET

Programa de Pós-Graduação em Ciência da Computação – PPGComp

Orientador: Dr. Rogério Luis Rizzi

Coorientador: Dr. Adair Santa Catarina

Cascavel-PR

2024

Christian Wendt dos Santos

Modelagem Computacional Dirigida por Dados para Diagnosticar Falhas de Rolamentos e Cavitação em Bombas Centrífugas/ Christian Wendt dos Santos. – Cascavel-PR, 2024-220p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Rogério Luis Rizzi

Dissertação (Mestrado)– Universidade Estadual do Oeste do Paraná – Unioeste – Cascavel.

Centro de Ciências Exatas e Tecnológicas – CCET

Programa de Pós-Graduação em Ciência da Computação – PPGComp, 2024.

1. Aprendizado de Máquina. 2. Redes Neurais Multicamadas de Perceptrons. 2. Redes Neurais Convolucionais. I. Rogério Luis Rizzi. II. Universidade Estadual Oeste do Paraná. III. Centro de Ciências Exatas e Tecnológicas CCET. IV. Modelagem Computacional para o Diagnóstico de Cavitação em Bombas Centrífugas: Uma Abordagem em Aprendizado de Máquina

Christian Wendt dos Santos

# **Modelagem Computacional Dirigida por Dados para Diagnosticar Falhas de Rolamentos e Cavitação em Bombas Centrífugas**

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós-Graduação em Ciência da Computação (PPGComp) da Universidade Estadual do Oeste do Paraná – Unioeste, campus de Cascavel.

Defesa da Dissertação. Cascavel-PR, 05 de setembro de 2024:

---

**Dr. Rogério Luis Rizzi**  
Orientador(a)

---

**Dr. Adair Santa Catarina**  
Coorientador(a)

---

**Dr(a). André Luiz Brun**  
UNIOESTE - Cascavel

---

**Dr(a). Thiago Berticelli Ló**  
IFPR - Cascavel

Cascavel-PR  
2024

*Este trabalho é dedicado às crianças adultas que,  
quando pequenas, sonharam em se tornar cientistas.*

# Agradecimentos

Neste momento de realização da defesa, é com imensa gratidão que expresso meus sinceros agradecimentos àqueles que contribuíram de maneira significativa para o sucesso desta pesquisa. Cada passo dado foi guiado por esforço, comprometimento e pelo apoio incansável de pessoas e instituições que merecem o meu mais profundo reconhecimento.

Primeiramente, dedico minha gratidão a Deus, fonte de toda sabedoria e inspiração, por me guiar ao longo desta jornada acadêmica e me conceder a força necessária para superar desafios até este momento.

Ao meu orientador prof. Rogério Luis Rizzi e coorientador prof. Adair Santa Catarina, expresso minha profunda admiração e reconhecimento. Sua orientação especializada, paciência e incentivo foram fundamentais para o desenvolvimento desta dissertação de mestrado.

À minha amada esposa, Jéssica Fróes de Brito Wendt, manifesto minha gratidão pelo apoio incondicional, compreensão e encorajamento durante toda essa trajetória.

À minha gestora, Alexandra da Silva Belini, cujo apoio foi fundamental para a concretização e finalização desta pesquisa.

Não posso deixar de expressar minha profunda gratidão ao Itaipu Parquetec por meio do Ceasb, cujo apoio e financiamento foram determinantes para a realização deste estudo. A confiança depositada em meu trabalho é uma honra que levo com grande responsabilidade, e espero ter correspondido às expectativas com dedicação e comprometimento.

A todos os professores, colegas e amigos que, de alguma forma, contribuíram para minha formação acadêmica e pessoal, deixo meu agradecimento sincero.

*“Não vos amoldeis às estruturas deste mundo,  
mas transformai-vos pela renovação da mente,  
a fim de distinguir qual é a vontade de Deus:  
o que é bom, o que Lhe é agradável, o que é perfeito.  
(Bíblia Sagrada, Romanos 12, 2)*

# Resumo

WENDT SANTOS, Christian. **Modelagem Computacional Dirigida por Dados para Diagnosticar Falhas de Rolamentos e Cavitação em Bombas Centrífugas**. Orientador: Dr. Rogério Luis Rizzi. 2024. 220f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual do Oeste do Paraná, Cascavel – Paraná, 2024.

A modelagem computacional baseada em aprendizado de máquina tem se destacado em aplicações no âmbito das Engenharias, especificamente no monitoramento e classificação da degradação de componentes críticos. Nesta pesquisa, foram desenvolvidos modelos de aprendizado de máquina para monitorar a ocorrência do fenômeno de cavitação em bombas centrífugas e defeitos em rolamentos por meio de sinais temporais registrados por sensores. Os dados experimentais são de origem da *Case Western Reserve University* (CWRU) e *UWA System Health Lab Prognostics Data Library* (SPDL) que envolvem os tipos e intensidades falhas em rolamentos esféricos e a presença do fenômeno de cavitação por meio de acelerômetros. O enfoque desta pesquisa está na construção da solução computacional à preparação dos dados e extração das características, utilizando a técnica de contagem de *hits* e a Transformada de Fourier de Curto Tempo (TFCT). A primeira envolve o cálculo de diversos parâmetros que definem a forma da onda no domínio do tempo, a exemplo da duração do *hit*, o tempo para atingir a maior amplitude e a energia do intervalo *hit*. A segunda transforma o sinal do domínio do tempo em um espectrograma de frequência por tempo e posteriormente em imagens. Na identificação e classificação dos padrões foram utilizados dois modelos de aprendizado de máquina baseados em redes neurais: uma Rede Neural Multicamadas de Perceptrons (RNMP), que utiliza parâmetros da técnica de contagem de hits, e uma Rede Neural Convolucional (RNC), que emprega imagens geradas pela TFCT. Os resultados obtidos com o modelo RNMP utilizando dados experimentais de vibração da *Case Western Reserve University*, mostraram uma acurácia de 69,50%, precisão de 67,07%, recall de 73,50% e *F1-score* de 68,63%. Já o modelo RNC apresentou uma acurácia de 93,90%, precisão de 93,41%, recall de 93,77% e *F1-score* de 92,98%. Enquanto que os resultados obtidos com os modelos RNMP para os dados de *UWA System Health Lab Prognostics Data Library* mostraram uma acurácia de 77,90%, precisão de 76,25%, recall de 75,68% e *F1-score* de 76,25%. Já o modelo RNC os valores foram 89,17% de acurácia, precisão de 87,57%, recall de 88,05% e *F1-score* de 87,80%. Esses resultados foram analisados utilizando validação cruzada *K-fold*, matriz de confusão e comparação de desempenho com os modelos de aprendizado de máquina do artigo de referência. Além disso, foi realizada uma análise estatística para definir parâmetros como limiar e janelamento do sinal, bem como para a criação de rótulos.

**Palavras-chave:** Parâmetros de Hits, Redes Neurais; Cavitação, Falhas de Rolamentos.

# Abstract

WENDT SANTOS, Christian. **Computational Modeling for Cavitation Diagnosis in Centrifugal Pumps: A Machine Learning Approach**. Orientador: Dr. Rogério Luis Rizzi. 2024. 220f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual do Oeste do Paraná, Cascavel – Paraná, 2024.

Computational modeling based on machine learning has stood out in engineering applications, specifically in monitoring and classifying the degradation of critical components. In this research, machine learning models were developed to monitor the occurrence of cavitation in centrifugal pumps and bearing defects through time signals recorded by sensors. The experimental data originate from the Case Western Reserve University (CWRU) and the UWA System Health Lab Prognostics Data Library (SPDL), involving types and intensities of faults in ball bearings and the presence of the cavitation phenomenon detected via accelerometers. The focus of this research is on constructing a computational solution for data preparation and feature extraction, using the hit counting technique and the Short-Time Fourier Transform (STFT). The first technique involves calculating several parameters that define the waveform in the time domain, such as hit duration, time to reach peak amplitude, and hit interval energy. The second transforms the time-domain signal into a time-frequency spectrogram and subsequently into images. Two machine learning models based on neural networks were used for pattern identification and classification: a Multilayer Perceptron Neural Network (MLP), which employs parameters from the hit counting technique, and a Convolutional Neural Network (CNN), which uses images generated by the STFT. The results obtained with the MLP model using vibration experimental data from the Case Western Reserve University showed an accuracy of 69.50%, precision of 67.07%, recall of 73.50%, and an F1-score of 68.63%. The CNN model, on the other hand, achieved an accuracy of 93.90%, precision of 93.41%, recall of 93.77%, and an F1-score of 92.98%. For the UWA System Health Lab Prognostics Data Library, the results obtained with the MLP model showed an accuracy of 77.90%, precision of 76.25%, recall of 75.68%, and an F1-score of 76.25%. The CNN model achieved an accuracy of 89.17%, precision of 87.57%, recall of 88.05%. These results were analyzed using K-fold cross-validation, confusion matrices, and performance comparison with machine learning models from the reference study. Additionally, a statistical analysis was conducted to define parameters such as signal thresholding and windowing, as well as to create labels.

**Keywords:** Hit Parameters; Neural Networks; Cavitation, Bearing Failures.

# Lista de ilustrações

|                                                                                                                                                      |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1 – Modelos aprendizado de máquina para a solução do problema motivador.                                                                      | 23 |
| Figura 2 – Organização e estrutura do trabalho.                                                                                                      | 26 |
| Figura 3 – Principais elementos hidromecânicos de uma bomba centrífuga.                                                                              | 48 |
| Figura 4 – Cavitação: (a) diagrama de fase, (b) nucleação, crescimento e colapso.                                                                    | 49 |
| Figura 5 – Escoamento multifásico em uma bomba centrífuga: (a) bolhas isoladas, (b) bolhas aglomeradas e (c) bolsões de gás, (d) separação do fluxo. | 50 |
| Figura 6 – Coeficiente de pressão: (a) $\sigma=0,463$ e (b) $\sigma=0,088$ .                                                                         | 51 |
| Figura 7 – Limiar de detecção: (a) 0,01 V, (b) 0,02 V.                                                                                               | 52 |
| Figura 8 – Parâmetros: (a) amplitude; (b) contagem; (c) contagem à pico; (d) energia.                                                                | 53 |
| Figura 9 – Parâmetros de Tempo: (a) duração, (b) tempo de subida.                                                                                    | 53 |
| Figura 10 – Sinal: (a) domínio do tempo, (b) domínio da frequência.                                                                                  | 55 |
| Figura 11 – Limitações da TF: (a) sinal temporal; (b) espectro de frequência.                                                                        | 56 |
| Figura 12 – Princípio da TFCT: (a) Aplicação da janela; (b) TFCT.                                                                                    | 57 |
| Figura 13 – Tipos de janelas.                                                                                                                        | 57 |
| Figura 14 – Aplicação da TFCT: (a) janela de 0,512s (b) janela de 0,064s.                                                                            | 58 |
| Figura 15 – Princípio da TW: (a) Aplicação da janela; (b) TW.                                                                                        | 58 |
| Figura 16 – Algoritmo com paradigma clássico e aprendizado de máquina.                                                                               | 59 |
| Figura 17 – Diagrama organizacional de IA, AM e AP.                                                                                                  | 60 |
| Figura 18 – Diagrama organizacional de tipos de aprendizado.                                                                                         | 61 |
| Figura 19 – Dados organizados.                                                                                                                       | 63 |
| Figura 20 – Regularidade do modelo: (a) <i>underfitting</i> ; (b) <i>overfitting</i> ; (c) <i>optimum</i> .                                          | 64 |
| Figura 21 – Erros relacionados a variância e <i>bias</i> .                                                                                           | 65 |
| Figura 22 – Elaboração de um modelo ML.                                                                                                              | 66 |
| Figura 23 – Elaboração de um modelo AP.                                                                                                              | 67 |
| Figura 24 – Estrutura do neurônio artificial.                                                                                                        | 70 |
| Figura 25 – Funções de ativação do tipo: a) limiar; b) rampa ; c) logística; d) linear; e) tangente hiperbólica; f) ReLU                             | 72 |
| Figura 26 – Rede Neural MLP                                                                                                                          | 73 |
| Figura 27 – Backpropagation na RNMP.                                                                                                                 | 77 |
| Figura 28 – Fluxo de processos da camada de convolução: (a) complexa; (b) simples.                                                                   | 80 |
| Figura 29 – Operação de convolução para estruturas de dados unidimensional (1D).                                                                     | 82 |
| Figura 30 – Operação de convolução para estruturas de dados unidimensional (2D).                                                                     | 82 |
| Figura 31 – Camada <i>pooling</i>                                                                                                                    | 83 |
| Figura 32 – Rede neural convolucional LeNet-5.                                                                                                       | 84 |
| Figura 33 – Camada de normalização: (a) pós ativação; (b) pré ativação.                                                                              | 86 |

|                                                                                                                                                                |     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Figura 34 – <i>Dropout</i> . . . . .                                                                                                                           | 88  |
| Figura 35 – Técnica <i>SMOTE</i> . . . . .                                                                                                                     | 89  |
| Figura 36 – Técnica ADASYN. . . . .                                                                                                                            | 90  |
| Figura 37 – Geração de dados sintéticos: (a) brutos ; (b) SMOTE; (c) SMOTE+ENN. . . . .                                                                        | 90  |
| Figura 38 – Decomposição do Valor Singular (DVS). . . . .                                                                                                      | 91  |
| Figura 39 – Fluxograma do processo de Otimização Bayesiana. . . . .                                                                                            | 92  |
| Figura 40 – Distribuições de probabilidade: (a) Priori; (b) Verrosimilhança; (c) Posteriori . . . . .                                                          | 93  |
| Figura 41 – Processo de otimização Bayesiana: (a) função objetivo a <i>apriori</i> ; (b) função objetivo <i>posteriori</i> ; (c) função de aquisição . . . . . | 94  |
| Figura 42 – Processo de otimização Bayesiana: . . . . .                                                                                                        | 95  |
| Figura 43 – Fluxograma metodológico dos modelos. . . . .                                                                                                       | 96  |
| Figura 44 – Fluxograma conceitual dos modelos RNMP e RNC. . . . .                                                                                              | 98  |
| Figura 45 – Ensaio de cavitação: (a) bomba centrífuga; (b) Posição dos acelerômetros. . . . .                                                                  | 98  |
| Figura 46 – Bancada experimental de falhas em rolamentos . . . . .                                                                                             | 101 |
| Figura 47 – Fluxograma conceitual da etapa Base de Dados dos modelos RNMP e RNC. . . . .                                                                       | 102 |
| Figura 48 – Fluxograma conceitual da preparação dos dados. . . . .                                                                                             | 103 |
| Figura 49 – Ampliação da base de dados com amostras sintéticas. . . . .                                                                                        | 103 |
| Figura 50 – Ampliação da base de dados com amostras sintéticas. . . . .                                                                                        | 104 |
| Figura 51 – Impacto do valor limiar: (a) Dados não normalizados; (b) Dados normalizados. . . . .                                                               | 104 |
| Figura 52 – Fluxograma conceitual da preparação dos dados do modelo RNC. . . . .                                                                               | 105 |
| Figura 53 – Fluxograma conceitual da extração de características. . . . .                                                                                      | 106 |
| Figura 54 – Fluxograma conceitual da transformação e balanceamento dos dados. . . . .                                                                          | 107 |
| Figura 55 – Fluxograma conceitual da transformação do dados para treinamento e teste. . . . .                                                                  | 108 |
| Figura 56 – Fluxograma conceitual de balanceamento dados. . . . .                                                                                              | 108 |
| Figura 57 – Aplicação da janela retangular: (a) sinal bruto; (b) sinal janelado. . . . .                                                                       | 109 |
| Figura 58 – Fluxograma conceitual da Extração de Características: TFCT. . . . .                                                                                | 109 |
| Figura 59 – Etapas do processo TFCT: (a) sinal no domínio do tempo; (b) sinal no domínio da frequência; (c) sinal no domínio tempo-frequência. . . . .         | 110 |
| Figura 60 – Fluxograma conceitual de AM do modelo RNMP e RNC. . . . .                                                                                          | 111 |
| Figura 61 – Fluxogramas das redes neural: RNC e RNMP. . . . .                                                                                                  | 112 |
| Figura 62 – Arquitetura da RNMP: (a) dados cavitação; (b) dados rolamentos. . . . .                                                                            | 114 |
| Figura 63 – Arquitetura da RNC para os dados de cavitação. . . . .                                                                                             | 115 |
| Figura 64 – Arquitetura da RNC para os dados de rolamento. . . . .                                                                                             | 116 |
| Figura 65 – Fluxograma conceitual da análise exploratória dos dados. . . . .                                                                                   | 117 |
| Figura 66 – Fluxograma conceitual do modelo RNMP. . . . .                                                                                                      | 118 |

|                                                                                                                                                                         |     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Figura 67 – Fluxograma conceitual do modelo RNC. . . . .                                                                                                                | 119 |
| Figura 68 – Fluxograma computacional do modelo RNMP. . . . .                                                                                                            | 120 |
| Figura 69 – Fluxograma computacional do módulo ARQ_BIN. . . . .                                                                                                         | 122 |
| Figura 70 – Fluxograma computacional do módulo PREPARA_DATA. . . . .                                                                                                    | 123 |
| Figura 71 – Fluxograma computacional do módulo FEATURES_EXTRACT. . . . .                                                                                                | 126 |
| Figura 72 – Fluxograma computacional do módulo IMBALANCE_DATA. . . . .                                                                                                  | 133 |
| Figura 73 – Fluxograma computacional do módulo RNMP. . . . .                                                                                                            | 136 |
| Figura 74 – Fluxograma computacional do modelo RNC. . . . .                                                                                                             | 142 |
| Figura 75 – Fluxograma computacional do módulo ARQ_BIN modelo RNC. . . . .                                                                                              | 144 |
| Figura 76 – Fluxograma computacional do módulo PREPARA_DATA do modelo<br>RNC. . . . .                                                                                   | 145 |
| Figura 77 – Fluxograma computacional do módulo RNC. . . . .                                                                                                             | 148 |
| Figura 78 – Fluxograma computacional do módulo <i>data_image(Dataset)</i> . . . . .                                                                                     | 149 |
| Figura 79 – Histogramas dos dados de potência com 0HP com $N/100$ . . . . .                                                                                             | 157 |
| Figura 80 – Gráficos $Q - Q$ para a potência de eixo de 0HP com $N/100$ . . . . .                                                                                       | 157 |
| Figura 81 – Rótulos considerando a normalidade dos dados: (a) $N/100$ ; (b) $N/1000$ . . . . .                                                                          | 160 |
| Figura 82 – Rótulos desconsiderando a normalidade dos dados: (a) $N/100$ ; (b) $N/1000$ . . . . .                                                                       | 160 |
| Figura 83 – Histogramas dos dados com carga: (a) $NRM$ ; (b) $B07$ ; (c) $IR07$ ; (d)<br>$ER21$ . . . . .                                                               | 162 |
| Figura 84 – Histogramas dos dados sem carga: (a) $NRM$ ; (b) $B07$ ; (c) $IR07$ ; (d)<br>$ER21$ . . . . .                                                               | 162 |
| Figura 85 – Sinal Bruto dos dados de rolamento para a carga 3HP. . . . .                                                                                                | 163 |
| Figura 86 – Sinal janelado de $NRM - 2HP$ : (a) 5s; (b) 0,11s; (c) 0,055s e (d) 0,02s. . . . .                                                                          | 168 |
| Figura 87 – Análise de sinais da condição normal ( $NRM$ ). . . . .                                                                                                     | 170 |
| Figura 88 – Análise de sinais da condição de falha na esfera do rolamento ( $B07$ ,<br>$B14$ e $B21$ ). . . . .                                                         | 171 |
| Figura 89 – Análise de sinais da condição de falha na pista interna do rolamento<br>( $IR07$ , $IR14$ e $IR21$ ). . . . .                                               | 172 |
| Figura 90 – Análise de sinais da condição de falha na pista externa do rolamento<br>( $ER07$ , $ER14$ e $ER21$ ). . . . .                                               | 173 |
| Figura 91 – Classe $B21$ 0,22s a 0,24s: (a) dados brutos; (b) dados normalizados. . . . .                                                                               | 174 |
| Figura 92 – Vizualização dos dados separados por classe com a técnica t-SNE: (a)<br>brutos; (b) com $DVS$ ; (c) com $DVS$ e SMOTE e (d) $DVS$ e SMOTE-<br>ENN. . . . .  | 176 |
| Figura 93 – Desempenho geral da RNMP nas fase de treino: (a) acurácia; (b) custo. . . . .                                                                               | 178 |
| Figura 94 – Aplicação das janelas retangulares: (a) sinal completo; (b) janela do<br>instante 0 a 0,2s; (c) janela do instante 0,8 a 1s; (d) janela 1,6 a 1,8s. . . . . | 181 |
| Figura 95 – Espectrogramas das condições da integridade estrutural dos rolamentos<br>(TFCT). . . . .                                                                    | 182 |

|                                                                                                                                                                              |     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Figura 96 – Imagens das condições da integridade estrutural dos rolamentos (cinza).                                                                                          | 183 |
| Figura 97 – Base de dados de rolamento para o modelo RNC. . . . .                                                                                                            | 183 |
| Figura 98 – Desempenho da RNC na fase de treino: (a) acurácia; (b) função de custo.                                                                                          | 184 |
| Figura 99 – Desempenho entre diferentes modelos de aprendizado de máquina. . . .                                                                                             | 188 |
| Figura 100 – Gráficos $Q - Q$ . . . . .                                                                                                                                      | 188 |
| Figura 101 – (a) <i>Boxplot</i> ; (b) Gráfico $Q - Q$ . . . . .                                                                                                              | 190 |
| Figura 102 – Histogramas para as condições com e sem cavitação com $N/100$ . . . .                                                                                           | 192 |
| Figura 103 – Gráficos $Q - Q$ para as condições com e sem cavitação com $N/100$ . . .                                                                                        | 193 |
| Figura 104 – Rótulos dos dados de cavitação: (a) $N/100$ ; (b) $N/1000$ . . . . .                                                                                            | 195 |
| Figura 105 – Histogramas dos dados com cavitação: (a) E4.Q55; (b) E4.Q108; (c) E4.Q161<br>e (d) E4.Q193. . . . .                                                             | 197 |
| Figura 106 – Histogramas dos dados sem cavitação: (a) E1.Q55; (b) E1.Q108; (c) E1.Q161<br>e (d) E1.Q193. . . . .                                                             | 197 |
| Figura 107 – Sinal Bruto. . . . .                                                                                                                                            | 198 |
| Figura 108 – Sinal janelado $ONCAV - E4.Q196$ : (a) 12s; (b) 0,11s; (c) 0,055s e (d)<br>0,02s. . . . .                                                                       | 201 |
| Figura 109 – Classe $OFFCAV - E0.Q55$ : (a) dados brutos; (b) dados normalizados.                                                                                            | 202 |
| Figura 110 – Vizualização dos dados de cavitação separados por classe com a técnica<br>t-SNE: (a) brutos; (b) com DVS; (c) com DVS e SMOTE e (d) DVS e<br>SMOTE-ENN. . . . . | 204 |
| Figura 111 – Desempenho geral da RNMP das fase de treino: (a) acurácia ; (b) custo.                                                                                          | 207 |
| Figura 112 – Condição $OFFCAV - Q55$ : (a) sinal completo; (b) janela do instante<br>0 a 0,2s; (c) janela do instante 5,0 a 5,02s; (d) janela 12 a 12,02s. . . .             | 208 |
| Figura 113 – Espectrogramas das condições de cavitação (TFCT). . . . .                                                                                                       | 209 |
| Figura 114 – Imagens das condições de cavitação (Gray). . . . .                                                                                                              | 210 |
| Figura 115 – Base de dados de cavitação para o modelo RNC. . . . .                                                                                                           | 210 |
| Figura 116 – Desempenho da RNC na fase de treino: (a) acurácia; (b) função de custo                                                                                          | 211 |

# Lista de quadros

|                                                                                 |     |
|---------------------------------------------------------------------------------|-----|
| Quadro 1 – Resultado da pesquisa: baixo nível de aderência. . . . .             | 29  |
| Quadro 2 – Resultado da Pesquisa: médio nível de aderência . . . . .            | 30  |
| Quadro 3 – Resultado da pesquisa: alto nível de aderência. . . . .              | 30  |
| Quadro 4 – Interação entre os módulos dos modelos RNMP e RNC. . . . .           | 67  |
| Quadro 5 – Arquiteturas de redes convolucionais. . . . .                        | 83  |
| Quadro 6 – Metadados dos arquivos de ensaio de 1730 rpm . . . . .               | 102 |
| Quadro 7 – Meta Dados dos arquivos dos ensaios de 1750, 1772 e 1796 rpm . . . . | 102 |
| Quadro 8 – Dados organizados de cavitação . . . . .                             | 107 |
| Quadro 9 – Descrição dos módulos do modelo RNMP. . . . .                        | 112 |
| Quadro 10 – Descrição dos módulos do modelo RNC. . . . .                        | 113 |
| Quadro 11 – Interação entre os módulos dos modelos RMNP e RNC. . . . .          | 113 |
| Quadro 12 – Descrição e interação entre os módulos do modelo RNMP. . . . .      | 121 |
| Quadro 13 – Descrição e interação entre os módulos do modelo RNC. . . . .       | 143 |

# Lista de tabelas

|                                                                                                          |     |
|----------------------------------------------------------------------------------------------------------|-----|
| Tabela 1 – Resultados relevantes dos trabalhos científicos. . . . .                                      | 29  |
| Tabela 2 – Arquitetura da RNC referente a Yan et al. (2016). . . . .                                     | 32  |
| Tabela 3 – Arquitetura da RNMP referente a Deng et al. (2017). . . . .                                   | 32  |
| Tabela 4 – Arquitetura da RNC de Zhang et al. (2020) . . . . .                                           | 33  |
| Tabela 5 – Hiperparâmetros da rede convolucional de Zhang et al. (2020) . . . . .                        | 34  |
| Tabela 6 – Arquitetura da rede convolucional de (KUMAR et al., 2020) . . . . .                           | 35  |
| Tabela 7 – Arquitetura da RNMP de He et al. (2021). . . . .                                              | 37  |
| Tabela 8 – Hiperparâmetros da RNMP de He et al. (2021). . . . .                                          | 37  |
| Tabela 9 – Arquitetura da RNMP de Tiwari, Bordoloi e Dewangan (2021) . . . . .                           | 38  |
| Tabela 10 – Hiperparâmetros da RNMP de Tiwari, Bordoloi e Dewangan (2021) . . . . .                      | 38  |
| Tabela 11 – Arquitetura da rede convolucional de Chao et al. (2020b) . . . . .                           | 39  |
| Tabela 12 – Hiperparâmetros da rede convolucional de Chao et al. (2020b) . . . . .                       | 39  |
| Tabela 13 – Arquitetura da rede convolucional de Chao et al. (2020a) . . . . .                           | 40  |
| Tabela 14 – Resultados de classificação dos modelos RNC e RNMP de Chao et al.<br>(2020a) . . . . .       | 40  |
| Tabela 15 – Arquitetura da RNMP de Kumar et al. (2021). . . . .                                          | 41  |
| Tabela 16 – Hiperparâmetros da RNMP de Kumar et al. (2021). . . . .                                      | 41  |
| Tabela 17 – Arquitetura da rede convolucional referente à Tang, Zhu e Yuan (2021) . . . . .              | 42  |
| Tabela 18 – Hiperparâmetros da rede convolucional referente à Tang, Zhu e Yuan<br>(2021). . . . .        | 42  |
| Tabela 19 – Arquitetura da rede convolucional de Tang, Zhu e Yuan (2022) . . . . .                       | 43  |
| Tabela 20 – Hiperparâmetros da rede convolucional de Tang, Zhu e Yuan (2022) . . . . .                   | 44  |
| Tabela 21 – Resultados de identificação de cavitação . . . . .                                           | 44  |
| Tabela 22 – Hiperparâmetros dos modelos de aprendizado de máquina dos artigos<br>de referências. . . . . | 47  |
| Tabela 23 – Nomes dos arquivos e organização dos dados de ensaio. . . . .                                | 99  |
| Tabela 24 – Organização dos dados de ensaio. . . . .                                                     | 100 |
| Tabela 25 – Nomes dos arquivos e organização dos dados para a condição de falha. . . . .                 | 101 |
| Tabela 26 – Arquitetura da RNMP. . . . .                                                                 | 114 |
| Tabela 27 – Hiperparâmetros do modelo RNMP. . . . .                                                      | 115 |
| Tabela 28 – Arquitetura da RNC. . . . .                                                                  | 116 |
| Tabela 29 – Hiperparâmetros da modelo RNC. . . . .                                                       | 116 |
| Tabela 30 – Tamanhos das amostras para o estudo estatístico. . . . .                                     | 155 |
| Tabela 31 – <i>Valores – p</i> do teste de Shapiro-Wilk para tamanho $N/100$ . . . . .                   | 155 |
| Tabela 32 – <i>Valores – p</i> do teste de Shapiro-Wilk para tamanho $N/1.000$ . . . . .                 | 156 |
| Tabela 33 – <i>Valores – p</i> do teste de Kolmogorov-Smirnov para tamanho $N/100$ . . . . .             | 156 |

|                                                                                                |     |
|------------------------------------------------------------------------------------------------|-----|
| Tabela 34 – Valores – $p$ do teste de Kolmogorov-Smirnov para tamanho $N/1.000$ .              | 156 |
| Tabela 35 – Valor – $p$ do teste de Levene com tamanho $N/100$ .                               | 158 |
| Tabela 36 – Análise de variância e teste Tukey para $NRM$ com $\alpha=0.05$                    | 158 |
| Tabela 37 – Análise de variância e teste Tukey para $B07$ , $B14$ e $B21$ com $\alpha=0.05$    | 158 |
| Tabela 38 – Análise de variância e teste Tukey para $IR07$ , $IR14$ e $IR21$ com $\alpha=0.05$ | 159 |
| Tabela 39 – Análise de variância e teste Tukey para $ER07$ , $ER14$ e $ER21$ com $\alpha=0.05$ | 159 |
| Tabela 40 – Testes Kruskal-Wallis e Dunn para $B07$ , $B14$ e $B21$ com $\alpha=5\%$           | 159 |
| Tabela 41 – Testes Kruskal-Wallis e Dunn para $IR07$ , $IR14$ e $IR21$ com $\alpha=5\%$        | 159 |
| Tabela 42 – Testes Kruskal-Wallis e Dunn para $ER07$ , $ER14$ e $ER21$ com $\alpha=5\%$        | 159 |
| Tabela 43 – Estatística descritiva dos dados brutos                                            | 161 |
| Tabela 44 – Valores dos intervalos de confiança para 68,2% e 99,74%.                           | 164 |
| Tabela 45 – Delineamento experimental incompleto para os dados de rolamento.                   | 164 |
| Tabela 46 – Duração por rótulo com dados brutos.                                               | 165 |
| Tabela 47 – Número de pico máximo por rótulos com dados brutos.                                | 166 |
| Tabela 48 – Amplitude por rótulo com dados brutos.                                             | 167 |
| Tabela 49 – Amplitude por rótulo combinado ao $DVS$ .                                          | 167 |
| Tabela 50 – Principais frequências dos sinais de acordo com o janelamento                      | 169 |
| Tabela 51 – Quantidade total de amostras de acordo com o tamanho da janela                     | 169 |
| Tabela 52 – Número de amostras: Aplicação do <i>data augmentation</i> (Algoritmo 8).           | 174 |
| Tabela 53 – Dados organizados para o modelo RNMP.                                              | 175 |
| Tabela 54 – Quantidade de amostras por classe para o modelo RNMP.                              | 175 |
| Tabela 55 – Quantidade de amostras do ensaio ao modelo RNMP.                                   | 175 |
| Tabela 56 – Dados organizados para o modelo RNMP.                                              | 177 |
| Tabela 57 – Quantidade de amostras por classe para o modelo RNMP.                              | 177 |
| Tabela 58 – Resultados dos conjuntos de dados de origem do ensaio e suas variações.            | 178 |
| Tabela 59 – Exploração do espaço e seleção dos hiperparâmetros                                 | 178 |
| Tabela 60 – Matriz confusão do modelo RNMP com dados de rolamento da fase de teste.            | 179 |
| Tabela 61 – Sensibilidade do modelo RNMP para os dados de rolamento.                           | 180 |
| Tabela 62 – Precisão do modelo RNMP para os dados de rolamento.                                | 180 |
| Tabela 63 – $F1-score$ do modelo RNMP para os dados de rolamento.                              | 180 |
| Tabela 64 – Valores frequência do sinal na classe $ONCAV - E4.Q108$ .                          | 181 |
| Tabela 65 – Quantidade de dados                                                                | 182 |
| Tabela 66 – Dados estruturados para a base de rolamento.                                       | 184 |
| Tabela 67 – Matriz confusão do modelo RNC com dados da fase de teste.                          | 185 |
| Tabela 68 – Precisão do modelo RNC para os dados de rolamento.                                 | 185 |
| Tabela 69 – Sensibilidade do modelo RNC para os dados de rolamento.                            | 185 |
| Tabela 70 – $F1-scoce$ do modelo RNC para os dados de rolamento.                               | 185 |

|                                                                                                       |     |
|-------------------------------------------------------------------------------------------------------|-----|
| Tabela 71 – Dados para a validação cruzada dos modelos DFCNN, MLP, RNMP e RNC. . . . .                | 186 |
| Tabela 72 – Validação cruzada do modelo RNMP para os dados de rolamento. . . .                        | 187 |
| Tabela 73 – Validação cruzada do modelo RNC para os dados de rolamento. . . .                         | 187 |
| Tabela 74 – Desempenho entre diferentes modelos de aprendizado de máquina. . . .                      | 187 |
| Tabela 75 – Testes de normalidade dos dados. . . . .                                                  | 188 |
| Tabela 76 – Comparação de desempenho entre diferentes modelos . . . . .                               | 189 |
| Tabela 77 – Comparação de desempenho entre diferentes modelos . . . . .                               | 190 |
| Tabela 78 – Tamanhos das amostras dos dados de cavitação para o estudo estatístico.                   | 191 |
| Tabela 79 – <i>Valores – p</i> do teste Shapiro-Wilk para os dados de cavitação com $N/100$ . . . . . | 192 |
| Tabela 80 – <i>Valor – p</i> do teste de Levene com tamanho $N/100$ . . . . .                         | 193 |
| Tabela 81 – Análise de variância e teste Tukey para <i>ONCAV</i> com $\alpha=0.05$ . . . .            | 193 |
| Tabela 82 – Análise de variância e teste Tukey para <i>OFFCAV</i> com $\alpha=0.05$ . . . .           | 194 |
| Tabela 83 – Testes Kruskal-Wallis e Dunn para <i>ONCAV</i> com $\alpha=0.05$ . . . . .                | 194 |
| Tabela 84 – Testes Kruskal-Wallis e Dunn para <i>OFFCAV</i> com $\alpha=0.05$ . . . . .               | 195 |
| Tabela 85 – Estatística descritiva dos dados de cavitação. . . . .                                    | 196 |
| Tabela 86 – Valores dos intervalos de confiança para 68, 2% 99, 74%. . . . .                          | 199 |
| Tabela 87 – Delineamento experimental incompleto para os dados de cavitação. . .                      | 199 |
| Tabela 88 – Duração e número de pico máximo por rótulo com dados brutos. . . .                        | 200 |
| Tabela 89 – Amplitude por rótulo com dados brutos e com DVS. . . . .                                  | 201 |
| Tabela 90 – Principais frequências dos sinais de cavitação acordo com o janelamento.                  | 202 |
| Tabela 91 – Quantidade de amostras de cavitação de acordo com a janela. . . . .                       | 202 |
| Tabela 92 – Aplicação do <i>data augmentation</i> aos dados de cavitação (Algoritmo 8).               | 203 |
| Tabela 93 – Dados brutos organizados da base de cavitação para o modelo RNMP.                         | 203 |
| Tabela 94 – Quantidade de amostras de cavitação para o modelo RNMP. . . . .                           | 203 |
| Tabela 95 – Amostras de cavitação utilizadas na modelagem RNMP. . . . .                               | 204 |
| Tabela 96 – Dados organizados de cavitação para o modelo RNMP. . . . .                                | 205 |
| Tabela 97 – Quantidade de amostras de cavitação por classe para o modelo RNMP                         | 205 |
| Tabela 98 – Resultados dos conjuntos de dados de cavitação. . . . .                                   | 206 |
| Tabela 99 – Exploração do espaço e seleção dos hiperparâmetros. . . . .                               | 206 |
| Tabela 100 – Matriz de confusão do modelo RNMP com dados da fase de teste. . . .                      | 207 |
| Tabela 101 – Resultados de precisão, sensibilidade e <i>F1-score</i> do modelo RNMP . . .             | 207 |
| Tabela 102 – Valores frequência do sinal em cada classe. . . . .                                      | 209 |
| Tabela 103 – Número de amostras: Aplicação do <i>data augmentaion</i> (Algoritmo 38)                  | 209 |
| Tabela 104 – Dados estruturados. . . . .                                                              | 211 |
| Tabela 105 – Matriz de confusão do modelo RNMP com dados da fase de teste . . .                       | 212 |
| Tabela 106 – Resultados de precisão, sensibilidade e <i>F1-score</i> do modelo RNMP . . .             | 212 |
| Tabela 107 – Validação cruzada do modelo RNC . . . . .                                                | 212 |

# Lista de abreviaturas e siglas

|       |                                               |
|-------|-----------------------------------------------|
| AM    | Aprendizado de Máquina                        |
| ACP   | Análise de Componentes Principais             |
| ADL   | Análise Discriminante Linear                  |
| AP    | Aprendizado Profundo                          |
| BP    | <i>Backpropagation</i>                        |
| BP-AG | <i>Backpropagation</i> -Algoritmo Genético    |
| CAD   | <i>Computer Aided Design</i>                  |
| CFD   | <i>Computacional Fluid Dynamics</i>           |
| CNN   | <i>Convolutional Neural Network</i>           |
| DBN   | <i>Deep Belief Network</i>                    |
| DCBD  | Descoberta de Conhecimento em Bancos de Dados |
| DEP   | Densidade Espectral de Potência               |
| DFC   | Dinâmica dos Fluidos Computacional            |
| DL    | <i>Deep Learning</i>                          |
| EA    | Emissão Acústica                              |
| FFT   | <i>Fast Fourier Transform</i>                 |
| GAN   | <i>Generative Adversarial Networks</i>        |
| GDE   | Gradiente Descendente Estocástico             |
| HIT   | Técnica de Contagem de <i>Hits</i>            |
| IA    | Inteligência Artificial                       |
| ICT   | Instituto de Ciência e Tecnologia             |
| KDD   | <i>Knowledge Discovery in Databases</i>       |
| LDA   | <i>Linear Discriminant Analysis</i>           |

|        |                                                   |
|--------|---------------------------------------------------|
| ML     | <i>Machine Learning</i>                           |
| MLP    | <i>Multilayer Perceptron</i>                      |
| MCP    | Multicamadas de Perceptrons                       |
| MGD    | Método do Gradiente Descendente                   |
| MVS    | Máquina de Vetor de Suporte                       |
| NPSH   | <i>Net Position Suction Head</i>                  |
| OB     | Otimização de Bayes                               |
| PCA    | <i>Principal Component Analysis</i>               |
| PTI-BR | Parque Tecnológico Itaipu Brasil                  |
| PSD    | <i>Power Spectral Density</i>                     |
| PSVM   | <i>Proximal Suport Machine</i>                    |
| RNC    | Rede Neural Convolutcional                        |
| RGB    | <i>Red-Green-Blue</i>                             |
| RNMP   | Rede Neural Multicamadas de Perceptrons           |
| ROC    | <i>Receiver Operating Characteristic</i>          |
| SAD    | Sistema de Aquisição de Dados                     |
| SMOTE  | <i>Synthetic Minority Over-sampling Technique</i> |
| STFT   | <i>Short Time Fourier Transform</i>               |
| SVM    | <i>Support Vector Machine</i>                     |
| TF     | Transformada de Fourier                           |
| TFCT   | Transformada de Fourier de Curto Tempo            |
| t-SNE  | <i>Distributed Stochastic Neighbor</i>            |
| TRF    | Transformada Rápida de Fourier                    |
| TW     | Transformada de Wavellet                          |
| VLSI   | <i>Very Large Scale Integration</i>               |

# Sumário

|          |                                                     |           |
|----------|-----------------------------------------------------|-----------|
| <b>1</b> | <b>INTRODUÇÃO</b>                                   | <b>22</b> |
| 1.1      | Formulação do Problema                              | 22        |
| 1.2      | Motivação                                           | 23        |
| 1.3      | Justificativa                                       | 24        |
| 1.4      | Objetivos                                           | 25        |
| 1.5      | Organização do Trabalho                             | 26        |
| 1.6      | Questão de Pesquisa e Contribuição                  | 27        |
| <b>2</b> | <b>PESQUISA BIBLIOGRÁFICA</b>                       | <b>28</b> |
| 2.1      | Crerérios e Classificação dos Trabalhos Científicos | 28        |
| 2.2      | Resultados Relevantes dos Trabalhos Científicos     | 31        |
| 2.3      | Dados Relevantes à Pesquisa                         | 44        |
| <b>3</b> | <b>FUNDAMENTAÇÃO TEÓRICA</b>                        | <b>48</b> |
| 3.1      | Equipamentos e Fenômenos Físicos                    | 48        |
| 3.1.1    | Bombas Centrífugas                                  | 48        |
| 3.1.2    | Fenômeno de Cavitação                               | 49        |
| 3.2      | Análise de Sinais e Transformação dos Dados         | 51        |
| 3.2.1    | Técnica de Contagem de <i>Hits</i>                  | 51        |
| 3.2.2    | Transformadas de Fourier                            | 54        |
| 3.3      | Aprendizados de Máquina e Profundo                  | 59        |
| 3.3.1    | Conceitos Fundamentais                              | 60        |
| 3.3.2    | Terminologias                                       | 62        |
| 3.3.3    | Técnicas para Extração de Características           | 65        |
| 3.3.4    | Aprendizado de Máquina × Aprendizado Profundo       | 66        |
| 3.4      | Redes Neurais                                       | 68        |
| 3.4.1    | Introdução                                          | 68        |
| 3.4.2    | Formulação Teórica e Matemática                     | 69        |
| 3.5      | Rede Neural Multicamadas de Perceptrons (RNMP)      | 72        |
| 3.5.1    | Introdução                                          | 72        |
| 3.5.2    | Arquitetura das RNMP                                | 73        |
| 3.5.3    | Etapas de Treinamento e Teste da Rede Neural MLP    | 75        |
| 3.5.4    | Fase de Propagação <i>Forward</i>                   | 76        |
| 3.5.5    | Fase de Retropropagação <i>Backward</i>             | 76        |
| 3.5.6    | Métodos de Otimização                               | 78        |
| 3.6      | Rede Neural Convolutacional (RNC)                   | 80        |

|            |                                                                 |            |
|------------|-----------------------------------------------------------------|------------|
| 3.6.1      | Introdução . . . . .                                            | 80         |
| 3.6.2      | Operação de Convolução . . . . .                                | 80         |
| 3.6.3      | Camada <i>Pooling</i> . . . . .                                 | 82         |
| 3.6.4      | Tipos Arquitetura de RNC . . . . .                              | 83         |
| <b>3.7</b> | <b>Técnicas e Métricas ao Desempenho das Redes</b> . . . . .    | <b>84</b>  |
| 3.7.1      | Problemas de Aprendizado em Redes Neurais . . . . .             | 84         |
| 3.7.2      | Técnicas de Inicialização dos Pesos das Redes Neurais . . . . . | 85         |
| 3.7.3      | Técnica de Normalização em Lotes ( <i>Batches</i> ) . . . . .   | 86         |
| 3.7.4      | Técnicas de Regularização . . . . .                             | 87         |
| 3.7.5      | Técnicas de Balanceamento dos Dados . . . . .                   | 89         |
| 3.7.6      | Método Decomposição do Valor Singular . . . . .                 | 90         |
| 3.7.7      | Otimização Bayesiana: Cálculo de Hiperparâmetros . . . . .      | 92         |
| <b>4</b>   | <b>MATERIAIS E MÉTODOS</b> . . . . .                            | <b>96</b>  |
| <b>4.1</b> | <b>Introdução aos Modelos: Uma Visão Geral</b> . . . . .        | <b>97</b>  |
| <b>4.2</b> | <b>Base de Dados</b> . . . . .                                  | <b>98</b>  |
| 4.2.1      | Dados de Cavitação via Acelerômetros . . . . .                  | 98         |
| 4.2.2      | Dados de Rolamento via Acelerômetros . . . . .                  | 100        |
| <b>4.3</b> | <b>Dados</b> . . . . .                                          | <b>102</b> |
| <b>4.4</b> | <b>Preparação dos Dados</b> . . . . .                           | <b>103</b> |
| 4.4.1      | Modelo RNMP . . . . .                                           | 103        |
| 4.4.2      | Modelo RNC . . . . .                                            | 105        |
| <b>4.5</b> | <b>Extração de Características</b> . . . . .                    | <b>105</b> |
| 4.5.1      | Modelo RNMP . . . . .                                           | 105        |
| 4.5.2      | Modelo RNC . . . . .                                            | 109        |
| <b>4.6</b> | <b>Aprendizado de Máquina</b> . . . . .                         | <b>110</b> |
| <b>4.7</b> | <b>Análise de Resultados</b> . . . . .                          | <b>117</b> |
| 4.7.1      | Métricas de Desempenho . . . . .                                | 117        |
| 4.7.2      | Análise Exploratória dos Dados . . . . .                        | 117        |
| <b>4.8</b> | <b>Fluxograma conceitual completo dos modelos</b> . . . . .     | <b>117</b> |
| <b>5</b>   | <b>DESENVOLVIMENTO DOS ALGORITMOS</b> . . . . .                 | <b>120</b> |
| <b>5.1</b> | <b>Modelo RNMP</b> . . . . .                                    | <b>120</b> |
| 5.1.1      | Dados . . . . .                                                 | 122        |
| 5.1.2      | Preparação dos Dados . . . . .                                  | 123        |
| 5.1.3      | Extração de Características . . . . .                           | 126        |
| 5.1.4      | Transformação dos Dados . . . . .                               | 133        |
| 5.1.5      | Aprendizado de Máquina . . . . .                                | 136        |
| <b>5.2</b> | <b>Modelo RNC</b> . . . . .                                     | <b>142</b> |
| 5.2.1      | Dados . . . . .                                                 | 144        |

|            |                                                              |            |
|------------|--------------------------------------------------------------|------------|
| 5.2.2      | Preparação dos Dados . . . . .                               | 144        |
| 5.2.3      | Aprendizado de Máquina . . . . .                             | 148        |
| <b>6</b>   | <b>ANÁLISE DE RESULTADOS . . . . .</b>                       | <b>154</b> |
| <b>6.1</b> | <b>Análise Exploratória dos Dados de Rolamento . . . . .</b> | <b>154</b> |
| 6.1.1      | Definição dos Rótulos . . . . .                              | 154        |
| 6.1.2      | Definição do Valor Limiar . . . . .                          | 161        |
| 6.1.3      | Análise dos parâmetros da Contagem de <i>Hits</i> . . . . .  | 165        |
| 6.1.4      | Análise de Frequência do Sinal . . . . .                     | 168        |
| <b>6.2</b> | <b>Modelo RNMP . . . . .</b>                                 | <b>174</b> |
| <b>6.3</b> | <b>Modelo RNC . . . . .</b>                                  | <b>180</b> |
| <b>6.4</b> | <b>Desempenho dos modelos RNMP e RNC . . . . .</b>           | <b>186</b> |
| <b>6.5</b> | <b>Dados de Cavitação . . . . .</b>                          | <b>191</b> |
| 6.5.1      | Análise Exploratória dos Dados . . . . .                     | 191        |
| 6.5.2      | Definição do Valor Limiar . . . . .                          | 196        |
| 6.5.3      | Análise dos parâmetros da Contagem de <i>Hits</i> . . . . .  | 200        |
| 6.5.4      | Análise de Frequência do Sinal . . . . .                     | 201        |
| <b>6.6</b> | <b>Modelo RNMP . . . . .</b>                                 | <b>202</b> |
| <b>6.7</b> | <b>Modelo RNC . . . . .</b>                                  | <b>208</b> |
| <b>7</b>   | <b>CONCLUSÃO . . . . .</b>                                   | <b>213</b> |
|            | <b>REFERÊNCIAS . . . . .</b>                                 | <b>216</b> |

# 1 Introdução

## 1.1 Formulação do Problema

As bombas centrífugas, responsáveis pelo transportes de fluidos, apresentam desgastes em suas estruturas em razão do seu uso e, também, pelo surgimento de cavitação hidráulica. O desgaste provocado pela cavitação é detectável durante a desmontagem do equipamento quando são observados erosões, trincas e desgastes superficiais que acabam comprometendo o desempenho e a vida útil da bomba (KARASSIK et al., 2008).

Para enfrentar esse desafio, profissionais da área têm buscado constantemente soluções para minimizar o impacto estrutural e operativo causados pela cavitação em bombas centrífugas. Entre as estratégias adotadas, destaca-se a melhoria do projeto, escolha adequada de materiais e o próprio processo de fabricação das bombas, tornando-as mais resistentes a situações anômalas, como a cavitação (KARASSIK et al., 2008). Porém, tais abordagens são efetivadas durante o processo de confecção do equipamento, ficando a cargo da correta operação e das manutenções sobre o equipamento a prevenção da ocorrência do fenômeno e as correções dos danos estruturais.

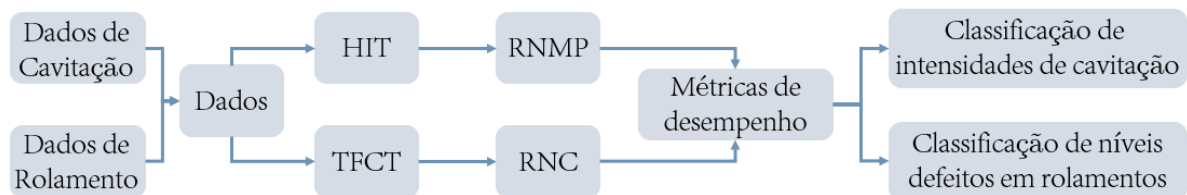
Essas avarias frequentemente afetam o rotor e os mancais de rolamentos, que são essenciais para absover as cargas dinâmicas, reduzindo o atrito entre as superfícies em movimento relativo e suportando esforços mecânicos radiais, axiais e combinados, (HARRIS; KOTZALAS, 2006), geradas durante operação que são potencializadas com presença de fenômenos hidráulicos. Essa vibração excessiva introduz falhas nos rolamentos que acabam por sua vez reduzindo a sua vida útil.

A manutenção baseada no monitoramento contínuo dos dados operativos viabiliza uma intervenção proativa, uma vez que os indícios de falhas são detectados precocemente por meio dos sensores instalados no equipamento. Com base nesses dados, pode-se gerar informações úteis para os responsáveis tomarem as decisões adequadas antes da evolução dos danos evitando paradas não programadas e reduzindo os custos de manutenção (COANDĂ; AVRAM; CONSTANTIN, 2020).

Nesse contexto, os modelos de Aprendizado de Máquina (AM) podem ser utilizados como ferramenta na identificação de padrões e na tomada de ações sem serem explicitamente programados, pois fundamentam-se na ideia de que esse tipo de abordagem aprende a partir de dados. Isso viabiliza a sua utilização quando novos conjuntos de dados são constantemente coletados, tornando-os relevantes no presente trabalho, cuja função é detectar o fenômeno de cavitação e a integridade estrutural dos rolamentos do conjunto girante, garantindo a eficiência e segurança do equipamento (SOUZA et al., 2022).

A proposta dessa pesquisa é utilizar modelos computacionais inteligentes para realizar a classificação de falhas em rolamentos e diagnosticar a presença de cavitação por meio de dados registrados por acelerômetros, ambos instalados nos mancais das bombas centrífugas. Os dados de ensaios são disponibilizados pelo Instituto *Case Western Reserve University* para o caso dos rolamentos e *UWA System Health Lab Prognostics Data Library (PDL)*<sup>1</sup> no caso do fenômeno de cavitação, cabendo a essa pesquisa o desenvolvimento dos modelos computacionais, os quais são constituídos por duas técnicas de transformação dos dados, contagem de *hits* e a Transformação de Fourier de Curto Tempo (TFCT), duas redes neurais, a saber: as Redes Neurais Multicamadas de Perceptrons (RNMP) e a Rede Neural Convolucional (RNC), modelos estatísticos para a interpretação e análise exploratória dos dados e métricas de desempenho para a avaliação dos resultados. Na Figura 1 é ilustrado o fluxograma dos métodos envolvidos na pesquisa.

Figura 1 – Modelos aprendizado de máquina para a solução do problema motivador.



Fonte: Autor.

O primeiro modelo corresponde a uma RNMP, que utiliza parâmetros resultantes das técnicas de contagem de *hits* como dados de entrada (técnica de caracterização da forma da onda no domínio do tempo). O segundo modelo emprega uma RNC aplicada aos dados dos acelerômetros tratado pela TFCT, diminuindo a necessidade do especialista na etapa de extração de características do sinal em alto nível.

## 1.2 Motivação

Nos últimos anos, a área de AM tem testemunhado um crescimento expressivo no desenvolvimento e aplicações técnicas cotidianas, tornando-se uma poderosa ferramenta no diagnóstico de falhas de equipamentos, estruturas e vários outros instrumentos focados em aplicações de engenharia. Esse crescimento se deve, em grande parte, ao aumento e disponibilidade dos dados e no avanço das técnicas de AM (LEI et al., 2020). Nesse contexto, a motivação para desenvolver um modelo de AM nesta dissertação é impulsionada por um projeto de pesquisa desenvolvido entre a parceria de instituição de pesquisa, universidade e empresa com olhares aos avanços tecnológicos na área de turbomáquinas (PARQUETEC, 2023).

<sup>1</sup> Base de dados web destinado a pesquisadores, indústrias e acadêmicos cuja a finalidade é testar modelos prognósticos.

A Itaipu Parquetec está desenvolvendo o projeto de pesquisa chamado Monitoramento Online de Turbinas por Emissão Acústica (MOMEA) para criar uma metodologia que viabiliza o monitoramento contínuo e a detecção de cavitação e trincas em rotores de turbinas hidráulicas. O projeto se fundamenta em três pilares: compreensão dos fenômenos físicos (cavitação e trincamento), desenvolvimento do sistema de aquisição de dados adequado às técnicas de ensaio de emissão acústica e utilização de modelos numéricos para diagnosticar a presença dos fenômenos (PARQUETEC, 2023) <sup>2</sup>.

O motivo para essa pesquisa decorre do fato de que a cavitação em bombas hidráulicas impacta negativamente o desempenho operacional e pode causar diversos problemas estruturais, como ruído excessivo, vibrações acima do limite permitido, redução de potência/eficiência e danos nos próprios mancais de rolamento. Tais anomalias fornecem indícios da ocorrência desse fenômeno, identificados durante inspeções visuais e manutenções (FERNÁNDEZ-OSETE et al., 2024).

Para coletar e armazenar os dados de maneira apropriada, é crucial empregar um sistema de aquisição alinhado com as especificações técnicas do fenômeno e das técnicas de ensaio. O uso de acelerômetros para monitoramento de vibração, caracterizado como um método não-destrutivo, oferece várias vantagens. Não expõe o equipamento a condições críticas e não altera suas propriedades físicas, químicas, mecânicas ou dimensionais. Essa técnica é capaz de detectar anomalias em elementos estruturais como rolamentos, e operacionais, como cavitação, tanto em monitoramentos intermitentes (ensaios) quanto contínuos (operação dos equipamentos).

A proposta de desenvolver modelos computacionais que utilizam técnicas de ensaios convencionais para monitorar fenômenos de cavitação e degradação de componentes mecânicos, como rolamentos, é valiosa. Esse enfoque permite aos especialistas tomar decisões com base em uma abordagem alinhada a sua experiência. Além disso, a geração de conhecimento científico ocorre ao explorar abordagens tradicionais, técnicas avançadas e combinações de algoritmos nas diversas áreas de conhecimento relacionadas a este tema. Tal exploração não apenas amplia a compreensão teórica e prática em aprendizado de máquina, mas também possibilita novas pesquisas e aplicações.

## 1.3 Justificativa

As bombas centrífugas são equipamentos vitais para a operação do setor industrial, seja pela fabricação de produtos, máquinas e ferramentas. A paralisação desses equipamentos industriais pode gerar consequências desastrosas para a manufatura de uma fábrica. A cavitação é um fenômeno hidrodinâmico que pode ocasionar danos severos a vários componentes mecânicos, incluindo o impulsor (que sofre erosão e desgaste devido ao colapso

<sup>2</sup> Pesquisa da aplicabilidade da técnica de emissão acústica no diagnóstico de cavitação em turbinas.

das bolhas de vapor), a carcaça da bomba (afetada por *pitting* e degradação superficial), os mancais e rolamentos (desgaste prematuro e falhas devido às vibrações e cargas dinâmicas), os vedantes e juntas (prejudicados por vazamentos e falhas), e o eixo (sujeito a fadiga do material), resultando em uma redução na eficiência da bomba, aumento dos custos de manutenção e tempo de inatividade (KARASSIK et al., 2008).

Tradicionalmente, a detecção e o diagnóstico de anomalias dependem de manutenções periódicas, monitoramento baseado em sensores e análise de sinais realizadas por especialistas. A aplicação de técnicas de aprendizado de máquina como ferramenta de apoio ao diagnóstico oferece maior eficiência ao processar grandes volumes de dados operacionais coletados por sensores de vibração, pressão, acústica ou fluxo. Esses métodos permitem a identificação e a correlação desses padrões com a presença e a intensidade de danos estruturais em componentes mecânicos causados pela cavitação. Isso viabiliza não apenas uma detecção precoce e precisa do fenômeno, mas também facilita a manutenção desses componentes mecânicos, aumentando a robustez e a confiabilidade dos sistemas de monitoramento. A capacidade de adaptar e calibrar modelos de acordo com os dados específicos de cada bomba e ambiente operacional aumenta ainda mais a precisão do diagnóstico, proporcionando uma solução eficaz (ALVAREZ, 2020).

Dentro desse contexto, os modelos computacionais baseados em dados podem ser utilizados para classificar defeitos mecânicos e intensidade de cavitação por meio de dados de monitoramento, com o objetivo de identificar possíveis problemas e agir preventivamente sobre o equipamento. Esse modelo de diagnóstico representa um passo significativo em direção à solução do problema motivador.

## 1.4 Objetivos

### Objetivo Geral:

Desenvolver um modelo computacional baseado em aprendizado de máquina para a identificação de cavitação em bombas centrífugas e classificação de falhas em seus rolamentos utilizando acelerômetros.

### Objetivos Específicos

- Realizar pesquisa bibliográfica de trabalhos correlatos, com os temas de rolamentos, cavitação, acelerômetros, bombas centrífugas e AM;
- Desenvolver uma metodologia para a extração características destinada a contagem de *hits* e para a transformada de Fourier de Curto Tempo;
- Elaborar algoritmos relacionados as etapas de tratamento, transformação, extração de características, otimização de hiperparâmetros e análise estatística para subsidiar

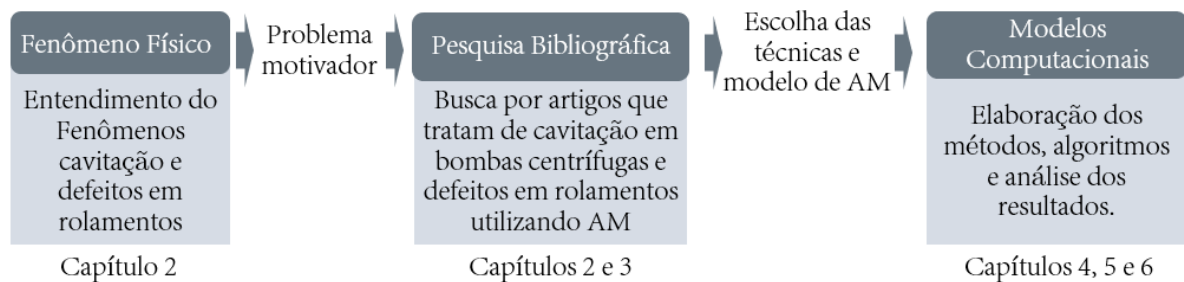
a modelagem em AM;

- Elaborar algoritmos, a partir da biblioteca do Pytorch, para a criação dos modelos RNMP e RNC;
- Obter valores de hiperparâmetros com base nas referências da pesquisa bibliográfica e técnica de otimização *Bayes*;
- Avaliar o desempenho dos modelos RNMP e RNC empregando bases de dados consolidadas e comparando com os artigos de referências.

## 1.5 Organização do Trabalho

A estrutura deste trabalho está centrada no problema motivador. Isso abrange desde a compreensão do fenômeno físico e mecanismos de desgastes, a investigação de trabalhos correlatos com o foco em aplicações em sistemas similares e o desenvolvimento dos modelos computacionais. Na Figura 2, é ilustrada essa organização e a relação entre os conteúdos gerados nos respectivos capítulos.

Figura 2 – Organização e estrutura do trabalho.



Fonte: Autor.

Na seção de pesquisa bibliográfica (Capítulo 2), buscaram-se trabalhos correlatos a proposta dessa pesquisa, com o intuito de identificar modelos de aprendizado de máquina, modelos de otimização para o cálculo dos hiperparâmetros e maneiras de correlacionar os dados de ensaio com os rótulos.

Na seção de fundamentação teórica (Capítulo 3), abordam-se os principais conceitos físicos, assim como os fundamentos matemáticos e computacionais do aprendizado de máquina. Explora-se temas como redes multicamadas de perceptrons e convolucionais, juntamente com técnicas para a parametrização dos modelos, incluindo aspectos como a inicialização dos pesos, normalização, regularização e otimização tanto dos parâmetros das redes quanto dos hiperparâmetros, visando aprimorar a eficácia e o desempenho desses modelos.

Na seção de materiais e métodos (Capítulo 4) é apresentado o processo metodológico da pesquisa, que contempla as bases de dados de cavitação e rolamentos, os

fluxogramas dos modelos estatísticos, de *Hits*, TFCT, RNMP e RNC. Além disso, são apresentados os valores iniciais dos hiperparâmetros para a otimização de Bayes.

Na seção de desenvolvimento (Capítulo 5), são expostos os algoritmos implementados em linguagem de programação Python, os quais abrangem o modelo computacional destinado ao diagnóstico de cavitação e falhas em rolamento esféricos.

Na seção de análise de resultados (Capítulo 6) foi realizada uma análise exploratória dos dados provenientes do ensaio (em sua forma bruta) para definir o valor limiar utilizando métricas estatísticas e o tamanho da janela retangular deslizante para o aumento da base de imagens por meio de uma análise de sinais por Transformada de Fourier (TF) e a qualidade dos dados provenientes da técnica de contagem de *hits*. Nas seções seguintes são apresentados os resultados das etapas da modelagem RNC e RNMP a classificação de cavitação baseada em as métricas de matriz de confusão, acurácia, precisão e *F1-score*.

## 1.6 Questão de Pesquisa e Contribuição

Como problema de pesquisa, foi realizada uma comparação entre o desempenho computacional e a significância dos resultados dos modelos RNC e RNMP utilizando dados provenientes de ensaios e trabalhos de referência. Além disso, a pesquisa contribuiu com métodos estatísticos e testes de hipóteses sobre os valores médios dos sinais para a definição de classes em modelos de AM supervisionados, oferecendo uma alternativa em situações onde não há uma definição dessas classes. Além disso, foi avaliada a influência de técnicas de transformação utilizando a Decomposição do Valor Singular (DVS) e o balanceamento dos dados por meio da combinação do *Synthetic Minority Over-sampling Technique* (SMOTE) com o *Edited Nearest Neighbors* (ENN).

## 2 Pesquisa Bibliográfica

Neste capítulo apresenta-se a revisão bibliográfica realizada na área de aprendizado de máquina, com foco na cavitação em bombas centrífugas. O objetivo foi identificar debates, lacunas e avanços recentes. Resultados relevantes de outros estudos e abordagens são apresentados para contribuir com o estado da arte nessa área de conhecimento.

A pesquisa resulta na seleção de modelos para classificar falhas e condições operacionais anormais em bombas centrífugas, com definição de hiperparâmetros para cada modelo e sua respectiva aplicação. Além disso, um mapeamento cronológico das referências destaca a evolução do tema em relação ao problema e à questão de pesquisa. Portanto, as seções deste capítulo são divididas em uma perspectiva panorâmica da pesquisa bibliográfica, seguida da síntese dos artigos científicos focados no problema e formulação e escolha dos métodos envolvidos nesta pesquisa e suas contribuições.

### 2.1 Critérios e Classificação dos Trabalhos Científicos

A revisão bibliográfica da pesquisa foi realizada com base em publicações de revistas científicas, estando disponíveis nas bases *Science Direct*, *Scopus* e Google. A busca foi baseada em um conjunto de palavras chaves aplicadas sobre os títulos dos artigos científicos e tal escolha está vinculada aos objetivos gerais e específicos dessa dissertação<sup>1</sup>. Então foi construída uma base de dados com artigos classificados de forma cronológica e numerados de acordo com sua aderência, conforme a seguinte padronização:

- **Baixo:** Artigos relacionados a aprendizado de máquina em bombas centrífugas;
- **Médio:** Artigos que tratam de aprendizado de máquina para o diagnóstico de cavitação em bombas centrífugas com outros tipos de sensores;
- **Alto:** Trabalhos de aprendizado de máquina para o diagnóstico de cavitação em bombas centrífugas com sensores de emissão acústica;
- **Apoio:** Trabalhos que tratam de métodos ou modelos numéricos referentes ao pré e pós processamento de dados e técnicas de otimização.

Os termos de busca incluíram palavras-chave como *Pump*, *Acoustic*, *Cavitation*, *Artificial neural network*, *Machine Learning*, *Deep Learning*, *Convolutional Neural Network* e RNC e suas combinações. O resultado da pesquisa, conforme apresentado na Tabela 1, revela que foram encontrados 194 artigos científicos com as condições de busca. No entanto,

<sup>1</sup> Essa pesquisa é válida até o dia 1/4/22

apenas 44 destes artigos atenderam aos critérios de classificação (baixo, médio e alto). A pesquisa incluiu 29 trabalhos científicos publicados na área de aprendizado de máquina aplicados a bombas centrífugas, 9 trabalhos relacionados ao aprendizado de máquina aplicado a bombas centrífugas e cavitação, e, por fim, 6 artigos científicos relacionados ao tema desta pesquisa.

Tabela 1 – Resultados relevantes dos trabalhos científicos.

| Palavras Chaves               | Pesquisa | Nível de Aderência |       |       |       |      |
|-------------------------------|----------|--------------------|-------|-------|-------|------|
|                               |          | Total              | Apoio | Baixa | Média | Alta |
| <i>Acoustic</i>               | 3        | 1                  | 1     | -     | -     | -    |
| <i>Cavitation</i>             | 95       | 17                 | 13    | 1     | 3     | -    |
| <i>IA</i>                     | 81       | 36                 | -     | 27    | 2     | 2    |
| <i>Acoustic+Cavitation</i>    | 5        | 0                  | -     | -     | -     | -    |
| <i>Cavitation+IA</i>          | 8        | 7                  | -     | 1     | 4     | 2    |
| <i>Acoustic+Cavitation+IA</i> | 2        | 2                  | -     | -     | -     | 2    |
| <b>Total</b>                  | 194      | 63                 | 14    | 29    | 9     | 6    |

Fonte: Autor.

São apresentados os trabalhos científicos selecionados da pesquisa bibliográfica, conforme critérios de classificação (baixa, média e alta), que serviram como base para a modelagem e escolhas das técnicas numéricas. As Tabelas 1, 2 e 3, nomeiam esses trabalhos de acordo com a classificação de aderência e oferecem um breve resumo que justifica a relevância da classificação. Embora a pesquisa tenha identificado 29 trabalhos científicos de baixa aderência, na Seção 3.2 optou-se por detalhar apenas os trabalhos relacionados a RNC, Emissão Acústica (EA), Transformada de Wavellet (TW), totalizando 5 trabalhos.

Quadro 1 – Resultado da pesquisa: baixo nível de aderência.

| Título                                                                                                       | Equipamento, sensor e método                                                                                     |
|--------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| 1. Research on Fault Diagnosis of Hydraulic Pump Using Convolutional Neural Network (YAN et al., 2016)       | Diagnóstico estrutural de uma bomba centrífuga com acelerômetros e aprendizagem de máquina (RNC e MVS)           |
| 2. Research on Fault Diagnosis of Mud Pump Fluid end Based Acoustic Emission (DENG et al., 2017)             | Diagnóstico estrutural de uma bomba com sensores acústicos e transformada de Wavelet, algoritmo genético e RNMP. |
| 3. A New Bearing Fault Diagnosis Method Based on Modified Convolutional Neural Networks (ZHANG et al., 2020) | Diagnóstico estrutural dos mancais de rolamento de uma bomba com acelerômetros e RNC.                            |
| 4. Improved Deep CNN for the Identification of Defects in Pumps Using Acoustic Images (KUMAR et al., 2020)   | Diagnóstico estrutural de uma bomba com sensores acústicos e a transformada de Wavelet e uma RNC.                |
| 5. Fault Diagnosis in Centrifugal Pump Using M and ANN (RANAWAT; KANAKAR; MIGLANI, 2020)                     | Diagnóstico estrutural de uma bomba centrífuga com acelerômetros e modelos de classificação MVS e RNMP           |

Quadro 2 – Resultado da Pesquisa: médio nível de aderência

| Título                                                                                                                                                         | Equipamento, sensor e método                                                                                               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| 6. Prediction of Flow Blockages and Impending Cavitation in Centrifugal Pumps using SVM Algorithms Based on Vibration (PANDA; RAPUR; TIWARI, 2018)             | Diagnóstico de cavitação em bombas centrífugas utilizando acelerômetros e <i>Support Vector Machine</i> (SVM).             |
| 7. Gas-Liquid Two-Phase Flow Pattern Identification of a Centrifugal Pump Based in ANN (HE et al., 2021)                                                       | Detecção e classificação de um fluxo multifásico em bombas centrífugas com Redes Neurais Artificiais (ANN).                |
| 8. Blockage and Cavitation Detection In Centrifugal Pumps from Pressure Signal Using Deep Learning (TIWARI; BORDOLOI; DEWANGAN, 2021)                          | Detecção de bloqueio e cavitação em bombas hidráulicas com sensores de pressão e RNMP.                                     |
| 9. Identification of Cavitation Intensity for Aviation Hydraulic Pumps Using 2D CNN with na Input of RGB-based Vibration Data (CHAO et al., 2020b)             | Classificação de intensidade de cavitação com acelerômetros e redes neurais convolucionais 2D (RNC).                       |
| 10. Cavitation Intensity Recognition for High-Speed Axial Piston Pumps Using CNN 1-D with Multi-Channel Inputs of Vibration Signals (CHAO et al., 2020a)       | Classificação de intensidade de cavitação com acelerômetros e Redes Neurais Convolucionais 1D (RNC).                       |
| 11. Identification Inlet Pipe Blockage Level in Centrifugal Pump Over a Range of Speed By Deep Learning Algorithm Using multi-Source Data (KUMAR et al., 2021) | Detecção de bloqueio e cavitação em bombas hidráulicas com sensores de pressão, acelerômetros e sensores elétricos e RNMP. |

Quadro 3 – Resultado da pesquisa: alto nível de aderência.

| Título                                                                                                                                     | Equipamento, sensor e método                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| 12. An Improved CNN with an Adaptable Learning rate towards Multi-Signal Fault Diagnosis of Hydraulic Piston (TANG; ZHU; YUAN, 2021)       | Detecção estrutural em bombas hidráulicas com sensores acústicos e Rede Neural Convolucional.    |
| 13. A nove adaptive convolutional neural network for fault diagnosis of hydraulic piston Pump with acoustic images (TANG; ZHU; YUAN, 2022) | Detecção estrutural em bombas hidráulicas com sensores de acústicos e Rede Neural Convolucional. |

## 2.2 Resultados Relevantes dos Trabalhos Científicos

Essa pesquisa bibliográfica teve como propósito encontrar artigos científicos com um grau de aderência ao escopo do trabalho e que fossem capazes de fornecer informações relevantes à escolha e parametrização dos modelos de aprendizado de máquina.

Dentro desse contexto, os artigos de baixa aderência apoiam-se em aplicações de aprendizado de máquina utilizados nos diagnósticos de falhas de componentes estruturais, como rolamento, gaxeta, quebras de cilindros desgastes de válvulas de bombas hidráulicas, centrífugas e deslocamento positivo por meio de Rede Neural com Múltiplas Camadas de Perceptrons (RNMP), Redes Neurais Convolucionais (RNC) e Máquina de Vetor de Suporte (MVS) com qualquer tipo de sensor.

Já os artigos de média aderência tiveram contribuições nas aplicações das técnicas de aprendizado de máquina com o MVS, RNMP e RNC utilizando sensores de pressão, elétricos e vibração, além da utilização de modelos numéricos que viabilizam a aplicação de aprendizado profundo na etapa de extração de características e pré-processamento como a TW.

Por fim, são apresentados os trabalhos de alta aderência que tratam exatamente do objeto de estudo desse trabalho, ou seja, a classificação do fenômeno de cavitação em bombas centrífugas utilizando modelo de aprendizado de máquina como as redes neurais convolucionais por meio de dados provenientes de sensores acústicos.

A seguir, são sintetizados os 15 trabalhos científicos, seguindo a ordem de aderência, que mostraram relevância para esta pesquisa e que serviram de base para o desenvolvimento das seções subsequentes.

### 1. *Research on Fault Diagnosis of Hydraulic Pump using Convolutional Neural NetWork:*

No trabalho se desenvolveu uma abordagem para diagnosticar falhas mecânicas em bombas hidráulicas a partir dos sinais de vibração registrados durante o seu monitoramento. As falhas que foram classificadas pelo modelo computacional correspondem respectivamente: i) quebra do cilindro, ii) desgaste da válvula, iii) quebra do êmbolo, iv) mancal solto v) desgaste do pino e vi) a quebra da mola (YAN et al., 2016).

As modelagens em aprendizado de máquina utilizadas para a classificação das falhas, e posteriormente comparados, foram a Máquina de Vetor de Suporte (MVS), Rede Neural com o Algoritmo de *Backpropagation* (BP) e Rede Neural Convolucional 1D (RNC). No caso da rede convolucional foi proposta uma versão modificada, ou seja, a inclusão de uma camada entre a camada densa e a camada de saída, cuja função foi concentrar informações dos pesos da primeira camada densa. A RNC é formada por 8 camadas. Entre elas estão, a camada de entrada, camadas de convoluções, camadas

*pooling*, camadas totalmente conectadas e a camada de saída, cuja arquitetura é apresentada na Tabela 2.

Tabela 2 – Arquitetura da RNC referente a Yan et al. (2016).

| N | Tipo da camada           | Kernel-stride | Qndt. kernel | Mapa característica |
|---|--------------------------|---------------|--------------|---------------------|
| 1 | Camada de entrada        | -             | 1            | 2048                |
| 2 | Convolução 1             | 99 - 1        | 8            | $1950 \times 8$     |
| 3 | Pooling 1                | 2 - 2         | 8            | $975 \times 8$      |
| 4 | Convolução 2             | 96 - 1        | 16           | $880 \times 16$     |
| 5 | Pooling 2                | 40 - 40       | 16           | $22 \times 16$      |
| 6 | Camada densa             | 352           | 1            | -                   |
| 7 | Camada <i>weight sum</i> | 9             | 1            | -                   |
| 8 | Camada de saída          | 6             | 1            | -                   |

Fonte: Adaptado de Yan et al. (2016).

Os resultados obtidos pelos três modelos foram comparados por meio do cálculo da acurácia, número de acertos preditos. A RNC apresentou um valor médio de acerto 96%, enquanto que o MVS e o RNMP apresentaram 84% e 89% de acurácia média, respectivamente. Esses valores foram calculados por meio da média aritmética no número de acertos sobre cada falha.

## 2. *Research on Fault Diagnosis of Mud Pump Fluid and Based Acoustic Emission:*

Este estudo utilizou técnicas de emissão acústica para identificar os modos de falhas de uma bomba de lama focando no vazamento do disco da válvula, na quebra da mola e no desgaste do pistão. Os parâmetros característicos do sinal de emissão acústica foram extraídos por meio da TW, que decompõe o sinal original em várias resoluções de frequência e calcula a energia referente a cada autovetor. Esses autovetores são normalizados em função da intensidade de energia e são parâmetros de entrada para um modelo de aprendizado de máquina, que é capaz de classificar a presença de defeitos (DENG et al., 2017).

Para a classificação dos defeitos foi utilizado uma RNMP, com 8 nós de entrada e 1 camada oculta com 17 neurônios e uma camada de saída com 3 neurônios, conforme apresenta do na Tabela 3.

Tabela 3 – Arquitetura da RNMP referente a Deng et al. (2017).

| Camada | Tipo                      | Número de Nós | Função de Ativação |
|--------|---------------------------|---------------|--------------------|
| 1      | Camada de entrada         | 8             | -                  |
| 2      | Primeira camada escondida | 17            | -                  |
| 3      | Camada de saída           | 3             | -                  |

Fonte: Adaptado de Deng et al. (2017).

Para o cálculo dos pesos da rede utilizou-se o algoritmo de BP combinado com o Algoritmo Genético (BP-AG). Como resultado de desempenho o algoritmo BP-

AG teve uma acurácia média de 96,5%, enquanto que somente o algoritmo BP apresentou uma acurácia próxima de 92,8%.

### 3. *A New Bearing Fault Diagnosis Method Based on Modified Convolutional Neural Networks:*

Na pesquisa empregou-se um método de conversão dos dados temporais em imagens de duas dimensões, para serem utilizadas com as redes neurais convolucionais. Esses dados brutos em forma de imagem diminuem a dependência da experiência de especialistas no processo de extração e seleção de características. O modelo de aprendizado de máquina desenvolvido foi submetido a um conjunto de dados de falhas em mancais de rolamento de uma motobomba registrados via acelerômetros. Tais falhas se fazem presentes nas pistas de rodagem interna, externa e nas esferas de rolagem dos rolamentos. Além disso, foi comparado o desempenho dessa rede convolucional com três outros modelos de aprendizado de máquina, com o intuito de verificar a sua eficácia e precisão, sendo esses modelos o MVS, MLP e *Deep Belief Network* (DBN) (ZHANG et al., 2020).

A arquitetura RNC é formada por 2 camadas de convolução, 2 camadas *pooling* (maxPool), 2 camadas totalmente conectadas e funções de ativação *Rectified Linear Unit* (ReLU) para as camadas intermediárias e a função de ativação *Softmax* para a camada de saída para várias classes. Essas informações, que representam a arquitetura da rede, são apresentadas na Tabela 4.

Tabela 4 – Arquitetura da RNC de Zhang et al. (2020)

| N | Tipo da camada    | Kernel         | Qndt. kernel | Mapa característica       |
|---|-------------------|----------------|--------------|---------------------------|
| 1 | Camada de entrada | -              | 1            | $160 \times 160 \times 1$ |
| 2 | Convolução 1      | $10 \times 10$ | 8            | $160 \times 160 \times 8$ |
| 3 | <i>Pooling</i> 1  | -              | 8            | $160 \times 160 \times 8$ |
| 4 | Convolução 2      | $3 \times 3$   | 16           | $24 \times 24 \times 16$  |
| 5 | <i>Pooling</i> 2  | -              | 16           | $24 \times 24 \times 16$  |
| 4 | Convolução 3      | $3 \times 3$   | 32           | $12 \times 12 \times 32$  |
| 5 | <i>Pooling</i> 3  | -              | 32           | $12 \times 12 \times 32$  |
| 4 | Convolução 4      | $3 \times 3$   | 32           | $6 \times 6 \times 32$    |
| 5 | <i>Pooling</i> 4  | -              | 32           | $6 \times 6 \times 32$    |
| 6 | Camada densa 1    | 100            | 1            | -                         |
| 7 | Camada densa 2    | 10             | 1            | -                         |
| 8 | Camada de saída   | 10             | 1            | -                         |

Fonte: Adaptado de Zhang et al. (2020)

Na Tabela 5 são apresentados os hiperparâmetros utilizados no cálculo dos parâmetros das matrizes de pesos e *bias* de redes neurais, além das técnicas para melhorar a convergência e validação dos resultados.

Tabela 5 – Hiperparâmetros da rede convolucional de [Zhang et al. \(2020\)](#)

| Hiperparâmetros                     | Informações           |
|-------------------------------------|-----------------------|
| Funções de ativação                 | ReLU e <i>Softmax</i> |
| Épocas                              | 30                    |
| Taxa de aprendizagem                | 0,0016                |
| Otimizador                          | Adam                  |
| Função de custo                     | Cross-Entropy         |
| Técnica de regularização            | <i>Dropout</i>        |
| <i>Batch normalization</i>          | Sim                   |
| <i>MiniBatch</i>                    | 20                    |
| Quantidade de dados por classe      | 200                   |
| Tamanho da base de dados            | 2000                  |
| <i>K</i> do <i>Cross Validation</i> | 4                     |
| Treinamento—Teste                   | 75% - 25%             |

Fonte: Adaptado de [Zhang et al. \(2020\)](#).

Para avaliar o desempenho das redes neurais, foram empregadas as métricas de acurácia de validação cruzada e a técnica de visualização de dados t-SNE. Nesse âmbito, os resultados médios de acurácia obtidos para cada método foram 90,5% para a RNC com *dropout*, 80% para a RNC, para os modelo MLP, DBN e MVS as acurácias foram, respectivamente, 75,5%, 75,7% e 66%.

#### 4. *Improved Deep Convolutional Neural Network of the Identification of Defects in the Pumps using Acoustic Images*

O trabalho de [Kumar et al. \(2020\)](#) apresentou um modelo de diagnóstico de defeitos estruturais de bombas centrífugas utilizando redes neurais convolucionais com sinais acústicos registrados por um microfone e uma nova função de custo baseada em técnicas de regularização. A base de dados foi construída a partir de um conjunto de ensaios em que se conhecia o tipo e a intensidade do defeito. Portanto, esse modelo de aprendizado de máquina é capaz de classificar 5 condições, sendo elas a condição operativa normal (df), defeito na pista interna de rolamento (ir), defeito na pista externa de rolamento (or), entupimento (clog) e quebra do rotor (bi).

Os sinais acústicos foram pré-processados por uma TW e convertido em estruturas de dados de duas dimensões (frequência por tempo) de escala cinza, chegando à 200 imagens para cada tipo de falha com tamanho de  $74 \times 1200$  *pixels*. Na Tabela 6 é apresentada a arquitetura das RNC.

Para a avaliação dos resultados foi realizada a matriz de confusão e o cálculo da acurácia para os modelos MVS, RNMP, RNC e RNC com a nova função de custo.

Os resultados alcançados por esses quatro modelos de aprendizado de máquina foram, respectivamente, 86,4%, 93,0%, 96,8% e 100% ([KUMAR et al., 2020](#)).

Tabela 6 – Arquitetura da rede convolucional de (KUMAR et al., 2020)

| N | Tipo da camada    | Kernel | Qndt. kernel | Mapa característica |
|---|-------------------|--------|--------------|---------------------|
| 1 | Camada de Entrada | -      | 1            | 74×1200             |
| 2 | Convolução 1      | 3×3×1  | 8            | -                   |
| 3 | Pooling 1         | 2×2    | 8            | -                   |
| 4 | Convolução 2      | 3×3×8  | 16           | -                   |
| 5 | Pooling 2         | 2×2    | 16           | -                   |
| 6 | Convolução 3      | 3×3×16 | 32           | -                   |
| 7 | Camada Densa      | 13     | 1            | -                   |
| 8 | Camada de saída   | 5      | 1            | -                   |

Fonte: Adaptado de Kumar et al. (2020).

5. *Fault Diagnosis in Centrifugal Pump using Support Vector Machine and Artificial Neural Network.*

O estudo de Ranawat, Kankar e Miglani (2020) apresenta uma metodologia para diagnosticar a falha de uma bomba centrífuga usando modelos de aprendizado de máquina, como MVS e Rede Neural Artificial (RNA). Diferentes características estatísticas são extraídas no domínio do tempo e da frequência do sinal de vibração para diferentes condições de trabalho da bomba. Para diminuir a dimensionalidade dessas características são empregados os métodos  $\chi^2$ , ReliefF e XGBoost.

A base de dados para a construção dos modelos foi concebida a partir de cinco condições de trabalhos diferentes, sendo elas a operação normal, desgaste no rotor, desgastes na esfera, pista interna e pista externa do rolamento, conforme detalhado na pesquisa.

O modelo de rede neural foi considerado o mais eficiente na classificação de falhas em uma bomba centrífuga em comparação ao MVS. No caso do modelo MVS para as respectivas técnicas de dimensionalidade os resultados foram 97%, contra 98% para o modelo neural. Os resultados apresentados neste estudo mostraram que uma abordagem de aprendizado de máquina baseada em rede neural a partir das características  $\chi^2$  e XGBoost podem ser usadas para o diagnóstico de falhas em bombas centrífugas.

6. *Prediction of Flow Blockages and Impending Cavitation in Centrifugal Pumps using Support Vector Machine Algorithms Based on Vibration Measurements.*

O trabalho de Panda, Rapur e Tiwari (2018) teve como foco a elaboração de um modelo de aprendizado de máquina para o diagnóstico de obstrução do fluxo em tubulação e formação iminente de bolhas de vapor na entrada da bomba, com base em sinais de vibração. O modelo de aprendizado de máquina utilizado foi o algoritmo Máquina de Vetor de Suporte (MVS) com variáveis estatísticas (média, desvio pa-

drão, curtose, assimetria e entropia) extraídos do sinal temporal como parâmetros de entrada para o modelo.

O experimento corresponde a um sistema hidráulico que contém uma bomba centrífuga, mancais de rolamento e uma válvula reguladora de vazão. A partir de dois acelerômetros triaxiais colocados no invólucro da bomba e no suporte do rolamento são coletadas as assinaturas de vibração em diferentes condições de obstrução de fluxo (B0: 0%, B1: 16,7%, B2: 33,3%, B3: 50% e B4: 66,6% de obstrução) e diferentes velocidades de rotação (40Hz até 65Hz) no início da formação de bolhas de vapor (PANDA; RAPUR; TIWARI, 2018).

Observou-se que os resultados da acurácia de cavitação foram de 70,9% à 97,1% para as velocidades de 40Hz e 65Hz, enquanto que as outras velocidades ficaram dentro desse intervalo de acerto. Esses valores foram obtidos a partir das médias dos valores de acerto dos cinco níveis de obstrução.

#### 7. *Gas-Liquid Two-Phase Flow Pattern Identification of a Centrifugal Pump Based in Artificial Neural Network.*

No artigo desenvolveu-se um modelo de aprendizado de máquina para realizar a identificação de fluxo bifásico gás-líquido dentro do rotor de uma bomba centrífuga. Nesse experimento foram usados água e ar comprimido como fluidos de trabalho. O ar comprimido foi regulado e injetado na fase líquida, e ambos os fluxos foram misturados antes de entrar na bomba. Após a saída da bomba, o gás e o líquido foram separados, e os parâmetros de fluxo, pressão e temperatura foram medidos.

Esse modelo é capaz de classificar 4 padrões de fluxo, sendo eles, fluxo de bolhas, fluxo de bolhas aglomeradas, fluxo de bolsa de gás e fluxo segregado, a partir dos parâmetros operativos do equipamento, como, pressão, vazão volumétrica de água, rotação da bomba e fração volumétrica de gás.

Para superar o desequilíbrio dos dados oriundos das quatro condições de fluxo e evitar uma redução significativa no desempenho de classificação do modelo, foi realizada uma sobreamostragem utilizando abordagem denominada pelo artigo por algoritmo SMOTE.

Segundo He et al. (2021), o modelo de classificação corresponde a uma Rede Neural Multicamadas de Perceptrons (RNMP) em sua camada de entrada as variáveis operativas, duas camadas escondidas com 512 neurônios cada e uma camada de saída representada pelos quatro padrões de fluxos, ou seja, 4 neurônios. A arquitetura da rede é apresentada na Tabela 7, enquanto que os hiperparâmetros utilizados são apresentados na Tabela 8.

Tabela 7 – Arquitetura da RNMP de He et al. (2021).

| Camada | Tipo                      | Número de Nós | Função de Ativação |
|--------|---------------------------|---------------|--------------------|
| 1      | Camada de entrada         | 4             | -                  |
| 2      | Primeira camada escondida | 512           | ReLU               |
| 3      | Segunda camada escondida  | 512           | ReLU               |
| 4      | Camada de saída           | 4             | Sigmoide           |

Fonte: Adaptado de He et al. (2021)

Tabela 8 – Hiperparâmetros da RNMP de He et al. (2021).

| Hiperparâmetros                        | Valores              |
|----------------------------------------|----------------------|
| Épocas                                 | <i>Early Stop</i>    |
| Taxa de aprendizagem                   | 0,01; 0,001; 0,0001  |
| Técnica de regularização               | $L_2$                |
| Função de custo                        | <i>Cross-Entropy</i> |
| Quantidade de dados por classe (SMOTE) | 72                   |
| Tamanho da base de dados (SMOTE)       | 288                  |
| Treinamento-Teste                      | 80-20                |

Fonte: Adaptado de He et al. (2021)

Os autores verificaram, em sua análise de resultados, as métricas de taxa de acerto, matriz de confusão, curva ROC, coeficientes *Kappa*, *Micro-F1* e *Macro-F1*. Os resultados alcançados para os dados de teste com o algoritmo SMOTE foram 93,18% para a taxa de acerto, 0,90 para o coeficiente kappa, 0,90 *Micro-F1* e 0,93 *Macro-F1*. Desse modo a utilização do algoritmo SMOTE melhorou os resultados em aproximadamente 15% (HE et al., 2021).

#### 8. Blockage and Cavitation Detection In Centrifugal Pumps from Dynamic Pressure Signal Using Deep Learning Algorithm.

O trabalho de Tiwari, Bordoloi e Dewangan (2021) apresentou um modelo computacional de aprendizado máquina para identificar a presença de bloqueio na tubulação e severidade de cavitação nos rotores de bombas centrífugas. Para a criação desse modelo utilizaram-se informações da pressão do escoamento, métricas estatísticas para a extração das características e uma rede neural multicamadas de perceptrons.

Dentro do escopo desta pesquisa, foi montado um experimento composto por uma bomba centrífuga, tubulações, reservatórios, válvulas e sensores de pressão, juntamente com um sistema de aquisição de dados. O objetivo foi analisar o efeito da obstrução da área transversal do fluxo na região de sucção da bomba, fazendo o uso de uma válvula mecânica modular com seis níveis de abertura ( $B0$ : 0%,  $B1$ : 16,7%,  $B2$ : 33,3%,  $B3$ : 66,6%,  $B4$ : 83,33% e  $B5$ : 100%) em combinação com oito velocidades de rotação da bomba (30Hz, 35Hz, 40Hz, 50Hz, 60Hz e 65Hz). Esse deli-

neamento experimental viabilizou a análise dos efeitos de cavitação para diferentes obstruções e velocidades de rotação (TIWARI; BORDOLOI; DEWANGAN, 2021). O modelo de classificação utilizado foi uma rede neural multicamadas de perceptrons com duas camadas ocultas e até seis parâmetros de entrada. Esses parâmetros são extraídos do sinal de pressão e correspondem às métricas estatísticas da média, desvio padrão, curtose, assimetria, variância e o valor RMS. A arquitetura da rede é apresentada na Tabela 9, enquanto que os hiperparâmetros utilizados são apresentados na Tabela 10.

Tabela 9 – Arquitetura da RNMP de Tiwari, Bordoloi e Dewangan (2021)

| Camada | Tipo                      | Número de Nós | Função de Ativação |
|--------|---------------------------|---------------|--------------------|
| 1      | Camada de entrada         | 1             | -                  |
| 2      | Primeira camada escondida | 6             | ReLU               |
| 3      | Segunda camada escondida  | 3             | ReLU               |
| 4      | Camada de saída           | 6             | Sigmoide           |

Fonte: Adaptado de Tiwari, Bordoloi e Dewangan (2021)

Tabela 10 – Hiperparâmetros da RNMP de Tiwari, Bordoloi e Dewangan (2021)

| Hiperparâmetros                | Valores       |
|--------------------------------|---------------|
| Épocas                         | 30            |
| Taxa de aprendizagem           | 0,1 e 0,01    |
| <i>Momentum</i>                | 0,1 e 0,01    |
| Técnica de regularização       | L2            |
| Função de custo                | Cross-Entropy |
| Mini <i>batch</i>              | 50            |
| Quantidade de dados por classe | 350           |
| Tamanho da base de dados       | 2000          |
| Treinamento-Teste              | 80-20         |

Fonte: Adaptado de Tiwari, Bordoloi e Dewangan (2021)

Segundo os autores a rede neural é treinada e testada para os diferentes níveis de bloqueio ( $B0$  até  $B5$ ) e diferentes valores de velocidade rotação ( $30Hz$  até  $65Hz$ ), alcançando uma acurácia média total de 92% para a identificação de cavitação com todas as variáveis estatísticas ( $\sigma$ - $\mu$ - $\chi$ - $\kappa$ ) como parâmetro de entrada para a RNMP.

#### 9. Identification of Cavitation Intensity for High-Speed Aviation Hydraulic Pumps Using 2D Convolutional Neural Networks with an Input of RGB-based Vibration Data

No artigo de Chao et al. (2020b) se apresenta um modelo de aprendizado de máquina, a fim de classificar as condições de cavitação de uma bomba hidráulica com base nos sinais de vibração medidos por meio de acelerômetros.

Os dados de vibração são coletados em três canais e transformados em imagens RGB, chegando a 3.456 imagens para a fase de treinamento e 384 imagens para a

fase de teste. Os dados foram coletados a partir de um experimento sobre um sistema hidráulico fechado com controle de temperatura, velocidade de rotação e pressão na entrada da bomba, sendo medidos por meio dos fluxos de descarga teórica e sucção real para inferir os níveis de cavitação, sendo eles sem, baixo, médio e severo. Foi realizada a criação dos rótulos e identificados quatro valores de perda volumétrica adimensional ( $\nabla\eta$ ) para a criação das classes de intensidade de cavitação.

Esses dados correspondem ao conjunto de treinamento que fundamenta o cálculo dos parâmetros da rede neural convolucional, cuja sua arquitetura é apresentada na Tabela 11. Os hiperparâmetros para esse modelos foram as funções de ativações ReLU e *softmax* e a função de *Cross-Entropy* estão apresentadas na Tabela 12 .

Tabela 11 – Arquitetura da rede convolucional de [Chao et al. \(2020b\)](#)

| N | Tipo da camada    | Kernel-stride    | Qndt. kernel | Mapa característica    |
|---|-------------------|------------------|--------------|------------------------|
| 1 | Camada de entrada | -                | 3            | $8 \times 8 \times 3$  |
| 2 | Convolução 1      | $2 \times 2 - 1$ | 32           | $8 \times 8 \times 32$ |
| 3 | <i>Pooling</i> 1  | $2 \times 2 - 2$ | 32           | $4 \times 4 \times 32$ |
| 4 | Convolução 2      | $2 \times 2 - 1$ | 32           | $4 \times 4 \times 32$ |
| 5 | Camada densa      | 512              | 1            | -                      |
| 6 | Camada densa      | 32               | 1            | -                      |
| 7 | Camada de saída   | 4                | 1            | -                      |

Fonte: Adaptado de [Chao et al. \(2020b\)](#).

Tabela 12 – Hiperparâmetros da rede convolucional de [Chao et al. \(2020b\)](#)

| Hiperparâmetros                | Informações           |
|--------------------------------|-----------------------|
| Funções de ativação            | ReLU e <i>Softmax</i> |
| Função de custo                | <i>Cross-Entropy</i>  |
| <i>Batch normalization</i>     | Sim                   |
| Quantidade de dados por classe | 960                   |
| Tamanho da base de dados       | 3.840                 |
| Treinamento-Teste              | 90-10                 |

Fonte: Adaptado de [Chao et al. \(2020b\)](#).

Para avaliar o desempenho da rede neural, foram utilizados os cálculos de acurácia, matriz de confusão e visualização dos dados por meio do modelo *t-Distributed Stochastic Neighbor Embedding* (t-SNE). Além disso, foram feitas comparações entre o uso de imagens em escala RGB e escala de cinza, análise de sensibilidade em relação ao número de *pixels* da imagem de entrada.

#### 10. *Cavitation Intensity Recognition for High-Speed Axial Piston Pumps Using CNN 1-D with Multi-Channel Inputs of Vibration Signals*

No artigo de [Chao et al. \(2020a\)](#) se propôs um modelo de aprendizado de máquina baseado em uma rede neural convolucional 1-D para classificar a intensidade de

cavitação em bombas de pistão axial. Os dados são registrados por meio de acelerômetros triaxiais (direções x, y e z) e rotulados em quatro classes, de acordo com a perda volumétrica adimensional ( $\nabla\eta$ ). Já estas classes são escoamento sem cavitação ( $\nabla\eta = 1\%$ ), nível baixo de cavitação ( $\nabla\eta = 2\%$ ), nível médio de cavitação ( $\nabla\eta = 8\%$ ), nível severo de cavitação ( $\nabla\eta = 76\%$ ).

Os dados são normalizados e separados em um conjunto para a fase de treinamento (80.000 amostras) e outro para a fase de teste (20.000 amostras). Além disso, na atualização dos pesos e *bias* foi utilizado o algoritmo Gradiente Descendente Estocástico (GDE) para a fase de treinamento e um *minibatch* de 2.000 subamostras com 40 pontos de comprimento para cada *batch*.

A arquitetura da estrutura RNC é formada por 4 camadas de convolução, 2 camadas pooling (maxPool), uma camada totalmente conectada e suas funções de ativação são ReLU e *Softmax*, conforme apresentada na Tabela 13.

Tabela 13 – Arquitetura da rede convolucional de Chao et al. (2020a)

| N | Tipo da camada  | kernel / stride | Qndt. kernels | Mapa característica |
|---|-----------------|-----------------|---------------|---------------------|
| 1 | Convolução 1    | 3 - 1           | 32            | 32×38               |
| 2 | Convolução 2    | 3 - 1           | 32            | 32×36               |
| 3 | Pooling 1       | 2 - 2           | 32            | 32×18               |
| 4 | Convolução 3    | 3 - 1           | 32            | 32×16               |
| 5 | Convolução 4    | 3 - 1           | 32            | 32×14               |
| 6 | Pooling 2       | 2 - 2           | 32            | 32×7                |
| 7 | Camada densa 1  | 224             | 1             | -                   |
| 8 | Camada densa 2  | 256             | 1             | -                   |
| 9 | Camada de saída | 4               | 1             | -                   |

Fonte: Adaptado de Chao et al. (2020a).

A RNMP foi composta por uma camada de entrada compatível com o *minibatch* e 4 camadas ocultas com 32 neurônios em cada camada (CHAO et al., 2020a).

Tabela 14 – Resultados de classificação dos modelos RNC e RNMP de Chao et al. (2020a)

| Sinal de entrada   | Acurácia [%] |        |            |        |
|--------------------|--------------|--------|------------|--------|
|                    | SNB (0dB)    |        | SNB (10dB) |        |
|                    | RNC          | RNMP   | RNC        | RNMP   |
| Média das direções | 44%          | 49,97% | 99,47%     | 86,03% |

Fonte: Adaptado de Chao et al. (2020a).

Os modelos RNC e RNMP apresentaram resultados para o reconhecimento da intensidade de cavitação, quando o sinal de vibração registrado possui um ruído branco Gaussiano de baixa intensidade de ruído de 0 dB e alto nível de ruído 10 dB. Na Tabela 14 é apresentada a acurácia dos modelos para as condições que relacionam a intensidade do ruído em frente ao sinal original para os valores de 0 e 10 dB.

No caso do modelo RNC, foi elaborada uma matriz de confusão para identificar os falso positivos nas três direções e um mapa de visualização das características dos dados para as camadas de entrada, característica e saída, por meio t-SNE.

11. *Identification Inlet Pipe Blockage Level in Centrifugal Pump Over a Range of Speed By Deep Learning Algorithm Using multi-Source Data*

O artigo de [Kumar et al. \(2021\)](#) trata do desenvolvimento de um modelo de aprendizado máquina para identificar o bloqueio do escoamento na tubulação e o nível de cavitação, a partir dos dados provenientes de diferentes tipos de sensores, gerando uma base heterogênea. O procedimento experimental é o mesmo realizado por [Tiwari, Bordoloi e Dewangan \(2021\)](#) com a inclusão de dois novos sensores para o registro das informações, a saber: acelerômetros e medidores de corrente elétrica.

O modelo utilizado foi uma RNMP com duas camadas ocultas e uma camada de entrada com parâmetros extraídos dos sinais de pressão, vibração e corrente elétrica. Esses parâmetros correspondem às métricas das estatísticas descritivas do sinal, com a sua média, desvio padrão, curtose, assimetria, variância e valor eficaz. A arquitetura da rede está apresentada na Tabela 15, enquanto que os hiperparâmetros utilizados estão na Tabela 16.

Tabela 15 – Arquitetura da RNMP de [Kumar et al. \(2021\)](#).

| Camada | Tipo                      | Número de nós | Função de ativação |
|--------|---------------------------|---------------|--------------------|
| 1      | Camada de entrada         | 3             | -                  |
| 2      | Primeira camada escondida | 100           | ReLU               |
| 3      | Segunda camada escondida  | 100           | ReLU               |
| 4      | Camada de saída           | 6             | Sigmoide           |

Fonte: Adaptado de [Kumar et al. \(2021\)](#).

Tabela 16 – Hiperparâmetros da RNMP de [Kumar et al. \(2021\)](#).

| Hiperparâmetros                | Valores              |
|--------------------------------|----------------------|
| Épocas                         | 20, 40 e 100         |
| Taxa de aprendizagem           | 0,001                |
| <i>Momentum</i>                | 0,9 e 0,99           |
| Função de custo                | <i>Cross-Entropy</i> |
| Otimizador                     | Adam                 |
| Mini <i>batch</i>              | 50                   |
| Quantidade de dados por classe | 150                  |
| Tamanho da base de dados       | 7.200                |
| Treinamento-Teste              | 80-20                |

Fonte: Adaptado de [Kumar et al. \(2021\)](#)

A RNMP foi preparada para detectar a condição de bloqueio com dados dos sensores separados e suas combinações. Portanto, as RNMP possuem parâmetros na

camada de entrada conforme o número de variáveis estatísticas e sensores. Assim foram construídas as matrizes de confusão para avaliação da acurácia e identificação dos falsos positivos. A acurácia média obtida para o modelo RNMP com sensores de pressão (SP), elétrico (SE), acelerômetros (AC) foi de 94% e com todos os parâmetros estatísticos foram (SP+SE+AC) foi de 99% (KUMAR et al., 2021). Percebe-se que ao utilizar mais de uma fonte de dados independentes como condição de entrada para os modelos de classificação, há um aumento no desempenho do modelo.

12. *An Improved Convolutional Neural Network with an Adaptable Learning rate Towards Multi-Signal Fault Diagnosis of Hydraulic Piston*

No artigo de Tang, Zhu e Yuan (2021) foi proposto um modelo de RNC aprimorado com o intuito de diagnosticar defeitos em componentes de uma bomba centrífuga. O aprimoramento foi alcançado por meio da modificação da função de custo, na utilização de uma função de divergência com técnica de regularização e, por fim, imagens acústicas em escala de cinza por meio da utilização da Transformada de Wavelet Analítica (AWT). Os dados são oriundos de um ensaio de fonte acústica por meio de microfones próximos aos mancais de rolamento da bomba centrífuga. Nas Tabelas 17 e 18 são apresentados a arquitetura da RNC.

Tabela 17 – Arquitetura da rede convolucional referente à Tang, Zhu e Yuan (2021)

| N | Tipo da camada  | Kernel | Qndt. kernels | Mapa característica |
|---|-----------------|--------|---------------|---------------------|
| 1 | Entrada         | -      | 3             | 64×64×3             |
| 2 | Convolução 1    | 5      | 8             | 64×64×8             |
| 3 | Pooling 1       | 2      | 8             | 32×32×8             |
| 4 | Convolução 2    | 5      | 17            | 32×32×17            |
| 5 | Pooling 2       | 2      | 17            | 16×16×17            |
| 6 | Camada densa 1  | 394    | 1             | -                   |
| 7 | Camada densa 2  | 146    | 1             | -                   |
| 8 | Camada de saída | 5      | 1             | -                   |

Fonte: Adaptado de Tang, Zhu e Yuan (2021)

Tabela 18 – Hiperparâmetros da rede convolucional referente à Tang, Zhu e Yuan (2021).

| Hiperparâmetros                | Informações           |
|--------------------------------|-----------------------|
| Funções de Ativação            | ReLU e <i>Softmax</i> |
| Épocas                         | 67                    |
| Taxa de aprendizagem           | 0,002906              |
| <i>MiniBatch</i>               | 45                    |
| Quantidade de dados por classe | 1.200                 |
| Tamanho da base de dados       | 6.000                 |
| Treinamento-Teste              | 30-70                 |

Fonte: Adaptado de Tang, Zhu e Yuan (2021)

Os resultados experimentais alcançados durante o diagnóstico de defeitos da bomba centrífuga mostram que a RNC aprimorada proposta apresenta uma melhora significativa na precisão de identificação de cerca de 3,2% em relação ao RNC tradicional que foi de 93%.

13. *A Novel Adaptive Convolutional Neural Network for Fault Diagnosis of Hydraulic Piston Pump with Acoustic Images*

No artigo de [Tang, Zhu e Yuan \(2022\)](#) discutiu-se a modelagem de uma rede neural convolucional customizada aos dados registrados por sensores acústicos, por meio da técnica de Otimização de Bayes (OB). Na construção desse modelo de aprendizagem de máquina foi utilizado a Transformada de Wavelet sobre os sinais temporais para a criação dos diagramas com tempo-frequência e uma técnica de OB para o cálculo ótimo dos hiperparâmetros que minimiza a função custo.

Os dados dessa base são categorizados em condição normal e 5 condições de falha (*swash plate wear*, *loose slipper failure*, *slipper wear*, *central spring wear*). Cada condição operativa é composta por 240 imagens que são divididas em conjuntos para as fases de treinamento e teste. As arquiteturas das RN e RNMP são apresentadas na Tabela 19 enquanto que os hiperparâmetros com os respectivos valores são apresentados na Tabela 20.

Tabela 19 – Arquitetura da rede convolucional de [Tang, Zhu e Yuan \(2022\)](#)

| N | Tipo da camada  | Filtro | Qndt. Filtro | Mapa Característica      |
|---|-----------------|--------|--------------|--------------------------|
| 1 | Entrada         | -      | 3            | $64 \times 64 \times 3$  |
| 2 | Convolução 1    | 5      | 8            | $64 \times 64 \times 8$  |
| 3 | Pooling 1       | 2      | 8            | $32 \times 32 \times 8$  |
| 4 | Convolução 2    | 5      | 17           | $32 \times 32 \times 17$ |
| 5 | Pooling 2       | 2      | 17           | $16 \times 16 \times 17$ |
| 6 | Camada Densa 1  | 394    | 1            | -                        |
| 7 | Camada Densa 2  | 146    | 1            | -                        |
| 8 | Camada de saída | 5      | 1            | -                        |

Fonte: Adaptado de [Tang, Zhu e Yuan \(2022\)](#)

Ambas as redes neurais convolucionais são comparadas a uma estrutura com parametrização definida e denominada LetNet-5, sendo formada por uma camada de entrada de tamanho de  $32 \times 32$ , por 3 camadas de convolução, 2 camadas *polling*, 2 camadas totalmente conectadas e, por fim, uma camada de saída. A arquitetura dessa RNC é pré-definida por um número de neurônios e parâmetros conhecidos ([LECUN et al., 1998](#)).

Para a avaliação de desempenho das redes neurais utilizou-se como métricas a matriz de confusão, acurácia e a técnica de visualização dos dados t-SNE. Dentro desse contexto, são apresentados na Tabela 21 os resultados da acurácia média para uma

Tabela 20 – Hiperparâmetros da rede convolucional de Tang, Zhu e Yuan (2022)

| Hiperparâmetros                | Informações           |
|--------------------------------|-----------------------|
| Funções de Ativação            | ReLU e <i>Softmax</i> |
| Épocas                         | 67                    |
| Taxa de aprendizagem           | 0,002906              |
| <i>MiniBatch</i>               | 45                    |
| Quantidade de dados por classe | 1.200                 |
| Tamanho da base de dados       | 6.000                 |
| Treinamento-Teste              | 30-70                 |

Fonte: Adaptado de Tang, Zhu e Yuan (2022)

condição de 10 repetições. Percebe-se que os modelos de 10dB apresentaram resultados superior à 99% de acurácia.

Tabela 21 – Resultados de identificação de cavitação

| Sinal de entrada                | Acurácia [%] |        |           |        |            |        |
|---------------------------------|--------------|--------|-----------|--------|------------|--------|
|                                 | SNB (0dB)    |        | SNB (5dB) |        | SNB (10dB) |        |
|                                 | RNC          | RNMP   | RNC       | RNMP   | RNC        | RNMP   |
| direção <i>x</i>                | 51%          | 50,90% | 76,02%    | 78,40% | 99,20%     | 92,70% |
| direção <i>y</i>                | 50%          | 55,50% | 76,07%    | 79,20% | 99,80%     | 93,60% |
| direção <i>z</i>                | 32%          | 43,50% | 38,05%    | 57,10% | 99,40%     | 71,80% |
| direções <i>x, y</i> e <i>z</i> | 53,45%       | -      | 91,42%    | -      | 100%       | -      |

Fonte: Adaptado de Tang, Zhu e Yuan (2022)

## 2.3 Dados Relevantes à Pesquisa

Esta seção apresenta uma síntese do entendimento do problema de classificação de cavitação em bombas centrífugas, utilizando a técnica de EA para capturar os dados e os modelos de aprendizado de máquina para a identificação de padrões, conforme proporcionado por esta pesquisa bibliográfica. O objetivo foi o de incorporar as conclusões e fatos mais relevantes de outras pesquisa no desenvolvimento deste modelo.

Nesta síntese foram considerados primeiramente a viabilidade em aplicar as técnicas utilizadas nos trabalhos ao problema motivador, independente, do nível de aderência. Outro fato abordado é o grau de detalhamento do conteúdo, pois percebeu-se que trabalhos com grau de aderência menor apresentaram métodos e técnicas promissoras para o desenvolvimento da solução. Também foram abordados os modelos de aprendizado de máquina, métodos para melhorar o desempenho do modelo como balanceamento das bases de dados, técnicas de extração de características e otimização de hiperparâmetros.

- **Modelos de aprendizado de máquina:** Os modelos que apareceram com mais frequência foram MVS, RNMP e RNC. As RNC apareceram em uma única di-

mensão, duas dimensões em escala de cinza e RGB, otimizadas ao problema e as parametrizadas como por exemplo a *LetNet5*.

Considerando todos os artigos, observou-se que as redes RNMP e RNC trataram pelo menos o problema de cavitação ou o sensoriamento acústico, com percentuais de acerto acima de 90% sob condições ideais. Nesses trabalhos apareceram outros métodos como *Deep Belief Network* (DBN), *Distributed Stochastic Neighbor* (t-SNE) e rede neural multicamada de perceptrons e algoritmo de genético (BP-AG) e um *Proximal Suport Machine* (PSVM). Entre esses métodos o t-SNE, mostrou-se uma ótima opção para a visualização dos resultados nas fases de convolução.

- **Técnicas de extração de características:** Para cada tipo de modelo de aprendizado de máquina, foram adotadas abordagens específicas de extração de características. No caso das RNMP, utilizou-se uma decomposição do sinal empregando métricas estatísticas descritivas, como média, desvio padrão, curtose e assimetria, além de parâmetros operativos do equipamento, como pressão, vazão e rotação. Foi empregado também o cálculo da decomposição de energia do sinal, combinado com uma Análise de Componentes Principais (ACP).

Em contraste, devido às RNC realizarem a própria extração de características dos dados, os artigos adotaram etapas de transformação nos sinais temporais para destacar padrões presentes nas séries. Para tal, foram empregadas transformadas Wavelet, com o objetivo de criar um espectro de tempo-frequência, juntamente com o uso de imagens em escalas de tons de cinza e RGB. Ambas as transformações buscaram criar uma estrutura de dados bidimensional semelhante a uma imagem, para facilitar o processo de aprendizado pela rede.

No caso das RNC-1D foram utilizados os dados brutos, sem a necessidade de realizar qualquer transformação.

- **Técnicas de otimização de hiperparâmetros e balanceamento das base de dados:** A única técnica de otimização de hiperparâmetros que apareceu nos artigos foi a otimização Bayesiana, que precisa de uma faixa de variação dos hiperparâmetros para determinar o ótimo local.

Com relação ao balanceamento das bases de dados, observou-se o algoritmo SMOTE - *Synthetic Minority Over-sampling Technique* - que funciona sintetizando exemplos adicionais da classe minoritária, criando dados sintéticos por meio de interpolação entre exemplos existentes. Logo essa técnica é promissora na aplicação das RNMP.

Com relação ao modelo RNC, pode-se utilizar as redes adversárias (*Generative Adversarial Networks* - GAN), que destacam-se pela sua capacidade de gerar dados sintéticos que se assemelham a dados reais, como geração de imagens de condições com cavitação (KIRSCHNER; RIEDELBAUCH et al., 2023).

- **Análise dos resultados:** Os métodos de análise de resultados que adotados foram acurácia, precisão, *F1-score*, matriz de confusão e curva ROC.

Com o objetivo de direcionar a fundamentação teórica, definiu-se com base na revisão bibliográfica, aprofundar estudos em rede neural convolucional, rede neural multicamadas de perceptrons, transformadas de Wavelet e de tempo curto e técnicas de otimização de Bayes, levando à elaboração da Tabela 22 contendo os limites dos hiperparâmetros.

Tabela 22 – Hiperparâmetros dos modelos de aprendizado de máquina dos artigos de referências.

| RNC                         | BAIXO   |               |            | MÉDIO         |           |           | ALTO      |                |  | RANGE |
|-----------------------------|---------|---------------|------------|---------------|-----------|-----------|-----------|----------------|--|-------|
|                             | [1]     | [3]           | [4]        | [9]           | [10]      | [12]      | [13]      |                |  |       |
| Camada de entrada           | 2048×1  | 160×160       | 74×1200    | 8×8           | -         | 64×64     | 64×64     | 8 à 2.048      |  |       |
| N de camadas de convolução  | 2       | 4             | 3          | 2             | 4         | 2         | 2         | 2 à 4          |  |       |
| Tamanhos dos Filtros        | 99 e 96 | 10 e 3        | 3          | 3             | 3         | 5         | 5         | 3 à 10         |  |       |
| N de <i>kernels</i>         | 8 e 16  | 8, 16 e 32    | 8, 16 e 32 | 32            | 32        | 8 e 17    | 8 e 17    | 8 à 32         |  |       |
| Tamanho do <i>pooling</i>   | 2 e 40  | -             | 2          | 2             | 2         | 2         | 2         | 2              |  |       |
| N de camadas conectadas     | 2       | 2             | 1          | 2             | 2         | 2         | 2         | 1 à 2          |  |       |
| N de neurônios por camada   | 352     | 100 e 10      | 13         | 512 e 32      | 224 e 256 | 394 e 146 | 394 e 146 | 10 à 512       |  |       |
| Camada de saída             | 6       | 10            | 5          | 4             | 4         | 5         | 5         | 4 à 10         |  |       |
| Época                       | -       | 30            | -          | -             | -         | 67        | 67        | 30 à 67        |  |       |
| Taxa de aprendizagem        | -       | 0,0016        | -          | -             | -         | 0,002906  | 0,002906  | (1,6 à 2,9)E-3 |  |       |
| <i>Batch normalization</i>  | -       | Sim           | -          | Sim           | -         | -         | -         | Sim            |  |       |
| Inicialização               | -       | -             | -          | -             | -         | -         | -         | -              |  |       |
| Regularização               | -       | Dropout       | -          | -             | -         | -         | -         | Dropout        |  |       |
| Função custo                | -       | Cross-Entropy | -          | Cross-Entropy | -         | -         | -         | Cross-Entropy  |  |       |
| Otimizador                  | -       | Adam          | -          | -             | -         | -         | -         | Adam           |  |       |
| <i>Bath size</i>            | -       | 20            | -          | -             | 40        | 45        | 45        | 20 à 45        |  |       |
| Tamanho da Base de Dados    | -       | 2.000         | -          | 3.840         | 2.000     | 6000      | 6000      | 2 à 6 mil      |  |       |
| Relação Treinamento - Teste | -       | 25-75         | -          | 90-10         | 80-20     | 70-30     | 70-30     | 70:30 à 90:10  |  |       |

*Nota:* [10], [12] e [13] Canais RGB. Fonte: Adaptado das referências da Seção 3.

## 3 Fundamentação Teórica

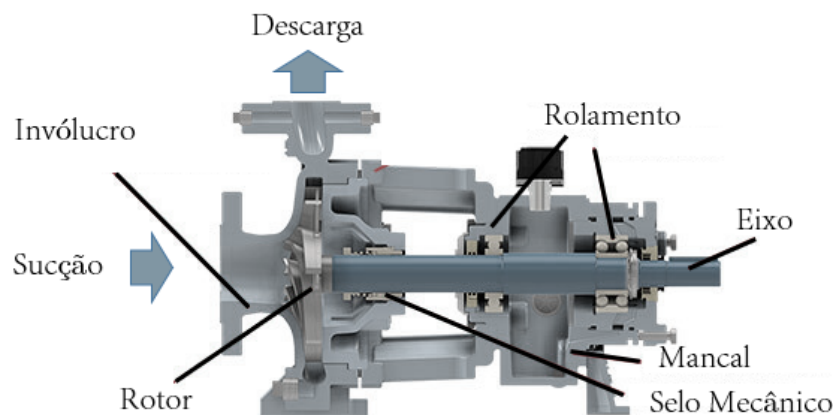
Esta seção está organizada em três frentes principais: a descrição dos equipamentos e fenômenos físicos, incluindo a cavitação e os tipos de falhas em rolamentos de motobombas; as técnicas de transformação de dados, como a Transformada de Fourier e a Contagem de *Hits*; e a modelagem por A, abordando conceitos, terminologias e a utilização de redes neurais na classificação de dados. São apresentadas a arquitetura, formulações matemáticas e pseudo-algoritmos das redes multicamadas de perceptrons e convolucionais. Além disso, são apresentadas as técnicas e métricas na construção dos modelos RNMP e RNC, como o balanceamento das bases de dados por *Synthetic Minority Over-sampling Technique* (SMOTE), a Decomposição do Valor Singular (DVS) e o método de otimização Bayesiano.

### 3.1 Equipamentos e Fenômenos Físicos

#### 3.1.1 Bombas Centrífugas

As bombas centrífugas são dispositivos mecânicos utilizados para ocasionar o escoamento de fluidos, como líquidos ou gases por meio de um conjunto giratório de rotores. O funcionamento básico de uma bomba centrífuga envolve a conversão de energia mecânica, presente no eixo, em energia cinética para o fluido, aumentando sua pressão e impulsionando-o através de um sistema de tubulação. A bomba centrífuga consiste em um rotor dentro de uma carcaça, que é conectada ao sistema de tubulação (INGRAM, 2009). Na Figura 3 é ilustrada uma bomba centrífuga com destaque aos componentes hidromecânicos.

Figura 3 – Principais elementos hidromecânicos de uma bomba centrífuga.



Fonte: Adaptado de Bombas (2023).

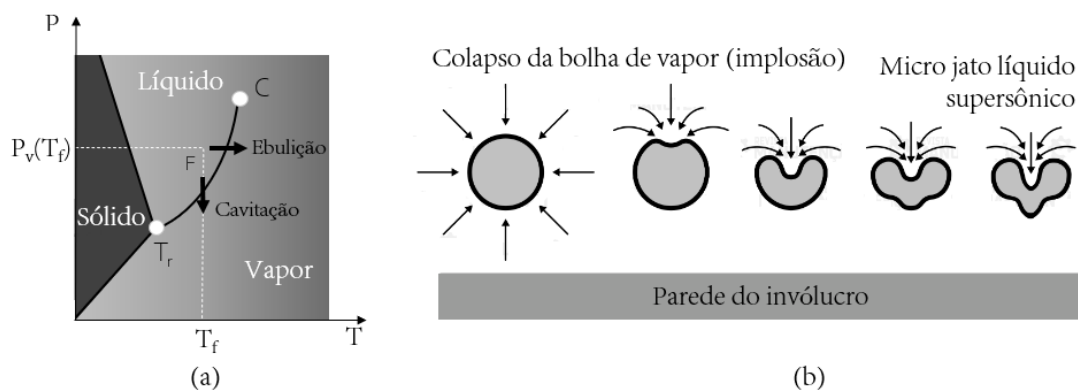
Em resumo, as bombas centrífugas são dispositivos fundamentais em muitos processos industriais, com ampla aplicação em vários segmentos, incluindo as áreas de petróleo, química, alimentos, agricultura, tratamento de água, energia, entre outras.

### 3.1.2 Fenômeno de Cavitação

O fenômeno de cavitação corresponde ao surgimento de bolhas de vapor dentro de um meio líquido inicialmente homogêneo e com temperatura constante, quando a pressão local do fluido de trabalho se torna inferior à pressão de vaporização em um processo isotérmico (JEAN-PIERRE; PIERRE, 2004). Essas bolhas são arrastadas pelo fluxo e quando encontram regiões de alta pressão, ocorrem as implosões, as quais podem causar danos significativos, especialmente quando estão próximas às paredes dos equipamentos ou componentes que confinam o fluido. Portanto, a cavitação contempla a nucleação, o crescimento e colapso das bolhas (ESCALER et al., 2006).

Na Figura 4-(a) é apresentado um diagrama de fase de pressão por temperatura para a água, que remete ao processo de formação de vapor por ebulição, ou seja aumento de temperatura a uma pressão constante e também por um processo de cavitação, queda de pressão a uma temperatura constante. O valor de pressão na qual ocorre essa transformação de fase é denominada como pressão de vapor. Na Figura 4-(b), é apresentada a formação da cavidade de vapor até o colapso (JEAN-PIERRE; PIERRE, 2004).

Figura 4 – Cavitação: (a) diagrama de fase, (b) nucleação, crescimento e colapso.



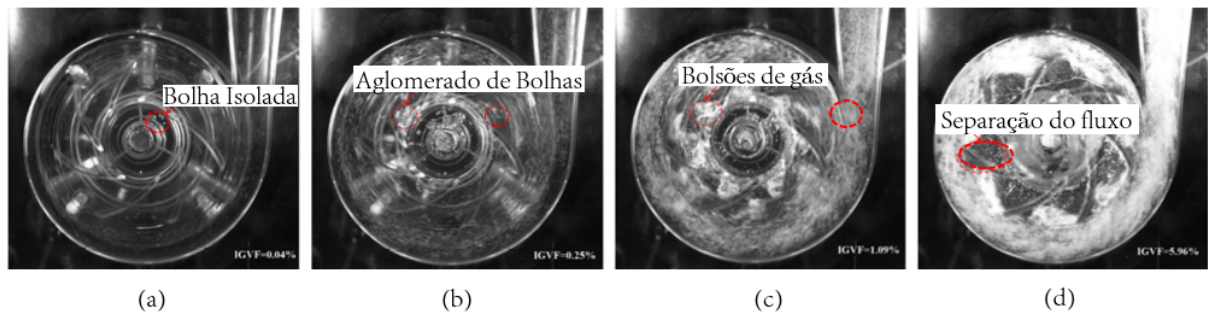
Fonte: Adaptado de Jean-Pierre e Pierre (2004)

Para compreender o fenômeno de cavitação em bombas centrífugas, é essencial entender como esse processo ocorre nos perfis hidrodinâmicos, que correspondem às lâminas das máquinas rotativas. A velocidade de rotação do rotor desempenha um papel significativo, pois influencia diretamente a velocidade do escoamento do fluido e afeta os valores de pressão dentro da bomba. Outro fator crucial para a ocorrência da cavitação é a queda de pressão no sistema de sucção, ou seja, o valor de pressão líquida na entrada da bomba, conhecido como NPSH (*Net Positive Suction Head*). Quando o NPSH disponível é menor

que o NPSH requerido (pressão do líquido bombeado se encontra abaixo da pressão necessária para superar as perdas na entrada do rotor), a pressão do fluido pode cair abaixo de seu ponto de vapor, causando a formação de bolhas de vapor dentro da bombas. Essas bolhas se formam nas regiões de baixa pressão da bomba, geralmente perto do impulsor ou nas pás do rotor. Quando as bolhas de vapor são transportadas para regiões de alta pressão da bomba, elas colapsam violentamente, causando a sua implosão. Esse colapso repentino gera ondas de choque, jatos de alta velocidade e liberação de energia que podem causar danos às partes internas da bomba (MACINTYRE, 1980).

Na Figura 5-(a) é ilustrado o início do fenômeno de cavitação em bomba centrífuga de um rotor aberto, em que as primeiras bolhas são nucleadas no centro do rotor até ocorrer a separação do fluxo, Figura 5-(d).

Figura 5 – Escoamento multifásico em uma bomba centrífuga: (a) bolhas isoladas, (b) bolhas aglomeradas e (c) bolsões de gás, (d) separação do fluxo.



Fonte: Adaptado de He et al. (2021)

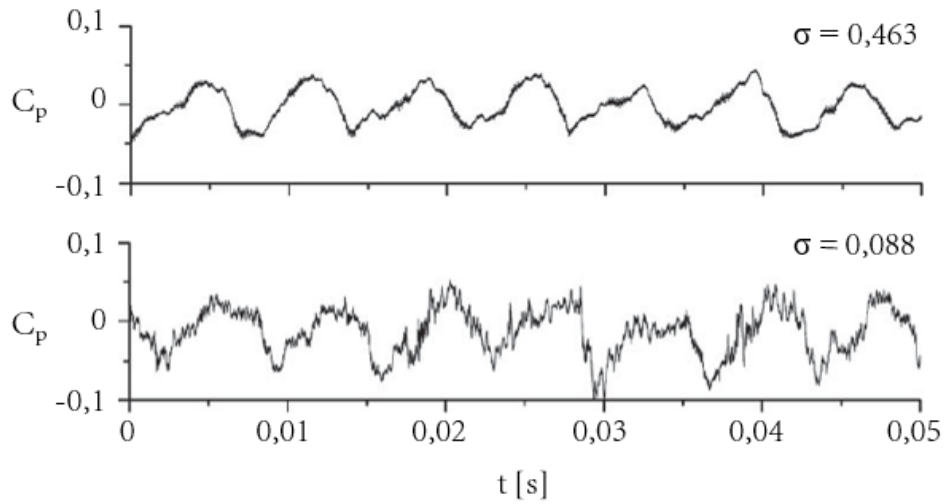
Os efeitos da cavitação incluem desgaste prematuro das partes da bomba, ruídos anormais, vibração excessiva e redução da eficiência da bomba (KARASSIK et al., 2008). Essas variáveis são mensuradas por meio de sensores de deslocamento e pressão, indicando um comportamento anormal do escoamento. Na Equação 3.1 é apresentado o coeficiente de pressão utilizado no monitoramento, em que  $C_p$  é o coeficiente de pressão,  $p$  é a pressão local,  $p_{ref}$  pressão de referência,  $V$  é a velocidade local e  $\rho$  é a densidade do fluido.

$$C_p = \left( \frac{p - p_{ref}}{\frac{1}{2}\rho V^2} \right) \quad (3.1)$$

Na Figura 6 é ilustrado o comportamento temporal de um sensor de pressão para a condição sem cavitação,  $\sigma=0,463$  ( $NPSH_d > NPSH_r$ ) e com cavitação  $\sigma=0,088$  ( $NPSH_d \approx NPSH_r$ ) sob uma determinada taxa de fluxo. Observa-se que para um número de cavitação alto, por exemplo  $\sigma=0.463$ , os sinais de pressão mostram características quase periódicas com frequência correspondente à interação rotor-estator entre o impulsor e a língua da voluta. Quando o número de cavitação diminui para  $\sigma=0,088$  abaixo do ponto crítico, ou seja, queda de 3% no ganho de pressão da bomba, a pressão do escoamento é afetada pela

ocorrência de bolhas de vapor, ocasionando uma modulação no sinal com um novo valor de frequência de maior intensidade e menor amplitude (ZHANG et al., 2018).

Figura 6 – Coeficiente de pressão: (a)  $\sigma=0,463$  e (b)  $\sigma=0,088$ .



Fonte: Adaptado de Zhang et al. (2018)

## 3.2 Análise de Sinais e Transformação dos Dados

Na interseção entre a Ciência dos Dados e a Análise Matemática, a contagem de *hits* e a TF emergem como ferramentas na compreensão de fenômenos físicos complexos. A contagem de *hits* revela padrões e identifica frequências por meio de eventos, enquanto que a TF identifica a composição fundamental de sinais, decompondo-os em suas componentes senoidais com frequências e amplitudes definidas. Em ambas as técnicas cria-se um conjunto de dados tratados que ao combina-las potencializa a eficiência do monitoramento de equipamentos como turbomáquinas (FERNÁNDEZ-OSETE et al., 2024).

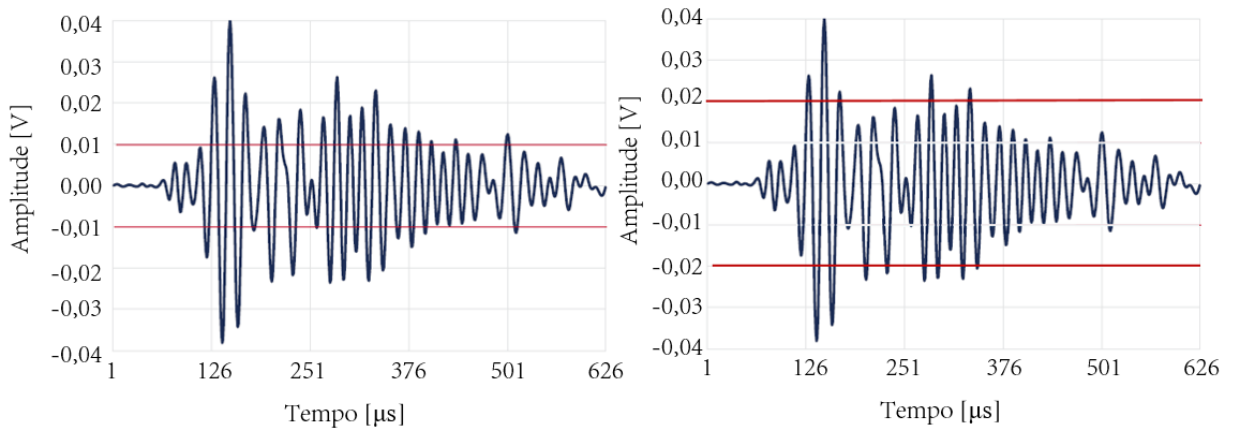
### 3.2.1 Técnica de Contagem de *Hits*

A técnica de emissão acústica é um método não destrutivo amplamente utilizado para identificar e monitorar a presença de anomalias em estruturas e equipamentos. Ela se baseia na detecção de anomalias através da propagação de ondas acústicas, também conhecidas como ondas elásticas transientes. Essas ondas são geradas quando o material é submetido a esforço ou deformação, e sua detecção oportuniza identificar potenciais áreas de falha ou desgaste. Essa análise envolve a extração de valores numéricos do sinal acústico registrado durante o teste. Esses valores são então atribuídos às variáveis ou parâmetros que caracterizam o comportamento do material sob teste. Existem diferentes tipos de parâmetros, alguns baseados no evento de detecção, conhecidos como parâmetros *Hit*, enquanto outros são baseados no tempo (GROSSE et al., 2021).

Os parâmetros *Hit* são utilizados para identificar a localização e a natureza das fontes de emissão acústica, indicando a presença de microfissuras e outros tipos de defeitos. Esses parâmetros são subdivididos em parâmetros de magnitude que são formados pela máxima amplitude, contagem de picos acima do limiar, contagem de picos até amplitude máxima e energia do sinal, parâmetros de tempo que correspondem a duração do *Hit* e o tempo para alcançar o pico máximo. Por fim, existem os parâmetros de frequências, como de iniciação, média e reverberação (FILIPPIN et al., 2017).

Na Figura 7, é ilustrado um sinal de origem acústica cujas linhas em vermelho determinam o limiar de detecção e, portanto, somente os valores acima dessa referência serão processados. Na Figura 7-(a) o valor limiar é limitado entre  $\pm 0,01$  V, sendo processados 17 picos, enquanto que um valor limiar de  $\pm 0,02$  V, o número de picos passou a ser 5 conforme observa-se na Figura 7-(b).

Figura 7 – Limiar de detecção: (a) 0,01 V, (b) 0,02 V.



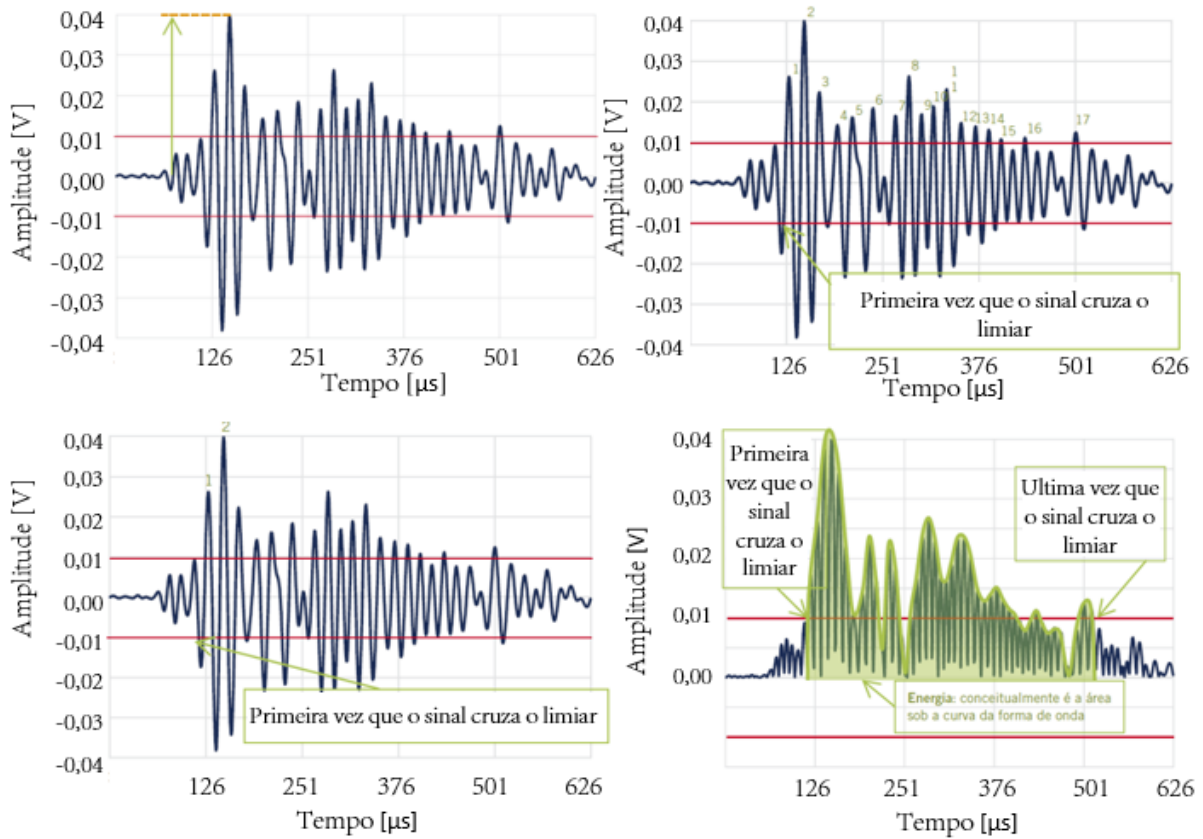
Fonte: Adaptado de FILIPPIN et al. (2017).

- Parâmetros de Magnitude:

O parâmetro amplitude corresponde ao maior valor em módulo de crista ou vale da forma de onda, a partir de um valor de referência, conforme observa-se na Figura 8-(a). O parâmetro contagem está relacionado ao número de vezes que as cristas da forma de onda atingem um valor superior ou igual ao limiar de detecção, conforme ilustrado na Figura 8-(b). A contagem ao pico é um subconjunto do parâmetro de contagem e trata do número de vezes que as cristas da forma de onda atingem um valor superior ou igual ao limiar de detecção antes do parâmetro amplitude, Figura 8-(c).

Por fim os parâmetros de energia e energia absoluta que representam a “força do sinal” sendo obtida por meio do cálculo da área sobre a forma de onda retificada. Esse parâmetro é obtido por meio de uma integral numérica sobre a duração, Figura 8-(d).

Figura 8 – Parâmetros: (a) amplitude; (b) contagem; (c) contagem à pico; (d) energia.

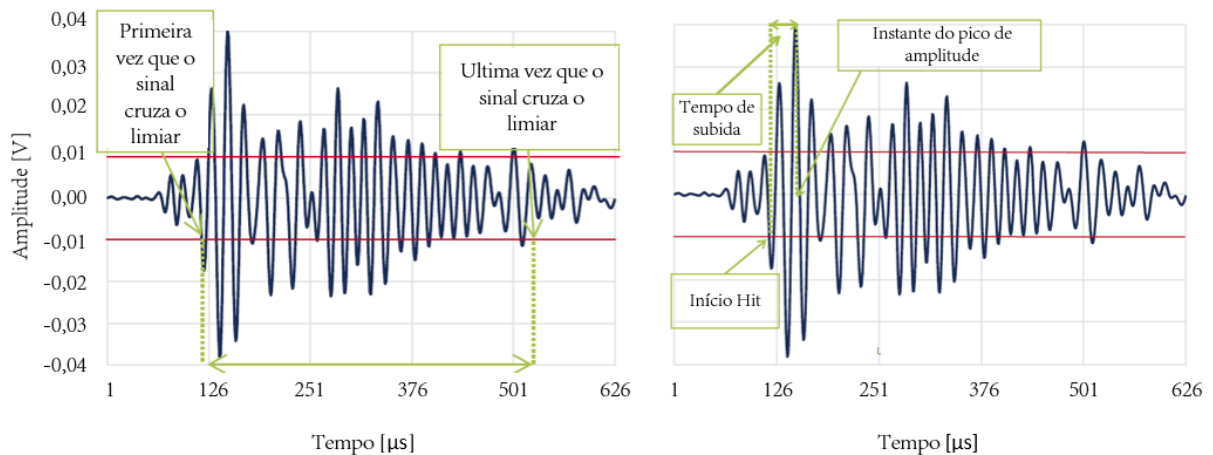


Fonte: Adaptado de FILIPPIN et al. (2017).

- Parâmetros de Tempo

No caso dos parâmetros de EA relacionados com a variável temporal, estão a duração do *Hit* e o tempo para alcançar o pico da forma de onda, conforme a Figura 9.

Figura 9 – Parâmetros de Tempo: (a) duração, (b) tempo de subida.



Fonte: Adaptado de FILIPPIN et al. (2017).

O parâmetro duração quantifica o tempo de duração do *hit*, por meio do registro temporal da primeira e última vez que o sinal supera o limiar de detecção, (Figura 9-(a)). Já o parâmetro tempo de subida é o tempo decorrido entre a primeira passagem da forma de onda pelo limiar de detecção até o instante de ocorrência do pico de maior amplitude (Figura 9-(b)).

- Parâmetros de Frequência

Os parâmetros de frequência não representam a frequência real do sinal de EA, mas fornecem uma estimativa das frequências fundamentais e são obtidos pela combinação linear entre os parâmetros de amplitude e tempo. Em geral, são três novos parâmetros de frequências, a saber: iniciação ( $F_i$ ), média ( $F_m$ ) e reverberação ( $F_r$ ).

Frequência de iniciação: É um parâmetro que indica o comportamento da onda no início do *hit*, ou seja do instante em que há o rompimento do limiar de detecção até a ocorrência do pico de amplitude. Na Equação 3.2 é apresentada a expressão matemática para o cálculo da frequência de iniciação.

$$F_i = \frac{\text{Contagem a pico}}{\text{Tempo de subida}} \quad (3.2)$$

Frequência média: É um parâmetro que trata do comportamento dinâmico do *Hit*, e como o próprio nome expressa, indica as médias das frequências e pode ser calculada pela razão do parâmetro de contagem por duração, conforme se observa na Equação 3.3.

$$F_m = \frac{\text{Contagem}}{\text{Duração}} \quad (3.3)$$

Frequência de reverberação: É um parâmetro que indica o comportamento final da forma de onda, ou seja do pico da amplitude até o último retrocesso pelo limiar de detecção calculado pela Equação 3.4.

$$F_r = \frac{\text{Contagem} - \text{Contagem a pico}}{\text{Duração} - \text{Tempo de subida}} \quad (3.4)$$

### 3.2.2 Transformadas de Fourier

A transformada de Fourier (TF) é uma técnica matemática que descreve como uma função ou sinal no domínio do tempo pode ser decomposta em suas componentes de frequências. Portanto, a TF é a transformação linear dos dados representados no domínio do tempo para o domínio da frequência, por meio de uma função elementar  $\psi(t) = e^{-j\omega t}$ , medindo a similaridade entre o sinal  $s(t)$  e uma série de funções senoidais e harmônicas  $e^{-j\omega t}$  (FOLLAND, 2009).

Na Equação 3.5 é apresentada a função de transformação do sinal que se encontra no domínio do tempo para o domínio da frequência, enquanto que na Equação 3.6 é apresentado a transformação inversa. Essas duas operações são conhecidas como transformada de Fourier e transformada inversa de Fourier (FOLLAND, 2009).

$$S(\omega) = \int_{-\infty}^{\infty} s(t)e^{-j\omega t} dt \quad (3.5)$$

$$S(t) = \int_{-\infty}^{\infty} s(\omega)e^{j\omega t} d\omega \quad (3.6)$$

A transformada discreta de Fourier (TFD), é utilizada quando o sinal é obtido de forma discreta. O sinal na sua forma discreta é representado pela série finita:  $x_r$ . A TF é obtida através da versão discreta da Equação 3.7, em que  $X_k$  é a TFD de  $x_r$ , enquanto que a Equação 3.8 trata da TFD.

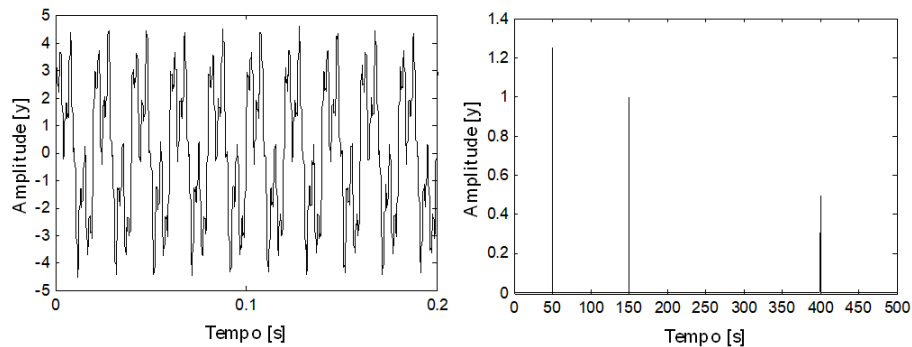
$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, \dots, N-1 \quad (3.7)$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}, \quad n = 0, 1, \dots, N-1 \quad (3.8)$$

A Transformada Rápida de Fourier (TRF) é um método computacional para a implementação da transformada de Discreta Fourier e que reduz o número de operações de cálculo  $O(N^2)$  para  $N \cdot \log(N)$ , ao dividir o problema TFD em dois. A resolução da TRF é dada pela razão entre a frequência de amostragem do sinal ( $FS$ ) e o número de pontos amostrados ( $N$ ) (PRESS, 1992).

Na Figura 10 é ilustrada uma série temporal representada no domínio do tempo e da frequência. Percebe-se que é difícil obter informações com relação a periodicidade do sinal quando expressado no domínio do tempo, porém, no domínio da frequência, notam-se claramente três valores de frequência  $50Hz$ ,  $150Hz$  e  $400Hz$  (PENHA, 1999).

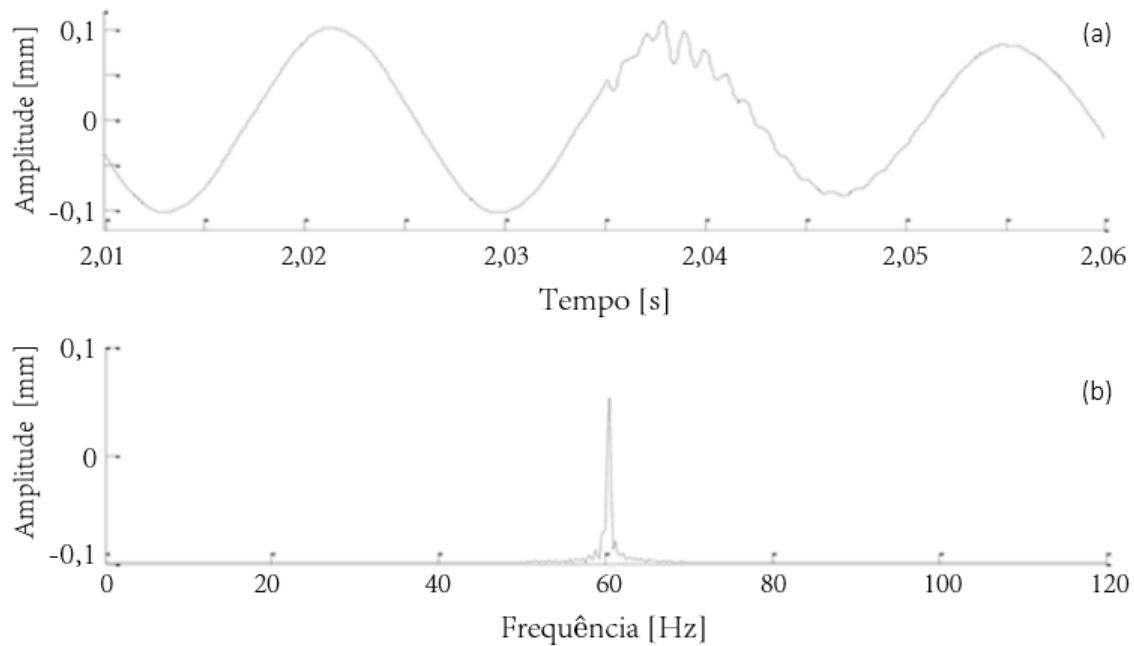
Figura 10 – Sinal: (a) domínio do tempo, (b) domínio da frequência.



Fonte: Adaptado de Penha (1999).

A limitação das TF, FTD e a TRF, é a impossibilidade de saber o instante que ocorre determinadas frequências; a frequência existe durante a janela de análise. Na Figura 11 é ilustrado um sinal com duas frequências, uma de  $60\text{Hz}$  que está presente durante a janela de ensaio e outro de  $2\text{KHz}$  com o início no instante de  $,035\text{s}$  e finaliza à  $2,05\text{s}$ . Ao observar o espectro de frequência da Figura 11-(b) é identificado somente uma frequência.

Figura 11 – Limitações da TF: (a) sinal temporal; (b) espectro de frequência.

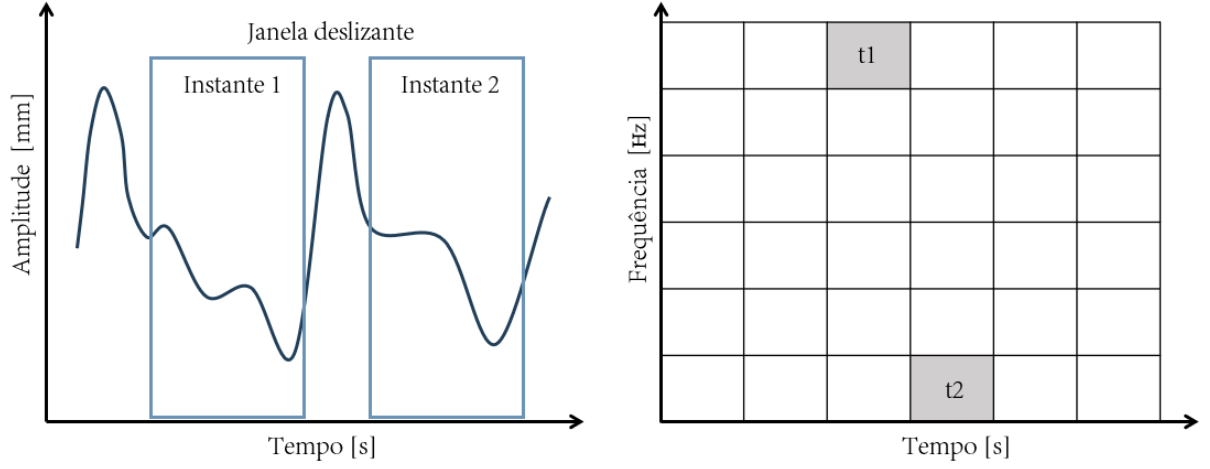


Fonte: Adaptado de [Penha \(1999\)](#).

A TF, tanto no domínio contínuo quanto no domínio discreto, fornece informações somente sobre as frequências do sinal, ou seja, não contempla informações temporais. Isso significa que esse método considera que o sinal possui uma frequência de oscilação constante e invariante no tempo ([FOLLAND, 2009](#)). Por este motivo, a TF e o espectro de potência são ferramentas mais utilizadas na análise de frequência de sinais estacionários. Para contornar essa deficiência da TF, [Gabor \(1946\)](#) propôs que o sinal analisado fosse comparado com uma função elementar localizada no domínio da frequência e do tempo.

Segundo [Penha \(1999\)](#), a Transformada de Fourier de Curto Tempo (TFCT), também conhecida como transformada de Fourier janelada, calcula a transformada de Fourier de um sinal dentro de uma janela centrada no instante de tempo  $t$ , resultando em propriedades estacionárias dentro dessa janela. Esta janela, conhecida como janela deslizante, é transladada ao longo do sinal, permitindo obter um espectro de frequência ao longo do tempo, conforme ilustrado na Figura 12. A TFCT é particularmente útil para a análise de sinais não estacionários, em que as características de frequência mudam ao longo do tempo. Ao aplicar uma janela móvel ao sinal original, a TFCT fornece uma representação tempo-frequência, viabilizando a visualização da evolução do conteúdo espectral do sinal.

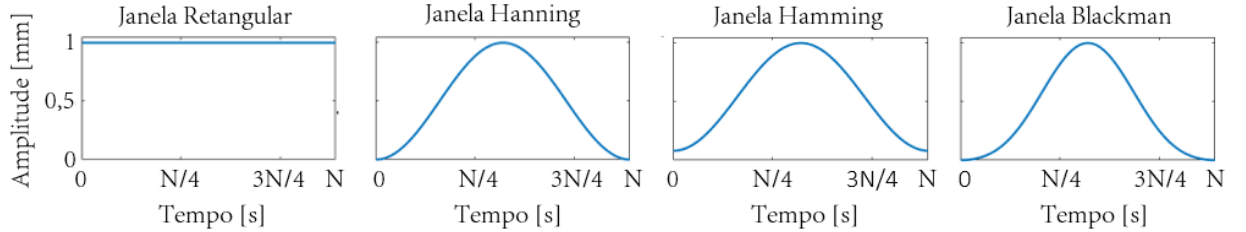
Figura 12 – Princípio da TFCT: (a) Aplicação da janela; (b) TFCT.



Fonte: Adaptado de [Penha \(1999\)](#).

A escolha da janela (tipo e largura) é crucial para a qualidade da análise. Uma janela estreita oferece uma boa resolução temporal, mas pode comprometer a resolução em frequência, enquanto uma janela larga proporciona uma boa resolução em frequência, mas pode não capturar detalhes temporais finos. Entre as janelas comumente utilizadas estão as janelas de Hamming, Hanning e Gaussiana, conforme ilustrado na Figura 13.

Figura 13 – Tipos de janelas.



Fonte: Adaptado de [Mallat \(1999\)](#).

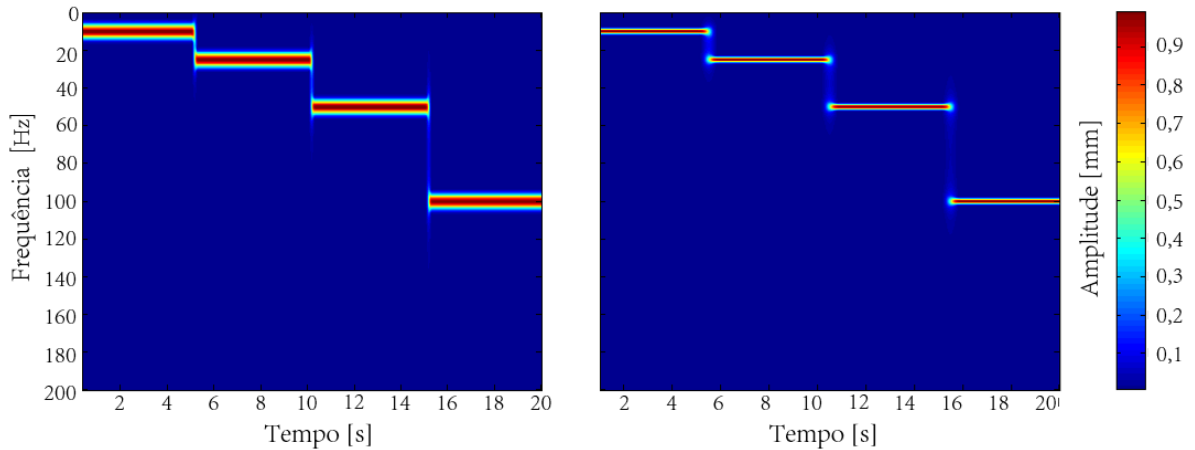
A Equação 3.9, representa a equação matemática para a aplicação da transformada de Fourier considerando a aplicação de uma janela, em que  $s(\gamma)$  é o sinal,  $\gamma(\tau - t)$  é a função de retardo e por fim a  $e^{-j\omega t}$  corresponde a função elementar.

$$S(t, \omega) = \int_{-\infty}^{\infty} s(\tau) \gamma(\tau - t) e^{-j\omega t} d\tau \quad (3.9)$$

Uma limitação da TFCT para a análise de sinais não-estacionários é o compromisso em manter resoluções de tempo e frequências adequadas. Essas duas variáveis são controladas pela largura da janela, que geralmente é definida por uma frequência fundamental. Na Figura 14 é ilustrada essa relação entre  $\Delta t$  e  $\Delta w$ , para três larguras de janela. Percebe-se que o tamanho de janela da Figura 14-(a), conseguiu definir o instante de

tempo que houve a transição entre a mudança de frequência do sinal, enquanto que para uma janela da Figura 14-(b) evidenciou melhor os valores de frequência, porém penalizando os valores no instante de transição uma vez que não existem duas frequências no mesmo instante (PENHA, 1999).

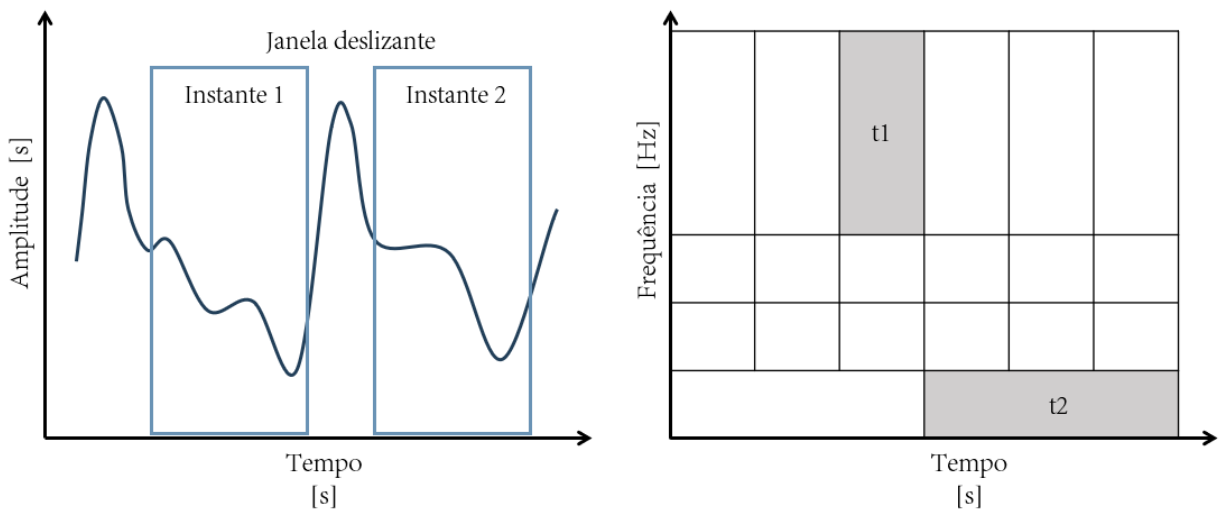
Figura 14 – Aplicação da TFCT: (a) janela de 0,512s (b) janela de 0,064s.



Fonte: Adaptado de Penha (1999).

Para evitar esse tipo de problema, pode-se utilizar a Transformada de Wavelet, que possui duas variáveis de resolução independentes (MALLAT, 1999). A Transformada de Wavelet (TW) decompõe um sinal em componentes de diferentes escalas e localizações no tempo. Isso é feito por meio do uso de funções base conhecidas como Wavelets, que são funções oscilatórias localizadas tanto no domínio do tempo quanto no domínio da frequência, conforme observa-se na Figura 15.

Figura 15 – Princípio da TW: (a) Aplicação da janela; (b) TW.



Fonte: Adaptado de Penha (1999).

### 3.3 Aprendizados de Máquina e Profundo

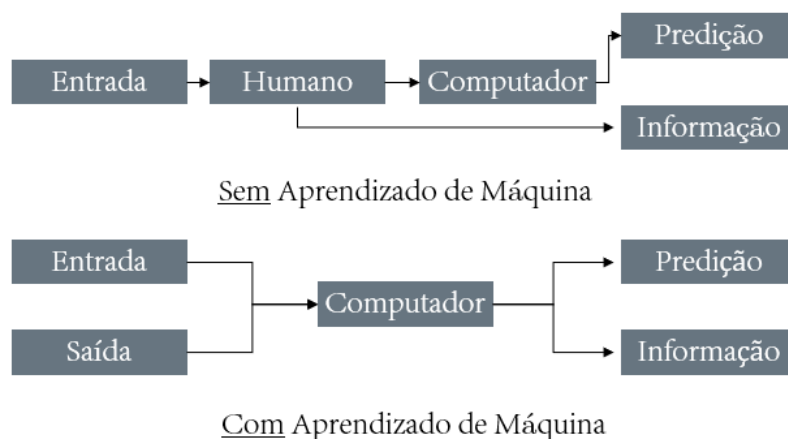
A Inteligência Artificial (IA) refere-se a um campo da tecnologia que engloba sistemas ou estruturas não naturais, como máquinas ou agentes, capazes de exibir formas de inteligência comparáveis às humanas. Essa capacidade envolve a realização de tarefas que normalmente exigiriam inteligência humana, como raciocínio, aprendizado, resolução de problemas e tomada de decisões (RUSSELL, 2010).

Entre essas abordagens estão os modelos computacionais utilizados para realizar tarefas baseada em conhecimento e em dados. O sucesso da abordagem baseada em conhecimento está na forma como as regras são criadas, pois devem ser simples ao ponto de cobrir o maior número de condições e cenários realistas e ao mesmo tempo evitar contradições em situações específicas. Enquanto que a abordagem baseada em dados os próprios algoritmos são capazes de criar automaticamente suas regras por meio de suas experiências a partir de determinado conjunto de amostras (JIANG, 2021).

Segundo Jiang (2021), essa última abordagem é conhecida por AM e envolve o desenvolvimento de algoritmos capazes de realizar ajustes em seus parâmetros com base em exemplos fornecidos, proporcionando a capacidade de fazer previsões, tomar decisões ou realizar ações sem serem explicitamente programadas.

O modelo de AM foca no desenvolvimento de algoritmos e modelos capazes de identificar padrões, realizar previsões e tomar decisões sem serem explicitamente programados para cada tarefa. Utiliza métodos probabilísticos e matemáticos para ajustar e otimizar o desempenho dos modelos com base nos dados disponíveis, buscando generalizar o aprendizado para novos conjuntos de dados. Na Figura 16, é ilustrado um paradigma de programação clássica, cujo programador com suas observações, gera um programa de cálculo da predição, enquanto que o algoritmo de AM corresponde a um programa criado a partir de dados para realizar uma predição (HEATON, 2018).

Figura 16 – Algoritmo com paradigma clássico e aprendizado de máquina.

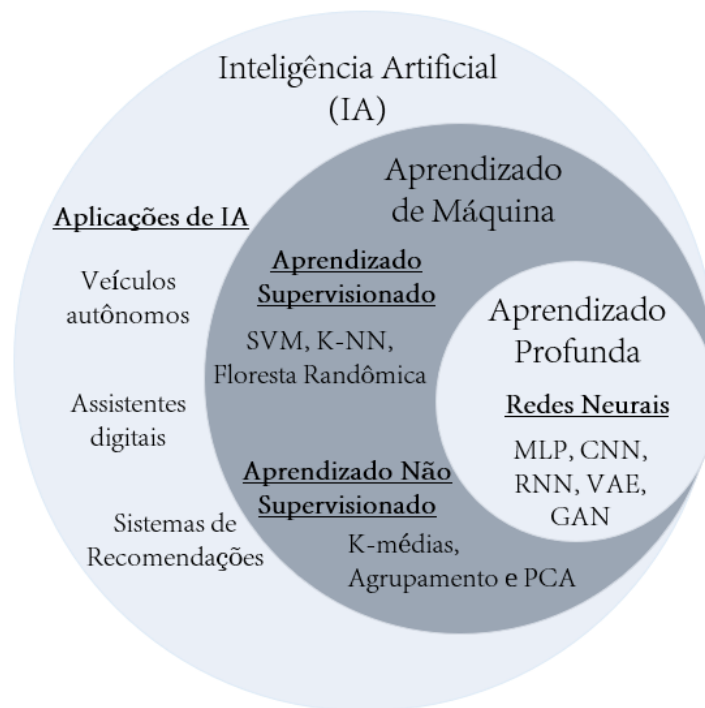


Fonte: Adaptado de Sebastian (2019).

Segundo [Goodfellow, Bengio e Courville \(2016\)](#) o Aprendizado Profundo (AP) é uma área específica de AM que se concentra em treinar modelos computacionais utilizando as redes neurais artificiais para aprender e realizar tarefas complexas. Nesse modelo, o processo de extração de característica é realizado pelo próprio algoritmo e deve haver um volume considerável de amostras para conseguir trabalhar com textos, imagens e sons ([MICROSOFT, 2023](#)).

Essa relação hierárquica destaca como a IA fornece o contexto mais amplo para abranger o ML, que por sua vez inclui a AP como uma de suas abordagens mais avançadas. Na Figura 17, é apresentado um diagrama organizacional para a relação hierárquica entre IA, AM e AP e os modelos computacionais que cada conceito engloba. No caso da IA, são exemplos de aplicações os veículos autônomos, assistentes digitais e sistemas de recomendações; no caso do AM estão os modelos MVS, floresta randômica, agrupamentos, PCA e enquanto que o AP são modelos baseados em redes neurais como RNMP, RNC, RNN e GAN ([HEATON, 2018](#)).

Figura 17 – Diagrama organizacional de IA, AM e AP.



Fonte: Adaptado de [Sebastian \(2019\)](#).

### 3.3.1 Conceitos Fundamentais

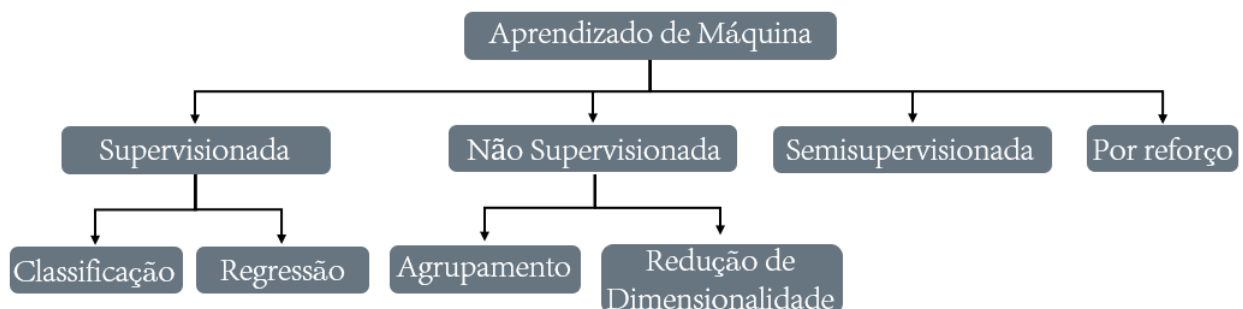
Os modelos de aprendizados utilizam uma abordagem em que eles próprios são capazes de criar suas regras por meio de sua experiência com um conjunto de dados, com o intuito de executar uma determinada tarefa. Essa abordagem é uma subárea da Ciência

da Computação conhecida por A e possui como etapa a coleta de dados, geração de características e modelo de treinamento. As definições foram baseadas em (JIANG, 2021).

- **Coleta dos Dados:** Essa etapa corresponde a coleta dos dados sob as condições da aplicação em que o modelo computacional será implementado. Esses dados devem ser rotulados manualmente para que a máquina aprenda as regras corretamente. Essa etapa é fundamental no desempenho do algoritmo computacional.
- **Geração de Características:** É uma etapa de pré-processamento capaz de extrair informações relevantes da base de dados. Essas informações são transformadas em novos parâmetros, denominadas por características, cuja finalidade é servir como entrada ao modelo computacional representando todos os dados.
- **Modelo de Treinamento:** Corresponde ao modelo matemático construído a partir das características dos dados para abstrair padrões, conforme modelo e estratégia de aprendizado. Os modelos de A são algoritmos que analisam dados e efetuam previsões no âmbito da classificação ou regressão.

Esses modelos são sub-divididos em aprendizado supervisionado, não supervisionado, semi-supervisionado e por reforço. Na Figura 18 são apresentados as quatro classes que possuem tipos de abordagens que dependem do conjunto de dados.

Figura 18 – Diagrama organizacional de tipos de aprendizado.



Fonte: Adaptado de Burkov (2020).

- **Aprendizado supervisionado:** Nesse tipo de aprendizado os dados devem ser rotulados  $(x_1, y_1), \dots, (x_i, y_i), \dots, (x_N, y_N)$ , e organizados em uma matriz *design* de dimensionalidade  $N$ . Essa matriz consiste em pares de dados de entrada (*features*) e saída (*target*). O objetivo dessa abordagem é utilizar o conjunto de dados característicos para produzir um modelo, realizar previsões e compará-los com os rótulos. No caso de problemas de classificação, a saída é formada por um conjunto finito de classes  $(1, 2, 3, \dots, C)$  ou por um número decimal quando se trata de problemas de regressão (BURKOV, 2020).

- **Aprendizado não-supervisionado:** Essa abordagem não possui rótulos no vetor característica, aumentando a dificuldade do modelo por falta de informação. Sua classificação ocorre de acordo com a semelhança dos dados, pois espera-se que dados semelhantes possuam os mesmos rótulos (JIANG, 2021). Nesse tipo de aprendizagem há estratégia de agrupamento para encontrar grupos e objetos similares em conjunto de dados, redução de dimensiolidade quando deseja-se reduzir o tamanho do vetor característica e detecção de *outlayer* para situações em que deseja-se detectar uma anomalia ou novidade sobre o vetor característica (BURKOV, 2020).
- **Aprendizado semi-supervisionado:** Adota-se esse tipo de estratégia quando há dados com e sem rótulos. O objetivo dessa estratégia é similar aos algoritmos supervisionados, porém os dados não rotulados devem fornecer informações adicionais para obter um modelo mais robusto e generalizável (BURKOV, 2020).
- **Aprendizado por reforço:** A aprendizagem por reforço é uma sub-área do A frequentemente utilizada em agentes computacionais que são entidades que interagem com um ambiente sendo capazes de perceber o estado desse ambiente por meio de um vetor de características. O agente pode executar ações em estados não terminais, e cada ação resulta em diferentes recompensas que podem levar o agente a um novo estado no ambiente. Um objetivo comum dos algoritmos de aprendizagem por reforço é aprender uma estratégia de ações que maximize as recompensas durante o treinamento (BURKOV, 2020).

### 3.3.2 Terminologias

- Modelos Simples  $\times$  Complexos:

A complexidade do modelo depende da forma funcional dos dados e do número de parâmetros livres. Os modelos lineares são modelos simples, pois trabalham com padrões de fácil separação e precisam de poucos parâmetros para a sua caracterização. Os modelos complexos, correspondem aos modelos não-lineares e portanto possuem formas funcionais que inviabilizam uma análise teórica (JIANG, 2021).

- Dados Não Estruturados  $\times$  Estruturados:

Os dados não estruturados, ou dados brutos, são uma coleção de entidades em sua forma natural que muitas vezes não podem ser diretamente utilizados para A. Esses dados podem incluir textos, imagens, áudios, vídeos e outros tipos de informações que não seguem um formato predefinido. Por outro lado, os dados estruturados são organizados em estruturas tabulares, como planilhas (*DataFrames*), onde cada linha representa um exemplo (instância) e cada coluna representa um atributo (característica) do dado, conforme ilustrado na Figura 19.

Figura 19 – Dados organizados.

| Atributos             |                       |     |                       | Exemplos              |                       |     |                       |
|-----------------------|-----------------------|-----|-----------------------|-----------------------|-----------------------|-----|-----------------------|
| Atributo 1            | Atributo 2            | ... | Atributo N            | Atributo 1            | Atributo 2            | ... | Atributo N            |
| Exemplo 1             | Exemplo 2             | ... | ExemploN <sub>1</sub> | Exemplo 1             | Exemplo 2             | ... | ExemploN <sub>1</sub> |
| Exemplo 4             | Exemplo 5             | ... | ExemploN <sub>2</sub> | Exemplo 4             | Exemplo 5             | ... | ExemploN <sub>2</sub> |
| ...                   | ...                   | ... | ...                   | ...                   | ...                   | ... | ...                   |
| ExemploN <sub>1</sub> | ExemploN <sub>2</sub> | ... | ExemploN <sub>N</sub> | ExemploN <sub>1</sub> | ExemploN <sub>2</sub> | ... | ExemploN <sub>N</sub> |

Fonte: Adaptado de [Burkov \(2020\)](#).

- Dados de Treinamento  $\times$  Validação  $\times$  Teste:

O conjunto de dados, normalmente, é dividido em grupos de treinamento e teste. Porém, existem situações em que cria-se um terceiro grupo de dados, denominado validação, quando se deseja aplicar a técnica de regularização *early stop*.

O conjunto de treinamento, normalmente, corresponde a maior porção dos dados e é utilizado para construir o modelo. Durante a fase de treinamento utiliza-se os dados de validação para verificar o desempenho do modelo com relação aos dados que não interferem na atualização dos parâmetros e evitar o *overfitting*. Por fim, o conjunto de teste é utilizado para avaliar a escolha do algoritmo de aprendizado e encontrar os melhores valores dos hiperparâmetros ([BURKOV, 2020](#)).

- Aprendizagem Superficial  $\times$  Profunda:

Um algoritmo de aprendizado superficial aprende os parâmetros do modelo a partir das características dos *labels* correspondente aos dados de treinamento. No AP os parâmetros dos modelos são aprendidos a partir da camada de saída em direção às primeiras camadas e ainda possuem mais de três camadas ([BURKOV, 2020](#)).

- *Underfitting*  $\times$  *Overfitting*:

Os modelos de AM são baseados em dados, que por sua vez contém regularidades que devem ser identificadas e aprendidas. Inevitavelmente os dados possuem informações irrelevantes (ruídos), *dados = sinal + ruído*, originados por erros e limitações do processo de aquisição dos dados. O desafio em construir um modelo de AM capaz de distinguir o sinal do ruído e com isso obter uma boa capacidade de generalização ([JIANG, 2021](#)).

A generalização do modelo refere-se à capacidade de realizar previsões precisas e consistentes sobre dados ainda não vistos, evitando assim um superajuste (*overfitting*) ou um subajuste (*underfitting*). Um modelo de A deve ser capaz de extrapolar padrões e relações aprendidas durante a etapa de treinamento para novos conjuntos de

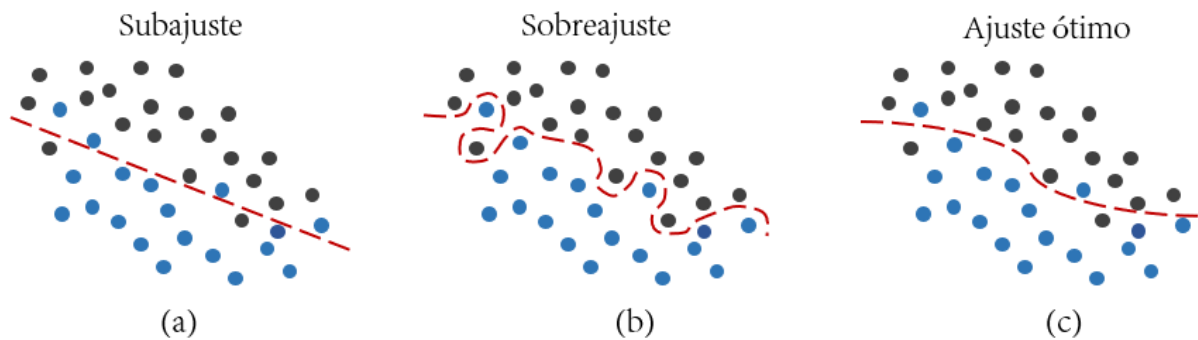
dados, como as etapas de teste, validação e aplicação. Dentro desse contexto, existe a complexidade do modelo pode levá-lo a problemas de *underfitting* ou *overfitting*.

O *underfitting* corresponde a um modelo com poucas variáveis pouco tempo de treinamento, quantidade insuficiente de dados e baixa qualidade dos dados, para capturar todas as regularidades no componente de sinal e por isso o modelo aprendido produzirá resultados ruins mesmo nos dados de treinamento. Para evitar esse tipo de problema deve-se aumentar a complexidade do modelo (JIANG, 2021).

O *overfitting* ocorre quando o modelo busca capturar todas as regularidades do sinal, incluindo o ruído, situação, situação que é altamente flutuante e distorce os resultados de aprendizado, especialmente para modelos complexos. Nesse caso, o modelo apresenta resultados ruins nas fases de validação e teste. Para evitar esse tipo de problema deve-se aumentar a base de dados, reduzir a complexidade do modelo e utilizar técnicas de regularização (JIANG, 2021).

Na Figura 20<sup>1</sup>, é possível observar um conjunto de dados composto por duas classes. Ao aplicar um modelo de classificação linear a esse conjunto, foi constatada uma baixa acurácia, indicando a presença de *underfitting*. Também foi identificado um modelo não-linear de alta complexidade, que alcançou uma taxa de acerto de 100%, sugerindo possibilidade de *overfitting*. Além disso, um modelo não linear com um número reduzido de parâmetros apresenta habilidade em capturar o padrão subjacente dos dados.

Figura 20 – Regularidade do modelo: (a) *underfitting*; (b) *overfitting*; (c) *optimum*.



Fonte: Adaptado de Jiang (2021).

- Viés × Variância:

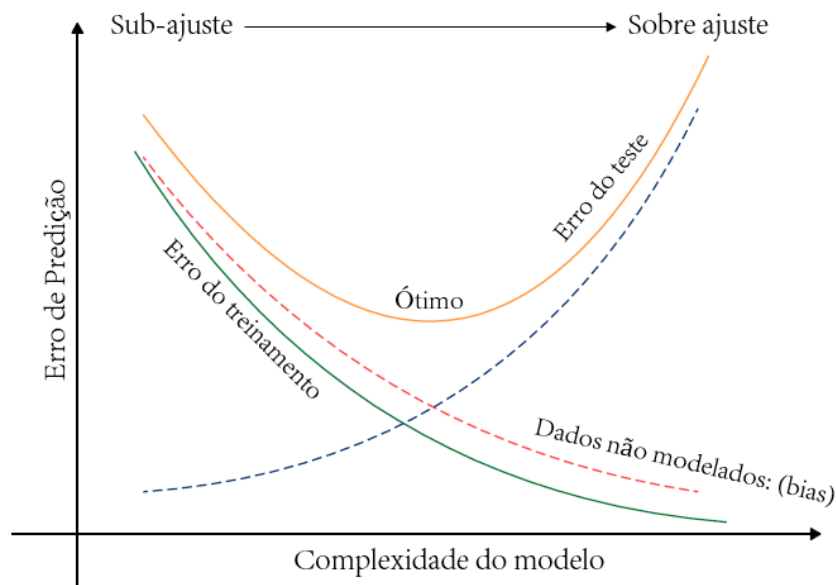
O erro total esperado em um algoritmo de A é uma combinação de dois componentes principais: variância e viés. Este erro pode ser expresso conforme a Equação 3.10,

$$erro = vies^2 + variancia. \quad (3.10)$$

<sup>1</sup> Acc - Acurácia

Segundo [Jiang \(2021\)](#) o erro de viés (*bias*) indica uma incapacidade de capturar toda a complexidade dos dados, ou seja o algoritmo é muito simples para abstrair a estrutura subjacente dos dados. Enquanto que o erro de variância mede a variabilidade ou sensibilidade das previsões dos algoritmos. Um alto erro de variância indica que o algoritmo é muito sensível a ruídos e flutuações aleatórias. Portanto, o objetivo é encontrar um equilíbrio entre os erros de viés e variância, com o intuito de obter uma generalização ótima dos dados de teste, conforme Figura 21.

Figura 21 – Erros relacionados a variância e *bias*.



Fonte: Adaptado de [Brunet \(2010\)](#).

- *Parâmetros × Hiperparâmetros:*

Os hiperparâmetros são parâmetros de entrada nos algoritmos de A que influenciam o desempenho do modelo. Esses parâmetros não podem ser aprendidos com os dados de treinamento. Por outro lado, que os parâmetros correspondem às variáveis do modelo e são diretamente modificadas pelo algoritmo de aprendizado a partir dos dados de treinamento ([BURKOV, 2020](#)).

### 3.3.3 Técnicas para Extração de Características

A extração de características desempenha um papel fundamental no A, possibilitando a representação eficaz e informativa dos dados originais. Esse tipo de técnica visa reduzir a complexidade dos dados, mantendo inclusive a relevância das informações. Esse processo compreende a transformação de dados brutos em um formato estruturado, contendo os padrões e informações relevantes que podem ser facilmente interpretados pelos algoritmos de AM.

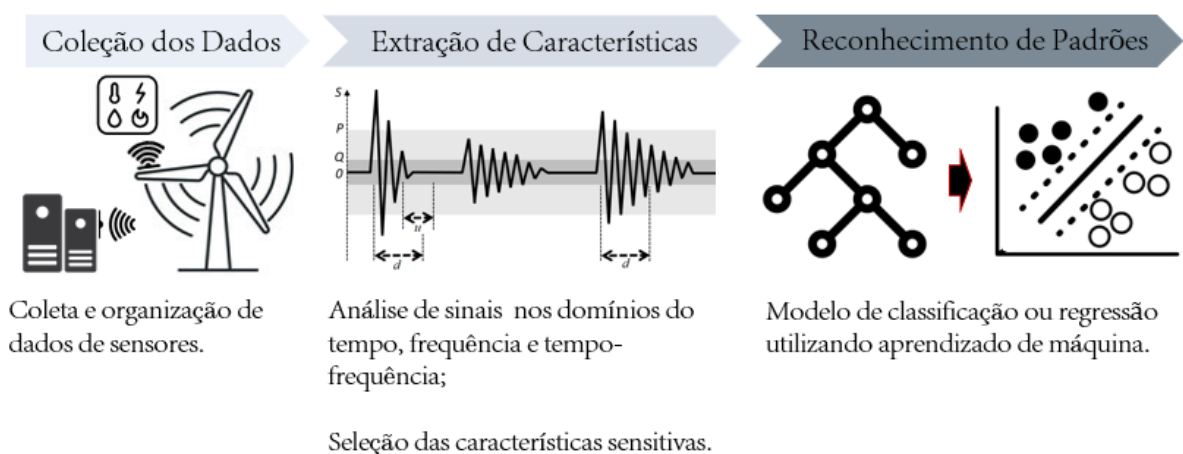
Para cada tipo de dado e problema existem técnicas adequadas para a realizar seleção das características. Uma abordagem comum é a relevância estatística sobre os dados como, por exemplo, a aplicação de estatística descritiva. Outra forma é aplicação de operações matemáticas que capturem melhor as relações intrínsecas com os dados, ou seja técnica de transformação. Nesse conjunto encontra-se Análise de Componentes Principais (ACP), Análise de Discriminante Linear (LDA), Transformada de Fourier (TF) e Transformada de Wavellet (TW) (JIANG, 2021) e (BISHOP; NASRABADI, 2006).

### 3.3.4 Aprendizado de Máquina × Aprendizado Profundo

O AM basicamente tem por objetivo aprender com os dados e realizar tarefas específicas por meio dessa experiência. Para a elaboração desse modelo não há necessidade de uma grande quantidade de dados e, conseqüentemente, não há necessidade de um poder computacional elevado. Deve-se fazer um pré-processamento dos dados para obter os recursos e alimentar o modelo de AM para produzir uma saída numérica capaz de distinguir classes e efetuar previsões de valores, ou seja, realizar classificação ou regressão (SHANTHAMALLU; SPANIAS, 2021).

Na Figura 22 são ilustradas as principais etapas do modelo ML, com início na coleta de dados por meio de sensores previamente instalados na máquina. Na sequência é realizado um pré-processamento sobre a base de dados que inclui a etapa de extração e outra da escolha das características. Por fim, utiliza um modelo de AM para realizar a classificação dos conjuntos de dados (LEI et al., 2020).

Figura 22 – Elaboração de um modelo ML.



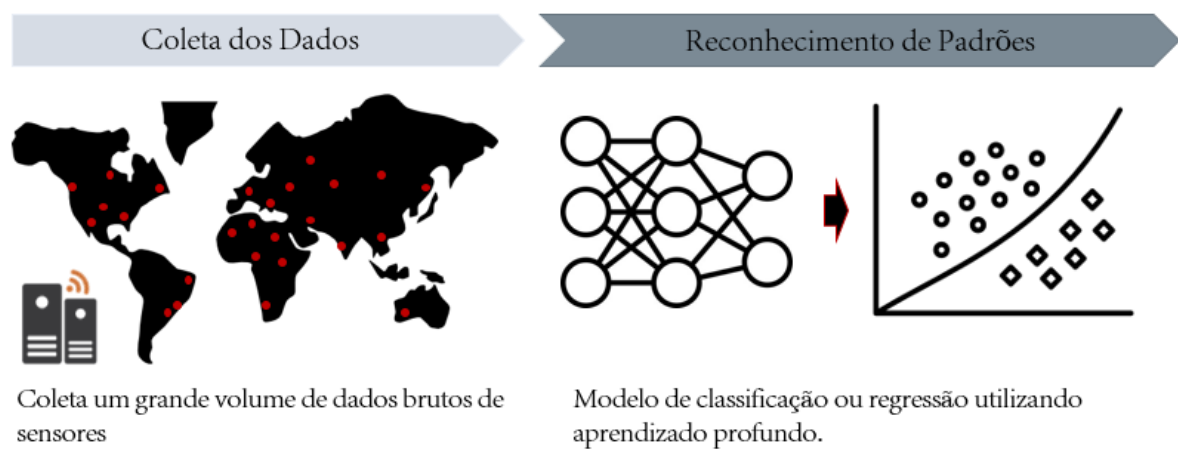
Fonte: Adaptado de Lei et al. (2020).

O Aprendizado Profundo (AP) é uma vertente do AM baseado em redes neurais, cuja arquitetura deve conter várias camadas para representar automaticamente as

características e estabelecer uma relação entre os dados de entrada e a saída desejada (SHANTHAMALLU; SPANIAS, 2021).

Nesse tipo de aprendizagem, deve-se ter uma grande quantidade de dados para fazer o treinamento dos modelos, o que implica em uma forte dependência do hardware de processamento. O processo de extração de características é realizado pela próprio modelo e os resultados de predição podem estar nos formatos de textos, som e imagens (LEI et al., 2020). Na Figura 23 é ilustrado o processo para a criação de modelos utilizando AP e no Quadro 4, são comparadas as principais características das duas abordagens.

Figura 23 – Elaboração de um modelo AP.



Fonte: Adaptado de Lei et al. (2020).

Quadro 4 – Interação entre os módulos dos modelos RNMP e RNC.

| Medidas                    | Modelos AM                                                                  | Modelos AP                                                   |
|----------------------------|-----------------------------------------------------------------------------|--------------------------------------------------------------|
| Número de dados            | Pode usar poucos dados                                                      | Precisa de grande quantidade de dados                        |
| Processo de personalização | Recursos devem ser identificados e criados no pré-processamento             | Aprende e criar os recursos durante a etapa de processamento |
| Abordagem de aprendizagem  | Divide o processo de aprendizado em etapas menores combinando os resultados | Percorre todo o processo de aprendizado                      |
| Tempo de execução          | Tempo de treinamento pequeno, de segundos à horas                           | Tempo de treinamento elevado podendo ser de horas a dias     |
| Saída                      | Corresponde a um valor numérico, regressão ou classificação                 | Pode ter vários formatos, como texto, som, imagem e etc.     |

Fonte: Adaptado de Microsoft (2023).

## 3.4 Redes Neurais

### 3.4.1 Introdução

Segundo Haykin (2001) as redes neurais são modelos computacionais inspirados no comportamento do sistema nervoso com a capacidade de adquirir e manter o conhecimento oriundo dos dados. Sua estrutura corresponde a conexões paralelamente distribuídas e responsáveis por propagar informações, a partir de um estímulo, para obter uma resposta e atuação. Tais modelos computacionais são úteis em aplicações complexas tendo em vista que são adaptáveis, generalistas, capazes de solucionar problemas não-lineares, intolerante a falhas, capacidade de realizar o mapeamento entrada-saída e permite a implementação *Very Large Scale Integration* (VLSI).

Segundo Haykin (2001) as principais características das redes neurais, são:

- **Adaptável:** As redes neurais possuem a capacidade de se adaptar à pequenas modificações operativas oriundas do ambiente, especialmente em condições não estacionárias, modificando os seus pesos em tempo real;
- **Generalista:** A natureza generalista das redes neurais é resultado de sua arquitetura flexível e da capacidade de aprender padrões sob diversos conjuntos de dados;
- **Não linearidade:** Um neurônio pode ser linear ou não linear, enquanto que uma rede é considerada não-linear quando suas conexões possuem ligações não lineares e distribuídas por toda a estrutura neural;
- **Tolerante a falha:** Esse tipo de estrutura precisa de um dano generalizado para que a resposta global seja comprometida, uma vez que o armazenamento da informação é de natureza distribuída. Esse tipo de degradação ocorre frequentemente em estruturas implementadas em nível de hardware;
- **Mapeamento entrada-saída:** Cada exemplo da amostra gera um único sinal de entrada e uma resposta de saída correspondente;
- **Implementação VLSI:** As redes neurais possuem a natureza maciçamente paralela potencializando a utilização de tecnologia de Integração de Grande Escala (VLSI) para capturar comportamentos complexos de forma hierárquica quando envolvem projetos de hardware.

Um modelo de rede neural é formado por neurônios artificiais que realizam o processamento de informação que em conjunto, são capazes de realizar uma representação do conhecimento<sup>2</sup> (HAYKIN, 2001).

<sup>2</sup> Conhecimento refere-se à informação armazenada ou máquina para interpretar, prever, responder o mundo externo.

Segundo [Haykin \(2001\)](#) existem três classes de arquiteturas de redes: as redes alimentadas adiante com uma única camada, as redes alimentadas adiante com múltiplas camadas e as redes recorrentes.

- **Redes com alimentação adiante com uma única camada:** É uma rede neural que possui uma camada de entrada e uma camada de saída, sendo alimentada adiante ou acíclica. Essa rede também é conhecida como rede de camada única.
- **Redes com alimentação adiante com múltiplas camadas:** É uma rede alimentada adiante que possui uma ou mais camadas intermediárias entre as camadas de entrada de saída. Essas camadas são conectadas entre si no sentido adiante, ou seja, os neurônios da primeira camada são conectados aos neurônios da segunda camada, os neurônios da segunda camada são conectados aos neurônios da terceira camada e assim sucessivamente até a última camada (camada de saída). Essas camadas intermediárias têm o papel de extrair estatísticas de ordem mais elevada.
- **Redes recorrentes:** É uma rede neural que possui pelo menos um laço de realimentação, ou seja, as informações provenientes de uma camada são ponderadas com as informações da sua entrada. Tais redes podem ter sua própria saída como entrada, sendo chamadas de conexões de retroalimentação.

Esses tipos de modelos são amplamente utilizados em diversas áreas do conhecimento devido à sua capacidade de se adaptar a diferentes tipos de problemas. Nesse contexto, as soluções possíveis incluem funções de aproximação, que ajudam a estimar valores contínuos; reconhecimento de padrões, que podem identificar e categorizar regularidades em dados; classificação, que atribui categorias a dados com base em suas características; agrupamento de dados, que organiza dados em grupos com características semelhantes; sistemas de predição, que fazem previsões baseadas em dados históricos; e associação por memória, que relaciona padrões e lembra associações previamente aprendidas ([SILVA et al., 2017](#)).

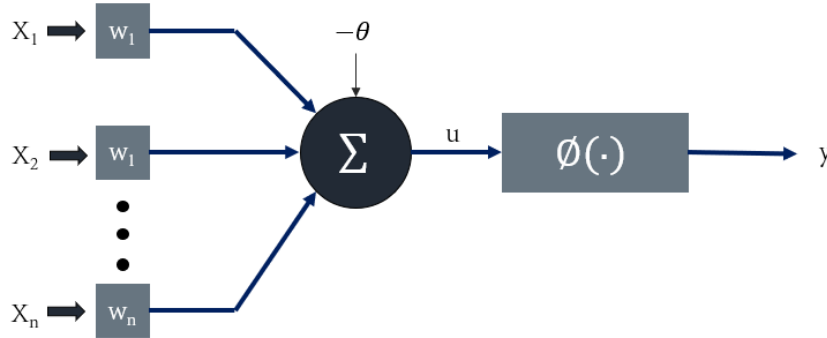
### 3.4.2 Formulação Teórica e Matemática

A estrutura de uma rede neural é composta por camadas de neurônios artificiais conectados entre si. Essas conexões têm pesos associados e ajustados de modo a ter um padrão de entrada a um determinado padrão de saída. Esse mapeamento da entrada/saída é conhecida como fase de treinamento da rede e utiliza do aprendizado supervisionado para realizar a alteração dos pesos sinápticos com o objetivo minimizar a diferença entre resultados obtidos ( $y$ ) e os esperados ( $y^*$ ) ([HAYKIN, 2001](#)).

Na Figura 24, é ilustrado um único neurônio artificial (perceptron) que possui sinais de entrada ( $x_1, x_2, \dots, x_n$ ), pesos correspondentes ( $w_1, w_2, \dots, w_n$ ), um operador

linear ( $\Sigma$ ), um limiar de ativação ( $\theta$ ), um potencial de ativação ( $u$ ), uma função de ativação ( $\sigma$ ) e um sinal de saída ( $y$ ) com os respectivos pesos.

Figura 24 – Estrutura do neurônio artificial.



Fonte: Adaptado de [Silva et al. \(2017\)](#)

Na Equação 3.11 a expressão matemática para esse modelo de neurônio.

$$y = \sigma(w_{11}x_1 + \dots + w_{1n}x_n + \theta) \quad (3.11)$$

O cálculo no neurônio artificial é iniciado pela soma do produto entre o sinal de entrada com os respectivos pesos sinápticos ( $\Sigma w_i x_i$ ) que, ao combinar com o limiar de ativação ( $\theta$ ), resulta no potencial de ativação linear ( $u$ ), ou seja, se ( $u \geq \theta$ ) então o neurônio produzirá uma excitação, caso contrário será inibido. Com isso, esse valor será direcionado a uma função de ativação cujo objetivo é limitar a saída do neurônio e introduzir um comportamento não linear. Na sequência são tratados os elementos básicos do neurônio artificial, de acordo com ([SILVA et al., 2017](#)):

- **Sinal de entrada** ( $x_1, x_2, \dots, x_n$ ): São amostras do ambiente externo representadas por variáveis numéricas;
- **Pesos sinápticos** ( $w_1, w_2, w_3$ ): São utilizados para ponderar a relevância das variáveis do sinal de entrada;
- **Agregador Linear** ( $\Sigma$ ): Corresponde ao produto interno entre o sinal de entrada com os pesos sinápticos, cujo objetivo é influenciar o potencial de ativação;
- **Limiar de ativação** ( $\theta$ ): É a variável utilizada para especificar o limiar de ativação do neurônio de saída, também conhecido como *bias*;
- **Potencial de Ativação** ( $u(.)$ ): Corresponde ao valor de ativação do neurônio. Se ( $u(.) = \Sigma w_i x_i - \theta \geq 0$ ), então o neurônio será ativado, caso contrário será inibido;
- **Função de Ativação** ( $\phi(.)$ ): O objetivo é limitar o valor de saída do neurônio para qualquer valor de entrada. Dependendo da função de ativação utilizada, os modelos

neurais passam a serem não-lineares. Além disso, têm um importante papel no cálculo de atualização dos pesos sinápticos;

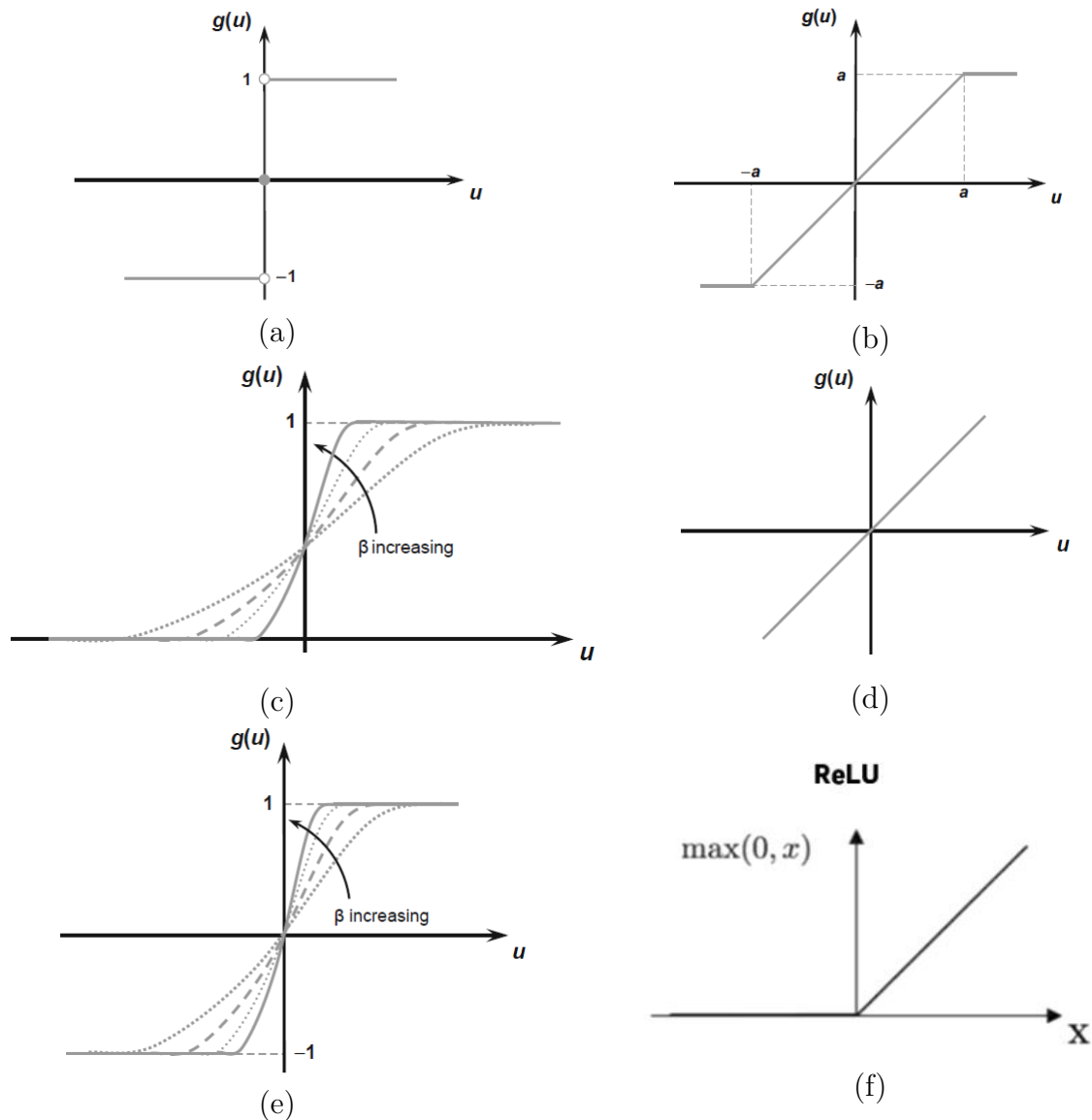
- **Sinal de Saída** ( $y(\cdot)$ ): Consiste no valor final dado pelo neurônio a partir do sinal de entrada. Esse sinal de saída pode ser utilizado para as entradas de novos neurônios da camada subsequente.

As funções de ativações têm por objetivo definir uma saída em termos de campo “induzido” e devem ser diferenciáveis em todo domínio de integração para que a primeira derivada exista e o cálculo de atualização dos pesos seja possível. Deste modo, as funções de ativação são classificadas em parcialmente e totalmente diferenciáveis. No grupo parcialmente diferenciáveis estão as funções que contêm descontinuidades no campo induzido  $u$ , como as funções de ativação limiar, bipolar e rampa simétrica, já no outro grupo estão as funções tangente hiperbólica, logística (SILVA et al., 2017). Nesse segundo grupo ainda enquadram-se as funções gaussiana e linear. A seguir são descritas as funções de ativação e ilustrado o seu comportamento nas regiões de transição.

- **Função limiar:** Também conhecida como função de *Heaviside*, assume o valor 1 quando o campo induzido é não-negativo, caso contrário é 0, (Figura 25-(a)). Essa função também é conhecida comportamento de tudo ou nada (HAYKIN, 2001).
- **Função rampa simétrica:** Nesta função o valor retornado é igual ao valor induzido quando respeitado o intervalo  $[-a, +a]$ . Caso contrário, os valores retornados são  $+a$  e  $-a$ , para valores fora desse intervalo, Figura 25-(b) (SILVA et al., 2017).
- **Função linear:** É definida pelo próprio potencial de ativação, ou seja, o valor retornado é o próprio valor induzido. O gráfico que representa a função é ilustrada na Figura 25-(d) (SILVA et al., 2017).
- **Função logística:** A função logística trabalha sobre o campo de ativação e corresponde a um valor entre 0 e 1, com a opção de alterar a inclinação do ponto de inflexão para intensificar a taxa de crescimento da função por meio do parâmetro  $\beta$ . Para valores de inclinação próximos ao infinito a função logística torna-se uma função limiar, porém, possui derivada em qualquer ponto do campo de ativação. Na Figura 25-(c) é ilustrada a função logística e a influência de  $\beta$  (HAYKIN, 2001).
- **Função tangente hiperbólica:** A função tangente hiperbólica produz uma saída entre -1 à 1, com as mesmas considerações da função logística, tendo um parâmetro que controla a inclinação. É diferenciável no campo de ativação e possui o mesmo comportamento da função bipolar, Figura 25-(e) (HAYKIN, 2001).
- **Função ReLU:** A função de ativação *Rectified Linear Activation* (ReLU) é linear para valores positivos e zero para valores negativos. Isso significa que para valores

de entrada positivos, a função ReLU mantém o valor de  $x$  como saída e para valores negativos de  $x$ , a saída é definida como zero, conforme pode ser observado na Figura 25-(f) (GOODFELLOW; BENGIO; COURVILLE, 2016).

Figura 25 – Funções de ativação do tipo: a) limiar; b) rampa ; c) logística; d) linear; e) tangente hiperbólica; f) ReLU



Fonte: Adaptado de Haykin (2001).

## 3.5 Rede Neural Multicamadas de Perceptrons (RNMP)

### 3.5.1 Introdução

Uma Rede Neural Multicamadas Perceptron (RNMP) é um tipo de modelo de rede neural artificial amplamente utilizado em tarefas de aprendizado de máquina supervisionado. Essa estrutura tem a capacidade de aprender padrões e realizar classificações

ou regressões a partir dos dados. O termo multicamadas perceptron refere-se à uma estrutura com várias camadas de neurônios totalmente conectadas, o que torna a modelagem de relações não lineares entre as variáveis de entrada e saída (SILVA et al., 2017).

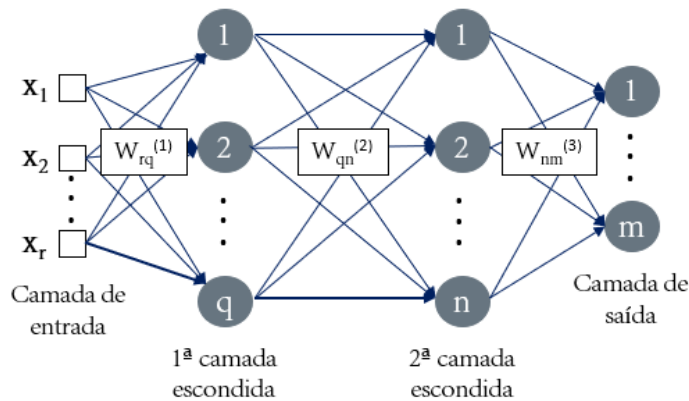
Durante o treinamento, a RNMP ajusta os pesos dessas conexões para minimizar o erro entre as saídas previstas (predição) e os valores reais (rótulos). Isso é feito usando algoritmos de aprendizado, como o cálculo de retropropagação do erro (*backpropagation*), funções de perda, métodos de otimização e utilização de métricas de desempenho. Nesse contexto, essa seção aborda a arquitetura da RNMP, as fases *forward*, *backward* e os métodos e algoritmos que realizam a aprendizagem (SILVA et al., 2017).

### 3.5.2 Arquitetura das RNMP

Segundo Silva et al. (2017), as RNMP são caracterizadas por possuir pelo menos uma camada de neurônios totalmente conectadas entre as camadas de entrada e saída, conhecida como camada escondida, intermediária ou oculta. Cada camada contém um conjunto de neurônios chamado de unidades ou perceptrons que estão interconectados por conexões ponderadas.

Esse tipo de rede possui uma arquitetura *feedforward* de múltiplas camadas, cujos sinais oriundos dos dados são abstraídos e parametrizados pelos pesos sinápticos presentes na ligação entre os neurônios, para estimular a rede a produzir um sinal de saída na última camada. Uma particularidade das redes neurais é que elas são tipicamente compostas por várias funções aninhadas,  $f_N(x) = f_3(f_2(f_1(x)))$ , sendo que cada função  $f_i(x)$  representa uma camada da rede e  $x$  o vetor com as características. Na Figura 26 é ilustrada uma estrutura de RNMP com duas camadas escondidas com  $q$  e  $n$  neurônios em cada camada e  $m$  neurônios na camada de saída, viabilizando a utilização desse modelo no reconhecimento de padrões com mais de uma categoria na classificação (SILVA et al., 2017).

Figura 26 – Rede Neural MLP



Fonte: Adaptado de Silva et al. (2017).

De forma similiar a estrutura do Perceptron, o sinal de ativação dos neurônios da RNMP é calculado pela soma dos sinais de entrada ao neurônio ponderadas pelos respectivos pesos sinápticos e viés (*bias*) (SILVA et al., 2017). Na Equação 3.12 é apresentada a expressão matemática para o neurônio  $n_i$  da primeira camada escondida da RNMP, em que  $a_i$  corresponde ao valor de saída deste neurônio após aplicar a função de ativação;  $\phi$  é a função de ativação  $n_i$ ;  $w_{ij}$  e  $b$  são os pesos sinápticos da camada de entrada e o *bias* do neurônio  $n_i$ . Por fim, o  $x_j$  corresponde ao dado de entrada.

$$y_i = \phi\left(\sum_{j=1}^n w_{ij}x_j + b_i\right) \quad (3.12)$$

A expressão matemática da arquitetura da RNMP da Figura 26 pode ser expressa em notação matricial. Nessa estrutura existe um vetor de entrada  $[X]_r$ , um vetor de saída  $[Y]_m$  e matrizes dos parâmetros (pesos e *bias*) que correspondem às ligações dos neurônios entre as camadas intermediárias,  $[W]_{r \times n}$ ,  $[W]_{q \times n}$  e  $[W]_{n \times m}$ , conforme a Equação 3.13.

$$\mathbf{Y}_{m \times 1} = \phi(\mathbf{W}_{m \times n}[\phi(\mathbf{W}_{n \times q}[\phi(\mathbf{W}_{q \times r}\mathbf{X}_{r \times 1})])]) \quad (3.13)$$

Essa arquitetura também pode ser apresentada no formato matricial não compactado e organizada de forma equivalente à estrutura da RNMP, de forma a perceber como cada neurônio em uma camada oculta aplica uma função de ativação não linear aos dados de entrada ponderados, ilustrando a complexidade do modelo e representações abstratas dos dados (KOLLMANNBERGER et al., 2021). A seguir, são apresentados os vetores e matrizes da RNMP da Figura 26.

$$\begin{aligned} [X] &= \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_r \end{bmatrix}, \quad [Y] = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \quad [W_1] = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \dots & w_{1r}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & \dots & w_{2r}^{(1)} \\ \vdots & \vdots & \dots & \vdots \\ w_{q1}^{(1)} & w_{q2}^{(1)} & \dots & w_{qr}^{(1)} \end{bmatrix} \\ [W_2] &= \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} & \dots & w_{1q}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} & \dots & w_{2q}^{(2)} \\ \vdots & \vdots & \dots & \vdots \\ w_{n1}^{(2)} & w_{n2}^{(2)} & \dots & w_{nq}^{(2)} \end{bmatrix} \quad \text{e} \quad [W_3] = \begin{bmatrix} w_{11}^{(3)} & w_{12}^{(3)} & \dots & w_{1m}^{(3)} \\ w_{21}^{(3)} & w_{22}^{(3)} & \dots & w_{2m}^{(3)} \\ \vdots & \vdots & \dots & \vdots \\ w_{n1}^{(3)} & w_{n2}^{(3)} & \dots & w_{nm}^{(3)} \end{bmatrix} \end{aligned}$$

Esse tipo de estrutura viabiliza a construção dos sistemas lineares e dos algoritmos computacionais, conforme apresentado nas Equações 3.14, 3.15 e 3.16. A Equação 3.14 representa as ligações entre os neurônios da camada de entrada e a primeira camada escondida, a Equação 3.15 as ligações entre a primeira e segunda camada escondida e, por fim, a Equação 3.16, que se refere à comunicação entre os neurônios da segunda camada escondida com os neurônios da camada de saída.

$$\begin{bmatrix} a_1^1 \\ a_2^1 \\ \vdots \\ a_q^1 \end{bmatrix} = \phi \left( \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \dots & w_{1r}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & \dots & w_{2r}^{(1)} \\ \vdots & \vdots & \dots & \vdots \\ w_{q1}^{(1)} & w_{q2}^{(1)} & \dots & w_{qr}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_r \end{bmatrix} \right) = \begin{bmatrix} \phi(w_{11}^{(1)}x_1 + w_{12}^{(1)}x_2 \dots + w_{1r}^{(1)}x_r) \\ \phi(w_{21}^{(1)}x_1 + w_{22}^{(1)}x_2 \dots + w_{2r}^{(1)}x_r) \\ \vdots \\ \phi(w_{q1}^{(1)}x_1 + w_{q2}^{(1)}x_2 \dots + w_{qr}^{(1)}x_r) \end{bmatrix} \quad (3.14)$$

$$\begin{bmatrix} a_1^2 \\ a_2^2 \\ \vdots \\ a_q^2 \end{bmatrix} = \phi \left( \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} & \dots & w_{1r}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} & \dots & w_{2r}^{(2)} \\ \vdots & \vdots & \dots & \vdots \\ w_{q1}^{(2)} & w_{q2}^{(2)} & \dots & w_{qr}^{(2)} \end{bmatrix} \begin{bmatrix} a_1^1 \\ a_2^1 \\ \vdots \\ a_q^1 \end{bmatrix} \right) = \begin{bmatrix} \phi(w_{11}^{(2)}a_1^1 + w_{12}^{(2)}a_2^1 \dots + w_{1r}^{(2)}a_r^1) \\ \phi(w_{21}^{(2)}a_1^1 + w_{22}^{(2)}a_2^1 \dots + w_{2r}^{(2)}a_r^1) \\ \vdots \\ \phi(w_{q1}^{(2)}a_1^1 + w_{q2}^{(2)}a_2^1 \dots + w_{qr}^{(2)}a_r^1) \end{bmatrix} \quad (3.15)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \phi \left( \begin{bmatrix} w_{11}^{(3)} & w_{12}^{(3)} & \dots & w_{1r}^{(3)} \\ w_{21}^{(3)} & w_{22}^{(3)} & \dots & w_{2r}^{(3)} \\ \vdots & \vdots & \dots & \vdots \\ w_{q1}^{(3)} & w_{q2}^{(3)} & \dots & w_{qr}^{(3)} \end{bmatrix} \begin{bmatrix} a_1^2 \\ a_2^2 \\ \vdots \\ a_q^2 \end{bmatrix} \right) = \begin{bmatrix} \phi(w_{11}^{(3)}a_1^2 + w_{12}^{(3)}a_2^2 \dots + w_{1r}^{(3)}a_r^2) \\ \phi(w_{21}^{(3)}a_1^2 + w_{22}^{(3)}a_2^2 \dots + w_{2r}^{(3)}a_r^2) \\ \vdots \\ \phi(w_{q1}^{(3)}a_1^2 + w_{q2}^{(3)}a_2^2 \dots + w_{qr}^{(3)}a_r^2) \end{bmatrix} \quad (3.16)$$

### 3.5.3 Etapas de Treinamento e Teste da Rede Neural MLP

As etapas de treinamento e teste são fundamentais na parametrização das redes neurais. Durante a etapa de treinamento o objetivo é ajustar os parâmetros da rede neural de forma a minimizar uma função custo. Esse processo ocorre de forma iterativa entre as fases de propagação *forward* e *backward*, conforme apresentado nos Algoritmo 1 nos passos 6 e 7. Esse processo é repetido por várias épocas<sup>3</sup> até que o modelo atinja o critério de parada no passo 5 do Algoritmo 1 (SILVA et al., 2017).

---

**Algorithm 1** Fase de Treinamento da RNMP.

---

- 1: Obter amostras de treinamento  $\mathbf{X}$  e  $\mathbf{Y}$  e iniciar os pesos  $\mathbf{W}$
  - 2: Especificar a taxa de aprendizagem  $\eta$
  - 3: Especificar precisão requerida  $e$  e número de épocas máxima  $epoch_{max}$
  - 4: Iniciar o contador de número de épocas ( $epoch \leftarrow 0$ )
  - 5: **while**  $|\text{ErroAtual} - \text{ErroAnt}| < e$  **AND**  $epoch < epoch_{max}$  **do**
  - 6:   Para todas as amostras de treinamento  $\mathbf{X}$  e  $\mathbf{Y}$
  - 7:   Etapa *Foward* (Obter  $\mathbf{I}$  e  $\mathbf{Y}$ )
  - 8:   Etapa de *Baward* (Obter  $\delta$  e  $\mathbf{W}$ )
  - 9:    $\text{ErroAnt} \leftarrow \text{ErroAtual}$
  - 10:    $epoch \leftarrow epoch + 1$
  - 11: **end while**
- 

<sup>3</sup> Unidade de tempo que representa uma passagem completa por todo o conjunto de dados de treinamento.

Uma vez treinada a rede neural, a etapa de teste é realizada para avaliar o desempenho deste modelo com dados ainda não vistos. Os dados de teste são diferentes dos dados de treinamento e são usados para verificar a capacidade de predição do modelo. Nesta etapa é realizada somente a fase de propagação *forward*, conforme pode ser observado no Algoritmo 2, em que se assume os pesos da rede obtidos na etapa de treinamento e calcula essa etapa no passo 3 (SILVA et al., 2017).

---

**Algorithm 2** Fase de Teste da RNMP.

---

- 1: Obter amostras de teste  $\mathbf{X}$
  - 2: Assumir  $\mathbf{W}$  ajustados no treinamento
  - 3: Execute a etapa *Foward* (Obter  $\mathbf{I}$  e  $\mathbf{Y}$ )
  - 4: Disponibilizar saídas da rede, as quais são dadas pelos elementos contidos em  $\mathbf{Y}$
- 

### 3.5.4 Fase de Propagação *Forward*

A fase de propagação *forward* de uma rede neural realiza o cálculo de predição na camada de saída. Durante essa fase é disponibilizado um conjunto de dados que são inseridos na camada de entrada como vetor  $\mathbf{X}$  e segue em um caminho unidirecional, passando por várias camadas intermediárias, até a camada de saída resultando no vetor  $\mathbf{Y}$ . As operações que ocorrem na rede nesta fase são as combinações lineares ponderadas pelos pesos sinápticos e as aplicações das funções de ativação Silva et al. (2017).

---

**Algorithm 3** Propagação para Foward

---

- 1: **Entrada:**  $X$  (dados de entrada),  $W$  (pesos),  $b$  (vieses)
  - 2: **Saída:**  $\hat{y}$  (previsão da rede)
  - 3: Inicialize a entrada da rede com os dados de entrada  $X$
  - 4: **for** cada camada  $l$  na rede neural **do**
  - 5:     Calcule  $z^{(l)} = W^{(l)} \cdot a^{(l-1)} + b^{(l)}$
  - 6:     Calcule a ativação  $a^{(l)} = \sigma(z^{(l)})$  ▷ onde  $\sigma$  é a função de ativação
  - 7: **end for**
  - 8:  $\hat{y} = a^{(L)}$  ▷ onde  $L$  é a última camada da rede
  - 9: **return**  $\hat{y}$
- 

### 3.5.5 Fase de Retropropagação *Backward*

O processo de treinamento da RNMP, conhecido como regra Delta Generalizada, ocorre em duas fases. A primeira corresponde à propagação adiante, denominada por *forward*, enquanto que a segunda fase trata da propagação reversa (*backward*) realizada pelo cálculo do erro ( $e$ ). O erro é calculado pela da diferença entre a resposta *forward* e os rótulos desejados (SILVA et al., 2017). A retropropagação é realizado da camada de saída até a camada de entrada, de forma a ajustar as matrizes de pesos visando minimizar o erro produzido por cada neurônio, por meio da diferenciação parcial e regra da cadeia. Na Equação 3.17, é apresentada a expressão matemática para o cálculo dos pesos.

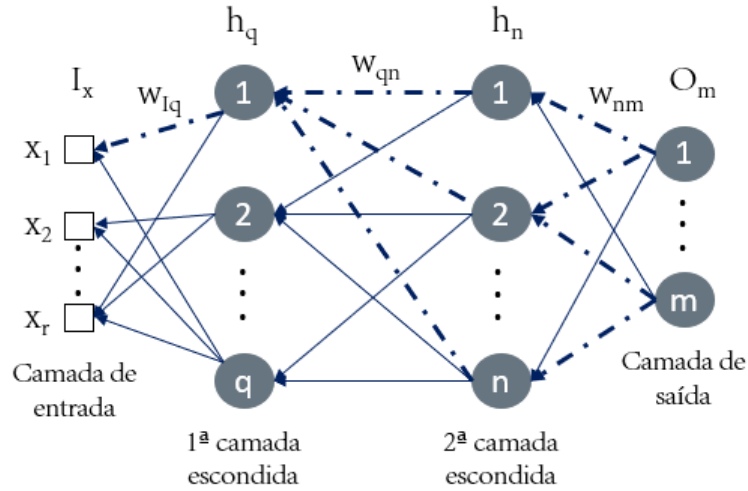
$$w(t+1) = w(t) - \eta \frac{\partial C_M}{\partial w} \quad (3.17)$$

Em que  $w(t)$  são os valores dos pesos atuais,  $w(t+1)$  os valores dos pesos atualizados,  $\eta$  a taxa de aprendizado e  $C_M$  a função custo médio. Em que a função custo é calculada pela Equação 3.18,

$$C_M = \frac{1}{2} \sum_{i=1}^n (y_i(k) - \hat{y}_i(k))^2 \quad (3.18)$$

Portanto, deve-se calcular variações dos custos de cada neurônio em função dos pesos da camada anterior, para aplicar sobre a Equação 3.17 e, com isso, definir os novos valores dos pesos. Na Figura 27 é ilustrada a fase *backpropagation*, com ênfase para o cálculo de atualização do primeiro peso que conecta a camada de entrada e a primeira camada oculta. As outras ligações representam as demais propagações *backward*. Enquanto que as derivadas de custo e as expressões para as atualizações dos pesos para a RNMP são tratadas na sequência, de acordo com (MARTINS, 2021).

Figura 27 – Backpropagation na RNMP.



Fonte: Adaptado de Martins (2021).

- **Ajuste dos pesos sináticos da camada de saída com a segunda camada:** Nesta camada o cálculo do erro é dado pela diferença entre o valor calculado e esperado, sobre as  $p$  amostras do conjunto de treinamento, conforme Equação 3.19.

$$\frac{\partial C_M}{\partial w_{ji}} = \frac{\partial C_M}{\partial e_j} \frac{\partial e_j}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} = e_j \sigma'(a_j) h_j \quad (3.19)$$

$$w(t+1) = w(t) + 2\eta \delta_r h_r \quad (3.20)$$

Em que  $\partial C_M$  é o custo médio,  $e$  é o erro do  $j$ -ésimo neurônio,  $\hat{y}$  é a saída fornecida pela rede no neurônio  $j$  da camada de saída.

- **Ajuste dos pesos sinápticos da segunda com a primeira camada oculta:** Para o cálculo dos pesos que ligam essas camadas utiliza-se a regra de diferenciação em cadeia sobre a função custo, conforme pode-se observar na Equação 3.21.

$$\frac{\partial C_M}{\partial W_{ji}^2} = \frac{\partial C_M}{\partial \hat{y}_j^2} \frac{\partial \hat{y}_j^2}{\partial a_j^2} \frac{\partial a_j^2}{\partial W_{ji}^2} = \eta \sigma'(a_j) h_q \sum_{i=1}^n \delta w_{ir} \quad (3.21)$$

$$w(t+1) = w(t) + \eta \delta_r h_q \quad (3.22)$$

- **Ajuste dos pesos sinápticos da primeira camada oculta com a camada de saída:** Para a atualização dos valores desses pesos utiliza-se, novamente a regra de diferenciação em cadeia, conforme a Equação 3.23.

$$\frac{\partial C_M}{\partial W_{ji}^2} = \frac{\partial C_M}{\partial \hat{y}_j^2} \frac{\partial \hat{y}_j^2}{\partial a_j^2} \frac{\partial a_j^2}{\partial W_{ji}^2} = \phi'(a_{h_q}) h_p \sum_{r=0}^R \delta w_{rq} \quad (3.23)$$

$$w(t+1) = w(t) + \eta \delta_q h_p \quad (3.24)$$

Em que  $\delta_r$  e  $\delta_q$  são os erros retropropagados nas respectivas camadas,  $\phi'$  é a derivada da função de ativação da camada oculta,  $h_p$  e  $h_q$  são as entradas para os neurônios das camadas ocultas, e  $w(t)$  são os pesos no tempo  $t$ .

Esse conjunto de equações deve ser combinado com a 3.17 para o cálculo dos pesos que é abstraída pelo *Algorithm 4* (SILVA et al., 2017),

---

**Algorithm 4** Fase Backward: Cálculo da Retropropagação do Erro

---

- 1: Input: Conjunto de dados para treinamento ( $\mathbf{X}$ ,  $\mathbf{Y}$ )
  - 2: Output: Parâmetros do modelo treinado  $\mathbf{W}$
  - 3:
  - 4: Calcular a predição  $\hat{y}^{(i)}$  pela propagação Forward
  - 5: Calcular a função custo  $L(y^{(i)}, \hat{y}^{(i)})$ ;  $i \leftarrow 1$  até  $N$
  - 6: Calcular o gradiente da função de custo:  $\partial \mathbf{L} / \partial \mathbf{W}$
  - 7: Atualizar os pesos para minimizar a função custo  $\mathbf{L}$
  - 8:  $w(t+1) = w(t) - \eta \frac{\partial L}{\partial W}$
- 

### 3.5.6 Métodos de Otimização

Nos paradigmas de aprendizado é essencial utilizar métodos de otimização na fase de treinamento, objetivando que os pesos se ajustem de forma a minimizar o erro e melhorar o tempo de treinamento, a convergência e o desempenho do modelo (JIANG, 2021). Dentro desse contexto destaca-se o Método do Gradiente Descendente (MGD) em razão da sua versatilidade, eficiência computacional e escalabilidade. Esse método segue

a direção do gradiente da função de perda com relação aos pesos ajusta-os na direção que diminui o erro. Além disso, o MGD possui ainda variações da sua implementação em que são introduzidos mecanismos para autoajustar as taxas de aprendizado, que são tratados na sequência.

- **Momento (*Momentum*)**: Adiciona uma componente de velocidade à atualização dos pesos, permitindo que o otimizador acumule a direção da descida do gradiente ao longo do tempo por meio das variações dos pesos (AGGARWAL et al., 2018),

$$\Delta w(t+1) = -\eta \nabla C(w_t) + \alpha \Delta w_t, \quad (3.25)$$

em que  $\Delta w_t$  é o momento acumulado na iteração anterior,  $\nabla C(w_t)$  a função gradiente descendente,  $\eta$  taxa de aprendizagem e  $\alpha$  coeficiente do *momentum*.

- **Root Mean Square Propagation (RMSprop)**: Ajusta as taxas de aprendizado de forma adaptativa para cada peso, resultando em um treinamento mais estável (AGGARWAL et al., 2018).
- **Adaptive Moment Estimation (Adam)**: Combina o momentum e o RMSprop, adaptando as taxas de aprendizado e mantendo momentos de primeira e segunda ordem dos gradientes (AGGARWAL et al., 2018).

Esses três modelos de otimização podem ser utilizados em três contextos: descida do gradiente completa (*full-batch gradient descent*), descida do gradiente em lote (*batch gradient descent*) e descida do gradiente em mini-lotes (JIANG, 2021).

- **Full-batch gradient descent**: Nesta abordagem são realizados os cálculos da perda, gradiente de perda e atualização dos pesos para cada uma das amostras da base de dados;
- **Batch gradient descent**: Também conhecida como *Stochastic Gradient Descent*, realiza os cálculos e a atualização dos pesos para um lote de dados. Neste caso, a atualização dos pesos é realizada para cada *batch*, que passa a ser um hiperparâmetro do modelo.
- **Mini-batch gradient descent**: Corresponde a uma sub-amostragem de cada *batch*.

## 3.6 Rede Neural Convolutacional (RNC)

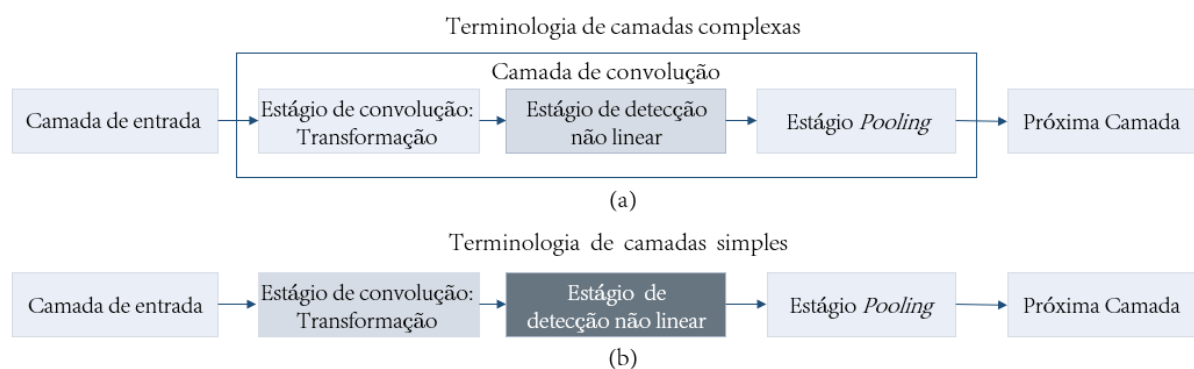
### 3.6.1 Introdução

Segundo [Goodfellow, Bengio e Courville \(2016\)](#) as RNC são modelos de AM preparados para lidar com estruturas de dados por meio da operação de convolução. Esse tipo de operação força a rede a tratar características locais das estruturas de dados (identificação de bordas e contornos em dados do tipo imagens), o compartilhamento de pesos e conexão esparsa entre os neurônios e os *kernels*, com o intuito de reduzir o número de parâmetros (forma de regularização) e melhorar o desempenho computacional.

Uma camada típica em uma RNC é processada em três etapas. A primeira corresponde a realização das operações de convolução com *kernels* para a produção dos mapas de características, a segunda etapa corresponde a aplicação da técnica de normalização (*Batch Normalization*) e a utilização da função de ativação não-linear, tratada na Seção 3.4.2, e por fim a terceira e última etapa trata da camada *pooling*. Potanto, uma RNC possui várias dessas estruturas de camadas convolucionais com inúmeros *kernels* para a operação de convolução e construção dos “*n*” mapas de características. Nesse tipo de estrutura há ainda a possibilidade que os dados de entrada sejam de mais de uma dimensão ([JIANG, 2021](#)).

Na Figura 28 são ilustradas duas terminologias para representar as camadas de convolução da rede neural, sendo a terminologia de simples camadas adotada neste documento ([GOODFELLOW; BENGIO; COURVILLE, 2016](#)).

Figura 28 – Fluxo de processos da camada de convolução: (a) complexa; (b) simples.



Fonte: Adaptado de [Goodfellow, Bengio e Courville \(2016\)](#).

### 3.6.2 Operação de Convolução

A operação de convolução é utilizada para conectar duas camadas da rede neural, transformando os dados da camada de entrada em estruturas que realçam suas caracte-

rísticas peculiares. Essa operação corresponde ao produto escalar entre os dados de uma camada com o *kernel* (JIANG, 2021).

Nesse tipo de operação o número de neurônios da segunda camada é definida como o resultado da expressão,  $n_2 = n_1 - k + 1$  para o caso de uma convolução unidimensional, (JIANG, 2021). Já no caso de uma convolução bidimensional a expressão é  $n_2 = (2p + n_1 - k)/s$ , em que:  $n_1$  e  $n_2$  são os números de neurônios das camadas de entrada e saída da operação de convolução,  $k$  tamanho dos *kernels*,  $s$  é o *stride* e  $p$  é o tamanho do *zero-padding* (GOODFELLOW; BENGIO; COURVILLE, 2016).

Nas Equações 3.26 e 3.27 são apresentadas as expressões matemáticas da operação de convolução discreta em uma notação indicial para as estruturas de uma e duas dimensões, respectivamente. Em que,  $y_i$  e  $y_{ij}$  correspondem aos valores dos neurônios das camadas de saída,  $w_i$  e  $w_{ij}$  os *kernels* e  $x_j$  e  $x_{ij}$  os valores dos neurônios da camada de entrada, para as condições unidimensional (JIANG, 2021).

$$y_j = \sum_{i=1}^k w_i \times x_{j+i-1} \quad , \quad \forall j = 1, 2, \dots, n \quad (3.26)$$

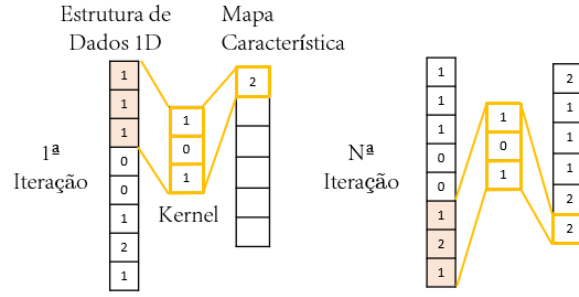
$$y_{ij} = \sum_{j=0}^k \sum_{i=0}^k w_{ij} \times x_{j+i-1} \quad , \quad \forall j = 1, 2, \dots, n \quad (3.27)$$

Sendo assim, dois hiperparâmetros fundamentais nas redes convolucionais são o *stride* e *zero-padding*. O *stride* corresponde ao passo do *kernel* sobre a matriz de característica (estrutura de dados) enquanto que o *zero-padding* realiza um tratamento de borda sobre as estruturas de dados de forma a manter fixas as dimensões do mapa de características após a operação de convolução (AGGARWAL et al., 2018).

Na Figura 29 é ilustrado o processo de cálculo da convolução unidimensional sobre uma estrutura de dados com  $N$  amostras. Esse processo é realizado deslizando o filtro ao longo da sequência de dados de entrada. Em cada posição, os elementos do filtro são multiplicados pelos elementos correspondentes da sequência dos dados de entrada e os resultados são somados para produzir um único valor na saída. Esse processo é repetido para cada posição na sequência de entrada, gerando uma nova sequência de saída com a capacidade de capturar características presentes nos dados originais (JIANG, 2021).

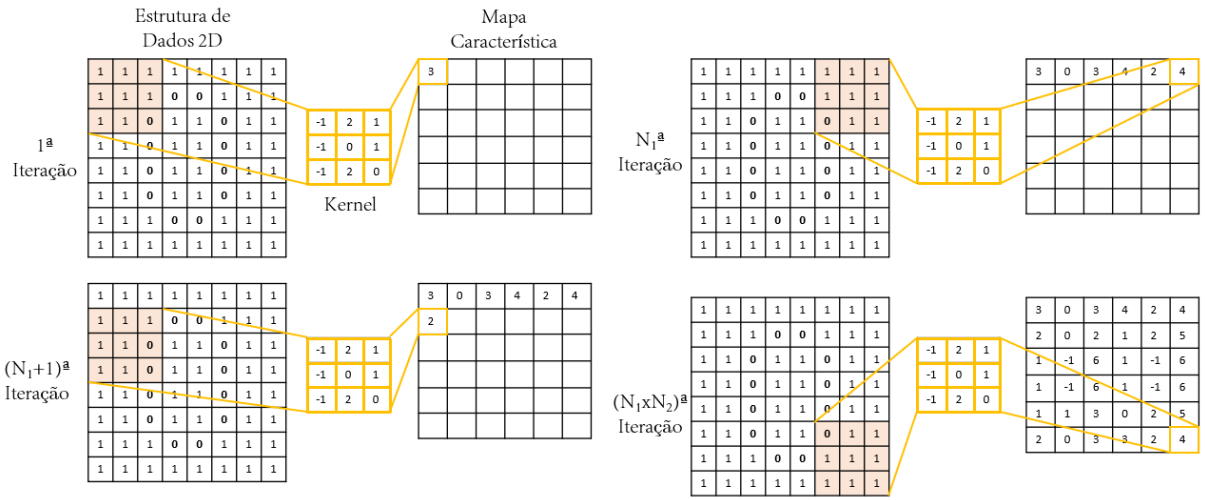
Na Figura 30 é ilustrado o processo de cálculo da convolução bidimensional sobre uma estrutura de dados de dimensão  $N_1 \times N_2$ . Neste caso, o filtro é deslizado sobre a matriz de dados de entrada. Em cada posição, os elementos do filtro são multiplicados pelos elementos correspondentes na região da matriz de entrada coberta pelo filtro. Os resultados dessas multiplicações são somados para produzir um único valor na saída. Esse processo é repetido para cada posição na matriz de entrada, resultando em uma nova matriz de saída (JIANG, 2021).

Figura 29 – Operação de convolução para estruturas de dados unidimensional (1D).



Fonte: Adaptado de Jiang (2021).

Figura 30 – Operação de convolução para estruturas de dados unidimensional (2D).

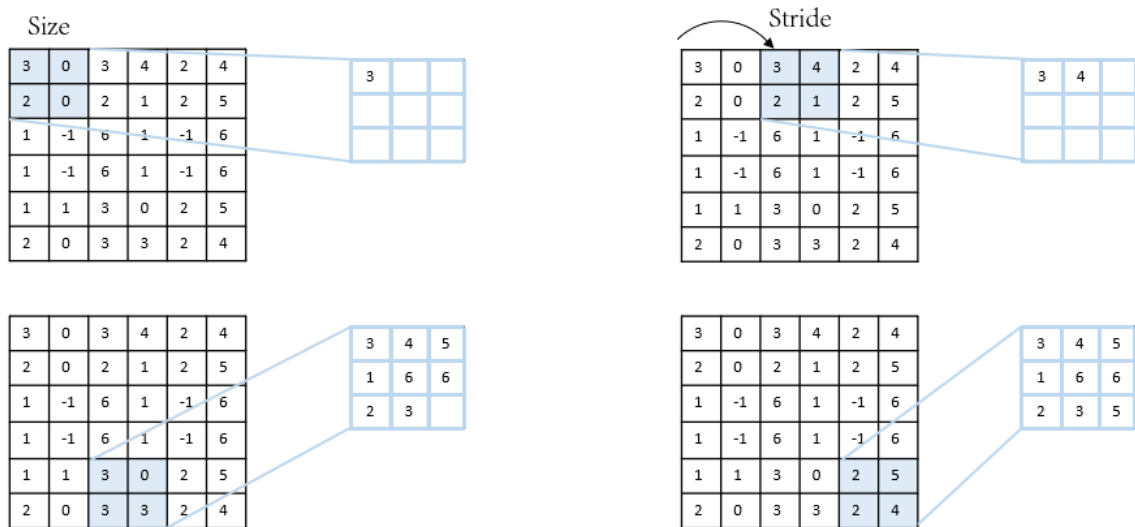


Fonte: Adaptado de Jiang (2021).

### 3.6.3 Camada Pooling

A camada de pooling reduz dimensionalidade dos dados ao selecionar valores representativos, mantendo características essenciais para processamento eficiente em redes neurais. Além disso, contribui a tornar a representação das características do mapa invariante a pequenas translações com os dados de entrada (GOODFELLOW; BENGIO; COURVILLE, 2016). Essa técnica resume-se em deslizar uma janela de tamanho  $m$  sobre os neurônios do mapa de características com um passo  $m$  e o valor máximo, médio ou mínimo é calculado nessa janela. A camada maxpooling é ilustrada na Figura 31 e representadas na Equação 3.28 em que  $\mathbf{X}$  é a matriz características de tamanho  $n \times n$  e  $y$  é o valor máximo dentro da janela de cálculo (JIANG, 2021).

$$y = \max(\mathbf{X}) \quad (3.28)$$

Figura 31 – Camada *pooling*

Fonte: Adaptado de [Jiang \(2021\)](#).

### 3.6.4 Tipos Arquitetura de RNC

Um panorama das arquiteturas de redes neurais convolucionais revela uma evolução das estruturas diante parametrização e testes em bases de dados voltados a visão computacional. Essas estruturas contém os elementos básicos da convolução e a inclusão de novas operações como a inserção de blocos residuais, blocos de identidades ([CANZIANI; PASZKE; CULURCIELLO, 2016](#)). No Quadro 5 são apresentados cinco redes convolucionais com o número de parâmetros, acurácia e contribuições. A base de dados utilizada foi a *ImageNet* que possui mais 14 milhões de imagens e 20 mil categorias.

Quadro 5 – Arquiteturas de redes convolucionais.

| CNN                 | N. de parâmetros  | Acurácia | Contribuição                                                             |
|---------------------|-------------------|----------|--------------------------------------------------------------------------|
| LetNet-5 (1998)     | $60 \times 10^3$  | -        | Modelo padrão de empilhamento de convoluções e MLP.                      |
| AlexNet (2012)      | $60 \times 10^6$  | 54%      | Utilizou a função de ativação ReLU e <i>Dropout</i> .                    |
| VGGNet-16 (2014)    | $138 \times 10^6$ | 71%      | Redes profundas com empilhamento de convoluções uniformes.               |
| ResNet-50 (2015)    | $25 \times 10^6$  | 76%      | Popularização do <i>skipconnections</i> e do <i>BatchNormalization</i> . |
| Inception-v3 (2015) | $23 \times 10^6$  | 78%      | Normalização em Lote.                                                    |

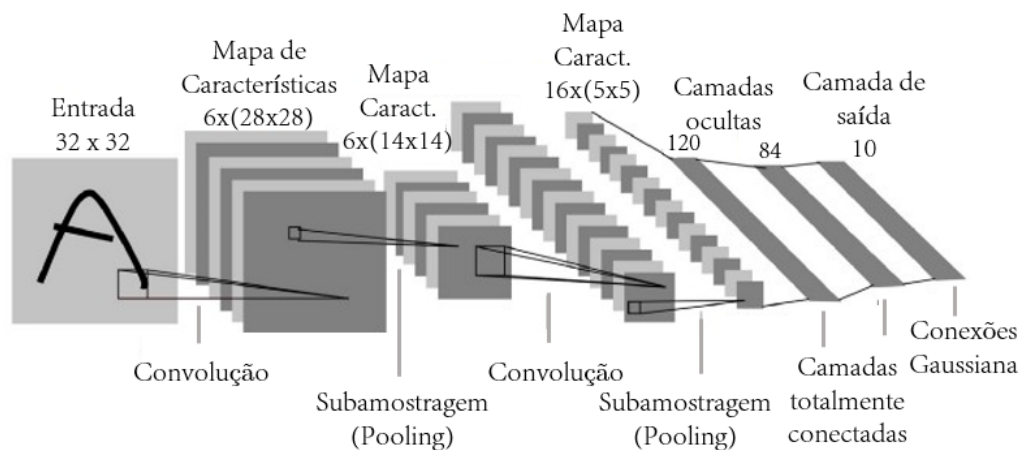
Fonte: Adaptado de [Canziani, Paszke e Culurciello \(2016\)](#).

*Nota:* LetNet-5 não fez parte do estudo do artigo [Canziani, Paszke e Culurciello \(2016\)](#).

Percebe-se que nesse quadro comparativo de desempenho dos modelos RNC, existe uma evolução e modificações sobre a arquitetura da RNC. Essas modificações implicam na inclusão de novos elementos e aumento percentual no número de acertos na classificação das imagens percentente à base de dados *ImageNet*. Entretanto, a pesquisa bibliográfica da Seção 2, mostrou a utilização recente das RNC do tipo LeNet-5 ao problema de classificação de cavitação em bombas centrífugas.

Segundo [LeCun et al. \(1998\)](#), a LeNet-5 é uma arquitetura de RNC projetada para reconhecimento de dígitos manuscritos e marcou o início do uso de redes convolucionais em visão computacional. A LeNet-5 possui camadas convolucionais seguidas por camadas *pooling* e camadas totalmente conectadas. Na Figura 32 é ilustrada essa RNC formada por duas camadas de convoluções de 6 e 16 *kernels*, duas camadas de *pooling* de ordem de subamostragem de  $2 \times 2$  e duas camadas totalmente conectadas, com 120 e 84 neurônios respectivamente. A camada de entrada corresponde a uma estrutura de dados bidimensional (escala de cinza) com dimensões de  $32 \times 32$  pixels, enquanto que a camada de saída possui dez neurônios, para realizar a classificação ([LECUN et al., 1998](#)).

Figura 32 – Rede neural convolucional LeNet-5.



Fonte: Adaptado de [LeCun et al. \(1998\)](#).

## 3.7 Técnicas e Métricas ao Desempenho das Redes

### 3.7.1 Problemas de Aprendizado em Redes Neurais

Segundo [Goodfellow, Bengio e Courville \(2016\)](#) a fase de treinamento de redes neurais com muitas camadas podem apresentar problemas relacionados à propagação dos gradientes durante o processo de retropropagação da informação. Esse processo é fundamental para ajustar os pesos da rede durante o aprendizado. Entre esses problemas estão o desvio de covariância (*covariate shift*), o desaparecimento dos gradientes (*vanishing gradient*) e a explosão dos gradientes (*exploding gradient*).

- O problema de desvio de covariância interna ocorre durante a fase de treinamento das redes neurais. Esse fenômeno descreve a mudança dos valores dos neurônios das camadas ocultas a medida que os parâmetros são atualizados. Isso acontece porque cada camada da rede depende dos valores de entrada provenientes das camadas anteriores, e à medida que os parâmetros são ajustados, as distribuições são alteradas, dificultando a convergência numérica. Esse tipo de problema ocasiona instabilidade numérica (dificuldade de convergência) e aumenta a sensibilidade da rede à inicialização dos pesos (IOFFE; SZEGEDY, 2015).
- O problema de desaparecimento dos gradientes ocorre quando os gradientes diminuem significativamente à medida que são retropropagados através das camadas da rede. Isso acontece porque a derivada da função de ativação (como a sigmóide ou a tangente hiperbólica) pode ser menor que 1 em algumas regiões, e quando várias dessas derivadas são multiplicadas juntas durante a retropropagação, os gradientes diminuem exponencialmente à medida que retrocedem para as camadas iniciais. Isso pode resultar em camadas iniciais da rede recebendo atualizações muito pequenas, o que torna o treinamento dessas camadas lento e dificulta a aprendizagem de padrões (GOODFELLOW; BENGIO; COURVILLE, 2016).
- O problema de explosão dos gradientes ocorre quando os gradientes aumentam exponencialmente à medida que são retropropagados pela rede. Isso pode ocorrer quando as derivadas das funções de ativação são maiores que 1, e ao multiplicá-las durante a retropropagação, os gradientes crescem à medida que retrocedem para as camadas iniciais. Isso pode levar a atualizações muito grandes nos pesos da rede, causando instabilidade no treinamento e resultando em oscilações ou divergência dos parâmetros (GOODFELLOW; BENGIO; COURVILLE, 2016).

Para tratar e solucionar esses tipos de problemas existem técnicas matemáticas e numéricas como a inicialização dos pesos, normalização em lote, técnicas de regularização e otimização de hiperparâmetro. Cada uma delas é apresentada na sequência.

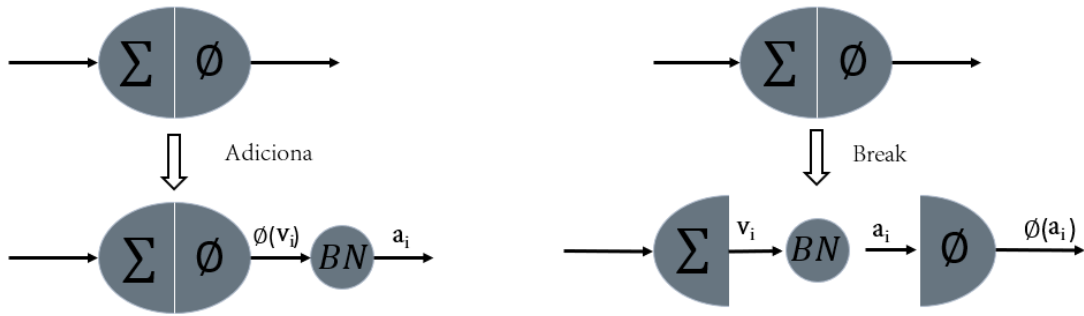
### 3.7.2 Técnicas de Inicialização dos Pesos das Redes Neurais

A etapa de inicialização dos pesos define os valores iniciais dos parâmetros da rede neural. Uma escolha adequada de inicialização não apenas acelera a convergência do treinamento, mas também influencia significativamente a capacidade da rede de evitar problemas como o desaparecimento ou explosão de gradientes, especialmente em redes profundas. Além disso, uma boa inicialização pode ajudar a rede a escapar de mínimos locais ou pontos de sela no espaço de otimização, melhorando a eficiência do processo de aprendizado e a qualidade final do modelo (AGGARWAL et al., 2018).

### 3.7.3 Técnica de Normalização em Lotes (*Batches*)

A normalização por lotes (*batch normalization*) é uma técnica amplamente utilizada em redes neurais profundas e trata problemas de desvio de covariância (*internal covariate shift*) e *vanishing and exploding gradients* (AGGARWAL et al., 2018). Essa normalização em *batch* visa minimizar os efeitos decorrente do gradiente, por meio da adição de camadas de normalização dos dados processados entre as camadas ocultas, criando características com variâncias similares. A inclusão dessa nova camada pode ser pós ou pré função de ativação, conforme ilustrado na Figura 33.

Figura 33 – Camada de normalização: (a) pós ativação; (b) pré ativação.



Fonte: Adaptado de Aggarwal et al. (2018).

Em cada unidade de processamento há uma média  $\beta$  e um desvio padrão  $\gamma$ , que são aprendidos durante o treinamento da rede. Esses parâmetros estão relacionados ao atual *mini-batch* e o potencial de ativação do neurônio  $v_i^{(m)}$  que corresponde a um *batch* com  $m$  instâncias (AGGARWAL et al., 2018). O *batch normalization* é aplicado sobre os potenciais de ativação dos neurônios de uma camada oculta (Equação 3.29), na sequência define a variância de cada neurônio por meio da Equação 3.30 e, por fim, é realizado o cálculo de normalização utilizando as Equações 3.31 e 3.32.

$$\mu_i = \frac{1}{m} \sum_{r=1}^m v_i^{(r)} \quad (v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(m)}) \quad (3.29)$$

$$\sigma_i^2 = \frac{1}{m} \sum_{r=1}^m (v_i^{(r)} - \mu_i)^2 \quad (3.30)$$

$$\hat{v}_i^{(r)} = \frac{v_i^{(r)} - \mu_i}{\sigma_i} \quad (3.31)$$

$$a_i^{(r)} = \gamma_i \hat{v}_i^{(r)} + \beta_i \quad (3.32)$$

Em resumo, a camada *batch normalization* estabiliza o treinamento, acelera a convergência, reduz o *overfitting* e viabiliza o uso de taxas de aprendizado mais altas.

### 3.7.4 Técnicas de Regularização

As estratégias de regularização visam contribuir para que o algoritmo de A diminua o erro proveniente da fase de teste sem aumentar o erro de treinamento, por meio do equilíbrio dos erros de variância e viés (KOLLMANNSEBERGER et al., 2021).

Existem várias abordagens para encontrar as regularidades do modelo. A solução simples com a ampliação da base de dados por meio de novos ensaios ou a criação de dados sintéticos, melhorando a capacidade de generalização e abstração do conteúdo presente nos dados e as soluções que tornam o problema de solução indeterminada em determinada por meio da penalização da função de custo ou a utilização de *dropout* (KOLLMANNSEBERGER et al., 2021). Dentro desse contexto, serão abordadas nessa seção, as técnicas de *dataset augmentation*, *early stop*, regularizações Lasso ( $L_1$ ) e Ridge ( $L_2$ ) e *Dropout*.

- *Dataset Augmentation*

A primeira maneira de melhorar a capacidade de generalização do modelo é utilizando mais dados. Na maioria dos casos, obter mais dados para a fase de treinamento pode ser considerada uma tarefa difícil, custosa ou inviável. Assim, uma opção é utilizar técnicas ou modelos matemáticos para sintetizar dados que apresentem o mesmo comportamento dos dados reais. Entre essas técnicas estão a inclusão de ruídos sobre um série temporal, rotação e/ou aumento de imagens e até a utilização de redes neurais concorrentes (KIRSCHNER; RIEDELBAUCH et al., 2023).

- *Early Stop*

Nesse tipo de técnica deve separar a base de dados nos grupos de treinamento, validação e teste. Nos grupos de treinamento e validação são realizados os cálculos dos parâmetros do modelo, enquanto que na base de teste verifica-se o ajuste ótimo do modelo (*underfitting* e *overfitting*) (HEATON, 2018).

O principal objetivo do *early stop* é interromper o processo de treinamento assim que o erro de validação aumentar, antes que o modelo entre em um regime de *overfitting*. Nessa interrupção do processamento são registrados e salvos os valores dos parâmetros (HEATON, 2018).

- Regularizações  $L_1$  e  $L_2$ :

As técnicas de regularização têm como objetivo limitar a capacidade da modelagem ao penalizar os parâmetros do modelo com a adição de informação ( $\Omega$ ). Esse termo penalidade é adicionado à função de custo ( $C$ ) de forma a reduzir os valores dos pesos, conforme observa-se na Equação 3.33.

$$C^* = C + \lambda\Omega \quad (3.33)$$

em que  $C$  é a função de custo,  $C^*$  é a função custo regularizada, o coeficiente  $\lambda$  é um hiperparâmetro que pondera o termo de penalidade ( $\Omega$ ). No caso das regularizações dos tipos  $L_1$  e  $L_2$  a penalidade ocorre sobre os pesos das redes, conforme observa-se nas Equações 3.34 e 3.35 (KOLLMANNSEBERGER et al., 2021).

$$C_{L1} = C + \frac{\lambda}{n} \sum_{i=j}^n |\omega_j| \quad (3.34)$$

$$C_{L2} = C + \frac{\lambda}{n} \sum_{i=j}^n ||\omega_j||^2 \quad (3.35)$$

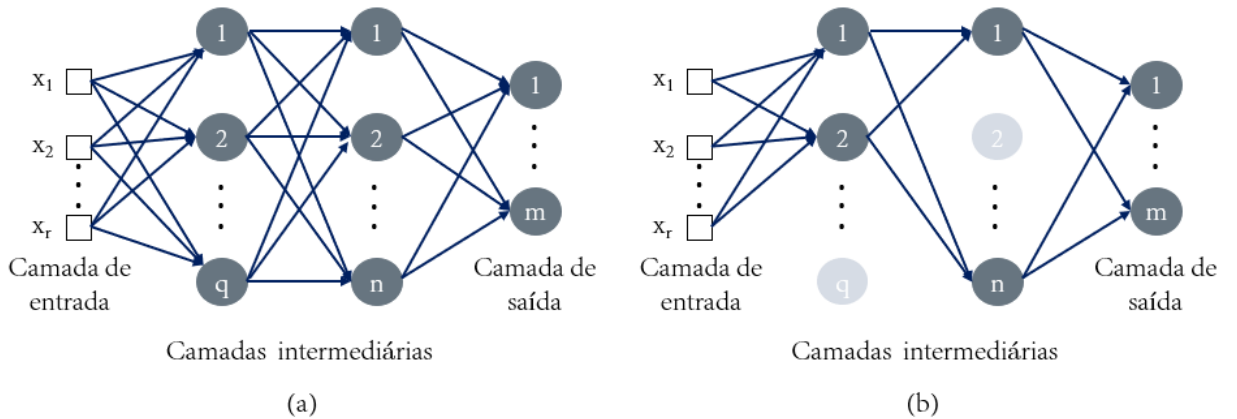
A utilização da regularização  $L_1$  resulta em matrizes esparsas, em que os pesos  $\mathbf{W}$  tendem a se igualar a zero. Por outro lado, a aplicação da regularização  $L_2$  incentiva os pesos  $\mathbf{W}$  a assumirem valores próximos de zero. Essa busca por pesos próximos de zero tem o efeito de mitigar a sensibilidade do modelo a flutuações nos valores de entrada, o que, por sua vez, reduz as oscilações significativas na saída do modelo.

No geral, a regularização com penalidades de norma de parâmetros pode reduzir o erro de generalização das redes neurais (KOLLMANNSEBERGER et al., 2021).

- *Dropout*

A regularização por *dropout* altera a própria rede em vez de modificar a função de custo. Antes de inicializar o fluxo de treinamento, metade dos neurônios ocultos são aleatória e temporariamente desativados, enquanto os neurônios de entrada e saída permanecem inalterados. Em seguida, são executadas as atualizações dos parâmetros por meio dos métodos baseados em MGD, em que a cada *mini-batch* dos dados de treinamento, os neurônios de *dropout* são restaurados e novos neurônios são desativados, repetindo o procedimento para o próximo mini-batch. Na Figura 34 é ilustrada uma RNMP com *dropout* (KOLLMANNSEBERGER et al., 2021).

Figura 34 – *Dropout*



Fonte: Adaptado de Kollmannsberger et al. (2021).

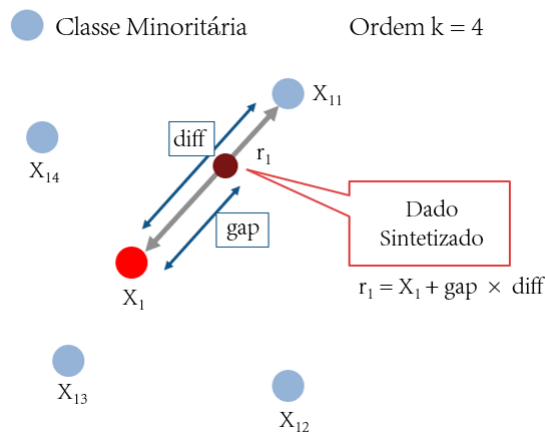
### 3.7.5 Técnicas de Balanceamento dos Dados

O desbalanceamento de classes é uma questão comum em conjuntos de dados que subsidiam modelos de A. Essa discrepância pode levar os modelos de A a resultados tendenciosos e com baixo desempenho na classificação das classes minoritárias (GHOSH et al., 2022). Para lidar com esse problema, pode-se realizar uma nova coleta dos dados, uma reamostragem dos dados, utilizar pesos nas classes para o caso de modelos de natureza como árvores de falha, MVS e redes neurais e técnicas de avaliação (*F1-score*).

Segundo Fernández et al. (2018) na reamostragem tem-se a subamostragem da classe majoritária (*undersampling*) ou a superamostragem da classe minoritária (*oversampling*). Entre essas abordagens existem as técnicas *Random Undersampling*, *Synthetic Minority Over-sampling Technique* (SMOTE) e *Adaptive Synthetic Sampling* (ADASYN).

O SMOTE é uma técnica de superamostragem que visa gerar amostras sintéticas para a classe minoritária, conforme ilustrado na Figura 35. Este algoritmo concentra-se no espaço de características, criando novas instâncias por meio de uma interpolação entre instâncias semelhantes que estão próximas umas das outras. O procedimento para calcular amostras sintéticas envolve o uso de uma métrica de distância para determinar os vizinhos mais próximos de cada amostra e um fator aleatório para ponderar a distância da amostra original (FERNÁNDEZ et al., 2018).

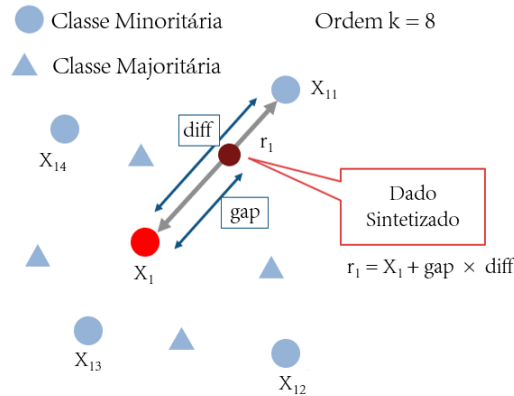
Figura 35 – Técnica SMOTE



Fonte: Adaptado de Vidhya (2020).

A técnica ADASYN trata da forma generalizada do algoritmo SMOTE. Similarmente ao SMOTE, seu objetivo é ampliar a amostragem da classe minoritária por meio da geração de instâncias sintéticas, conforme ilustrado na Figura 36. No entanto, sua distinção ao SMOTE consiste ao considerar uma distribuição de densidade que pondera o grau de dificuldade em criar novas amostras. Essa abordagem possibilita ao algoritmo adaptar uma fronteira de decisão, a partir da dificuldade de cada amostra (BATISTA; PRATI; MONARD, 2004).

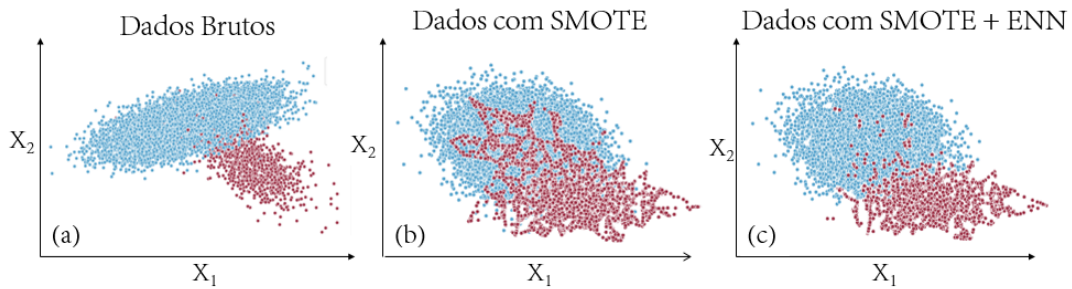
Figura 36 – Técnica ADASYN.



Fonte: Adaptado de Vidhya (2020)

O SMOTE+ENN é uma técnica híbrida que visa limpar pontos de dados sobrepostos para cada uma das classes distribuídas no espaço da amostra, uma vez que os clusters das classes podem invadir o espaço um do outro, resultando em um sobreajuste dos dados. Portanto, o SMOTE estima os vizinhos mais próximos de cada classe minoritária ao ponto de criar um superajuste, enquanto que o ENN remove as instâncias que encontram-se a grandes distâncias do núcleo, (BATISTA; PRATI; MONARD, 2004). Na Figura 37 é ilustrado um conjunto de dados com duas classes sem a aplicação de nenhuma técnica, ou seja dados brutos e com as técnicas SMOTE e SMOTE + ENN.

Figura 37 – Geração de dados sintéticos: (a) brutos ; (b) SMOTE; (c) SMOTE+ENN.

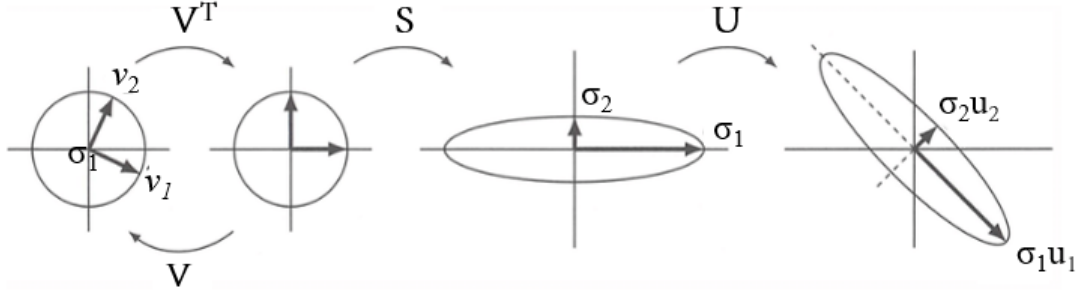


Fonte: Adaptado de Vidhya (2020).

### 3.7.6 Método Decomposição do Valor Singular

A Decomposição em Valores Singulares (DVS) é um método numérico da Álgebra Linear utilizado para decompor matrizes retangulares em três componentes principais: duas matrizes ortonormais, que representam os vetores singulares da esquerda e direita, e uma matriz diagonal que contém os autovalores. A combinação dessas matrizes compreende a transformação dos espaços vetoriais, conforme ilustrado na Figura 38, facilitando o discernimento de informações latentes e ruídos. A DVS tem aplicações em áreas de processamento e análise de sinais, viabilizando a identificação de componentes principais e a redução de dimensionalidade (GOLUB; KAHAN, 1965).

Figura 38 – Decomposição do Valor Singular (DVS).



Fonte: Adaptado de [Peters \(2019\)](#).

A Equação 3.36 trata dessa relação da matriz de dados  $\mathbf{A}_{m \times n}$  para as suas decomposições ortornomais,

$$\mathbf{A} = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^T. \quad (3.36)$$

- $\mathbf{U}$  é uma matriz ortogonal  $m \times r$  cujas colunas são os vetores singulares à esquerda.
- $\mathbf{\Lambda}$  é uma matriz diagonal  $r \times r$  de valores singulares ou autovalores.
- $\mathbf{V}^T$  é a transposta de uma matriz dos ortonormal  $r \times n$  cujas colunas são os vetores singulares à direita.

Para obter essas matrizes, realiza-se o cálculo dos autovalores e autovetores, conforme apresentado nas Equações 3.37 e 3.38, e aplica-se a relação  $\mathbf{u}_i = \frac{1}{\sqrt{\lambda_i}} \mathbf{A} \mathbf{v}_i$  para obter a matriz ortonormal  $\mathbf{U}$ , como observado na Equação 3.39.

$$\mathbf{\Lambda} = \begin{bmatrix} \sqrt{\lambda_1} & 0 & \dots & 0 \\ 0 & \sqrt{\lambda_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{\lambda_r} \end{bmatrix}_{r \times r} \quad (3.37)$$

$$\mathbf{V} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \dots \quad \mathbf{v}_n]_{r \times n} = \begin{bmatrix} \frac{w_{11}}{|w1|} & \frac{w_{21}}{|w2|} & \dots & \frac{w_{n1}}{|wr|} \\ \frac{w_{12}}{|w1|} & \frac{w_{22}}{|w2|} & \dots & \frac{w_{nr}}{|wr|} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{w_{1r}}{|w1|} & \frac{w_{2r}}{|w2|} & \dots & \frac{w_{nr}}{|wr|} \end{bmatrix}_{r \times n} \quad (3.38)$$

$$\mathbf{U} = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_n]_{m \times r} = \begin{bmatrix} u_{11} & u_{21} & \dots & u_{r1} \\ u_{12} & u_{22} & \dots & u_{r2} \\ \vdots & \vdots & \ddots & \vdots \\ u_{1m} & u_{2m} & \dots & u_{rm} \end{bmatrix}_{m \times r} \quad (3.39)$$

A partir dessas matrizes é possível ajustar o espaço amostral de um conjunto de dados de acordo com as exigências do modelo de A. Em outras palavras, a matriz de *design* pode ser representada em qualquer um desses espaços ilustrados na Figura 38.

Além disso, é possível reduzir a dimensionalidade dessas matrizes para representar suas principais componentes. Esse método é conhecido como Decomposição do Valor Singular Truncado, conforme exemplificado na Equação 3.40, em que observa-se um truncamento para duas componentes principais destacadas em negrito ([PETERS, 2019](#)).

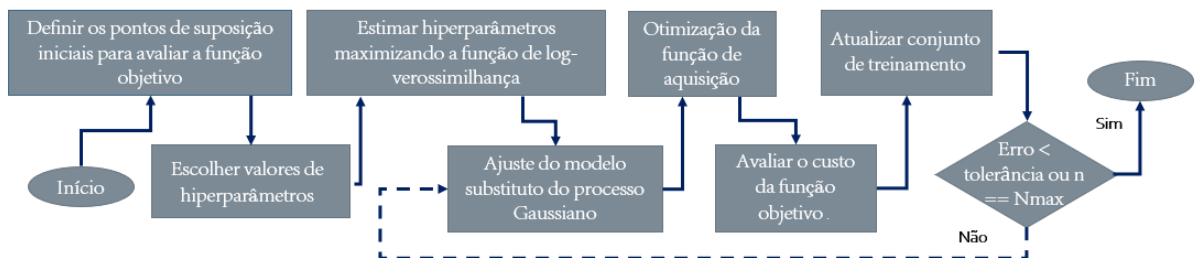
$$[A]_{m \times n} = \begin{bmatrix} \mathbf{u}_{11} & \mathbf{u}_{21} & \dots & u_{r1} \\ \mathbf{u}_{12} & \mathbf{u}_{22} & \dots & u_{r2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{1m} & \mathbf{u}_{2m} & \dots & u_{rm} \end{bmatrix} \begin{bmatrix} \sqrt{\lambda_1} & 0 & \dots & 0 \\ 0 & \sqrt{\lambda_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{\lambda_r} \end{bmatrix} \begin{bmatrix} \frac{\mathbf{w}_{11}}{|\mathbf{w}_1|} & \frac{\mathbf{w}_{21}}{|\mathbf{w}_2|} & \dots & \frac{\mathbf{w}_{n1}}{|\mathbf{w}_r|} \\ \frac{\mathbf{w}_{12}}{|\mathbf{w}_1|} & \frac{\mathbf{w}_{22}}{|\mathbf{w}_2|} & \dots & \frac{\mathbf{w}_{n1}}{|\mathbf{w}_r|} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{w_{1r}}{|w_1|} & \frac{w_{2r}}{|w_2|} & \dots & \frac{w_{nr}}{|w_r|} \end{bmatrix} \quad (3.40)$$

### 3.7.7 Otimização Bayesiana: Cálculo de Hiperparâmetros

A otimização Bayesiana é útil em contextos onde a função objetivo é custosa de calcular, não tem uma expressão bem definida e não oferece um mecanismo eficiente para estimar seu gradiente. Um exemplo motivador são os ajustes de hiperparâmetros em modelos complexos de A, como redes neurais profundas. Esse processo é caro em termos de tempo e energia, e os hiperparâmetros muitas vezes são discretos, impossibilitando o cálculo do gradiente ([GARNETT, 2023](#)).

Nesse processo de otimização é realizada uma sequência de decisões, em que é preciso determinar a próxima observação, com base no resultado obtido. Tais decisões são tomadas em meio à incerteza, uma vez que o resultado de uma observação não pode ser conhecido antecipadamente. A abordagem Bayesiana lida com essa incerteza de maneira sistemática, utilizando probabilidades e inferência Bayesiana para raciocinar sobre quantidades incertas, incluindo a própria função objetivo. Esse método possibilita avaliar o mérito de uma determinada localização de observação com base na distribuição de probabilidade sobre o valor que poderia ser observado ([GARNETT, 2023](#)).

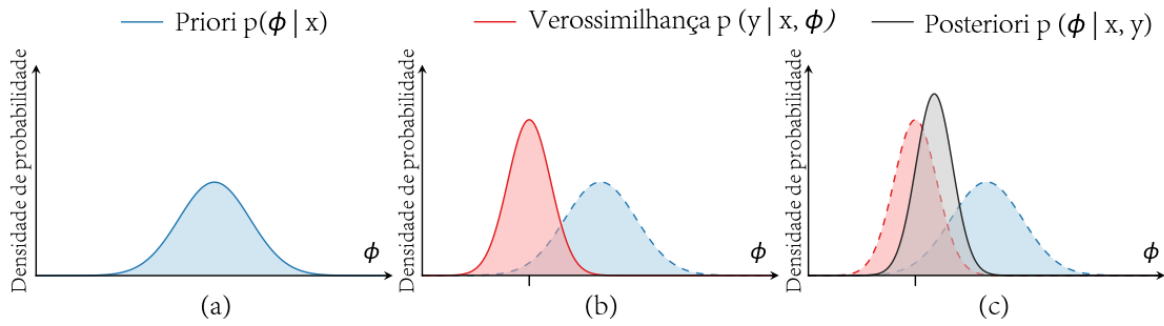
Figura 39 – Fluxograma do processo de Otimização Bayesiana.



Fonte: Adaptado de [Shende et al. \(2021\)](#).

Segundo Garnett (2023), a inferência Bayesiana é explicada como um processo iterativo, começando com uma distribuição *a priori* que expressa as incertezas iniciais sobre os parâmetros do modelo (Figura 40-a). A atualização das incertezas ocorre após a observação de dados, utilizando o modelo de observação para calcular a verossimilhança das observações sob diferentes valores dos parâmetros do modelo (Figura 40-b). Isso resulta na obtenção da distribuição *a posteriori* dos parâmetros (Figura 40-c), que é proporcional ao produto da distribuição *a priori* e da verossimilhança, normalizado pela constante de verossimilhança (Teorema de Bayes).

Figura 40 – Distribuições de probabilidade: (a) Priori; (b) Verossimilhança; (c) Posteriori



Fonte: Adaptado de Garnett (2023).

O teorema de Bayes (Equação 3.41) da origem a distribuição *a posteriori*  $p(y|x)$  reflete um compromisso entre as incertezas iniciais e as informações dos dados observados, que estão expressas na Equação 3.42 que foram organizadas do teorema de Bayes (GARNETT, 2023).

$$p(\phi|x, y) = \frac{p(\phi|x)p(y|x, \phi)}{p(y|x)} \quad (3.41)$$

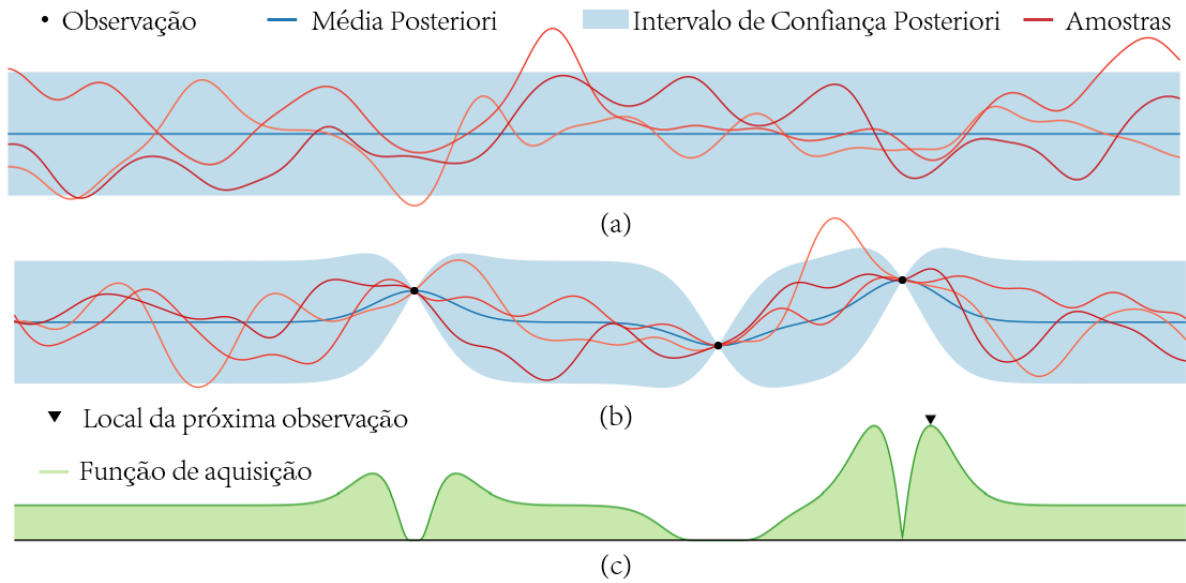
$$p(y|x) = \int p(y|x, \phi)p(\phi|x), d\phi. \quad (3.42)$$

Além do teorema de Bayes, a otimização bayesiana é conduzida por meio de uma função de aquisição, que atribui uma pontuação a possíveis locais de observação com base em seu potencial para melhorar o processo de otimização. As funções de aquisição são menos custosas para avaliar e têm gradientes tratáveis analiticamente, permitindo o uso de otimizadores padrão com a finalidade de explorar regiões onde a função objetivo é incerta e onde espera-se valores máximos, (GARNETT, 2023).

Dentro desse contexto, o processo de otimização Bayesiana inicia codificando a função objetivo, projetada para considerar a diferenciabilidade, escalas de comprimento e estacionariedade da série representada por  $p(f)$  e utiliza a função de aquisição para realizar a escolha ótima das regiões exploradas. Na Figura 41 é ilustrado o processo de Otimização Bayesiana em uma função objetivo unidimensional (GARNETT, 2023).

A Figura 41-a ilustra um processo Bayesiana *apriori* em uma função objetivo unidimensional, projetada para considerar a diferenciabilidade, escalas de comprimento e estacionariedade da série. Com o processo *apriori* estabelecido, considera-se um conjunto de observações em algumas localizações  $x$ , dado que se trata de um processo estocástico representado por  $p(\phi|x)$ , em que  $\phi$  corresponde a distribuição dos valores da função. Assim chega-se aos valores correspondentes  $D = (x, y)$ , os quais são agregados em um conjunto de dados. A inferência Bayesiana utiliza essas observações para formar o processo posterior  $p(f|D)$  ilustrado na Figura 41-b e com o apoio da função de aquisição (Figura 41-c) uma vez que regiões em que o valor da função é conhecida a incerteza é baixa e a próxima região de escolha é o maior valor de incerteza da função de aquisição. Portanto, a *posteriori* da função objetivo se ajusta aos valores observados, com a média interpolando através dos dados e os intervalos estatísticos refletindo uma menor incerteza em torno das localizações observadas (GARNETT, 2023).

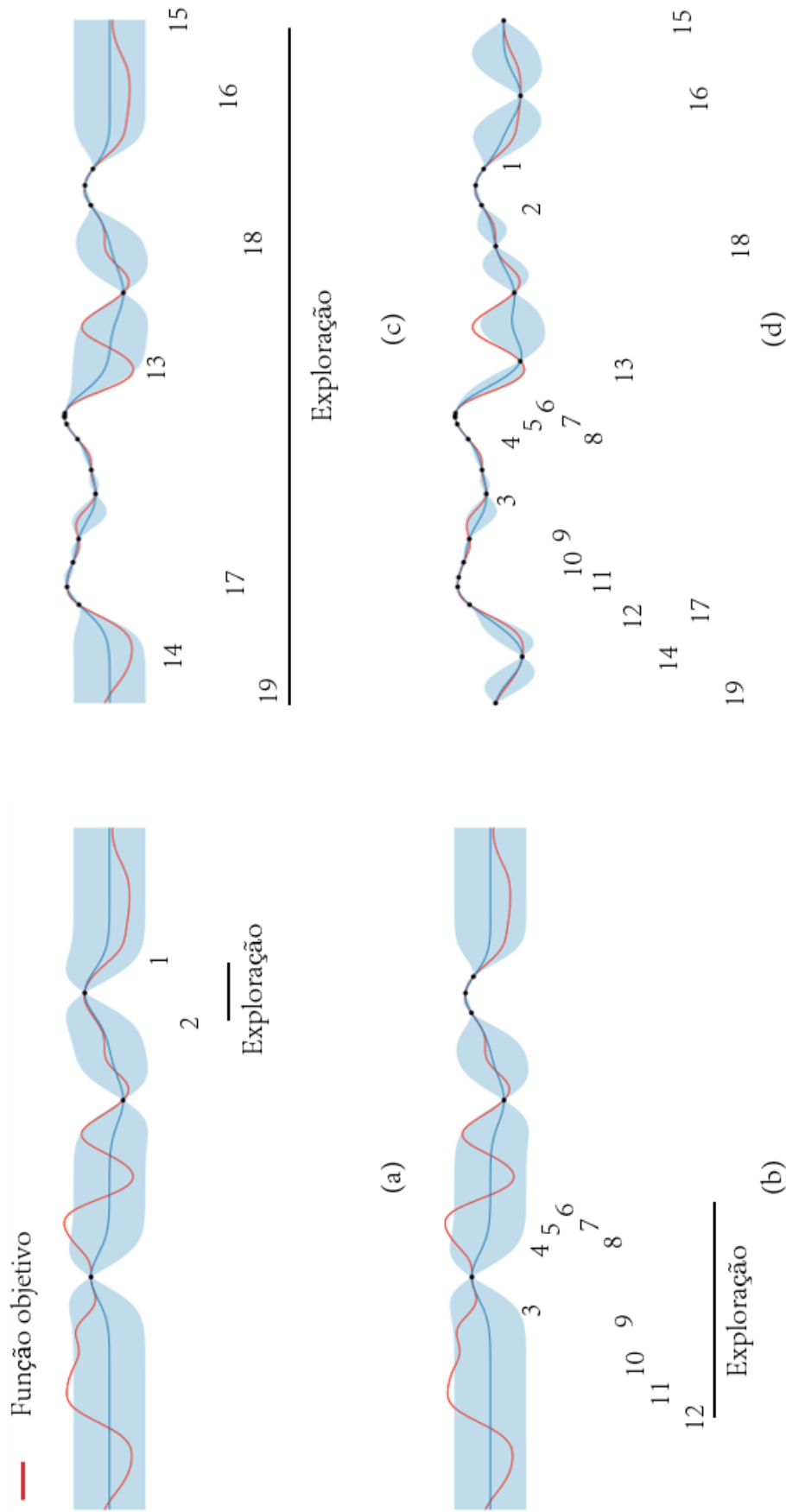
Figura 41 – Processo de otimização Bayesiana: (a) função objetivo a *apriori*; (b) função objetivo *posteriori*; (c) função de aquisição



Fonte: Adaptado de Vidhya (2020)

Na Figura 42 é ilustrado o processo iterativo de otimização Bayesiana, começando com uma incerteza inicial, devido a duas regiões conhecidas e progredindo com medições sucessivas. Realizou-se uma otimização no entorno da primeira e da segunda amostras conhecidas, reduzindo as incertezas nessa região (Figura 42 - a e b). Embora não ilustrada, com o apoio da função de aquisição é realiza a escolha de mais 7 pontos de otimização que acabaram por representar todo o espaço amostral, proporcionando um conhecimento prévio da função objetivo verdadeira (Figura 42-d) (GARNETT, 2023).

Figura 42 – Processo de otimização Bayesiana:



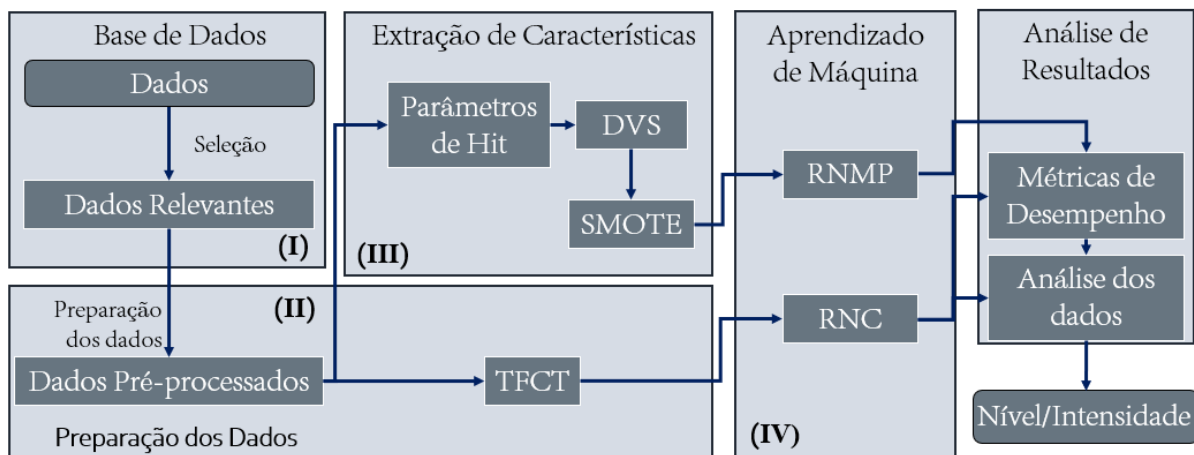
Fonte: Garnett (2023).

## 4 Materiais e Métodos

A metodologia adotada para o desenvolvimento dos modelos computacionais foi organizada e sistematizada conforme o método de Descoberta de Conhecimento em Bancos de Dados (DCBD). Este método envolve etapas de seleção, transformação, mineração de dados e a descoberta de conhecimento (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Destaca-se que as etapas de transformação e mineração de dados equivalem às etapas de extração de características e do modelo de aprendizado de máquina.

Com base na abordagem DCBD os modelos computacionais foram divididos nas etapas de criação da base de dados; preparação dos dados; extração de características; aprendizado de máquina; e análise de resultados. Na Figura 43 é ilustrado o fluxo metodológico do tratamento dos dados brutos à classificação em níveis de cavitação/falhas de rolamentos incluindo os métodos e procedimentos utilizados em cada etapa da metodologia. Nas seções subsequentes são tratados em detalhes esses métodos.

Figura 43 – Fluxograma metodológico dos modelos.



Fonte: Adaptado de Fayyad, Piatetsky-Shapiro e Smyth (1996).

- **Criação da Base de Dados:** Nesta etapa é apresentado o procedimento para a construção da base de dados de cavitação e de defeitos em rolamentos que subsidiou a criação do modelo de aprendizado de máquina. Após a aquisição dos dados é realizada a seleção conforme informações oriundas dos ensaios, tipos e posições de sensores de monitoramento e condições operativas no caso da bomba centrífuga. Com a conclusão dessa etapa obtém-se os dados para a parametrização do modelo.
- **Preparação dos Dados:** Esta etapa está relacionada à preparação dos dados obtidos da etapa anterior. Isso corresponde à ampliação da base de dados por meio

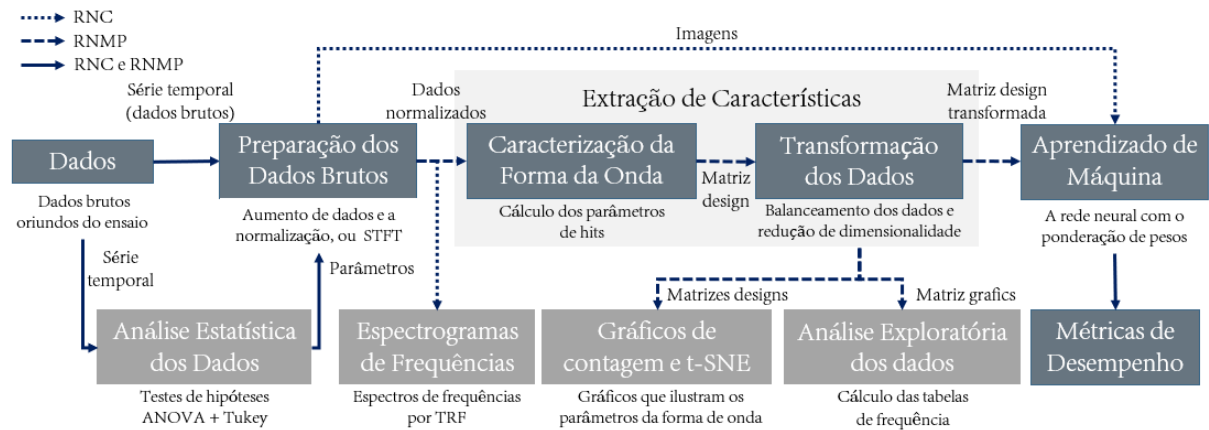
da criação de dados artificiais, na normalização dos dados e na correlação entre os dados e rótulos para cada condição operativa. No caso do modelo RNC é realizada a transformação do sinal do domínio do tempo para o domínio da frequência, por meio da Transformada de Fourier de Tempo Curto (TFCT) ou a Transformada de Wavellet (TW). Enquanto que no caso do modelo RNMP esse conjunto de dados pré-processados é utilizado na **extração de características** dos sinais.

- **Extração de Características:** A extração das características dos dados, foi aplicada somente para o modelo de aprendizado de máquina RNMP. Esse modelo é baseado na contagem dos *hits* e fornece valores que caracterizam a forma da onda no domínio do tempo. Além disso, utilizou-se uma abordagem transformação dos dados para uma base singular por meio da DVS e a aplicação de um balanceamento de dados utilizando a SMOTE. No caso do modelo RNC esta etapa é realizada pelo próprio modelo durante a fase de treinamento da rede, por meio dos *kernels* de convolução. Os dados de treinamento e teste são separados nessa fase, antes da criação dos dados sintéticos.
- **Aprendizado de Máquina:** Para a identificação de padrões e criação das classes, foram utilizados dois modelos de aprendizagem de máquina, a RNMP e a RNC. Ambas as redes recebem os dados transformados para realizar o cálculo de parâmetros. No caso da RNMP, utiliza-se os parâmetros baseados na contagem de *Hits*, enquanto que a RNC utiliza os espectros de frequência obtidos pela TFCT.
- **Análise de Resultados:** Esta etapa corresponde à análise dos resultados dos modelos de aprendizado de máquina baseados em redes neurais, por meio de métricas de desempenho, validação cruzada e análise estatística. Entre as métricas de desempenho estão acurácia, precisão, *F1-score*, matriz de confusão e curva ROC. Com relação às técnicas estatísticas utilizou-se uma análise exploratória dos dados.

## 4.1 Introdução aos Modelos: Uma Visão Geral

Ao aplicar a metodologia DCBD aos modelos RNMP e RNC, obtem-se o fluxograma conceitual ilustrado na Figura 44, destacando as informações de entrada, saída e processamento. O modelo RNMP é composto por 10 etapas, das quais seis são destinadas à classificação da intensidade de cavitação e/ou defeitos em rolamentos e duas à exploração dos dados. Vale destacar que neste modelo, a etapa de extração de características foi dividida em duas subetapas para facilitar a implementação dos algoritmos, a etapa de caracterização da forma da onda e a etapa de transformação dos dados. No caso do modelo RNC, existem somente 8 módulos, sendo quatro destinados ao diagnóstico. Cada módulo desse fluxo conceitual são abordados nas seções subsequentes.

Figura 44 – Fluxograma conceitual dos modelos RNMP e RNC.



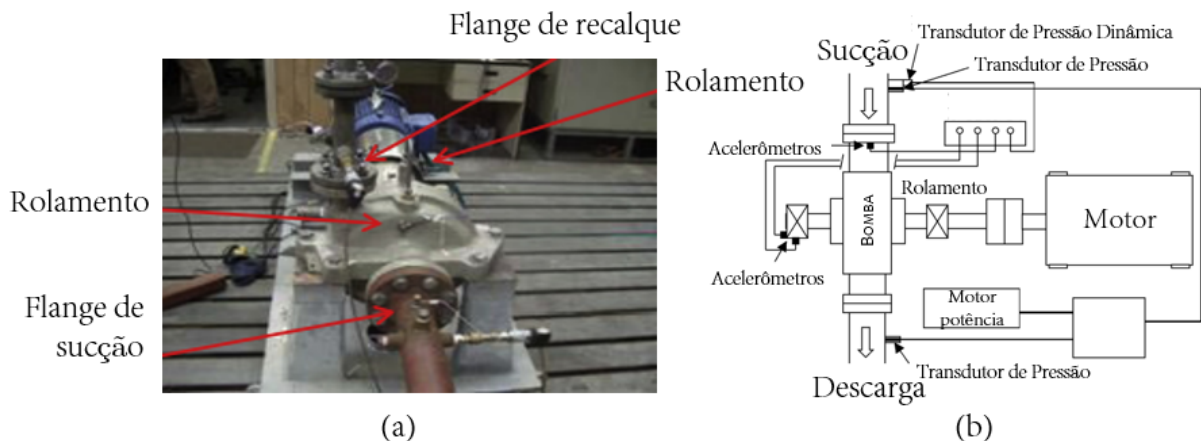
Fonte: Autor.

## 4.2 Base de Dados

### 4.2.1 Dados de Cavitação via Acelerômetros

A base de dados utilizada para a construção dos modelos de AM foi a *UWA System Health Lab Prognostics Data Library (SPDL)*, proveniente de um ensaio de cavitação em bombas centrífugas, com base na variável Net Position Suction Head (NPSH). Esse procedimento recomenda a indução de cavitação na sucção da bomba centrífuga por meio de manobras de abertura e fechamento das válvulas, além da aceleração da velocidade de rotação do rotor, o que altera o valor do NPSH e, conseqüentemente, a presença de cavitação. Na Figura 45 é ilustrada a bomba centrífuga do ensaio e o esquema com o posicionamento dos quatro acelerômetros nas regiões de sucção e descarga, e no rolamento próximo ao ventilador *FE* nos sentidos axial e transversal.

Figura 45 – Ensaio de cavitação: (a) bomba centrífuga; (b) Posição dos acelerômetros.



Fonte: Adaptado de Hodkiewicz e Pan (2004).

Os dados brutos provenientes dos ensaios são apresentados na Tabela 23, que inclui apenas as variáveis hidráulicas de interesse para esta pesquisa: vazão, rotação, NPSH e altura manométrica. Esta base de dados consiste em 20 arquivos .xls, cada um representando diferentes condições operativas. O fenômeno de cavitação ocorre quando o NPSH<sub>a</sub> é igual ao NPSH<sub>r</sub>, ou seja, quando  $NPSH_{\text{ration}} = 1$ . Nesses casos, ocorre uma redução superior a 3% na eficiência hidráulica. As demais condições podem apresentar escoamento sem o fenômeno de cavitação ou em uma situação de iminência, como no ensaio 3, teste 3, e no ensaio 2, teste 3, que apresentaram valores de 1,01 e 1,03, respectivamente.

Tabela 23 – Nomes dos arquivos e organização dos dados de ensaio.

| Arquivo<br>[–] | Ensaio<br>[–] | Teste<br>[–] | Vazão<br>[m <sup>3</sup> /h] | Rotação<br>[RPM] | NPSH <sub>a</sub><br>[mca] | NPSH <sub>ration</sub><br>[–] | NPSH <sub>r</sub><br>[mca] | Altura<br>[mca] |
|----------------|---------------|--------------|------------------------------|------------------|----------------------------|-------------------------------|----------------------------|-----------------|
| 01             | 3             | 0            | 161                          | 2,4              | 10,09                      | 2,57                          | 3,92                       | 124,89          |
| 02             | 3             | 1            | 161                          | 2,4              | 4,9                        | 1,25                          | 3,92                       | 124,80          |
| 03             | 3             | 2            | 161                          | 2,4              | 3,99                       | 1,02                          | 3,92                       | 123,94          |
| 04             | 3             | 3            | 161                          | 2,4              | 3,96                       | 1,01                          | 3,92                       | 121,97          |
| 05             | 3             | 4            | 161                          | 2,4              | 3,92                       | 1,00                          | 3,92                       | 119,40          |
| 06             | 4             | 0            | 193                          | 2,9              | 10,34                      | 2,72                          | 3,8                        | 114,25          |
| 07             | 4             | 1            | 193                          | 2,9              | 4,64                       | 1,22                          | 3,8                        | 114,61          |
| 08             | 4             | 2            | 193                          | 2,9              | 4,06                       | 1,07                          | 3,8                        | 114,08          |
| 09             | 4             | 3            | 193                          | 2,9              | 4                          | 1,05                          | 3,8                        | 112,94          |
| 10             | 4             | 4            | 193                          | 2,9              | 3,8                        | 1,00                          | 3,8                        | 109,33          |
| 11             | 2             | 0            | 108                          | 1,6              | 9,51                       | 4,32                          | 2,2                        | 134,39          |
| 12             | 2             | 1            | 108                          | 1,6              | 4,67                       | 2,12                          | 2,2                        | 134,13          |
| 13             | 2             | 2            | 108                          | 1,6              | 4,04                       | 1,84                          | 2,2                        | 134,05          |
| 14             | 2             | 3            | 108                          | 1,6              | 2,27                       | 1,03                          | 2,2                        | 132,02          |
| 15             | 2             | 4            | 108                          | 1,6              | 2,2                        | 1,00                          | 2,2                        | 120,08          |
| 16             | 1             | 1            | 55                           | 0,8              | 4,5                        | 4,09                          | 1,1                        | 138,35          |
| 17             | 1             | 2            | 55                           | 0,8              | 3,38                       | 3,07                          | 1,1                        | 138,36          |
| 18             | 1             | 3            | 55                           | 0,8              | 1,55                       | 1,41                          | 1,1                        | 138,48          |
| 19             | 1             | 4            | 55                           | 0,8              | 1,1                        | 1,00                          | 1,1                        | 131,62          |
| 20             | 1             | 0            | 55                           | 0,8              | 9,21                       | 8,38                          | 1,1                        | 138,30          |

Fonte: Adaptado de [Hodkiewicz e Pan \(2004\)](#).

Para esta pesquisa, foram consideradas apenas as condições de cavitação, definidas como  $NPSH_{\text{ration}} = 1$  e as condições sem cavitação, definidas como  $NPSH_{\text{ratio}} \neq 1$ . Com base nesse critério, os dados dos ensaios foram reorganizados conforme descrito a seguir:

- Sem cavitação (*OFFCAV*): Esta condição é considerada como a operação normal da bomba, na qual entende-se essa condição como condição normal de operação, na qual  $NPSH_a > NPSH_r$ . Isso indica a ausência de formação de vapor, resultando em um número de cavitação ( $\sigma$ ) igual a zero e, portanto, sem perda de eficiência.
- Com Cavitação (*ONCAV*): Essa condição implica na ocorrência do fenômeno de cavitação, na qual  $NPSH_a = NPSH_r$ , indicando a energia de aspiração na entrada

da bomba é igual as perdas de carga da entrada da bomba até a entrada da pá, indicando a presença de vapor e perda de eficiência.

Em cada ensaio da Tabela 23, registram-se informações temporais de vibração dos acelerômetros, as quais são utilizadas para a criação dos modelos de aprendizado de máquina. Esses ensaios foram organizados e renomeados para facilitar a leitura e a identificação das condições operativas abordadas. A nomenclatura adotada foi baseada na condição de vazão e no número do teste de cada ensaio, refletindo o valor de  $NPSH_{\text{ration}}$ . Na Tabela 24, apresenta-se a relação dos arquivos da base de dados e a nomenclatura proposta. Por exemplo, o Ensaio 01 é representado pela nomenclatura *OFFCAV – E0.Q161*, enquanto o arquivo 10 é identificado como *ONCAV – E4.Q193*.

Tabela 24 – Organização dos dados de ensaio.

| Classe | Ensaio | Vazão [ $m^3/h$ ] |           |           |           | N. de amostras |
|--------|--------|-------------------|-----------|-----------|-----------|----------------|
|        |        | Q55               | Q108      | Q161      | Q193      | Qndt.          |
| OFFCAV | E0     | Ensaio 16         | Ensaio 11 | Ensaio 01 | Ensaio 06 | 5.241.600      |
|        | E1     | Ensaio 17         | Ensaio 12 | Ensaio 02 | Ensaio 07 |                |
|        | E2     | Ensaio 18         | Ensaio 13 | Ensaio 03 | Ensaio 08 |                |
|        | E3     | Ensaio 19         | Ensaio 14 | Ensaio 04 | Ensaio 09 |                |
| ONCAV  | E4     | Ensaio 20         | Ensaio 15 | Ensaio 05 | Ensaio 10 | 1.310.400      |

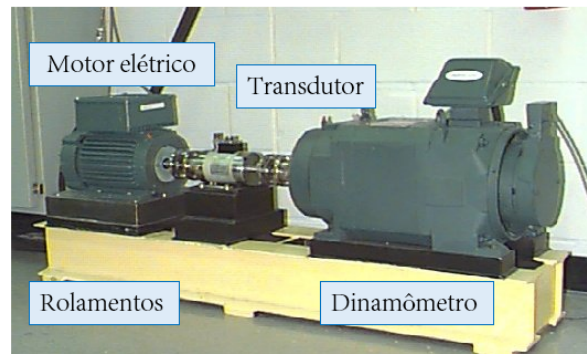
Fonte: Adaptado de [Hodkiewicz e Pan \(2004\)](#)

#### 4.2.2 Dados de Rolamento via Acelerômetros

A base de dados de falhas mecânicas em rolamentos é composta por sinais temporais que capturam o comportamento operacional, refletindo o estado estrutural dos rolamentos localizados próximos ao ventilador (*FE*) e ao motor (*DE*) na bancada experimental do motor elétrico. Esta base de dados, disponibilizada pela *Case Western Reserve University* (CWRU), está em formato de arquivos Matlab. A nomenclatura desses arquivos indica a posição do acelerômetro, a velocidade de rotação e o tempo de ensaio, conforme descrito em ([ZHANG et al., 2020](#)). Os dados foram coletados por acelerômetros com uma frequência de amostragem de  $48kHz$ , em um tempo de ensaio de 5s para a condição de 0HP e 10s para as demais condições de carga. Além disso, esses sinais estão rotulados, ou seja a integridade estrutural do rolamento está associada ao sinal, permitindo o desenvolvimento de métodos de diagnóstico com maior confiabilidade.

Na Figura 46, é ilustrada a bancada de teste, que é composta pelo motor, transdutor de torque, dinamômetro e pelos dois rolamentos.

Figura 46 – Bancada experimental de falhas em rolamentos



Fonte: Adaptado de [Center West Reserve University \(2018\)](#).

As falhas no rolamento são criadas por meio de descargas elétricas nas esferas, nas pistas de rodagem interna e externa com profundidades de  $7\mu''$ ,  $14\mu''$  e  $21\mu''$ . Ao correlacionar as falhas com as intensidades, chega-se a nove condições estruturais de falhas e uma condição saudável, totalizando 10 classes para os modelos de aprendizado de máquina realizarem a predição. Na Tabela 25 é apresentada a organização da base de dados, de acordo com o tipo de intensidade de falha em quatro carga de potência.

Tabela 25 – Nomes dos arquivos e organização dos dados para a condição de falha.

| Diâmetro da Falha | Torque no eixo | Esfera  | Pista Interna | Pista Externa |
|-------------------|----------------|---------|---------------|---------------|
| 0,000             | 0HP            |         | NRM – 0       |               |
|                   | 1HP            |         | NRM – 1       |               |
|                   | 2HP            |         | NRM – 2       |               |
|                   | 3HP            |         | NRM – 3       |               |
| 0,007             | 0HP            | B07 – 0 | IR07 – 0      | ER07 – 0      |
|                   | 1HP            | B07 – 1 | IR07 – 1      | ER07 – 1      |
|                   | 2HP            | B07 – 2 | IR07 – 2      | ER07 – 2      |
|                   | 3HP            | B07 – 3 | IR07 – 3      | ER07 – 3      |
| 0,014             | 0HP            | B14 – 0 | IR14 – 0      | ER14 – 0      |
|                   | 1HP            | B14 – 1 | IR14 – 1      | ER14 – 1      |
|                   | 2HP            | B14 – 2 | IR14 – 2      | ER14 – 2      |
|                   | 3HP            | B14 – 3 | IR14 – 3      | ER14 – 3      |
| 0,021             | 0HP            | B21 – 0 | IR21 – 0      | ER21 – 0      |
|                   | 1HP            | B21 – 1 | IR21 – 1      | ER21 – 1      |
|                   | 2HP            | B21 – 2 | IR21 – 2      | ER21 – 2      |
|                   | 3HP            | B21 – 3 | IR21 – 3      | ER21 – 3      |

Fonte: Adaptado de [Center West Reserve University \(2018\)](#)

Em cada arquivo da Tabela 25 existem os sinais dos rolamentos “DE” e “FE” e os metadados referentes aos ensaios. Nos Quadros 6 e 7 são apresentados quatro metadados da série, como: o nome da variável que está armazenado, o conteúdo do sinal, o número de amostras, a velocidade de rotação registrada e o tempo de ensaio.

Quadro 6 – Metadados dos arquivos de ensaio de 1730 rpm

| Classes | Arquivo      | Amostras [-] | Rotação [rpm] | Tempo [s] |
|---------|--------------|--------------|---------------|-----------|
| NORM    | X125_DE_time | 488.545      | 1.721         | 10.18     |
| B07     | X125_DE_time | 488.545      | 1.721         | 10.18     |
| B14     | X192_DE_time | 486.804      | 1.725         | 10,14     |
| B21     | X229_DE_time | 486.804      | 1.729         | 10,14     |
| IR07    | X112_DE_time | 485.643      | 1.721         | 10,12     |
| IR14    | X177_DE_time | 485.063      | 1.726         | 10,11     |
| IR21    | X217_DE_time | 489.125      | 1.727         | 10,19     |
| ER07    | X164_DE_time | 484.483      | 1.723         | 10,09     |
| ER14    | X204_DE_time | 488.545      | 1.723         | 10,17     |
| ER21    | X265_DE_time | 486.224      | 1.718         | 10,13     |

Fonte: Adaptado de [Center West Reserve University \(2018\)](#).

Quadro 7 – Meta Dados dos arquivos dos ensaios de 1750, 1772 e 1796 rpm

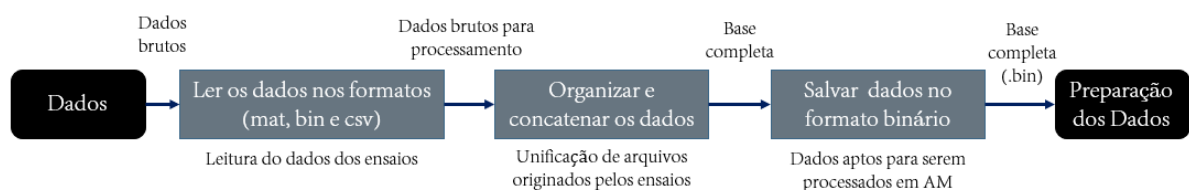
|             | Arquivo [_DE_time] |      |      | Rotação [rpm] |       |       | Tempo [s] |       |      |
|-------------|--------------------|------|------|---------------|-------|-------|-----------|-------|------|
|             | 1750               | 1772 | 1796 | 1750          | 1772  | 1796  | 1750      | 1772  | 1796 |
| <i>NRM</i>  | X124               | X123 | X122 | 1.747         | 1.772 | 1.796 | 10,14     | 10.15 | 5,10 |
| <i>B07</i>  | X124               | X123 | X122 | 1.747         | 1.772 | 1.796 | 10,14     | 10.15 | 5,10 |
| <i>B14</i>  | X191               | X123 | X189 | 1.748         | 1.772 | 1.797 | 10,15     | 10,12 | 5,19 |
| <i>B21</i>  | X228               | X190 | X226 | 1.754         | 1.774 | 1.797 | 10,15     | 10,14 | 5,08 |
| <i>IR07</i> | X111               | X227 | X109 | 1.748         | 1.772 | 1.796 | 10,12     | 10,13 | 5,08 |
| <i>IR14</i> | X176               | X110 | X173 | 1.751         | 1.772 | 1.796 | 10,17     | 7,95  | 1,73 |
| <i>IR21</i> | X215               | X175 | X213 | 1.752         | 1.774 | 1.796 | 10,23     | 10,10 | 5,09 |
| <i>ER07</i> | X163               | X214 | X161 | 1.749         | 1.772 | 1.796 | 10,06     | 10,05 | 2,71 |
| <i>ER14</i> | X203               | X162 | X201 | 1.749         | 1.772 | 1.796 | 10,14     | 10,09 | 5,11 |
| <i>ER21</i> | X264               | X263 | X262 | 1.746         | 1.771 | 1.796 | 10,14     | 10,13 | 2,71 |

Fonte: Adaptado de [Center West Reserve University \(2018\)](#).

### 4.3 Dados

Esta fase consiste no *download* dos dados brutos conforme o formato disponibilizado no ensaio e então processados e consolidados em uma única série temporal, que é armazenada em um arquivo com extensão binária, conforme ilustrado na Figura 47 o fluxograma conceitual dessa etapa.

Figura 47 – Fluxograma conceitual da etapa Base de Dados dos modelos RNMP e RNC.

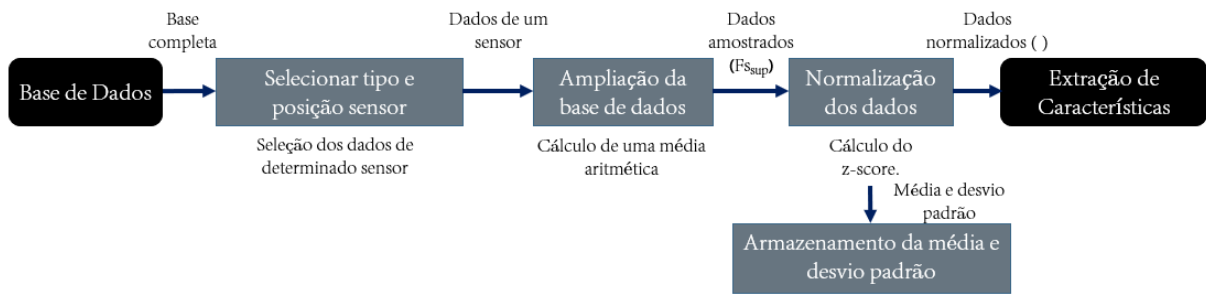


## 4.4 Preparação dos Dados

### 4.4.1 Modelo RNMP

Essa etapa envolve o aumento do volume de dados da série, visando aprimorar a precisão no cálculo das derivadas para identificar os valores máximos e mínimos. Além disso, inclui-se a normalização desse conjunto de dados utilizando o cálculo do *z-score*. Esse processo é composto por três etapas, conforme apresentado na Figura 48.

Figura 48 – Fluxograma conceitual da preparação dos dados.



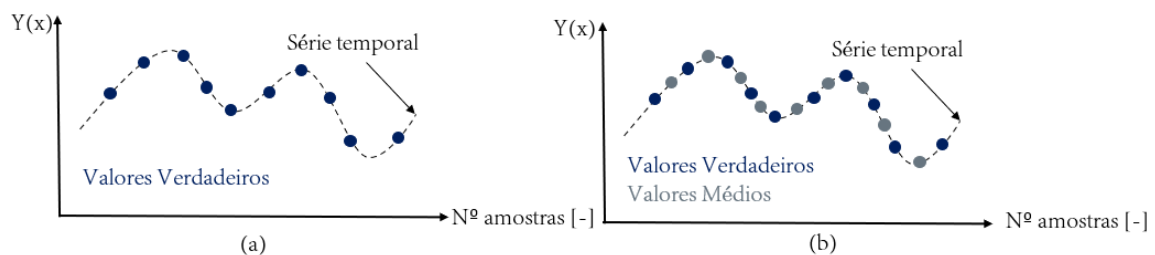
Fonte: Autor.

A primeira etapa corresponde a escolha do tipo e posição do sensor utilizado como fonte de dados. A etapa seguinte envolve o aumento do volume de dados da série temporal, conhecida como *data augmentation*. Este aumento permite uma maior precisão no cálculo das derivadas que são essenciais para identificar os valores máximos e mínimos da série, que por sua vez é a essência da técnica de contagem de *hits*. A Equação 4.1 apresenta a regra utilizada para esse aumento de dados.

$$Y_{i+\frac{1}{2}} = (Y_i + Y_{i+1})/2 \quad (4.1)$$

Na Figura 49-(a) são ilustradas um conjunto de amostras que forma uma série temporal, que após aplicar a Equação 4.1, o conjunto passa a possuir um volume maior de dados e consegue representar a posição de um pico ou vale com maior precisão, conforme também ilustrada na Figura 49-(b).

Figura 49 – Ampliação da base de dados com amostras sintéticas.



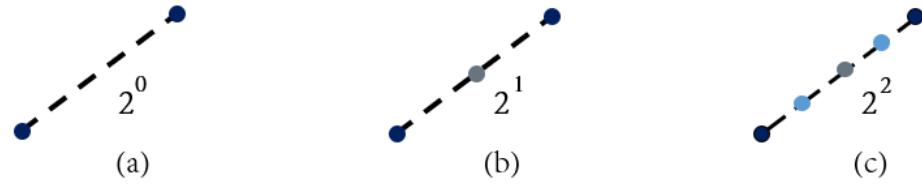
Fonte: Autor.

A Equação 4.2 representa o cálculo da frequência de aquisição sintética, utilizada para converter o conjunto de amostras em uma série temporal. Onde o  $FS$  é a frequência de aquisição do sistema físico,  $FS_{synt}$  é a frequência sintética e  $n$  o número de vezes que é criado um novo ponto médio.

$$FS_{synt} = (2^n) \cdot FS \quad (4.2)$$

Na Figura 50 são ilustrados três condições hipotéticas,  $n = 0$ ,  $n = 1$  e  $n = 2$ , que viabilizam o aumento exponencial no volume dos dados.

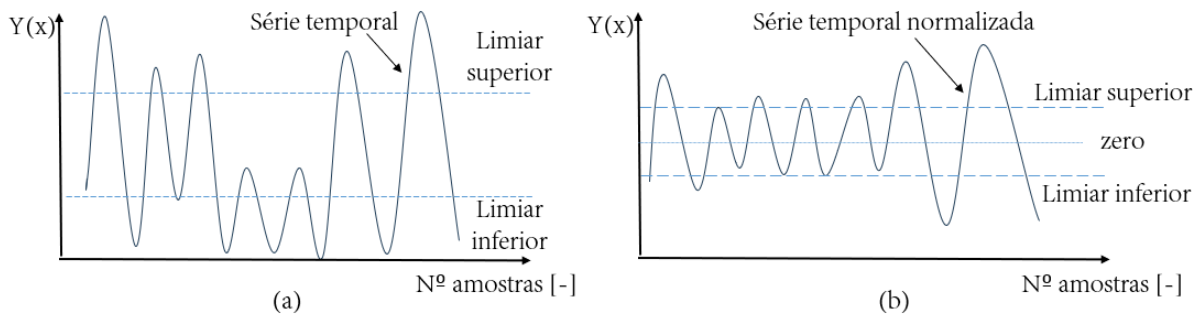
Figura 50 – Ampliação da base de dados com amostras sintéticas.



Fonte: Autor.

Após o aumento do volume de dados, é necessário normalizá-los para garantir que as escalas das características apresentem pesos equivalentes, reduzir o impacto de *outliers* e principalmente garantir que a média da série temporal encontra-se dentro do valor de limiar em todas as condições operativas ou de falha, conforme ilustrado na Figura 51.

Figura 51 – Impacto do valor limiar: (a) Dados não normalizados; (b) Dados normalizados.



Fonte: Adaptado de [Center West Reserve University \(2018\)](#).

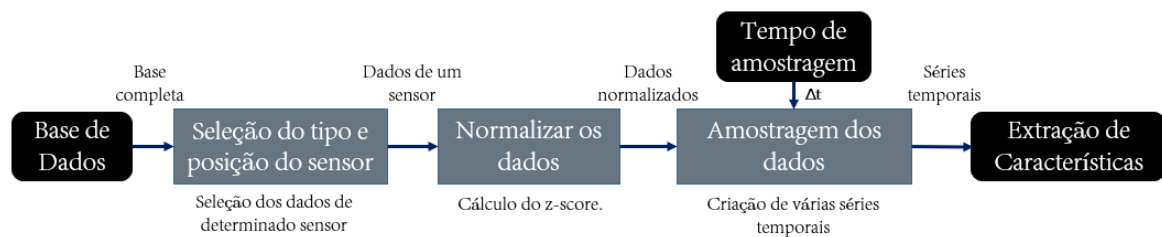
O cálculo do *z-score* para normalizar o conjunto de dados (Equação 4.3), é uma medida estatística que expressa o desvio de uma amostra em relação à média da distribuição dos dados. A Equação 4.3 trata-se desse cálculo em que  $\mu$  é a média e  $\sigma$  corresponde ao desvio padrão.

$$Z_{score} = \frac{Y(x_i) - \mu}{\sigma} \quad (4.3)$$

### 4.4.2 Modelo RNC

A preparação dos dados envolve a normalização por meio do cálculo do *z-score* e a subamostragem dos dados para criar múltiplas séries temporais de menor duração. Essa abordagem é necessária porque a técnica TFCT gera um único espectro de frequência (imagem) para cada série temporal. Dessa forma, um ensaio sob uma determinada condição operativa pode produzir vários espectros com o mesmo rótulo e sob as mesmas condições de teste, aumentando a robustez e a diversidade do conjunto de dados para análise. Na Figura 52, é ilustrado o fluxograma conceitual dessa etapa.

Figura 52 – Fluxograma conceitual da preparação dos dados do modelo RNC.



Fonte: Autor.

A partir da base de dados, seleciona-se o conjunto de dados com o qual se deseja trabalhar e realiza-se a especificação do sensor. Em uma base de dados com inúmeros sensores, essa seleção é crucial para a lógica dos algoritmos. Além disso, é essencial que o conjunto de dados inclua informações sobre a frequência de aquisição. Também é importante normalizar a série temporal para evitar erros no espectrograma ao convertê-lo em uma imagem com escala de intensidade de 0 a 255 pixels.

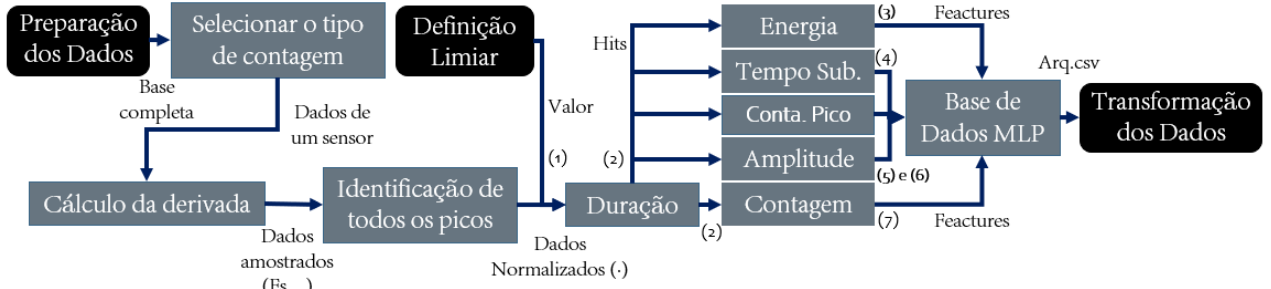
Para a amostragem da série, utiliza-se a TRF para calcular os valores de frequência. O tamanho da janela retangular envolve uma análise em relação aos valores de frequência, determinando o tamanho da série que não altera os valores de frequência do sinal.

## 4.5 Extração de Características

### 4.5.1 Modelo RNMP

Nesta etapa, ocorre a caracterização da forma da onda por meio da técnica de contagem de *hits*, conforme descrito na Seção 3.2.1. Os parâmetros considerados abrangem sete dos dez apresentados, incluindo o valor limiar, a duração, o tempo de subida, o número do pico máximo, a quantidade de picos, a amplitude e a energia. A partir dos dados normalizados, é gerada a matriz de *design*, fundamental para as etapas subsequentes na análise de aprendizado de máquina. Um fluxograma conceitual correspondente a esta etapa é apresentado na Figura 53, fornecendo uma representação visual do processo e auxiliando na compreensão das operações envolvidas na caracterização da forma da onda.

Figura 53 – Fluxograma conceitual da extração de características.



Fonte: Autor.

Em que: (1) valor do limiar; (2) tempo de duração do *hit*; (3) energia do *hit*; (4) tempo para atingir a amplitude máxima; (5) número de picos para atingir a amplitude máxima; (6) amplitude máxima no *hit* e (7) número de picos até atingir a amplitude no *hit*.

A etapa de extração das características inicia com a seleção do tipo de contagem, ou seja, a contagem de pico ou a contagem de vale. A partir dessa informação é realizado o cálculo da derivada por meio das diferenças finitas progressivas conforme a Equação 4.4 e a sua identificação conforme a Equação 4.5.

$$Z'(i) \approx \frac{Z(i+h) - Z(i)}{h} \quad (4.4)$$

Em que  $h$  representa o tamanho do passo. Por se tratar de um conjunto discreto o  $h$  é igual a uma unidade.

$$\begin{cases} Z'(i) > 0 & \text{e} & Z'(i+1) < 0 & \text{corresponde a um pico} \\ Z'(i) < 0 & \text{e} & Z'(i+1) > 0 & \text{corresponde a um vale} \end{cases} \quad (4.5)$$

O cálculo dos parâmetros de *hits* tem início com a definição do valor limiar. Essa variável exerce uma influência direta sobre todos os outros parâmetros de contagem. Valores inadequados podem resultar em cenários onde apenas alguns *hits* são detectados ou em situações em que milhares deles são registrados. Após a definição do limiar é possível identificar o número e as durações dos *hits* que ocorrem no sinal. A partir do parâmetro de duração, são calculados os parâmetros de tempo de subida, energia, amplitude, contagem e contagem de pico para cada *hit*, conforme ilustrado na Figura 53.

Cada *hit* possui um rótulo identificado durante as etapas de ensaio e/ou na construção da base de dados. Com base nesses dados anteriores e nos parâmetros de *hits*, foram elaboradas planilhas que contêm informações sobre a forma da onda (características do sinal). Essas planilhas, denominadas por dados organizados, são então disponibilizadas para o modelo RNMP. O Quadro 8 exemplifica o formato de saída fornecido por este mo-

delo de extração de características a RNMP. Cada coluna representa uma característica (*feature*) ou rótulo (*label*), enquanto as linhas representam exemplos individuais de *hits*.

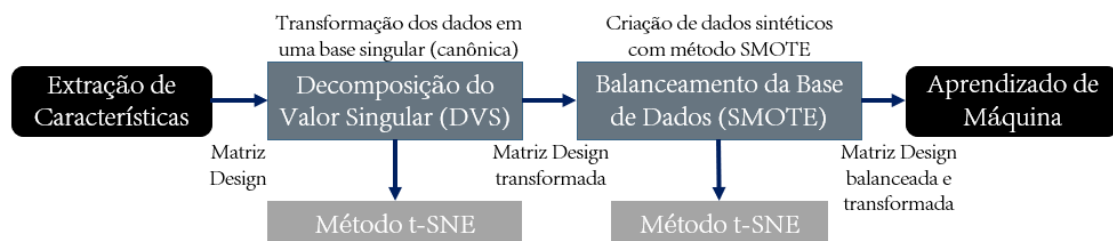
Quadro 8 – Dados organizados de cavitação

| N        | Duração<br>[s]  | Tempo de<br>Subida [s] | Amplitude<br>[dB] | Contagem<br>[-]  | Contagem<br>à Pico [-] | Energia<br>[-]  | Rótulo<br>[-]  |
|----------|-----------------|------------------------|-------------------|------------------|------------------------|-----------------|----------------|
| 1        | 0,0025          | 0,0015                 | 0.4786            | 7                | 5                      | 0,0005          | <i>NRM</i>     |
| ...      | ...             | ...                    | ...               | ...              | ...                    | ...             | ...            |
| <i>N</i> | <i>DuracaoN</i> | <i>TempoN</i>          | <i>AmplitudeN</i> | <i>ContagemN</i> | <i>ContagemN</i>       | <i>EnergiaN</i> | <i>RotuloN</i> |

Fonte: Autor.

Após a criação dos dados estruturados (matriz *design*), é empregada a técnica SMOTE para a geração de dados sintéticos e o método DVS para a redução de dimensionalidade. O processo inicia-se com a aplicação da técnica SMOTE para aumentar a quantidade de dados, especialmente em situações de desequilíbrio de classes, visando aprimorar a capacidade do modelo de aprendizado de máquina em lidar com diferentes cenários. Posteriormente, realiza-se a redução da dimensionalidade por meio do método DVS, que possibilita representar os dados em um espaço de menor dimensão, preservando as características fundamentais do conjunto de dados original. Ao término desta etapa, os dados são processados e estão prontos para serem utilizados na construção e treinamento do modelo de aprendizado de máquina. Na Figura 54 é ilustrado o fluxograma conceitual desta etapa de transformação dos dados, incluindo os métodos visualização por *clusters*.

Figura 54 – Fluxograma conceitual da transformação e balanceamento dos dados.



Fonte: Autor.

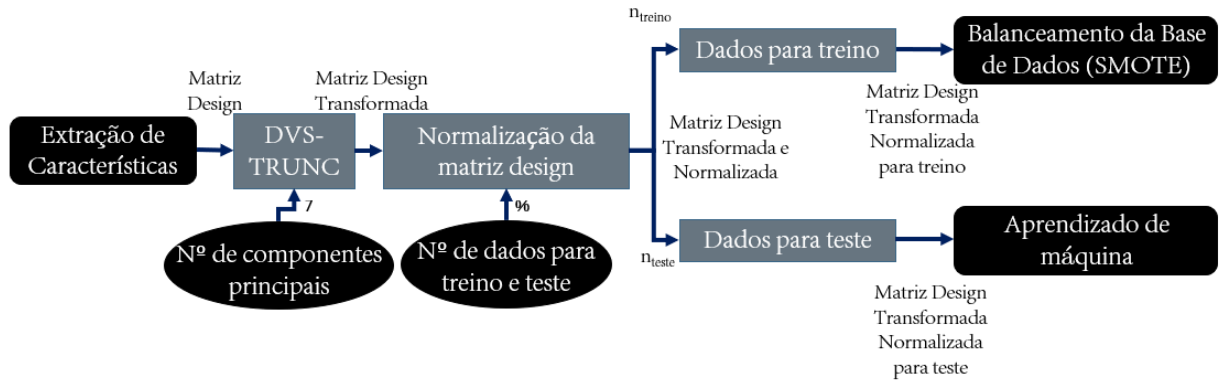
Ressalta-se que ambos os métodos alteram a matriz de *design* conforme a proposta dos métodos. Ademais, como resultado, ambos fornecem gráficos de clusters (t-SNE) para ilustrar a classificação após a aplicação de cada método.

- Método de Decomposição do Valor Singular (DVS):

O método de transformação de base utilizado nesta etapa de processamento foi o método de decomposição do valor singular truncado com seis componentes principais. A transformação da matriz *design* em duas matrizes unitárias e uma matriz diagonal singular permite reconstruir a matriz *design* em estruturas de dados, embora sem informação da física (duração negativa), mas sem ruídos e facilmente separáveis. Esse

modelo deve receber como entrada a matriz design e produz como saída a matriz design transformada tanto para a fase de treinamento quanto para a fase de teste, sendo que os dados de teste são disponibilizados para o modelo de Aprendizado de Máquina, conforme ilustrado na Figura 55.

Figura 55 – Fluxograma conceitual da transformação do dados para treinamento e teste.



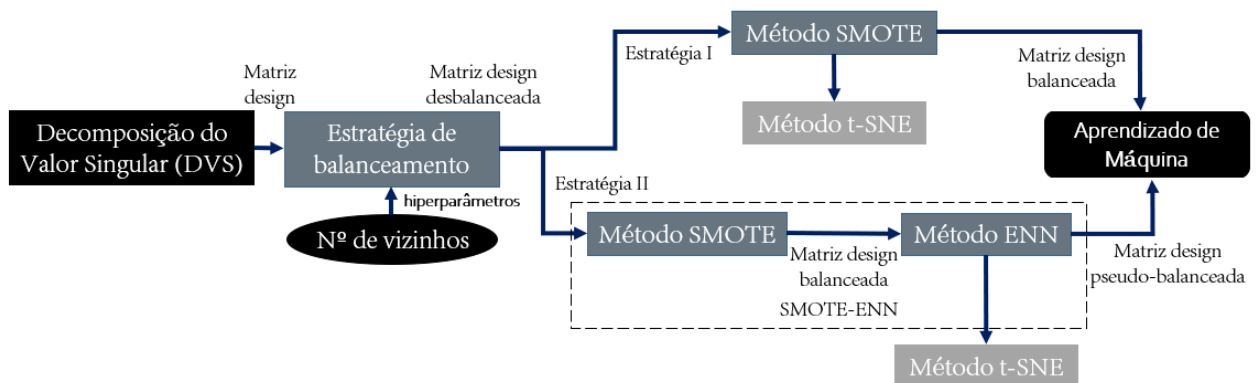
Fonte: Autor.

- Método de Balanceamento da Base de Dados (SMOTE):

O método de contagem de *hits* inevitavelmente causa um desequilíbrio na base de dados, já que não controla o número de ocorrências dos *hits*. Como resultado, tornou-se essencial empregar um método visando de equilibrar a base de dados, também conhecido na área da computação por método de balanceamento.

O método de balanceamento SMOTE foi aplicado a toda a base de dados com uma ordem  $K = 5$  e, também em combinação com o método *Edited Nearest Neighbors* (ENN) de ordem  $k = 3$ , resultando na abordagem SMOTE-ENN, conforme ilustrado na Figura 56. Existe a possibilidade de aplicar variações do método SMOTE, como o método ADASYN.

Figura 56 – Fluxograma conceitual de balanceamento dados.

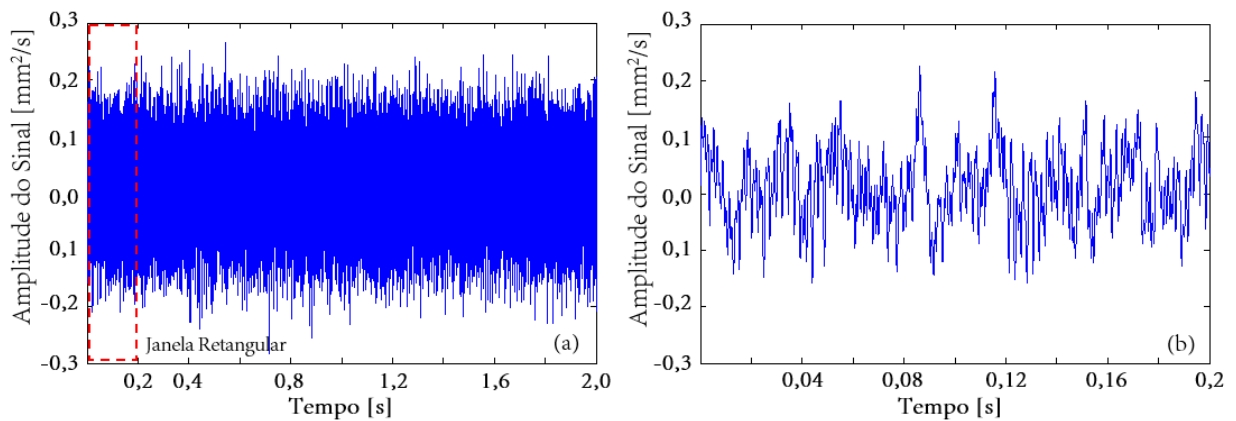


Fonte: Autor.

### 4.5.2 Modelo RNC

Na fase de preparação dos dados para o modelo RNC, foi empregada a técnica de processamento de sinais, como a TFCT, gerando como resultado um espectro de frequência por tempo. Esse método realiza a transformação dos dados de uma determinada classe em uma única imagem. Portanto, cada conjunto de dados oriundos de uma classe pode ser sub-dividido em várias partes iguais, com o objetivo de aumentar o número de imagens que foram fornecidas ao modelo RNC e, para isso, utilizou uma janela retangular que não alterasse os valores de frequência e que produzisse uma quantidade adequada para o modelo AP, conforme tratado na seção de resultados. Na Figura 57 é ilustrada a aplicação da janela retangular de tamanho de 0,2s sobre uma sinal de 2s.

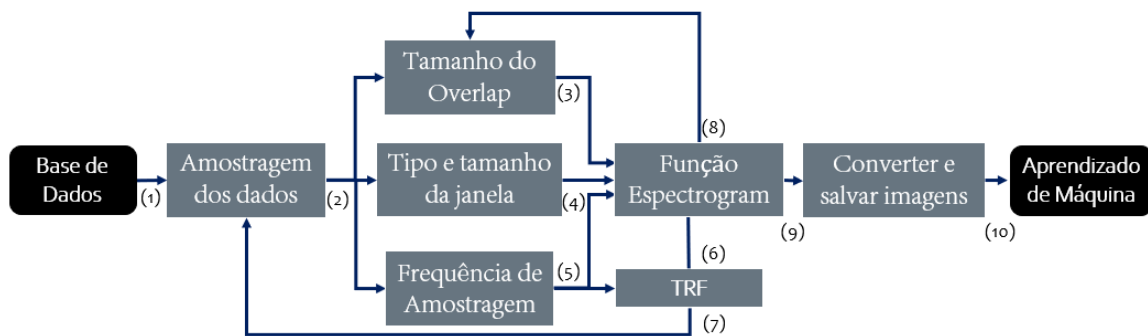
Figura 57 – Aplicação da janela retangular: (a) sinal bruto; (b) sinal janelado.



Fonte: Autor.

A parametrização da TFCT corresponde, na amostragem dos dados, à escolha do tipo e tamanho da janela, às sobreposições das janelas (*overlap*) e à frequência de amostragem. Na Figura 58, é ilustrado o fluxo para o cálculo da TFCT.

Figura 58 – Fluxograma conceitual da Extração de Características: TFCT.

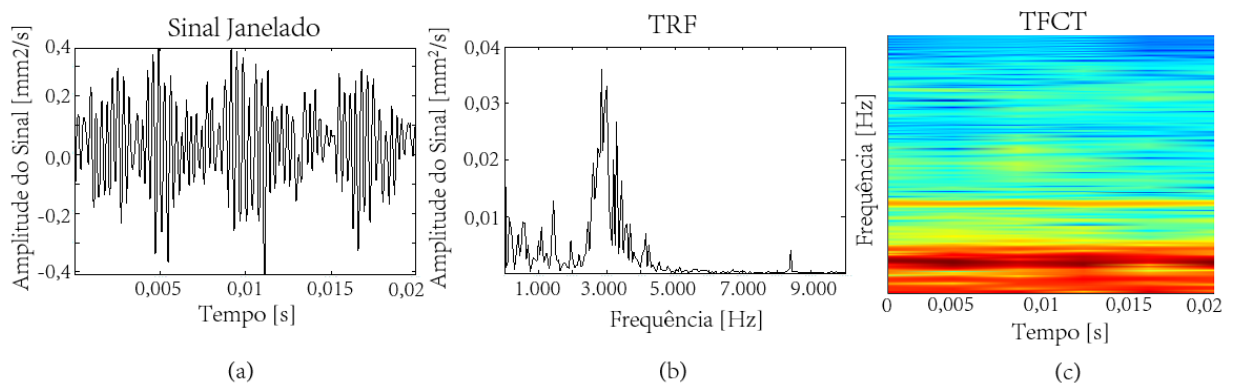


Fonte: Autor.

Em que correspondem aos (1) dados de ensaio; (2) dados selecionados a partir da janela retangular; (3) porcentagem de sobreposição entre as janelas (*overlap*); (4) tipo de tamanho da janela para o cálculo das Transformadas de Fourier; (5) frequência de amostragem de dados; (6) Transformada de Fourier (7) novo valor de tamanho da janela retangular (8) novo valor de *overlap*; (9) espectrogramas do sinal utilizando a função *Espectrogram* da biblioteca *Scipy*; (10) imagens dos espectrogramas.

Esses valores foram estabelecidos segundo testes, em que buscou-se o maior número de espectrogramas sem alterar significativamente os valores de frequência para a condição normal de operação, resultando na seguinte composição de dados: janela retangular de tamanho 0,02s sem *overlap*; frequência de amostragem: 48kHz; tipo e tamanho da janela: *Hamming* com 192 amostras e 75% de *overlap*. Na Figura 59, são ilustradas as etapas do processo de geração de imagens utilizando a abordagem TFCT para o modelo RNC.

Figura 59 – Etapas do processo TFCT: (a) sinal no domínio do tempo; (b) sinal no domínio da frequência; (c) sinal no domínio tempo-frequência.



Fonte: Autor.

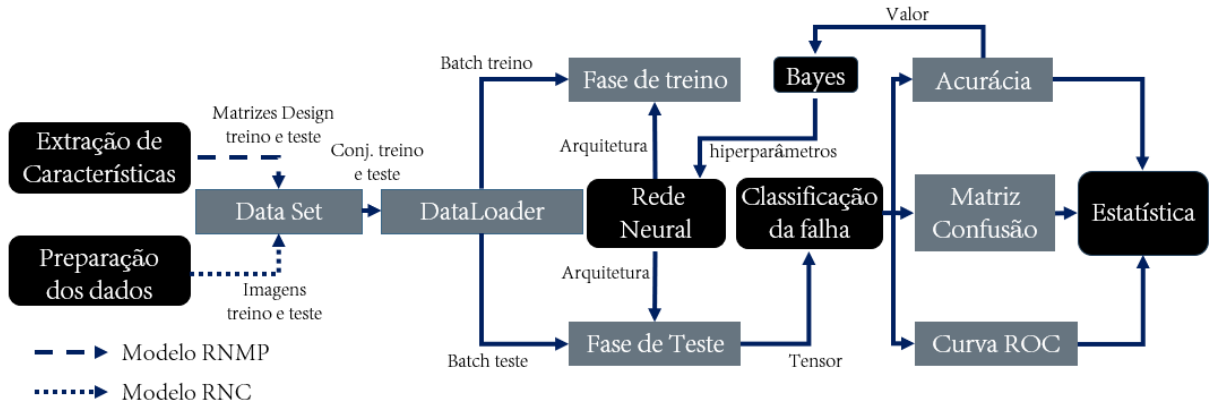
Na Figura 59-(a), é apresentada a etapa de amostragem dos dados com a utilização de uma janela retangular de tamanho 0,02 segundos. Nesta representação, é possível visualizar a amplitude do sinal em função do tempo. Na Figura 59-(b) é ilustrado o espectro de frequência neste sinal janelado, cujos eixos correspondem à amplitude média do sinal e frequência. Por fim, na Figura 59-(c) é observado o espectro de frequência por tempo, em que a amplitude do sinal é a escala de cores, cuja amplitude deve ser convertida na escala de imagem, ou seja, de 0 a 255.

## 4.6 Aprendizado de Máquina

Os modelos de AM adotados nesta aplicação foram os RNMP e RNC, cuja parametrização inicial está conforme Seção 2.3 deste documento e biblioteca do *PyTorch*. Ambas as modelagens são subdivididas em três fases: pré-processamento (envolve a preparação dos dados e a escolha dos hiperparâmetros), processamento (cálculo dos pesos e biases)

e pós-processamento (análise dos resultados). É importante ressaltar que este modelo requer uma adequação da matriz *design* para ser processada pelos algoritmos e *Frameworks* da biblioteca. Na Figura 60 é ilustrado o fluxograma conceitual da etapa de aprendizado de máquina do modelo RNMP.

Figura 60 – Fluxograma conceitual de AM do modelo RNMP e RNC.



Fonte: Autor.

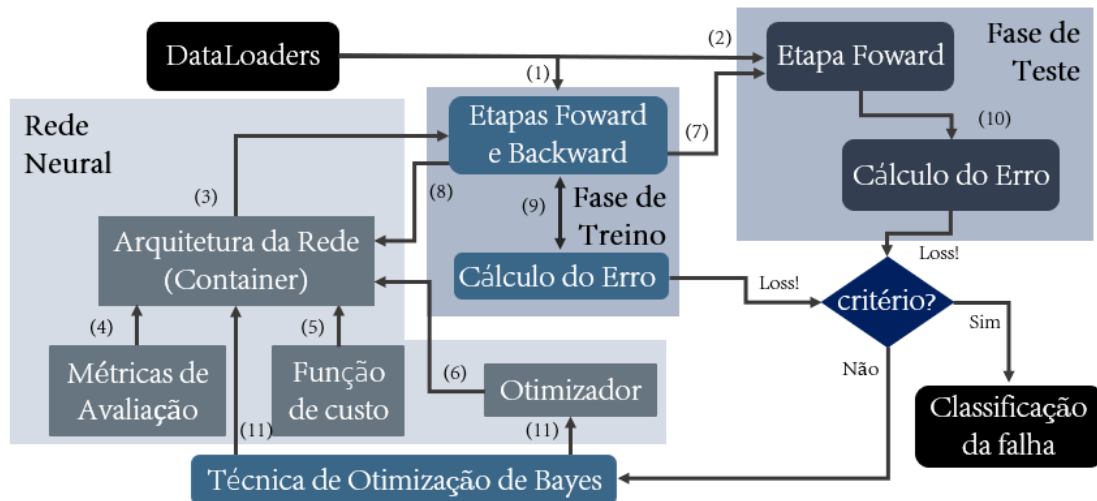
Essa fase inicia com a criação do *dataset* que consiste em selecionar em pares de entrada e saída do modelo com foco na organização dos dados de treino e teste de forma aleatória no formato de *dataFrames*. Essa estrutura de dados viabiliza a criação dos *DataLoaders* que são convenientes para carregar e iterar sobre os dados de forma eficiente, como por exemplo, na forma de *batches*.

As RNMP e RNC foram desenvolvidas com base em uma estrutura de classe, viabilizando a organização hierárquica das camadas das redes neurais para realizar a etapa *Forward*, conhecida como *container*. Na estrutura RNMP são definidos os números de camadas intermediárias, as funções de ativação e o número de neurônios pertencentes a cada camada, enquanto que na estrutura RNC são definidas as camadas de convolução, camadas pooling, *batch normalization*, quantidade e tamanhos dos filtros de convolução, números de camadas ocultas. Também, é necessário informar para ambos os modelos a métrica de avaliação, função de custo e otimizador. Além disso, os dados e rótulos foram embaralhados, separados em *batches* e no formato de tensores sendo 80% para o treinamento e 20% para teste. Dentre os dados de treinamento, 20% são reservados para a validação, resultando em uma distribuição final de 76% dos dados totais para treinamento, 4% validação e 20% teste.

O fluxograma utilizado para exemplificar as RNC e RNMP e a relação entre as etapas de treino e teste é apresentado na Figura 61. Este fluxo utiliza conjuntos de dados fornecidos pelos *DataLoaders* para as fases de treinamento e teste. Durante o treinamento, os parâmetros da rede são calculados por etapas de *forward* e *backward*, enquanto na fase de teste, apenas a etapa de *forward* é aplicada. Uma avaliação baseada em critérios de

aceitação é realizada, podendo encerrar o processo ou aplicar técnicas de otimização para os hiperparâmetros dos modelos matemáticos empregados.

Figura 61 – Fluxogramas das redes neural: RNC e RNMP.



Fonte: Adaptado de [Pytorch \(1996\)](#).

Nos Quadros 9 e 10, são delineados os processos que ocorrem em cada módulo nas condições dos modelos MLP e RNC, alinhando a representação visual da Figura 61. Enquanto que no Quadro 11 são tratadas as interações entre os módulos dos modelos RNMP e RNC, que estão exemplificados na Figura 61 por meio dos índices numéricos.

Quadro 9 – Descrição dos módulos do modelo RNMP.

| Módulos                   | Descrição:                                                                                                                                                                   |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Dados estruturados        | Neste módulo é realizado o <i>upload</i> do <i>DataFrame</i> com as características do sinal de EA, contagem de <i>hits</i> .                                                |
| Tratamento dos dados      | Os dados e rótulos, oriundos do <i>DataFrame</i> são embaralhados, separados em <i>batches</i> no formato de tensores, para as fases de treinamento e teste.                 |
| Rede neural: RNMP         | Neste módulo são configurados os números de parâmetros de entrada, camadas intermediárias e camada de saída, incluindo a quantidade de neurônios e suas funções de ativação. |
| Métrica de avaliação      | Cálculo da acurácia, F1-score, ROC e precisão.                                                                                                                               |
| Função de custo           | Seleção da função de perda por entropia cruzada.                                                                                                                             |
| Otimizador                | Otimizador <i>Adam</i> com as informações sobre os valores de taxa de aprendizagem e decaimento dos pesos.                                                                   |
| <i>Forward e backward</i> | Cálculo e validação dos parâmetros da RNMP e RNC.                                                                                                                            |

Fonte: Adaptado de [Pytorch \(1996\)](#).

Quadro 10 – Descrição dos módulos do modelo RNC.

| Módulos                          | Descrição:                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Dados estruturados               | Neste módulo ocorre o carregamento da base de dados de imagens juntamente com a planilha que estabelece a associação entre as imagens e os respectivos rótulos.                                                                                                                                                                                                                 |
| Tratamento dos dados             | São gerados os <i>datasets</i> e os <i>dataloades</i> e os <i>Batches</i> para o embaralhamento das imagens, transformação das imagens em tensores e separação dos dados em conjunto de treinamento e testes.                                                                                                                                                                   |
| Rede Neural: RNC                 | Este módulo está associado ao modelo RNC, no qual são determinados aspectos cruciais, incluindo o tamanho da imagem de entrada, número de camadas de convolução, quantidade e tamanhos do <i>kernels</i> , <i>strides</i> , <i>pooling</i> , a aplicação de <i>batch normalization</i> , número de camadas densas e quantidade de neurônios e a escolha de funções de ativação. |
| <i>Forward</i> e <i>backward</i> | Cálculo e validação dos parâmetros da rede RNC.                                                                                                                                                                                                                                                                                                                                 |

Fonte: Adaptado de [Pytorch \(1996\)](#).

*Nota:* Métricas de avaliação, função de custo e otimizador são os mesmos definidos no Quadro 9.

Quadro 11 – Interação entre os módulos dos modelos RMNP e RNC.

| Índice | RNMP                                                                                                                                     | RNC                                                           |
|--------|------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| (1)    | <i>DataFrame</i> da biblioteca Pandas                                                                                                    | Imagens e uma planilha com os nomes das imagens e dos rótulos |
| (2)    | São os dados e rótulos embaralhados, separados em <i>batches</i> e no formato de tensores sendo 80% para o treinamento e 20% para teste. |                                                               |
| (3)    | Matrizes e funções da Rede MLP                                                                                                           | Matrizes e funções da Rede CNN                                |
| (4)    | Modelo para o cálculo do acurácia.                                                                                                       |                                                               |
| (5)    | Modelo para o cálculo do erro, taxa de aprendizagem e decaimento de pesos.                                                               |                                                               |
| (6)    | Algoritmo <i>Adam</i> , taxa de aprendizagem e decaimento dos pesos.                                                                     |                                                               |
| (7)    | <i>Forward</i> da RNMP                                                                                                                   | <i>Forward</i> RNC.                                           |
| (8)    | <i>Backward</i> da RNMP                                                                                                                  | <i>Backward</i> da RNC.                                       |
| (9)    | Valores preditos pelos modelos e retorna o erro relacionado a cada saída.                                                                |                                                               |
| (10)   | <i>Forward</i> da RNMP                                                                                                                   | <i>Forward</i> da RNC.                                        |
| (11)   | Novos valores dos hiperparâmetros por meio do método de Otimização de Bayes (OB).                                                        |                                                               |

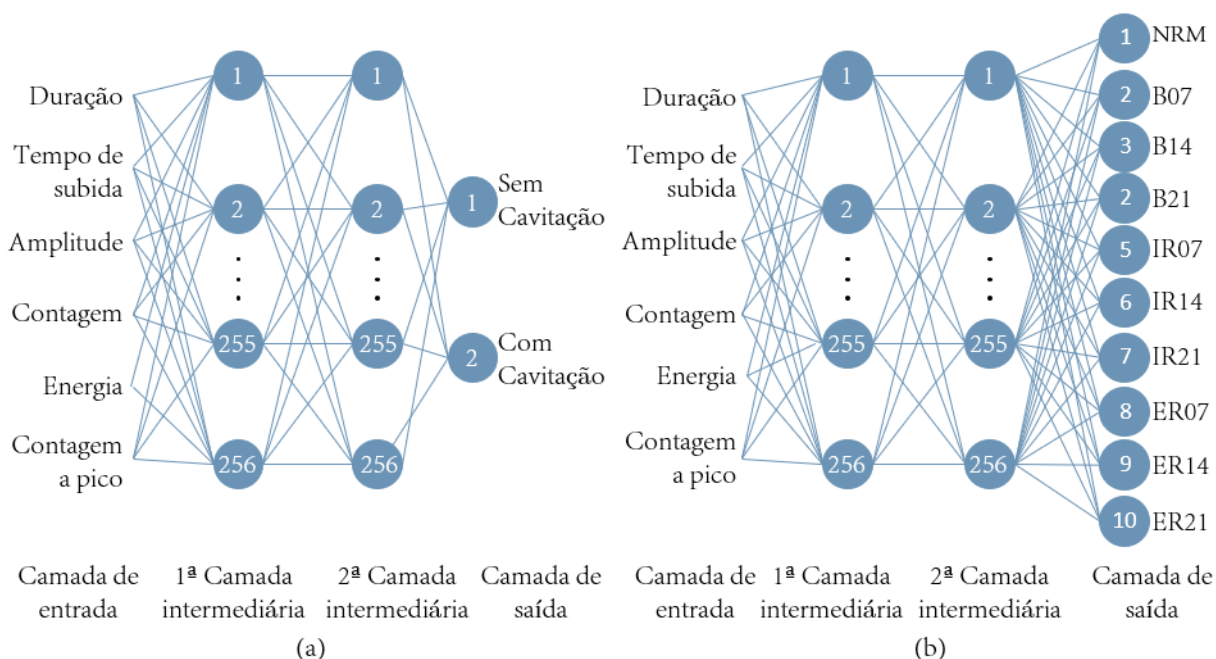
Fonte: Adaptado de [Pytorch \(1996\)](#).

- Modelo RNMP

Essa estrutura consiste em uma camada de entrada com **6 neurônios**, que correspondem às características da forma de onda: duração, tempo de subida, amplitude, contagem, contagem a pico e energia, seguida por duas camadas intermediárias, cada uma com 256 neurônios, e intercaladas por camadas de normalização (*BatchNormalization*). Finalmente, a camada de saída é composta por 3 neurônios para o caso do diagnóstico de cavitação e 10 neurônios para o caso das falhas em rolamentos. As funções de ativação escolhidas são *ReLU* para as camadas intermediárias e *Softmax* para a camada de saída.

A visualização da arquitetura para o conjunto de dados de cavitação está disponível na Figura 62-(a), enquanto que a arquitetura para o diagnósticos de falhas em rolamentos é ilustrada na Figura 62-(b), ambas as estruturas são detalhadas na Tabela 26. Além disso, outros hiperparâmetros, como número de épocas, taxa de aprendizagem, momento e tamanho do lote (*batch*), são apresentados na Tabela 27.

Figura 62 – Arquitetura da RNMP: (a) dados cavitação; (b) dados rolamentos.



Fonte: Autor.

Tabela 26 – Arquitetura da RNMP.

| Camada | Tipo                      | Número de Nós | Função de Ativação |
|--------|---------------------------|---------------|--------------------|
| 1      | Camada de entrada         | 6             | -                  |
| 2      | Primeira camada escondida | 256           | ReLU               |
| 3      | Segunda camada escondida  | 256           | ReLU               |
| 4      | Camada de saída           | 2 ou 10       | <i>Softmax</i>     |

---

Fonte: Autor.

Tabela 27 – Hiperparâmetros do modelo RNMMP.

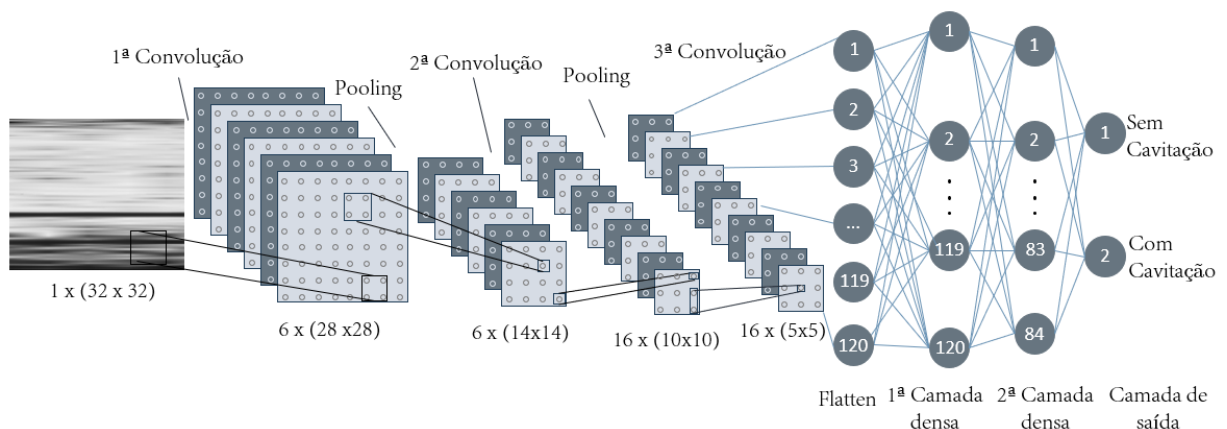
| Hiperparâmetros                  | Intervalo de busca |
|----------------------------------|--------------------|
| Camada de entrada                | 6                  |
| Número de camadas                | 2 à 8              |
| Número de neurônios por camada   | 32 à 256           |
| Funções de ativação              | ReLu ou Tanh       |
| Camada de saída                  | 10                 |
| Número de épocas                 | 100 à 300          |
| Taxa de aprendizado ( $lr$ )     | 1E-5 à 1E-3        |
| Decaimento de pesos ( $\alpha$ ) | 1E-5 à 1E-4        |
| Otimizador                       | Adam               |
| Função de Custo                  | Cross-Entropy      |
| Tamanho dos batches              | 256                |
| Treino-Validação-Teste           | 76%-4%-20%         |

- Modelo RNC

A arquitetura inicial da RNC é composta por três camadas convolucionais. Cada uma dessas camadas aplica filtros convolucionais (*kernels*) para extrair características dos dados. Após cada camada convolucional, há uma camada de *pooling*, que reduz a dimensionalidade dos dados e contribui para centralizar o conteúdo da imagem. Entre essas camadas, é aplicada a normalização em lote (*batch normalization*). Após as camadas convolucionais, é realizada a implementação da camada *Flatten*, que corresponde ao alinhamento dos neurônios antes das camadas densas. Essa arquitetura conta com duas camadas densas, que têm a função de realizar a classificação dos dados. A camada de saída possui um número de neurônios igual ao número de classes, ou seja conta com 2 neurônios, enquanto que para os dados de falha em rolamentos são 10 neurônios.

As funções de ativação escolhidas para este modelo são a tangente hiperbólica (*tanh*) para as camadas convolucionais e densas e a *softmax* para a camada de saída.

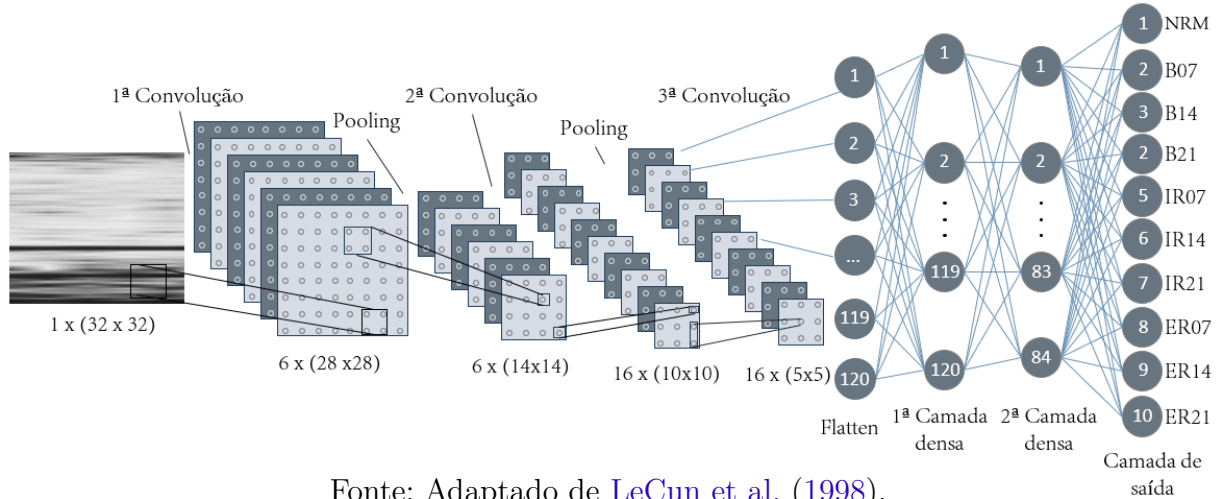
Figura 63 – Arquitetura da RNC para os dados de cavitação.



Fonte: Adaptado de [LeCun et al. \(1998\)](#).

A representação visual dessas arquiteturas pode ser vista nas Figura 63 e 64, enquanto os detalhes específicos da arquitetura da RNC, como o número de filtros, tamanhos dos *kernels*, tamanhos de *pooling*, e configurações das camadas densas, são apresentados na Tabela 28 e os valores iniciais dos hiperparâmetros na Tabela 29.

Figura 64 – Arquitetura da RNC para os dados de rolamento.



Fonte: Adaptado de LeCun et al. (1998).

Tabela 28 – Arquitetura da RNC.

| <i>Layers</i>           | <i>Feature Map</i> | <i>Size Map</i> | <i>Kernel</i> | <i>Stride</i> | <i>Activation</i> |
|-------------------------|--------------------|-----------------|---------------|---------------|-------------------|
| Camada de entrada       | 1                  | 32×32           | -             | -             | -                 |
| Camada de Convolução 1  | 6                  | 28×28           | 5×5           | 1×1           | <i>Tanh(.)</i>    |
| Camada <i>Pooling</i> 1 | 6                  | 14×14           | 2×2           | 2×2           | -                 |
| Camada de Convolução 2  | 16                 | 10×10           | 5×5           | 1×1           | <i>Tanh(.)</i>    |
| Camada <i>Pooling</i> 2 | 16                 | 5×5             | 2×2           | 2×2           | -                 |
| Camada de Convolução 3  | 120                | 1×1             | 5×5           | 1×1           | <i>Tanh(.)</i>    |
| Camada Densa 1          | -                  | 84              | -             | -             | <i>Tanh(.)</i>    |
| Camada Densa 2          | -                  | 120             | -             | -             | <i>Tanh(.)</i>    |
| Camada de saída         | -                  | 2 ou 10         | -             | -             | <i>Softmax(.)</i> |

Fonte: Adaptado de LeCun et al. (1998).

Tabela 29 – Hiperparâmetros da modelo RNC.

| Hiperparâmetros                    | Intervalo de busca |
|------------------------------------|--------------------|
| Épocas                             | 100 a 300          |
| Taxa de aprendizagem ( <i>lr</i> ) | 1E-5 à 1E-3        |
| Momentum ( $\alpha$ )              | 1E-5 à 1E-4        |
| Função de custo                    | Cross-Entropy      |
| Otimizador                         | <i>Adam</i>        |
| MiniBatch                          | 256                |
| Treinamento-Validação-Teste        | 76%-4%-20%         |

## 4.7 Análise de Resultados

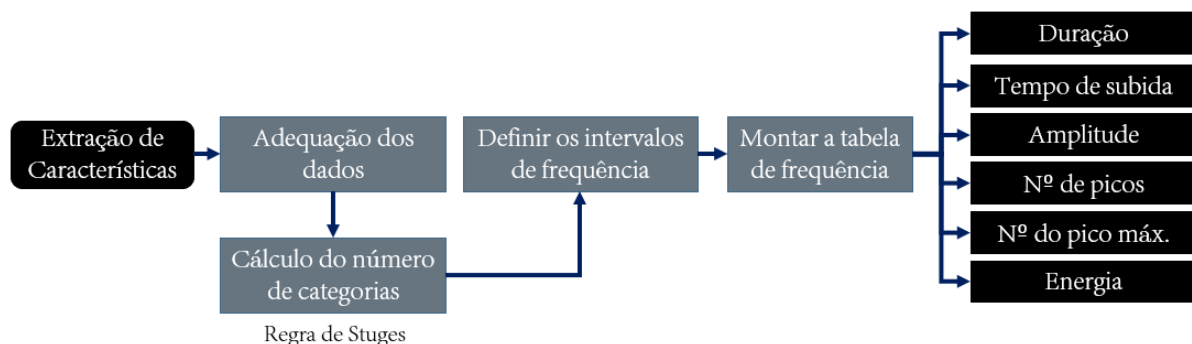
### 4.7.1 Métricas de Desempenho

Na etapa de análise de resultados foi utilizada uma validação cruzada *K-fold*, cujo valor *K* é um hiperparâmetro a ser otimizado e uma matriz de confusão. Além disso, foram utilizados dentro desse contexto, testes de hipóteses para comparar o desempenho dos modelos de aprendizado de máquina com as referências tendo como métricas os valores de acurácia, precisão, *recall*, *F1-score*.

### 4.7.2 Análise Exploratória dos Dados

A análise exploratória de dados desempenha um papel fundamental na compreensão e na interpretação de conjuntos de dados. Assim, utilizou-se as tabelas de frequência para tabular a distribuição dos valores dos parâmetros de hits, são esses: duração, tempo de subida, amplitude, número de picos, número de pico máximo e energia. Na Figura 65, é ilustrado o fluxograma conceitual para a criação das tabelas de frequências.

Figura 65 – Fluxograma conceitual da análise exploratória dos dados.

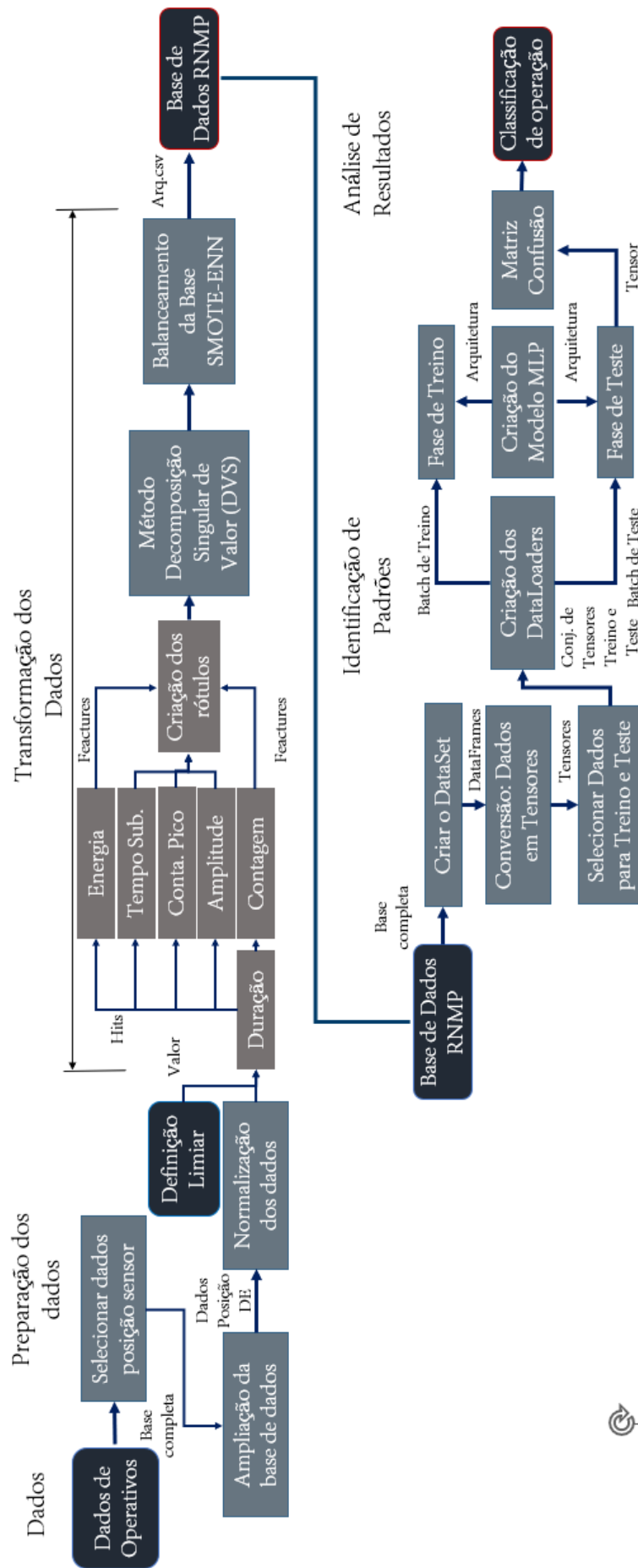


O procedimento inicia-se com a conversão dos dados do (matriz características), seguindo a preparação para a análise estatística. O segundo passo compreende a aplicação da regra de Sturges para determinar o número adequado de categorias em dados contínuos, seguida pela definição dos intervalos de frequência (STURGES, 1926). Finalmente, com base nessas etapas preparatórias, procede-se à construção das tabelas de frequência para cada característica, proporcionando uma representação tabular dos dados.

## 4.8 Fluxograma conceitual completo dos modelos

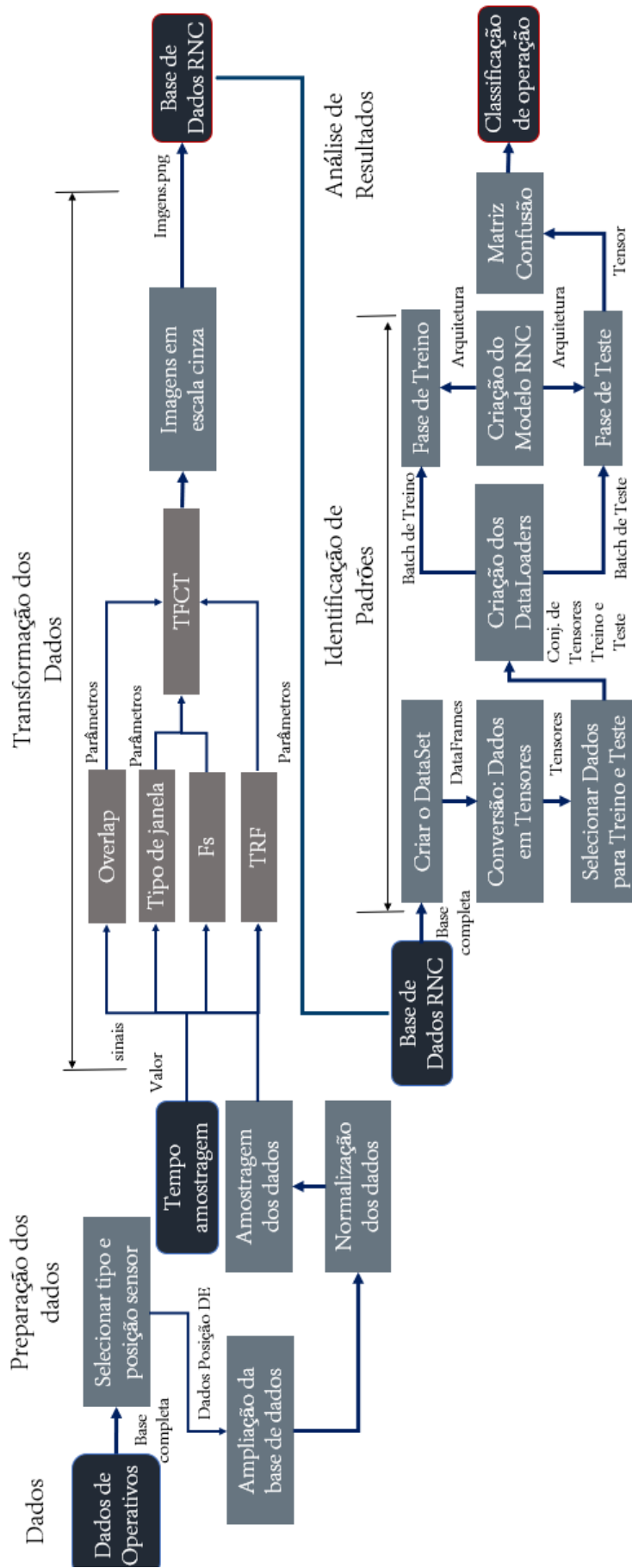
Estes fluxogramas fornecem uma representação visual abrangente das interações e relações entre diferentes componentes destes modelos, visando oferecer uma compreensão mais detalhada de seu funcionamento. A seguir, são apresentados os fluxogramas dos modelos RNMP (Figura 66) e RNC (Figura 67).

Figura 66 – Fluxograma conceitual do modelo RNMP.



Fonte: Autor.

Figura 67 – Fluxograma conceitual do modelo RNC.



Fonte: Autor.

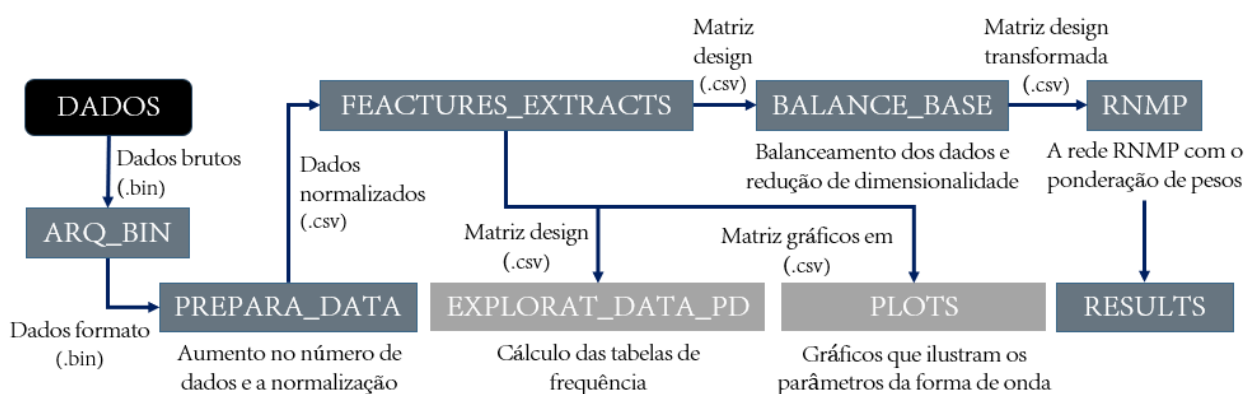
## 5 Desenvolvimento dos Algoritmos

Nesta seção, são apresentados os algoritmos dos modelos computacionais desenvolvidos. Esses algoritmos estão estruturados de acordo com os fluxos conceituais de cada etapa do processo DCBD. Para manter a consistência e a lógica, optou-se por apresentar os códigos e algoritmos por modelo, em vez de seguir o fluxo metodológico, uma vez que os dados de saída de um módulo são as condições de entrada para o módulo subsequente. Assim, as próximas seções correspondem aos modelos RNMP e RNC, oferecendo um fluxo lógico em relação à programação.

### 5.1 Modelo RNMP

Os algoritmos de aprendizado de máquina vinculados ao modelo RNMP são discutidos e detalhados nesta seção. Esses algoritmos são desenvolvidos para serem aplicados em conjuntos de dados compostos por séries temporais, com o objetivo de detectar desde falhas em rolamentos até níveis de cavitação, independente do tipo de sensor utilizado na coleta de dados. A Figura 68 ilustra o fluxograma computacional do modelo RNMP, alinhado com o fluxograma conceitual.

Figura 68 – Fluxograma computacional do modelo RNMP.



Fonte: Autor.

Este fluxo computacional abrange os algoritmos utilizados desde o *download* dos dados provenientes do ensaio até a aplicação das técnicas de avaliação dos resultados. Cada módulo deste fluxo representa estruturas de dados organizadas em classes, seguindo o paradigma de orientação a objetos, gerando saídas nas extensões apresentadas no fluxograma, como por exemplo binários (“*.bin*”) e arquivos separados por vírgula (“*.csv*”). No Quadro 12, são descritas as funções de cada módulo, necessárias para atender aos métodos e modelos apresentados na Seção 4.

Quadro 12 – Descrição e interação entre os módulos do modelo RNMP.

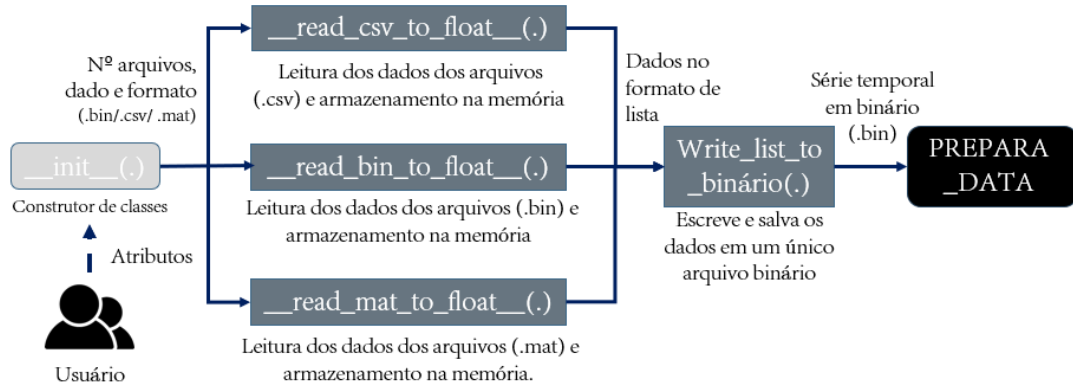
| Módulo           | Descrição                                                                                                                                                             | Funções                                                                                                                                                                                                                                          |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ARQ_BIN          | Realiza a leitura dos arquivos dos dados brutos nos formatos “.bin”, “.csv” e “.mat” e fornece como saída um único arquivo binário.                                   | As funções desse módulo são: <i>csv_bin</i> , <i>mat_bin</i> e <i>bin_bin</i> .                                                                                                                                                                  |
| PREPARA_DATA     | Realiza o aumento no número de amostras e a normalização da série, fornecendo arquivos binários como saída.                                                           | <i>select_data</i> , <i>frequency_synthetic</i> , <i>data_augmentation</i> , <i>data_normalization</i> e <i>export_data</i> .                                                                                                                    |
| FEACTURE_EXTRACT | Realiza a caracterização da forma da onda por meio da contagem de <i>hits</i> e fornece a matriz <i>design</i> em um arquivo “.csv”.                                  | <i>signal_limiar</i> , <i>derivada_foward</i> , <i>count_picos</i> , <i>duration</i> , <i>rise_time</i> , <i>count</i> , <i>count_peak</i> , <i>energy</i> , <i>data-frame</i> e <i>archive_name</i> .                                           |
| BALANCE_BASE     | Realiza o balanceamento da base de dados e a redução de dimensionalidade fornecendo ao próximo módulo um arquivo “.csv”.                                              | <i>import_feactures</i> , <i>array</i> , <i>tSNE</i> , <i>SMOTE</i> , <i>DVS</i> , <i>DVS_truncated</i> , <i>undersample</i> , <i>sample_data</i> e <i>SMOTE_PARTIAL</i> .                                                                       |
| RNMP             | Corresponde à parametrização do modelo e à realização das fases de treino e teste fornecendo como saída os valores dos pesos e <i>bias</i> em arquivo <i>python</i> . | <i>import_data</i> , <i>set_data</i> , <i>data_hit</i> , <i>set_loader</i> , <i>net_parameter</i> , <i>train_main</i> , <i>test_main</i> .                                                                                                       |
| RESULTS          | Realiza a leitura dos arquivos dos dados brutos nos formatos .bin, .csv e .mat e fornece como saída um único arquivo binário                                          | <i>csv_bin</i> , <i>mat_bin</i> e <i>bin_bin</i> .                                                                                                                                                                                               |
| EXPLORAT_DATA_PD | Realiza a análise exploratória dos dados a partir dos parâmetros contagem de <i>hits</i> e fornece como saída um único arquivo “.csv” com a frequência de ocorrência. | <i>import_feactures</i> , <i>data_rotulo</i> , <i>list_k_interval</i> , <i>freq_table</i> e <i>freq_k_table</i>                                                                                                                                  |
| PLOTS            | Realiza a ilustração dos parâmetros da contagem de <i>hits</i> sobre a série temporal, salvando imagens no formato “.eps” como arquivos de saída.                     | <i>import_data</i> , <i>import_feactures</i> , <i>plot_signal</i> , <i>plot_limiar</i> , <i>plot_hits</i> , <i>dataFrameDuration</i> , <i>dataFrameCount</i> , <i>dataFrameCountPeak</i> , <i>dataFrameAmplitude</i> , <i>dataFrame_contagem</i> |

Fonte: Autor.

### 5.1.1 Dados

Neste módulo, é realizada a leitura e adequação dos dados disponibilizados pelo ensaio em um ou mais arquivos, nos três formatos suportados: binário, separado por vírgulas (“csv”) e MATLAB. A Figura 69 ilustra o fluxograma computacional do módulo ARQ\_BIN, onde cada bloco representa uma função de código.

Figura 69 – Fluxograma computacional do módulo ARQ\_BIN.



Fonte: Autor.

Este fluxo requer como entrada a série temporal, juntamente com o número e formato dos arquivos. Como parte do processamento, ocorre a consolidação desses dados em uma única série temporal, culminando na geração de um único arquivo binário como saída do algoritmo.

O Algoritmo 5 corresponde ao módulo `read_csv_float`, que se aplica ao `read_bin_float` e `read_mat_float`, enquanto que o Algoritmo 5 refere-se ao módulo `Write_list_to_binario`.

---

#### Algorithm 5 Dados

---

```

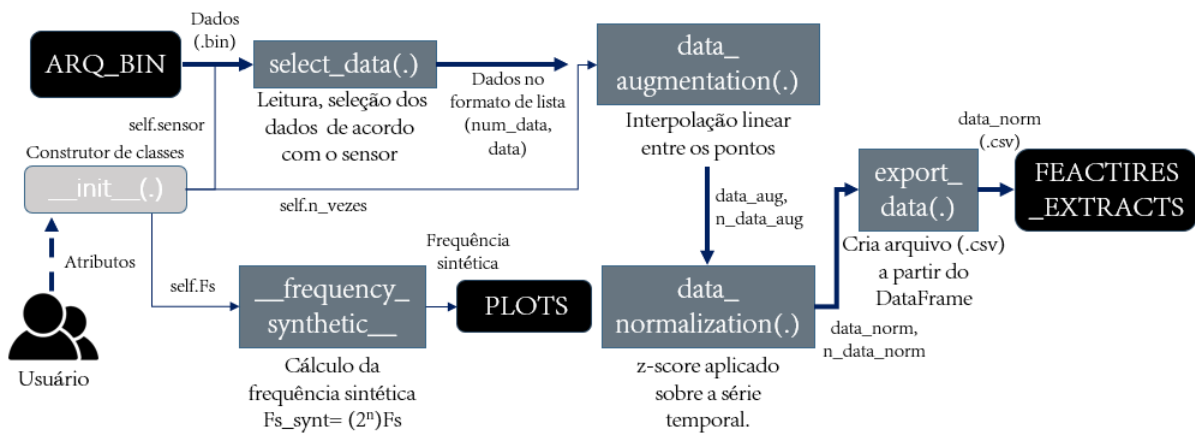
1: function SELECT_DATA(self)
2:                                     ▷ Download dos dados
3:   Import módulos os e scipy
4:   mat_data ← carregar arquivo mat (dir_arq)
5:   arq_name ← obter lista de chaves do dicionário mat_data
6:
7:   for all arq em arq_name do:
8:     Adicionar arq à lista_arq_name
9:   end for
10:
11:  for all arq em lista_arq_name do:
12:    if self.sensor = True then:
13:      Adicionar mat_data[arq][i][0] a data
14:    end if
15:  end for
16:  return (num_data, data)
17: end function
  
```

---

### 5.1.2 Preparação dos Dados

No que diz respeito à fase de preparação dos dados, incluem-se a seleção do sensores que deseja analisar, o aumento da base de dados por meio da inserção de dados sintéticos com interpolações lineares entre dois dados e, por fim, a normalização da série temporal. Essas unidades estão presentes no fluxograma computacional do módulo PREPARA\_DATA ilustrada na Figura 70 que requer como entrada a série temporal preparada no ARQ\_BIN, juntamente com o número de amostras e entrega, ao fim do processamento, os dados normalizados em um arquivo csv para a etapa FEACTURES\_EXTRACTS.

Figura 70 – Fluxograma computacional do módulo PREPARA\_DATA.



O módulo `__init__()` é responsável por concentrar os atributos fornecidos pelo usuário, que correspondem à informações, como: diretório, nomes dos arquivos, escolha do sensor, frequência de amostragem, número de vezes que será aumentada a série temporal e, por fim, o nome do arquivo onde são armazenados os dados processados no formato de csv. Essas informações estão presentes no Algoritmo 6.

---

**Algorithm 6** `__init__(self, cwd, FS, sensor, n_vezes, data_file, csv_file)`

---

- |                                            |                                               |
|--------------------------------------------|-----------------------------------------------|
| 1: <code>self.cwd ← cwd</code>             | ▷ Diretório que contém os arquivos            |
| 2: <code>self.data_file ← data_file</code> | ▷ Nome do arquivo externo                     |
| 3: <code>self.sensor ← sensor</code>       | ▷ Escolha do sensor                           |
| 4: <code>self.FS ← FS</code>               | ▷ Frequência de Amostragem                    |
| 5: <code>self.n_vezes ← n_vezes</code>     | ▷ Novo tamanho série temporal: $2^{n\_vezes}$ |
| 6: <code>self.csv_file ← csv_file</code>   | ▷ Nome do arquivo que será salvo              |
- 

O módulo `__frequency_synthetic__` é responsável por calcular um novo valor de frequência de amostragem ( $FS_{synt}$ ), com base nos atributos de frequência de amostragem (`self.FS`) e número de vezes que deseja aumentar (`self.n_vezes`) (Equação 4.2).

O módulo `select_data()` é responsável por carregar os dados na memória, ajustando os nomes dos arquivos de forma a torná-los aptos para a seleção a partir da escolha do sensor. Ao final desse processo, o algoritmo retorna o número de amostras (`num_data`) e os dados selecionados (`data`), conforme ilustrado no Algoritmo 7.

**Algorithm 7** select\_data(self)

---

```

1:                                     ▷ importar bibliotecas e criar variáveis
2: import os
3: from scipy import io
4:  $data \leftarrow []$  e  $lista\_arq\_name \leftarrow []$ 
5:                                     ▷ Carregar dados na memória
6:  $dir\_arq \leftarrow os.path.join(os.getcwd(), self.data\_file)$ 
7:  $mat\_data \leftarrow io.loadmat(dir\_arq)$ 
8:  $arq\_name \leftarrow list(mat\_data.keys())$ 
9:                                     ▷ Adequar nomes dos arquivos
10: for  $arq$  in  $arq\_name$  do
11:    $lista\_split \leftarrow arq.split("_")$ 
12:    $n \leftarrow lista\_split[0]$ 
13:   if  $len(n) \neq 0$  then
14:      $lista\_arq\_name.append(arq)$ 
15:   end if
16: end for
17:                                     ▷ Seleção dos dados de acordo com a escolha do sensor
18: for  $arq$  in  $lista\_arq\_name$  do
19:    $lista\_split2 \leftarrow arq.split("_")$ 
20:   if  $lista\_split2[1] == self.sensor$  then
21:      $arq\_name \leftarrow arq$ 
22:      $num\_data \leftarrow len(mat\_data[arq])$ 
23:     for  $i$  in  $range(num\_data)$  do
24:        $data.append(mat\_data[arq][i][0])$ 
25:     end for
26:   end if
27: end for
28:
29: return ( $num\_data$ ,  $data$ )                                     ▷ Numero de amostras e os dados

```

---

O módulo `data_augmentation(.)`, tratado no Algoritmo 8, aborda o processo de expansão do conjunto de dados, com o intuito de aprimorar a precisão na detecção dos valores máximos e mínimos do sinal. Os valores sintéticos são inseridos entre dois pontos de valores reais por meio de uma interpolação linear, a partir do cálculo da média, fornecendo como saída o número de dados ( $n\_aug$ ) e os dados aumentados ( $data\_aug$ ).

O módulo de `data_normalization(.)`, tratado no Algoritmo 9, corresponde à etapa de normalização dos dados por meio cálculo da média e desvio padrão do sinal (score-z). Essa transformação faz o sinal permanecer em um intervalo com uma média nula e desvio padrão unitário. Esse algoritmo deve receber os dados do `data_augmentation(self)` e o atributo `self.n_vezes`, fornecendo como saída o número de amostras ( $n\_norm$ ), dados normalizados ( $data\_norm$ ), média ( $mean$ ) e desvio padrão ( $std$ ).

O módulo `export_data(.)`, tratado no Algoritmo 10, representa a criação de um arquivo csv com os dados da série temporal normalizada, `data_normalization(.)`.

**Algorithm 8** data\_augmentation(self)

---

```

1:                                     ▷ Dados e tamanho da lista
2:  $n\_sample, data \leftarrow \text{SELECT\_DATA}(\text{self})$                                      ▷ Algoritmo 7
3:                                     ▷ Aumento no número de dados ( $2^{n\_vezes}$ );
4: for  $j \leftarrow 0$  to  $\text{self}.n\_vezes$  do
5:    $data\_augment \leftarrow []$ 
6:    $n\_augment \leftarrow \text{len}(data)$ 
7:   for  $i \leftarrow 0$  to  $n\_augment - 1$  do
8:      $data\_synt \leftarrow \frac{data[i] + data[i+1]}{2}$                                      ▷ Média
9:      $data\_augment.append(data[i], data\_synt)$ 
10:  end for
11:   $data \leftarrow data\_augment$ 
12: end for
13:  $n \leftarrow \text{len}(data)$ 
14: return  $(n, data)$                                      ▷ Dados no formato de lista e o tamanho dessa lista

```

---

**Algorithm 9** data\_normalization(self)

---

```

1: import numpy as np
2:                                     ▷ Dados com frequência sintética
3:  $n, data \leftarrow \text{DATA\_AUGMENTATION}(\text{self})$                                      ▷ Algoritmo 8
4:  $data\_norm \leftarrow []$ 
5:  $mean \leftarrow \text{np.mean}(data)$ 
6:  $std \leftarrow \text{np.std}(data)$ 
7:                                     ▷ Normalização dos dados
8: for  $i \leftarrow 0$  to  $n - 1$  do
9:    $zscore \leftarrow \frac{data[i] - mean}{std}$                                      ▷ Cálculo do z-score
10:   $data\_norm.append(zscore)$ 
11: end for
12:                                     ▷ N° de amostras, dados normalizados, média e desvio padrão
13: return  $(n, data\_norm, mean, std)$ 

```

---

**Algorithm 10** export\_data(self)

---

```

1:                                     ▷ Carregando dados na memória
2:  $n, data\_norm, mean, std \leftarrow \text{DATA\_NORMALIZATION}(\text{self})$                                      ▷ Algoritmo 9
3:  $data \leftarrow []$ 
4: for  $i \leftarrow 0$  to  $n - 1$  do
5:    $data.append([i, data\_norm[i], mean, std])$ 
6: end for
7:                                     ▷ Criação do arquivo CSV com dados normalizados
8: with open( $\text{self.csv\_file}$ , mode = "w", newline =) as  $arq\_csv$  :
9:    $escritor\_csv \leftarrow \text{csv.writer}(arq\_csv)$ 
10:   $escritor\_csv.writerow(["n", "data\_norm", "mean", "std"])$ 
11:
12: for  $linha$  in  $data$  do
13:    $escritor\_csv.writerow(linha)$ 
14: end for
15:
16: return  $(data)$                                      ▷ Tupla: dados, média e desvio padrão

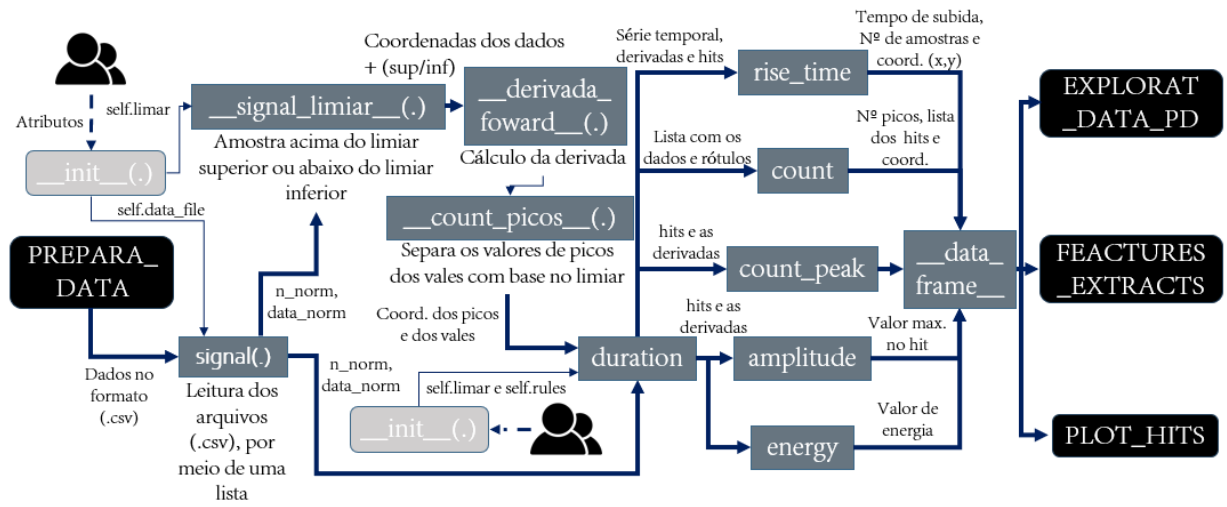
```

---

### 5.1.3 Extração de Características

A etapa de extração de características é composta principalmente pelos módulos encarregados de calcular os parâmetros relacionados à contagem de *hits*, conforme apresentado na Seção 4.5.1. Cada módulo referente à etapa extração de características é ilustrado na Figura 71 e corresponde à uma função do algoritmo. Essas funções incluem desde a preparação dos dados como a leitura do sinal, aplicação do limiar, cálculo das derivadas, contagem de todos os picos, até as medições da origem do *hit*, como: duração, tempo de subida, contagem absoluta, contagem de pico, amplitude e energia.

Figura 71 – Fluxograma computacional do módulo FEATURES\_EXTRACT.



O módulo `___init___.()` é responsável por concentrar os atributos informados pelo usuário, como: diretório dos arquivos provenientes da extração de características, frequência de amostragem sintética, valor do limiar, condição para iniciar um *hit*, rótulo e nome do arquivo com os dados normalizado e rotulados. Essas informações estão presentes no Algoritmo 11.

**Algorithm 11** `___init___.(self, fail_name, FS_synt, limiar, rules_hits, cwd)`

```

1: function ___INIT___.(self, fail_name, FS_synt, limiar, rules_hits, cwd):
2:   self.cwd ← cwd                                ▷ Diretório atual
3:   self.FS_synt ← FS_synt                          ▷ Frequência de amostragem sintética
4:   self.limiar ← limiar                            ▷ Valor do limiar
5:   self.rules_hits ← rules_hits                    ▷ Critério para a definição de um hit
6:   self.fail_name ← fail_name                      ▷ Rótulo
7:   self.data_file ← fail_name + '.csv'              ▷ Nome do arquivo
8: end function
  
```

O módulo `signal(.)` refere-se ao Algoritmo 13 e é responsável por ler o arquivo csv especificado por `self.data_file`, armazenar os dados na memória e registrar as coordenadas (x, y) de cada amostra. Como resultado da execução do algoritmo, são retornados o

número total de amostras e uma tupla contendo as coordenadas normalizadas do sinal, representadas por  $(n\_norm, data\_norm)$ .

O módulo `__signal_limiar__`(.), referenciado pelo Algoritmo 12, é responsável por identificar amostras que excedem o limiar superior e aquelas que estão abaixo do limiar inferior. Ele produz como saída o número de amostras e uma tupla contendo as coordenadas dos dados, rotuladas como “sup” ou “inf”  $(n\_sample, pst)$ .

---

**Algorithm 12** `signal(self)`


---

```

1: function SIGNAL(self):
2:   importar pandas como pd                                ▷ Bibliotecas e variáveis
3:   arq_name ← self.data_file
4:   data_norm ← []
5:                                                         ▷ Leitura dos dados normalizados
6:   df ← PD.READ_CSV(arq_name)
7:   data ← df.values.tolist()                                ▷ Converter dataframe em lista
8:   for cada linha em data do
9:     data_norm.append((linha[0], linha[1]))                  ▷ Coordenadas: dados
10:  end for
11:  n ← len(data_norm)
12:  retornar (n, data_norm)                                ▷ N de amostras e dados
13: end function

```

---



---

**Algorithm 13** `__signal_limiar__`(self)

---

```

1: function __SIGNAL_LIMIAR__(self):
2:   n_sample, signal ← SIGNAL(self)                          ▷ Algoritmo 13
3:   pst ← []
4:                                                         ▷ Amostras superior ao limiar_sup e inferior ao limiar_inf
5:   for cada amostra em signal do
6:     if amostra[1] ≥ self.limiar then
7:       pst.append(tupla(amostra[0], amostra[1], "sup"))
8:     else
9:       pst.append(tupla(amostra[0], amostra[1], "inf"))
10:    end if
11:  end for
12:  retornar (n_sample, pst)                                ▷ N amostras, tuplas de dados (x,y) rotulados
13: end function

```

---

O módulo `__derivada_forward__`(.) refere-se ao Algoritmo 10 responsável pelo cálculo das derivadas do sinal, produzindo como saída o número de amostras e o valor da derivada  $(len(dsign), dsign)$ .

O módulo `__count_picos__`(.), associado ao Algoritmo 14, é responsável por identificar os valores de pico e vale que excedem ou ficam abaixo dos respectivos limiares. Para isso, este módulo conta com o apoio dos Algoritmos 12 e 10, que combinados fornecem essa seleção, produzindo como saída uma lista das coordenadas de todos os picos e vales do sinal  $(vlr\_peak, vlr\_vale)$ .

**Algorithm 14** `__derivada_forward__`(self)

---

```

1: function __DERIVADA_FORWARD__(self):
2:    $n, sign \leftarrow \text{__SIGNAL\_LIMIAR__}(self)$  ▷ Algoritmo 12
3:    $step \leftarrow 1$ 
4:    $dsign \leftarrow []$ 
5:   ▷ Cálculo da derivada do sinal pelo método forward
6:   for  $i$  de 0 até  $n - 1$  do
7:      $dsign.append((sign[i + 1][1] - sign[i][1]) / (step))$  ▷ Método Forward
8:   end for
9:   retornar  $(len(dsign), dsign)$  ▷ N amostras, derivadas
10: end function

```

---

**Algorithm 15** `__count_picos__`(self)

---

```

1: function __DERIVADA_FORWARD__(self):
2:    $n\_samples, sign \leftarrow \text{__SIGNAL\_LIMIAR__}(self)$  ▷ Algoritmo 12
3:    $nd\_samples, dsign \leftarrow \text{__DERIVADA_FORWARD__}(self)$  ▷ Algoritmo 10
4:    $vlr\_peak, vlr\_vale \leftarrow [], []$ 
5:   ▷ Verificar no sinal o que é pico e o que é vale
6:   for  $i$  de 0 até  $nd\_samples - 1$  do
7:     if  $(dsign[i] > 0 \text{ e } dsign[i + 1] < 0)$  then
8:        $vlr\_peak.append(tupla([sign[i][0], sign[i][1]]))$ 
9:     else if  $(dsign[i] < 0 \text{ e } dsign[i + 1] > 0)$  then
10:       $vlr\_vale.append(tupla([sign[i][0], sign[i][1]]))$ 
11:     end if
12:   end for
13:   retornar  $(vlr\_peak, vlr\_vale)$  ▷ Lista com as coordenadas de picos e vales
14: end function

```

---

O módulo `duration(.)` corresponde ao Algoritmo 42 e foi concebido para calcular o intervalo de tempo entre o primeiro pico que ultrapassa o valor do limiar e o último pico antes que o sinal retorne a valores inferiores ao limiar. Esse algoritmo requer como entrada o sinal temporal bruto e o valor do limiar. Esse algoritmo opera em duas etapas: a primeira envolve a identificação do primeiro pico e do último pico, enquanto a segunda etapa aborda a correção do início do *hit* até o primeiro pico, seguido da conversão para a unidade de tempo. Em sua saída, são fornecidos o número de ocorrência de *hits* com as respectivas durações.

O módulo `rise_time(.)`, conforme delineado no Algoritmo 32, trata do intervalo de tempo necessário para alcançar a amplitude máxima dentro de um evento *Hit*. Assim como no algoritmo `duration(.)`, ele opera em duas etapas: a primeira envolve a identificação do pico máximo e do primeiro pico após o início do *hit*, enquanto a segunda etapa aborda a correção do início do *hit* até o primeiro pico. Esse algoritmo deve receber como entrada a série temporal normalizada, as derivadas do sinal e informações sobre a duração e fornece como saída o tempo em segundos para atingir o pico máximo, e o número de amostras com as respectivas coordenadas.

**Algorithm 16** `duration(self)`


---

```

1: function DURATION(self):
2:                                     ▷ Série temporal dos dados normalizados da memória
3:    $n, sign \leftarrow \text{___SIGNAL\_LIMIAR___}(\text{self})$                                      ▷ Algoritmo 12
4:    $vlr\_peak, vlr\_vale \leftarrow \text{___COUNT\_PICOS___}(\text{self})$                        ▷ Algoritmo 22
5:    $n\_hits \leftarrow 0$ 
6:    $hit \leftarrow \text{False}$ 
7:    $hits\_picos, hits\_time, hits\_sample \leftarrow [], [], []$ 
8:    $hits\_ini, hits\_end \leftarrow [], []$ 
9:                                     ▷ Definição da duração com base no primeiro pico e na “regra”
10:  for  $vlr\_pk$  em  $vlr\_peak$  do
11:    if  $\neg hit$  and  $vlr\_pk[1] \geq \text{self.limiar}$  then                                     ▷ Início do hit
12:       $hit \leftarrow \text{True}$ 
13:       $hits\_ini.append(vlr\_pk)$ 
14:       $cont \leftarrow 0$ 
15:    else if  $hit$  and  $vlr\_pk[1] < \text{self.limiar}$  then                                     ▷ Fim do hit
16:       $cont \leftarrow cont + 1$ 
17:      if  $cont \geq \text{self.rules\_hits}$  then
18:         $hits\_end.append(vlr\_pk)$ 
19:         $hit \leftarrow \text{False}$ 
20:      end if
21:    end if
22:  end for
23:                                     ▷ Corrigir o início do hit a partir do primeiro pico
24:    $hit\_ini\_limiar \leftarrow []$ 
25:   for  $x, y$  em  $hits\_ini$  do
26:      $x \leftarrow \text{int}(x)$ 
27:      $i \leftarrow 0$ 
28:     while  $sign[x - i][1] \geq \text{self.limiar}$  do
29:        $i \leftarrow i + 1$ 
30:     end while
31:      $hit\_ini\_limiar.append(\text{tupla}([sign[x - i][0], sign[x - i][1]]))$ 
32:   end for
33:                                     ▷ Salvar a lista com os hits
34:    $num\_hits \leftarrow \min(\text{len}(hits\_ini), \text{len}(hits\_end))$ 
35:   for  $i$  de 0 até  $num\_hits - 1$  do
36:      $n\_hits \leftarrow n\_hits + 1$ 
37:      $hits\_picos.append(\text{tupla}([hits\_ini[i][0], hits\_end[i][0]]))$ 
38:      $hits\_time.append(((1/\text{self.FS}) * (hits\_end[i][0] - hit\_ini\_limiar[i][0])))$ 
39:      $hits\_sample.append(\text{tupla}([hit\_ini\_limiar[i][0], hits\_end[i][0]]))$ 
40:   end for
41:    $\triangleright$  N do hit, primeiro e último pico do hit, duração [s] e coord. dos hits
42:   retornar ( $n\_hits, hits\_picos, hits\_time, hits\_sample$ )
43: end function

```

---

**Algorithm 17** `rise_time(self)`


---

```

1: function RISE_TIME(self):
2:                                     ▷ Entrada: Série normalizada, derivada do sinal e Hits
3:    $n\_samples, sign \leftarrow \text{__SIGNAL\_LIMIAR__}(self)$                                      ▷ Algoritmo 12
4:    $nd\_samples, dsign \leftarrow \text{__DERIVADA\_FORWARD__}(self)$                                      ▷ Algoritmo 10
5:    $n\_hits, hits\_pico, hits\_time, hits\_sample \leftarrow \text{DURATION}(self)$    ▷ Algoritmo 42
6:    $time\_rise, pts\_rise \leftarrow [], []$ 
7:                                     ▷ Processamento:
8:   for  $i$  de 0 até  $n\_hits - 1$  do
9:      $peak\_max \leftarrow 0$ 
10:    for  $j$  de 0 até  $nd\_samples - 1$  do
11:      if  $dsign[j] > 0$  and  $dsign[j + 1] < 0$  and  $sign[j][1] \geq self.limiar$  then
12:        if  $sign[j][0] \geq hits\_pico[i][0]$  and  $sign[j][0] \leq hits\_pico[i][1]$  then
13:          if  $peak\_max \leq sign[j][1]$  then
14:             $peak\_max \leftarrow sign[j][1]$ 
15:             $pos\_end \leftarrow sign[j][0]$ 
16:             $pos\_ini \leftarrow hits\_pico[i][0]$ 
17:             $coord\_x \leftarrow pos\_end$ 
18:             $coord\_y \leftarrow peak\_max$ 
19:          end if
20:        end if
21:      end if
22:    end for
23:     $rise\_picos\_pts \leftarrow pos\_end - pos\_ini$ 
24:     $rise\_ini\_pts \leftarrow hits\_pico[i][0] - hits\_sample[i][0]$ 
25:     $rise\_pts \leftarrow rise\_ini\_pts + rise\_picos\_pts$ 
26:     $rise\_time \leftarrow (1/self.FS) \times rise\_pts$ 
27:     $time\_rise.append(rise\_time)$ 
28:     $pts\_rise.append(tupla([rise\_pts, coord\_x, coord\_y]))$ 
29:  end for
30:                                     ▷ Saída: Tempo de subida [s], Número de amostras, coordenadas (x,y)
31:  retornar ( $time\_rise, pts\_rise$ )
32: end function

```

---

O módulo `count(.)` identificado como Algoritmo 18, desempenha a função de quantificar os picos que excedem o limiar predefinido dentro de cada *Hit*. Esse algoritmo é alimentado com os dados normalizados, os valores das derivadas e as informações referente ao módulo `duration(.)` (Algoritmo 18). Como saída são fornecidos o número de picos e uma lista das coordenadas com todos os picos em cada *Hit*.

O módulo `amplitude(.)` corresponde ao Algoritmo 20 e representa a maior amplitude dos picos que foram identificados no módulo `count(.)`. Esse algoritmo recebe como parâmetros de entrada informações dos dados normalizados, derivadas e a “duração”. O processo do cálculo do pico máximo resume-se em identificar todos os picos em cada *hit* e por meio da função “max” definir o maior de todos. Como saída, é disponibilizado o valor da amplitude para cada *hit*.

**Algorithm 18** count(self)

---

```

1: function COUNT(self):
2:    $n\_samples, sign \leftarrow \text{___SIGNAL\_LIMAR___}(self)$  ▷ Algoritmo 12
3:    $nd\_samples, dsign \leftarrow \text{___DERIVADA\_FORWARD___}(self)$  ▷ Algoritmo 10
4:    $n\_hits, hits, hits\_time, hits\_sample \leftarrow \text{DURATION}(self)$  ▷ Algoritmo 42
5:    $hit\_cont \leftarrow 1$ 
6:    $hits\_cont \leftarrow []$ 
7:    $num\_peak \leftarrow []$ 
8:   ▷ 1º if garante o pico; 2º if garante o hit correto
9:   for  $i$  de 0 até  $n\_hits - 1$  do
10:    for  $j$  de 0 até  $nd\_samples - 1$  do
11:      if  $dsign[j] > 0$  and  $dsign[j + 1] < 0$  and  $sign[j][1] \geq self.limiar$  then
12:        if  $sign[j][0] \geq hits[i][0]$  and  $sign[j][0] \leq hits[i][1]$  then
13:           $hits\_cont.append(tupla([i, hit\_cont, sign[j][0], sign[j][1]]))$ 
14:           $hit\_cont \leftarrow hit\_cont + 1$ 
15:        end if
16:      end if
17:    end for
18:     $num\_peak.append(hit\_cont - 1)$ 
19:     $hit\_cont \leftarrow 1$ 
20:  end for
21:  retornar ( $num\_peak, hits\_cont$ ) ▷ N de picos, lista dos hits com as coord.
22: end function

```

---

**Algorithm 19** amplitude(self)

---

```

1: function AMPLITUDE(self):
2:   ▷ Dados normalizados, derivada do sinal e os hits
3:    $n\_samples, sign \leftarrow \text{___SIGNAL\_LIMAR___}(self)$  ▷ Algoritmo 12
4:    $nd\_samples, dsign \leftarrow \text{___DERIVADA\_FORWARD___}(self)$  ▷ Algoritmo 10
5:    $n\_hits, hits, hits\_time, hits\_sample \leftarrow \text{DURATION}(self)$  ▷ Algoritmo 42
6:    $amplitudes \leftarrow []$ 
7:   ▷ 1º if garante o pico; 2º if garante o hit correto
8:   for  $i$  de 0 até  $n\_hits - 1$  do
9:      $vlr\_max \leftarrow []$ 
10:    for  $j$  de 0 até  $nd\_samples - 1$  do
11:      if  $dsign[j] > 0$  and  $dsign[j + 1] < 0$  and  $sign[j][1] \geq self.limiar$  then
12:        if  $sign[j][0] \geq hits[i][0]$  and  $sign[j][0] \leq hits[i][1]$  then
13:           $vlr\_max.append(sign[j][1])$ 
14:        end if
15:      end if
16:    end for
17:     $amplitudes.append(max(vlr\_max))$  ▷ Lista com todas as amplitudes
18:  end for
19:  retornar  $amplitudes$  ▷ Valor máximo do pico por hit
20: end function

```

---

O módulo `count_peak(.)`, designado como Algoritmo 22, aborda a contagem dos picos até alcançar o pico máximo em cada *hit*. Esse módulo recebe como entrada o número *hits*, informações da “duração” e “amplitude” e fornece como resultado de processamento o número de picos que ocorrem até o valor de maior amplitude.

O módulo `energia(.)` correspondente ao Algoritmo 21, que calcula a magnitude total do evento *Hit* que está relacionado, por sua vez, a “duração” do evento.

---

**Algorithm 20** `count_peak(self)`


---

```

1: function COUNT_PEAK(self):
2:    $n\_samples, sign \leftarrow \text{__SIGNAL\_LIMIAR__}(self)$  ▷ Algoritmo 12
3:    $nd\_samples, dsign \leftarrow \text{__DERIVADA\_FORWARD__}(self)$  ▷ Algoritmo 10
4:    $n\_hits, hits, hits\_time, hits\_sample \leftarrow \text{DURATION}(self)$  ▷ Algoritmo 42
5:   for  $i$  de 0 até  $n\_hits - 1$  do
6:      $peak\_max, cont\_max, contador \leftarrow 0$ 
7:     for  $j$  de 0 até  $nd\_samples - 1$  do
8:       if  $dsign[j] > 0$  and  $dsign[j + 1] < 0$  and  $sign[j][1] \geq self.limiar$  then
9:         if  $sign[j][0] \geq hits[i][0]$  and  $sign[j][0] \leq hits[i][1]$  then
10:           $contador \leftarrow contador + 1$  ▷ Conta todos os picos
11:          if  $peak\_max \leq sign[j][1]$  then
12:             $peak\_max \leftarrow sign[j][1]$ 
13:             $cont\_max \leftarrow contador$  ▷ Recebe o número do pico máximo
14:             $coord\_x, coord\_y \leftarrow sign[j][0], sign[j][1]$ 
15:          end if
16:        end if
17:      end if
18:    end for
19:     $cont\_pico.append(tupla([cont\_max, coord\_x, coord\_y]))$ 
20:  end for
21:  retornar  $cont\_pico$  ▷ Saída: Número do pico, coordenada x e coordenada y
22: end function

```

---



---

**Algorithm 21** `energia(self)`


---

```

1: function ENERGIA(self):
2:    $n\_samples, sign \leftarrow \text{__SIGNAL\_LIMIAR__}(self)$  ▷ Algoritmo 12
3:    $nd\_samples, dsign \leftarrow \text{__DERIVADA\_FORWARD__}(self)$  ▷ Algoritmo 10
4:   for  $j$  de 0 até  $nd\_samples - 1$  do
5:     if  $dsign[j] > 0$  and  $dsign[j + 1] < 0$  and  $sign[j][1] \geq self.limiar$  then
6:       if  $sign[j][0] \geq hits[i][0]$  and  $sign[j][0] \leq hits[i][1]$  then
7:          $contador \leftarrow contador + 1$  ▷ Conta todos os picos
8:       end if
9:     end if
10:  end for
11:  retornar  $cont\_pico$  ▷ Saída: Número do pico, coordenada x e coordenada y
12: end function

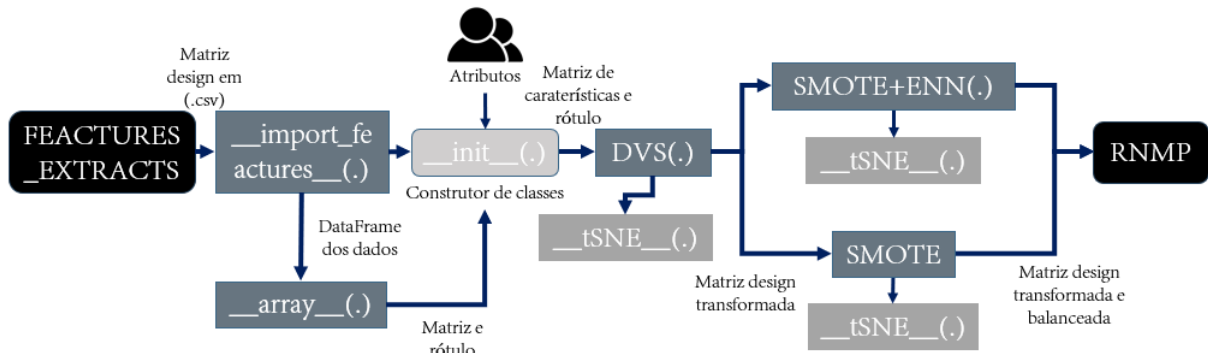
```

---

### 5.1.4 Transformação dos Dados

Esta seção trata dos algoritmos relacionados à segunda fase da extração de características, referida neste trabalho como “transformação dos dados”. Os módulos dessa etapa estão apresentados no fluxograma computacional da Figura 72. Esse fluxo inicia com a definição dos objetos e atributos na função `__init__()`, passando pelo balanceamento da base, que oferece três opções (`undersample()`, `SMOTE()`, `SMOTE_PARTIAL()`), embora que nessa pesquisa tenha sido utilizado foi utilizado somente a parcial e, por fim, utilizou-se a transformação DVS culminando na transformação dos dados em uma base unitária. Esse algoritmo requer como entrada a matriz de *design* e as definições das classes, produzindo como saída uma matriz de *design* transformada, com um número aumentado de amostras.

Figura 72 – Fluxograma computacional do módulo `IMBALANCE_DATA`.



Fonte: Autor.

O módulo `__init__()` é responsável por concentrar os objetos e os atributos informados pelo usuário, tais como as classes (`self.class`) e a matriz de design (`self.design`), assim como suas derivadas, a matriz de características (`self.mat_feact`) e os rótulos (`self.rotulo`).

Além disso, sua função principal é invocar as funções `__import_feactures__()` e `__array__()`, que têm o propósito de ler a matriz de *design* e separar as características dos rótulos em duas novas estruturas de dados, conforme observa-se no Algoritmo 22 quando as funções são atribuídas aos objetos.

---

**Algorithm 22** `__init__(self, desing_file, classes)`

---

- 1: **function** `__INIT__(self, desing_file, classes):`
  - 2:   `self.classesDict ← classes` ▷ Dicionário de classes
  - 3:   `self.classes ← list(classes.keys())`
  - 4:   `self.desing ← __import_feactures__(desing_file)` ▷ Algoritmo 23
  - 5:   `self.mat_feact, self.rotulo ← __array__(self.desing)` ▷ Algoritmo 24
  - 6: **end function**
- 

O módulo `__import_feactures__()` (Algoritmo 23) é responsável por importar as características do arquivo csv que contém a matriz *design*. O Algoritmo 24, denominado

módulo `__array(.)`, tem a função de separar uma matriz com as características e outras com os rótulos para o módulo `SMOTE_PARTIAL(.)`, que lida com o balanceamento parcial dos dados.

O Algoritmo 25, denominado módulo `__DVS__truncated(.)`, é responsável por aplicar a técnica de Decomposição de Valor Singular (DVS) truncado aos dados de entrada, a fim de reduzir a dimensionalidade do conjunto de características. Primeiramente, o algoritmo importa a classe *TruncatedSVD* do módulo *sklearn.decomposition*. Após isso, determina-se o número de componentes principais desejados, optando por utilizar o mesmo número de características presentes na matriz de *design*.

O algoritmo `SMOTE_PARTIAL` tem como objetivo realizar o balanceamento parcial dos dados utilizando a técnica SMOTE. Para isso, primeiramente, importa a biblioteca “pandas” para manipulação de dados tabulares.

Em seguida, são coletados o número de exemplos de cada classe presente na matriz de *design*. Com base numa busca, identifica-se o valor mínimo e máximo de amostras entre as classes. A partir disso, do fator  $x$  (0 a 1) é determinado o número de amostras sintéticas que são criadas (“ $n$ ”), que são inseridas nas classes minoritárias.

O algoritmo então percorre cada classe na matriz de *design*, separando as amostras em dois conjuntos: um para ser balanceado e outro que será responsável por criar o novo desbalanceamento, porém de menor amplitude, denominado por desbalanceamento parcial. Por fim, as amostras balanceadas são combinadas com o conjunto de amostras que criam o desbalanceamento parcial. O algoritmo retorna as características e os rótulos correspondentes das amostras balanceadas parcialmente.

---

**Algorithm 23** `__import_feactures__`(self, features\_file)

---

```

1: function __IMPORT_FEACTURES__(self, features_file)
2:   Import pandas como pd
3:    $df \leftarrow$  pd.read_csv(features_file)           ▷ Ler o arquivo CSV (DataFrame)
4:                                           ▷ Rótulo: String -> Integer
5:    $df['Rotulo'] \leftarrow df['Rotulo'].replace(self.classesDict)$ 
6:   return  $df$                                      ▷ Matriz Design na forma de um DataFrame
7: end function
```

---



---

**Algorithm 24** `__array__`(self, df\_design)

---

```

1: function __ARRAY__(self, df_design)
2:    $matrix\_features \leftarrow df\_design.drop('Rotulo', axis=1)$            ▷ Deletar o rótulo
3:    $rotulo \leftarrow df\_design['Rotulo'].values$                            ▷ Valor do rótulo
4:   return  $matrix\_features, rotulo$            ▷ Matriz Características e lista dos rótulos
5: end function
```

---

**Algorithm 25** DVS\_truncated(self, X\_feact, y\_rotulo)

---

```

1: function DVS_TRUNCATED(self, X_feact, y_rotulo)
2:   Import sklearn.decomposition.TruncatedSVD as TruncatedSVD
3:                                     ▷ Método DVS Truncado
4:    $num\_components \leftarrow 6$                                      ▷ Número de componentes principais (k)
5:    $dvs \leftarrow$  TRUNCATED_SVD( $n\_components=num\_components$ )
6:    $X\_dvs \leftarrow$  SVD.FIT_TRANSFORM( $X\_feact$ ) ▷ Aplicar o DVS Truncado na matriz
7:    $y\_dvs \leftarrow y\_rotulo$                                      ▷ Matriz Características e lista dos rótulos
8:   return  $X\_dvs, y\_dvs$ 
9: end function

```

---

**Algorithm 26** SMOTE\_PARTIAL(self)

---

```

1: function SMOTE_PARTIAL(self, matriz_desing, x)
2:   Import pandas as pd
3:    $list\_samples \leftarrow []$ 
4:   for  $i, class$  em ENUMERATE(self.classes, start=0) do
5:      $num\_samples \leftarrow$  matriz_desing['Rotulo'].value_counts()[ $i$ ]
6:     list_sampel.APPEND(list_samples, num_samples)
7:   end for
8:    $n\_min, n\_max \leftarrow$  min(list_samples), max(list_samples)
9:    $n \leftarrow$  int( $(1 - x) \times n\_max - n\_min$ ) ▷ Nenhuma classe deve ser maior que “n”
10:   $df1, df2 \leftarrow$  pd.DataFrame()
11:  for  $i, class$  em ENUMERATE(self.classes, start=0) do
12:                                     ▷ Só pega o rótulo da vez
13:     $feact\_class \leftarrow$  matriz_desing[matriz_desing['Rotulo'] == self.classesDict[class]]
14:
15:    if ( $len(feact\_class) - n \leq 0$ ) then
16:       $df1 \leftarrow df1.append(feact\_class, ignore\_index = True)$ 
17:    else
18:       $dif \leftarrow len(feact\_class) - n$ 
19:                                     ▷ DataFrame com dados a balancear
20:       $df1 \leftarrow df1.append(feact\_class.head(n), ignore\_index = False)$ 
21:
22:                                     ▷ DataFrame com dados que criarão o desbalanceamento parcial
23:       $df2 \leftarrow df2.append(feact\_class.tail(dif), ignore\_index = False)$ 
24:    end if
25:  end for
26:   $X\_feact, y\_rotulo \leftarrow$  __ARRAY__(self, df1)                                     ▷ Algoritmo 24
27:   $X\_smote, y\_smote \leftarrow$  SMOTE(self, X_feact, y_rotulo)
28:   $matrix\_desing\_smote \leftarrow X\_smote$ 
29:   $matrix\_desing\_smote['Rotulo'] \leftarrow y\_smote$ 
30:   $matrix\_desing\_smote\_partial \leftarrow$  matrix_desing_smote.append(df2)
31:   $X\_smote\_part, y\_smote\_part \leftarrow$  __ARRAY__(self, matrix_smote_part)
32:  return  $X\_smote\_parcial, y\_smote\_parcial$ 
33: end function

```

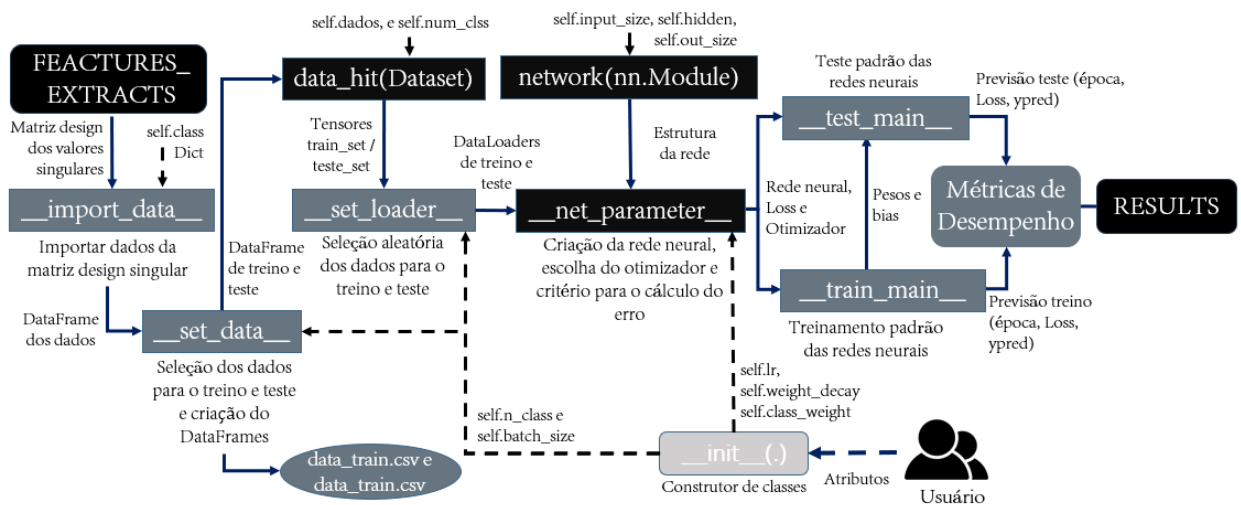
---

### 5.1.5 Aprendizado de Máquina

Esta seção aborda a estrutura do código para a construção do modelo de aprendizado de máquina (módulo RNMP), que utiliza redes neurais multicamadas de perceptrons como classificador. O desenvolvimento deste algoritmo está alinhado com detalhes da arquitetura e hiperparâmetros apresentados na Seção 4.6.

O fluxo computacional do módulo RNMP é ilustrado na Figura 73. Embora não esteja demarcado, esse fluxo é composto por etapas de pré-processamento, para preparar o conjunto de dados de acordo com os requisitos do *PyTorch* (`__import_features__()`, `set_data()`, `set_loader()`), processamento que envolve o cálculo dos pesos, bias e hiperparâmetros (`__net_parameter__()`, `__train__()` e `__test__()`), e, por fim, pós-processamento, por meio da avaliação de desempenho da rede feita pelo módulo *Métricas de Desempenho*. Além disso, essa estrutura interage com as classes `data_hit(DataSet)` e `network(nn.Module)` da biblioteca *PyTorch*, por meio dos módulos `set_dataLoader` e `net_parameter`.

Figura 73 – Fluxograma computacional do módulo RNMP.



Fonte: Autor.

A função `__init__()`, Algoritmo 27, é responsável por concentrar os objetos e atributos de todas as etapas desse módulo. Os parâmetros essenciais para o desenvolvimento desse algoritmo são inicializados, incluindo o tamanho do lote (`batch_size`), o número de épocas (`num_epochs`), a taxa de aprendizado (`lr`), o decaimento de pesos (`weight_decay`), e as características (`features`) extraídas do conjunto de dados. Além disso, o número de classes (`n_class`) é determinado com base no número de características.

Durante o processamento, os parâmetros de entrada (`input_size`), tamanho oculto (`hidden_size`) e tamanho de saída (`out_size`) do modelo são configurados. Adicionalmente, a função `__net_parameter__()` é chamada para inicializar a rede neural, o critério da função de custo e o otimizador, necessários para o treinamento do modelo.

**Algorithm 27** `__init__()`


---

```

1: function __INIT__(self, feactures_file)
2:                                     ▷ Hiperparâmetros: RNMP
3:   self.batch_size  $\leftarrow$  batch_size                                     ▷ Tamanho do batch
4:   self.num_epochs  $\leftarrow$  num_epochs                                     ▷ N de épocas
5:   self.lr  $\leftarrow$  lr                                                 ▷ Taxa de aprendizagem
6:   self.weight_decay  $\leftarrow$  weight_decay                             ▷ Decaimento dos pesos
7:   self.feactures  $\leftarrow$  feactures                                     ▷ Matriz de Características
8:   self.n_clss  $\leftarrow$  len(feactures)                                     ▷ N de classes
9:   self.classesDict  $\leftarrow$  classes
10:  self.classes  $\leftarrow$  list(classes.keys())
11:  self.input_size  $\leftarrow$  input_size                                     ▷ Tamanho da camada de entrada
12:  self.hidden_size  $\leftarrow$  hidden_size                                 ▷ Tamanho da camada oculta
13:  self.out_size  $\leftarrow$  out_size                                       ▷ Tamanho da camada de saída
14:                                     ▷ Funções: Algoritmos 29 e 33
15:  self.desing  $\leftarrow$  self.__import_feactures__(design_file)
16:  self.net, self.criterion, self.optimizer  $\leftarrow$  self.__net_parameter__(.)
17: end function

```

---

O Algoritmo 28 descreve o módulo `__import_feactures__()`, responsável por importar a matriz de design transformada contida em um arquivo “csv”. Utilizando a biblioteca “pandas”, o arquivo “csv” é lido por meio da função `read_csv`, e uma conversão de *strings* para *floats* é realizada. Ao término do processo, o *DataFrame* com rótulos numéricos é retornado pela função.

**Algorithm 28** `__import_feacture__()`


---

```

1: function __IMPORT_FEACTURES__(self, desing_file)
2:   import pandas as pd
3:   df  $\leftarrow$  pd.read_csv(desing_file)                                     ▷ Ler o arquivo CSV (DataFrame)
4:   df['Rotulo']  $\leftarrow$  df['Rotulo'].replace(self.classesDict)
5:   return df
6: end function

```

---

A função `__set_data__()` representada no Algoritmo 29 é a primeira etapa de preparação dos dados no ambiente *Pytorch* para realizar o aprendizado de máquina. A seguir é feita a descrição passo a passo do módulo:

- Importar a biblioteca *torch*;
- Definir uma semente para garantir a reprodutibilidade dos resultados;
- Gerar índices aleatórios para embaralhar os dados, utilizando a função `randperm` do *PyTorch*;
- Definir o tamanho do conjunto de treinamento;
- Dividir em conjuntos de treinamento, validação e teste, com base nos índices aleatórios gerados e gerar os respectivos conjuntos;

- Ajustar os tamanhos dos conjuntos de treino e teste para serem um múltiplo do tamanho do lote (*self.batch\_size*) e garantir uma divisão com números inteiros;
- Salvar os conjuntos de treinamento e teste ajustados em arquivos “csv” denominados “data\_train.csv” e “data\_test.csv”, respectivamente;
- Retornar os conjuntos de treinamento e teste, no formato de *DataFrame*.

---

**Algorithm 29** `__set_data__(.)`


---

```

1: function __SET_DATA__(self)
2:   Import torch
3:                                     ▷ Separação dos dados de treino e teste
4:   torch.manual_seed(1)
5:   indices ← torch.randperm(len(self.desing)).tolist()
6:   train_size ← int(0.8×len(self.desing))
7:   df_train ← self.desing.iloc[indices[:train_size]]
8:   df_test ← self.desing.iloc[indices[train_size:]]
9:                                     ▷ Definição dos tamanhos ajustados
10:  n_train ← len(df_train)
11:  n_test ← len(df_test)
12:  n_new_train ← n_train - n_train % self.batch_size
13:  n_new_test ← n_test - n_test % self.batch_size
14:                                     ▷ Criação dos DataFrames e arquivos “csv”
15:  df_train ← df_train.head(n_new_train)
16:  df_test ← df_test.head(n_new_test)
17:  df_train.to_csv('data_train.csv', index=False)
18:  df_test.to_csv('data_test.csv', index=False)
19:  return (df_train, df_test)                                     ▷ DataFrames de treino e teste
20: end function

```

---

A função *set\_loader(.)* representa a segunda etapa na preparação dos dados dentro do fluxo computacional RNMP. É essencial para garantir um carregamento eficiente dos dados, viabilizando o carregamento simultâneo por meio do parâmetro *num\_workers*, a divisão dos dados em lotes (*batches*) e a utilização de processamento paralelo, especialmente quando combinado com GPUs. A seguir são detalhadas essas etapas pertencentes ao Algoritmo 30.

- Importar a classe *DataLoader* do módulo *torch.utils.data*;
- Definir uma variável (dicionários) com o número de *threads* (*num\_workers*);
- Chamar a função `__set_data__(.)`, que é responsável por obter os conjuntos de dados de treinamento e teste (Algoritmo 29);
- Definir os dados treino e teste com a *data\_loader(Dataset)* (Algoritmo 31);
- Criar os objetos *DataLoader* com as opções de embaralhamento (*shuffle*), tamanho dos lotes (*batch\_size*) e número de *threads* (*num\_workers*).

**Algorithm 30** `set_loader(.)`


---

```

1: function SET_LOADER(self)
2:   from torch.utils.data import DataLoader
3:                                     ▷ Dataset: treino e teste
4:   df_train, df_test ← self.__set_data__()
5:   train_set ← data_hit(df_train, self.num_feact)
6:   test_set ← data_hit(df_test, self.num_feact)
7:                                     ▷ DataLoardes: Treino e teste
8:   train_loader ← DataLoader(train_set, batch_size=self.batch_size, shuffle=True)
9:   test_loader ← DataLoader(test_set, batch_size=self.batch_size, shuffle=True)
10:
11:   return (train_loader, test_loader)    ▷ Objetos DataLoader para treino e teste
12: end function

```

---

O módulo “`data_hit(.)`” corresponde à classe “`torch.utils.data.Dataset`” do *PyTorch* e é empregado para a criação de conjuntos de dados personalizados. Essa classe estabelece a conexão entre os dados e os rótulos. Além disso, também efetua a conversão dos dados, que estão originalmente em formato de *arrays*, para tensores, como ilustrado no Algoritmo 31.

**Algorithm 31** `data_hit(Dataset)`


---

```

1: function __INIT__(self, DataFrame, n_cls)
2:   self.dados ← DATAFRAME(.)TO_NUMPY(())
3:   self.n_cls ← n_cls
4: end function
5:
6: function __GETITEM__(self, idx)                                     ▷ Método para obter um item
7:   sample ← self.dados[idx][0:6]
8:   label ← self.dados[idx][-1:]
9:                                     ▷ Converte para tensor
10:  self.dados ← PD.READ_CSV(csv_path, n_rows=n).TO_NUMPY(())
11:  self.dados ← PD.READ_CSV(csv_path, n_rows=n).TO_NUMPY(())
12:  sample ← TORCH.FROM_NUMPY(sample).TYPE(torch.float32)
13:  label ← TORCH.FROM_NUMPY(label).TYPE(torch.float32)
14:  return sample, label
15: end function
16:
17: function __LEN__(self)
18:   return LEN(self.dados)
19: end function

```

---

O módulo *network()* representado no Algoritmo 32, trata da criação da estrutura da RNMP, sendo uma subclasse da classe *nn.Module* do *PyTorch*. O método `__init__(.)` é o construtor da classe e recebe como argumentos os tamanho da camada de entrada (*input\_size*), camada intermediária (*hidden\_size*) e saída (*out\_size*). Dentro desse método, são definidas as camadas da rede por meio da classe *nn.Sequential*, que permite

empilhar camadas de forma sequencial. As camadas consistem em operações lineares (*nn.Linear*) seguidas de normalização (*nn.BatchNorm1d*) e ativação ReLU (*nn.ReLU*). A saída da rede é gerada pela aplicação da função *softmax* (*nn.Softmax*) na camada de saída. O método *forward* é responsável por realizar a passagem direta dos dados pela rede neural, aplicando as camadas definidas anteriormente e retornando a predição da rede.

---

**Algorithm 32** *network*(*nn.Module*)

---

```

1: function __INIT__(input_size, hidden_size, out_size)
2:   super(network,self).__init__() ▷ Construtor da classe
3:   self.features ← nn.Sequential(
4:     nn.Linear(input_size, hidden_size), ▷ Matriz 1
5:     nn.BatchNorm1d(hidden_size),
6:     nn.ReLU(),
7:     nn.Linear(hidden_size, hidden_size), ▷ Matriz 2
8:     nn.BatchNorm1d(hidden_size),
9:     nn.ReLU()
10:  )
11:
12:  self.out ← nn.Linear(hidden_size, out_size) ▷ Matriz 3
13:  self.softmax ← nn.Softmax()
14: end function
15:
16: function FORWARD(X)
17: ▷ Método para a frente
18:   X ← X.view(X.size(0), -1) ▷ Lineariza o X
19:   feature ← self.features(X)
20:   output ← self.softmax(self.out(feature))
21:   return output
22: end function

```

---

O Algoritmo 33 trata do módulo *\_\_net\_parameter\_\_*(.) que tem por objetivo requisitar o módulo *network*(*nn.Module*) e estabelecer os critérios para calcular o erro, o otimizador empregado e a otimização dos hiperparâmetros.

---

**Algorithm 33** *\_\_net\_parameter\_\_*(.)

---

```

1: function __NET_PARAMETER__(self, lr, weight_decay, class_weights)
2:   import torch
3:   from torch import nn, optim
4:   net ← network(self.input_size, self.hidden_size, self.out_size)
5:   criterion ← nn.CrossEntropyLoss(weight=torch.Tensor(class_weights))
6:   optimizer ← optim.Adam(net.parameters(), lr=lr, weight_decay=weight_decay)
7:   return (net, criterion, optimizer) ▷ Arquitetura RNMP, função custo e
   otimizador
8: end function

```

---

Por fim, são feitos comentários sobre os módulos que abordam as etapas de treino, teste e métricas de avaliação. Os módulos de *\_\_train\_\_*(.) e *\_\_test\_\_*(.) são detalhados

nos Algoritmos 34 e 35, respectivamente. Percebe-se uma semelhança em suas estruturas, exceto pelo fato de que na etapa de treinamento envolve o cálculo do *backpropagation* e a atualização dos pesos. Em ambos os casos, calcula a predição da rede empregando as arquiteturas das redes com os dados organizados em lotes, seguido do cálculo do erro. Para a classificação, define-se o índice da maior probabilidade.

---

**Algorithm 34** `__train__()`


---

```

1: function TRAIN(train_loader, net, epoch)
2:   NET.TRAIN(())
3:
4:   for each batch in train_loader do
5:     Ver linhas 6 à 13 do Algoritmo 34
6:
6:                                     ▷ Backward
7:     OPTIMIZER.ZERO_GRAD              ▷ Zera gradientes
8:     LOSS.BACKWARD                    ▷ Backpropagation
9:     OPTIMIZER.STEP                   ▷ Atualiza parâmetros
10:  end for
11:  return (ypred_array, rotulo_array, Acc)
12: end function

```

---



---

**Algorithm 35** `__test__()`


---

```

1: function TEST(test_loader, net, epoch)
2:   NET.EVAL(())
3:   with TORCH.NO_GRAD(():
4:
5:     for each batch in test_loader do
6:       dado, rotulo ← batch
7:       dado ← dado.TO(torch.float32)
8:       rotulo ← rotulo.TO(torch.long)
9:       rotulo ← rotulo.SQUEEZE
10:
11:       y_pred ← net(dado)                                     ▷ Forward
12:       loss ← criterion(y_pred, rotulo)
13:       max_v, idx_max ← torch.max(y_pred, dim = 1)           ▷ A maior probabilidade
14:     end for
15:     return (ypred_array, rotulo_array, Acc)
16: end function

```

---

Com relação ao módulo de Métricas de Desempenho, tem-se o Algoritmo 36 que calcula a curva ROC, matriz de confusão, acurácia, precisão e *F1-Score*. Esse algoritmo recebe como parâmetro o número de classes do problema *self.n\_cls*, para ser utilizado posteriormente nos cálculos da matriz de confusão. Dentro desse módulo há duas funções, a função *accuracy\_score* calcula a pontuação de acurácia do modelo a partir das previsões (*ypred\_array*) e dos rótulos verdadeiros (*rotulo\_array*) e a função *matriz\_confusao* é responsável por gerar uma matriz que pontua o número de acertos por classe, baseada nas previsões da rede (*list\_ypred*) e rótulos verdadeiros (*list\_rotulo*).

**Algorithm 36** Métricas de Desempenho

---

```

1: function INIT(n_clss)
2:   self.n_clss  $\leftarrow$  n_clss
3: end function
4:
5: function ACCURACY_SCORE(ypred_array, rotulo_array)
6:   Ntot  $\leftarrow$  len(rotulo_arr)
7:   for i in range(Ntot) do
8:     if (ypred_arr[i] == rotulo_arr[i]) then
9:       Nacc += 1
10:    end if
11:  end for
12:  return (Nacc/Ntot)  $\times$  100
13: end function
14:
15: function MATRIZ_CONFUSAO(n_samples, list_ypred, list_rotulo)
16:  mat_confusao  $\leftarrow$  np.zeros(shape=(self.n_clss, self.n_clss), dtype=int)
17:  for i in range(n_samples) do
18:    mat_confusao[vet_ypred[i], vet_rotulo[i]] += 1
19:  end for
20:  return (mat_confusao)
21: end function

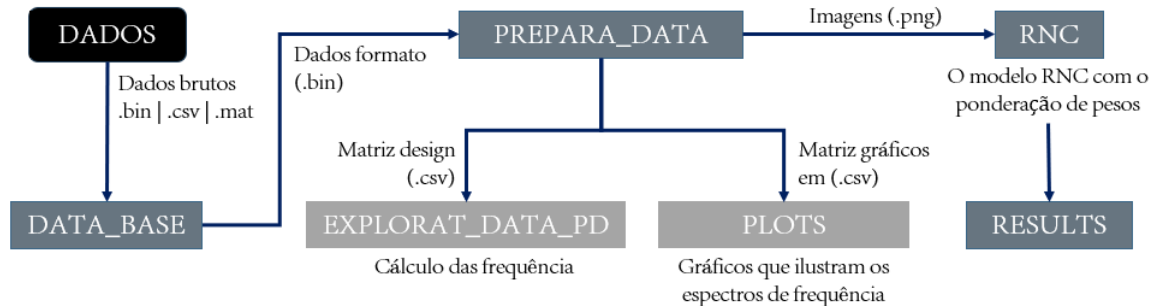
```

---

## 5.2 Modelo RNC

Nesta seção, são apresentados os algoritmos de aprendizado de máquina associados ao modelo RNC. Esses algoritmos foram concebidos para análise de conjuntos de dados compostos por imagens, visando detectar desde falhas em rolamentos até níveis de cavitação, sem depender do tipo de sensor utilizado na coleta de dados. A Figura 74 apresenta o fluxograma computacional do modelo RNC, alinhado ao fluxograma conceitual.

Figura 74 – Fluxograma computacional do modelo RNC.



Fonte: Autor.

Este fluxo computacional abrange os algoritmos utilizados desde o *download* dos dados provenientes do ensaio até a aplicação de técnicas de avaliação dos resultados. Os módulos *PREPARA\_DATA* e *PLOTS* foram desenvolvidos no software *Matlab* em um paradigma procedural, enquanto que os módulos *ARQ\_BIN*, *EXPLORAT\_DATA*,

*RNC* e *RESULTS* são estruturas de dados organizadas em classes, seguindo o paradigma de orientação a objetos. Cada módulo possui requisitos de entrada, realiza um processamento específico e produz uma saída compatível com o próximo módulo.

No Quadro 13, são detalhadas as funções específicas de cada módulo, essenciais para a implementação dos métodos e modelos discutidos na Seção 4.4.2.

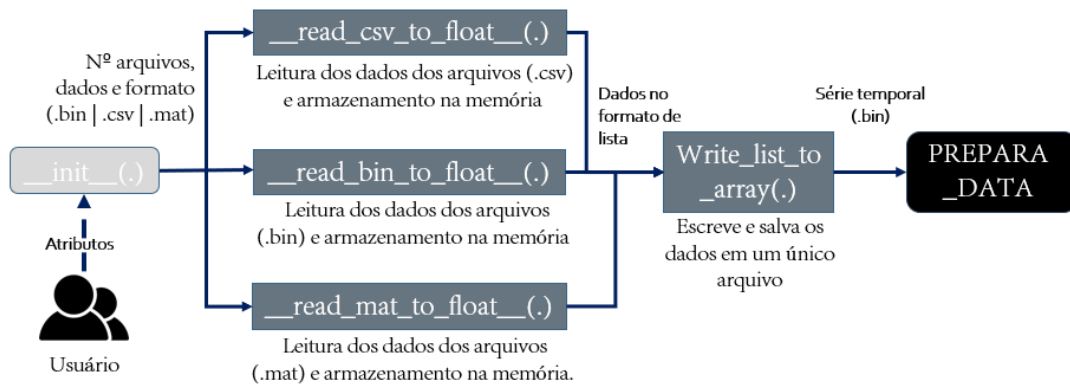
Quadro 13 – Descrição e interação entre os módulos do modelo RNC.

| Módulo        | Descrição                                                                                                                                                                                                                         | Funções                                                                                                                                                                                                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ARQ_BIN       | Realiza a leitura dos arquivos dos dados brutos nos formatos “.bin”, “.csv” e “.mat” e fornece como saída um único arquivo matlab.                                                                                                | As funções desse módulo são:<br><code>__read_bin_to_float__()</code> ,<br><code>__read_csv_to_float__()</code> ,<br><code>__read_mat_to_float__()</code><br><code>__write_list_to_binary__()</code>                                                                                                           |
| PREPARA_DATA  | Realiza a normalização da série, o aumento no número de amostras por janelamento do sinal e a transformação do sinal para o domínio da frequência criando imagens como saída incluindo um arquivo “.csv” com o nomes das imagens. | <code>select_data()</code> , <code>__frequency_synthetic__()</code> ,<br><code>data_augmentation()</code> ,<br><code>data_normalization()</code> , <code>export_data()</code>                                                                                                                                 |
| RNC           | Corresponde a parametrização do modelo e a realização das fases de treino e teste fornecendo como saída os valores de pesos e <i>bias</i> em arquivo <i>python</i> .                                                              | <code>__import_features__()</code> ,<br><code>set_data()</code> , <b><code>data_hit()</code></b> ,<br><code>set_loader()</code> , <code>net_parameter</code> ,<br><code>__train__()</code> , <code>__test__()</code> ,<br><b><code>data_image(Dataset)</code></b> ,<br><b><code>network(nn.Module)</code></b> |
| RESULTS       | Realiza a avaliação do modelo numérico por meio de métricas de desempenho como acurácia, precisão e <i>F1-score</i> no formato médio e matriz confusão                                                                            | <code>acuracia_score()</code> , <code>precision_score()</code> , <code>F1_score</code> ,<br><code>ROC_curve()</code> e <code>matriz_confusao()</code>                                                                                                                                                         |
| EXPLORAT_DATA | Realiza a análise exploratória dos dados a partir dos valores de frequências providenciados pela TRF e fornece como saída um único arquivo “.csv” com o valor de frequência e tamanho da janela retangular                        | <code>__import_features__()</code> ,<br><code>data_rotulo</code> , <code>list_k_interval()</code> ,<br><code>freq_table()</code> e <code>freq_k_table()</code> ,<br><code>freq_analitic()</code>                                                                                                              |

### 5.2.1 Dados

Nesse módulo é realizada a leitura e adequação dos dados disponibilizados pelo ensaio em um ou mais arquivos e em três tipos de formatos (binário, separados por vírgula e matlab). A Figura 75 ilustra o fluxograma computacional do módulo ARQ\_BIN, em que cada bloco representa uma função de código.

Figura 75 – Fluxograma computacional do módulo ARQ\_BIN modelo RNC.



Fonte: Autor.

Este fluxo requer como entrada a série temporal, juntamente com o número e formato dos arquivos. Como parte do processamento, ocorre a consolidação desses dados em uma única série temporal, culminando na geração de um único arquivo de saída do algoritmo.

O Algoritmo 37 corresponde ao módulo *read\_csv\_float*, embora que se aplica ao *read\_bin\_float* e *read\_mat\_float*, enquanto que o Algoritmo 37 refere-se ao módulo *Write\_list\_to\_array*.

### 5.2.2 Preparação dos Dados

A etapa de preparação dos dados foi desenvolvido com o apoio dos softwares *Matlab* e *Python* e inicia com o tratamento da série temporal utilizando o algoritmo denominado por *FFT\_STFT*, cuja função é normalizar os dados, aplicar uma janela retangular, calcular e salvar os espectros de frequência *FFT* e *STFT*. Na sequência é realizada a transformação dos espectros de frequência em imagens, eliminação das bordas, registro das imagens com os rótulos correspondentes e por fim a criação da base de dados em formato adequado ao modelo RNC. Cada uma dessas etapas corresponde a um módulo do Figura 76.

O Algoritmo 38 inicia definindo um diretório de trabalho e carrega arquivos de dados que contêm informações sobre as condições em análise, armazenando-os em uma variável denominada “Faul”. Em seguida, define um vetor de dados (“VetX”) para sensor de extremidade de acionamento (DE) ou de extremidade do ventilador (FE). O algoritmo

**Algorithm 37** ARQ\_BIN(.)

---

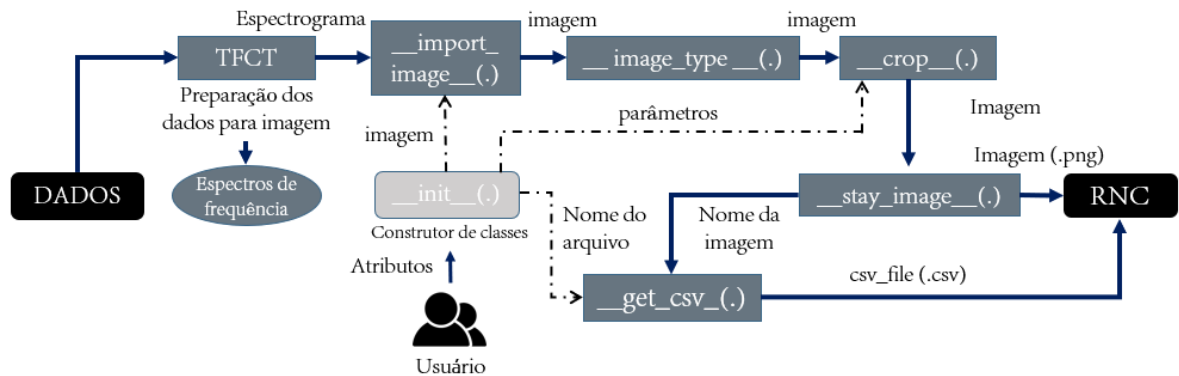
```

1: function __READ_MAT_TO_FLOAT__(self)
2:   Import módulos os e scipy
3:                                     ▷ Download dos dados
4:   mat_data ← carregar arquivo mat (dir_arq)
5:   arq_name ← obter lista de chaves do dicionário mat_data
6:
7:   for all arq em arq_name do:
8:     Adicionar arq à lista_arq_name
9:   end for
10:
11:  for all arq em lista_arq_name do:
12:    if self.sensor = True then:
13:      Adicionar mat_data[arq][i][0] a data
14:    end if
15:  end for
16:  return (num_data, data)
17: end function

```

---

Figura 76 – Fluxograma computacional do módulo PREPARA\_DATA do modelo RNC.



Fonte: Autor.

então itera sobre cada classe de falha, realizando os cálculos de *FFT* e *STFT*. Após cada cálculo, são gerados gráficos e os resultados são salvos constando em seus nomes informações sobre a classe e o intervalo de tempo.

O Algoritmo 39 denominado “image\_type(.)” recebe dois parâmetros: “gray” e “extensao”. Ele itera sobre todos os arquivos no diretório especificado e, para cada arquivo, abre a imagem correspondente. Dependendo do valor do parâmetro “gray”, a imagem é convertida para escala de cinza ou mantida em RGB. Em seguida, a imagem é salva com o nome especificado e a extensão fornecida. Por fim, a imagem é fechada e o processo é repetido para todos os arquivos no diretório.

O Algoritmo 40 realiza o tratamento das bordas por meio de processamento de imagens. Para cada documento presente no diretório especificado, o algoritmo abre a imagem e realiza o corte com base no atributo (*self.crop*) resultando em uma nova imagem

sem as bordas. Esse algoritmo realiza o processo em toda a base de imagens facilitando a análise subsequente dos documentos.

O Algoritmo 41 é responsável pelo processo de cadastro do rótulo da imagem com a imagem, visando criar um arquivo “csv” contendo informações sobre as imagens e seus respectivos rótulos. Essa organização e registro ocorre em um arquivo de formato *csv*.

---

**Algorithm 38** FFT\_STFT(.)
 

---

```

1: Definir DIR como “Diretório”
2: Change o diretório para DIR
3: Fault  $\leftarrow$  load(Arquivos)  $\therefore$  Fault contém B007, ..., OR021
4: define VetX como sensor DE ou FE
5:
6: Tini  $\leftarrow$  0.00
7: FS  $\leftarrow$  Frequência de aquisição de dados
8: DeltaT  $\leftarrow$  Tamanho da janela retangular
9: Tend_Janela  $\leftarrow$  DeltaT
10:
11: for class = B007, ..., OR021 do
12:   Tend  $\leftarrow$  Tempo do experimento
13:   N  $\leftarrow$  Tend/FS
14:
15:   for i = 1, 2, ..., N do
16:     VetT_Janela  $\leftarrow$  CriarVetor(Tini, Tend_janela)
17:     VetX_Janela  $\leftarrow$  VetX[Tini : Tend_janela]
18:
19:     CalculoFFT(Vet_X, FS)
20:     CalculoSTFT(Vet_X, FS)
21:
22:     plotarGrafico(VETX)
23:     plotarGrafico(FFT)
24:     plotarGrafico(STFT)
25:
26:     As variaveis class, VetT(i), VetT(i + 1) estão no formato string
27:     NameTime  $\leftarrow$  Concatenar(class, VetT(i), VetT(i + 1), 'TIME')
28:     NameFFT  $\leftarrow$  Concatenar(class, VetT(i), VetT(i + 1), 'FFT')
29:     NameSTFT  $\leftarrow$  Concatenar(class, VetT(i), VetT(i + 1), 'STFT')
30:
31:     SalvarGrafico(NameTime)
32:     SalvarGrafico(NameFFT)
33:     SalvarGrafico(NameSTFT)
34:
35:     Tini  $\leftarrow$  Tini + DeltaT
36:     Tend_Janela  $\leftarrow$  Tend_Janela + DeltaT
37:   end for
38:
39: end for

```

---

---

**Algorithm 39** `__image_type__()`

---

```

1: function __IMAGE_TYPE__(self, gray, extensao)
2:   for all arq in ListarDiretorio() do
3:     image_emf  $\leftarrow$  AbrirImagem(dir_arq)
4:     if gray = Falso then
5:       image  $\leftarrow$  ConverterParaRGB(image_emf)
6:     else
7:       image  $\leftarrow$  ConverterParaL(image_emf)
8:     end if
9:     SalvarImagem(image, image_name, extensao)
10:    FecharImagem(image)
11:   end for
12: end function

```

---



---

**Algorithm 40** `__crop__()`

---

```

1: function TRATAR_BORDAS
2:   for all doc in ListarDiretorio() do
3:     image  $\leftarrow$  AbrirImagem(dir_Arq)
4:     crop_size  $\leftarrow$  (330, 410)
5:     center_crop  $\leftarrow$  CriarTransformacaoCorteCentro(crop_size)
6:     cropped_image  $\leftarrow$  AplicarTransformacao(image, center_crop)
7:     doc_crop  $\leftarrow$  'CROP_' + doc
8:     SalvarImagem(cropped_image, dir_crop)
9:   end for
10: end function

```

---



---

**Algorithm 41** `__get_csv__()`

---

```

1: function CRIAR_CSV
2:   lista_bearing  $\leftarrow$  []
3:   for all arq in ListarDiretorio() do
4:     lista_split  $\leftarrow$  DividirArquivo(arq, underline_)
5:     for i de 0 até Comprimento(classes) do
6:       if lista_bearing  $\neq$  'desktop.ini' then
7:         RemoverArquivo('desktop.ini')
8:       end if
9:       if lista_split[1] = classes[i] then
10:        lista_bearing.Adicionar([arq, classes[i]])
11:      end if
12:    end for
13:  end for
14:  saida  $\leftarrow$  AbrirArquivo(self.csv_file, 'w', sem_quebra_de_linha)
15:  escrever  $\leftarrow$  CriarEscritorCSV(saida)
16:  for i de 0 até Comprimento(lista_bearing) do
17:    escrever.EscreverLinha(lista_bearing[i])
18:  end for
19:  FecharArquivo(saida)
20: end function

```

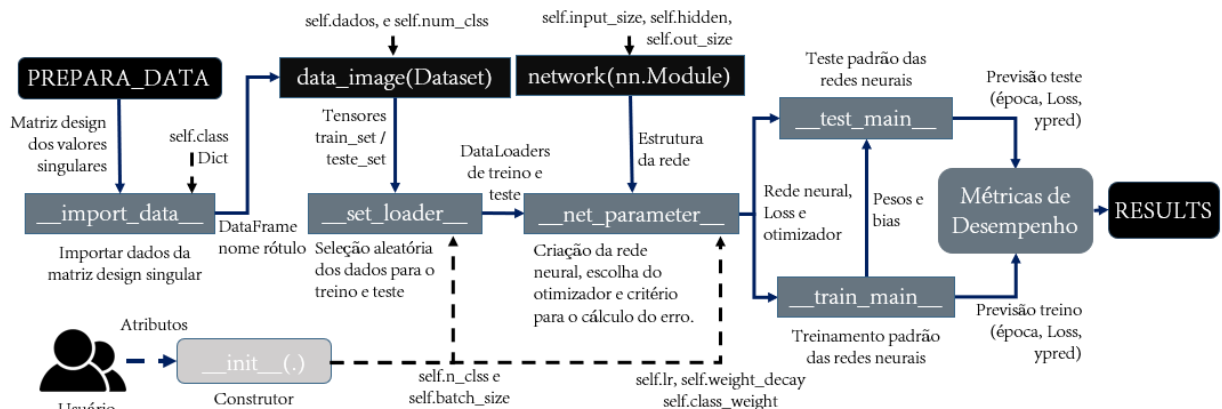
---

### 5.2.3 Aprendizado de Máquina

Esta seção aborda a estrutura do código para o modelo de aprendizado de máquina (módulo RNC), que utiliza redes neurais convolucionais como classificador. O desenvolvimento deste algoritmo está alinhado com detalhes da arquitetura e hiperparâmetros apresentados na **seção 4.6**.

O fluxo computacional do módulo RNC é ilustrado na Figura 77. Esse fluxo se assemelha ao fluxo RNMP, devido a *FrameWork PyTorch*. Essa estrutura possui uma etapa de pré-processamento, para preparar o conjunto de dados (`__import_image__()`, `data_image(Dataset) set_loader()`), processamento que envolve a parametrização e cálculo dos pesos, bias e hiperparâmetros (`network(nn.Module)`, `__net_parameter__()`, `__train__()` e `__test__()`), e, por fim, pós-processamento, por meio da avaliação de desempenho da rede realizada pelo módulo *Métricas de Desempenho*. Deve-se ressaltar, que os módulos `data_image(DataSet)` e `network(nn.Module)` são estruturas de classes externas da biblioteca PyTorch, que interagem com os módulos `set_loader` e `net_parameter`.

Figura 77 – Fluxograma computacional do módulo RNC.



Fonte: Autor.

A função `__init__()`, no Algoritmo 42, é responsável por concentrar os objetos e atributos de todas as etapas desse módulo. Os parâmetros essenciais para o desenvolvimento desse algoritmo são inicializados, incluindo o tamanho do lote (`batch_size`), o número de épocas (`num_epochs`), a taxa de aprendizado (`lr`), o decaimento de pesos (`weight_decay`), e as características (`features`) extraídas do conjunto de dados. Além disso, o número de classes (`n_cls`) é determinado com base no número de características.

Durante o processamento, os parâmetros referentes à arquitetura da rede são configurados. São eles: número de entrada (`input_size`), número de camadas de convolução (`hidden_size`), camadas de normalização (`hidden_size`), camadas *pooling* (`hidden_size`), funções de ativação (`hidden_size`), número de camadas e de neurônios intermediários (`hidden_size`), (`hidden_size`) e tamanho de saída (`out_size`). Adicionalmente, a função

`__net_parameter__()` é chamada para inicializar a rede neural, o critério da função de custo e o otimizador, necessários para o treino do modelo.

---

**Algorithm 42** `__init__()`


---

```

1: function __INIT__(self, feactures_file)
2:                                     ▷ Hiperparâmetros: RNC
3:   self.batch_size  $\leftarrow$  batch_size                                     ▷ Tamanho do batch
4:   self.num_epochs  $\leftarrow$  num_epochs                                     ▷ N de épocas
5:   self.lr  $\leftarrow$  lr                                                 ▷ Taxa de aprendizagem
6:   self.weight_decay  $\leftarrow$  weight_decay                             ▷ Decaimento dos pesos
7:   self.feactures  $\leftarrow$  feactures                                     ▷ Matriz de Características
8:   self.n_clss  $\leftarrow$  len(feactures)                                   ▷ N de classes
9:   self.classesDict  $\leftarrow$  classes
10:  self.classes  $\leftarrow$  list(classes.keys())
11:  self.input_size  $\leftarrow$  input_size                                     ▷ Tamanho da camada de entrada
12:  self.hidden_size  $\leftarrow$  hidden_size                                 ▷ Tamanho da camada oculta
13:  self.out_size  $\leftarrow$  out_size                                       ▷ Tamanho da camada de saída
14:                                     ▷ Funções: Algoritmos 43 e 50
15:  self.desing  $\leftarrow$  self.__import_feactures__(design_file)
16:  self.net, self.criterion, self.optimizer  $\leftarrow$  self.__net_parameter__()
17: end function

```

---

O Algoritmo 43 descreve o módulo `__import_data__()`, responsável por importar a matriz de *design* transformada contida em um arquivo “csv”. O arquivo “csv” é lido por meio da função `read_csv`, e uma conversão de *strings* para *floats* é realizada.

---

**Algorithm 43** `__import_data__()`


---

```

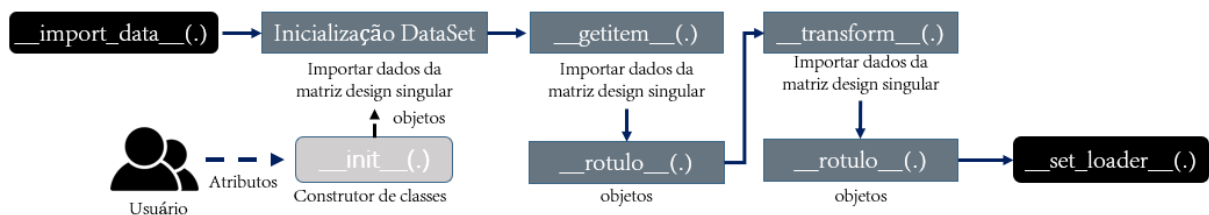
1: function __IMPORT_FEACTURES__(self, desing_file)
2:   image  $\leftarrow$  read_image(img_path)
3:   return image
4: end function

```

---

Com relação ao módulo `data_image(Dataset)` existe um conjunto de funções para a transformação em uma imagem com escala de cores (0 a 255) e conversão para tensor, como o redimensionamento, recortes e conversão de escala cinza ou RGB. A Figura 78, detalha o procedimento para o tratamento dos dados e criação em imagens.

Figura 78 – Fluxograma computacional do módulo `data_image(Dataset)`.



**Algorithm 44** Inicialização do Dataset

---

```

1: function __INIT__(annotations_file, img_dir, train, classes, transform=None, tar-
   get_transform=None)
2:   os.chdir(img_dir)
3:   self.img_labels = pd.read_csv(annotations_file)
4:   self.img_dir = img_dir
5:   self.transform = transform
6:   self.target_transform = target_transform
7:   self.train = train
8:   self.classes = classes
9: end function

```

---

A função `__getitem__()` do módulo `image_data(Dataset)` acessa as imagens e os índices do arquivo `csv` ao converter em uma lista. A imagem é então lida usando a função `read_image` e o rótulo é extraído do `Dataset`. Após isso, a imagem passa por uma série de transformações definidas pela função `__transforms__()` para prepará-la para ser usada no treinamento do modelo. O rótulo é obtido pela função `__rotulo__()`. Finalmente, a função retorna a imagem transformada (`imageT`) e o rótulo correspondente (`rotulo`).

**Algorithm 45** Obtenção de Item do Dataset

---

```

1: function __GETITEM__(self, idx)
2:   if torch.is_tensor(idx) then
3:     idx = idx.tolist()
4:   end if
5:   img_path = os.path.join(self.img_dir, self.img_labels.iloc[idx, 0])
6:   image = read_image(img_path)
7:   label = self.img_labels.iloc[idx, 1]
8:   imageT = self.__transforms__(image)
9:   rotulo = self.__rotulo__(label)
10:  return imageT, rotulo
11: end function

```

---

O módulo `__transform__()` que está representado pelo Algoritmo 46, corresponde a terceira etapa da criação do `image_data(Dataset)`. Esse algoritmo realiza transformações sobre a imagem, como o redimensionamento de  $32 \times 32$  pixels, recorte das bordas e conversão para a escala cinza.

A função `__rotulo__()` é responsável por atribuir um rótulo numérico a partir de uma classe especificada, conforme apresentado no Algoritmo 47. Por fim, o Algoritmo 48, responsável por separar as imagens de treino e as imagens de teste de forma aleatória por meio dos nomes listados no arquivo `csv`.

**Algorithm 46** Função `__transforms__()`


---

```

1: function __TRANSFORMS__(self, image)
2:   # Transformações Individuais:
3:   resize_transform = T.Resize(size=(32,32))
4:   crop_transform = T.RandomCrop((32, 32))
5:   gray_transform = T.Grayscale(num_output_channels=1)
6:
7:   # Transformação Composta:
8:   transform = T.Compose([
9:       resize_transform,
10:      crop_transform,
11:      gray_transform
12:   ])
13:
14:   # Transformações na imagem:
15:   image_transformed = transform(image)
16:   return image_transformed
17: end function

```

---

**Algorithm 47** Obtenção do Rótulo

---

```

1: function __ROTULO__(self, label)
2:   rot_image = 0
3:   for i in range(len(self.classes)) do
4:     if self.classes[i] == label then
5:       rot_image = i
6:     end if
7:   end for
8:   return rot_image
9: end function

```

---

**Algorithm 48** Dados de Treino ou Teste

---

```

1: function __TRAIN_DATA__(self)
2:   torch.manual_seed(1)
3:   df = pd.read_csv(self.img_labels)
4:   indices = torch.randperm(len(df)).tolist()
5:   train_size = int(1 * len(df))
6:   if self.train == True then
7:     df_train = df.iloc[indices[: train_size]]
8:     return df_train
9:   else
10:    df_test = df.iloc[indices[train_size :]]
11:    return df_test
12:   end if
13: end function

```

---

O módulo `network()` representado no Algoritmo 49, trata da criação da estrutura da RNC, sendo uma subclasse da classe `nn.Module` do *PyTorch*. O método `__init__()`

é o construtor da classe e recebe como argumentos o tamanho e o número de *kernels*, tamanho dos filtros *pooling*, funções de ativação, número de neurônios das camadas intermediárias (*hidden\_size*) e saída (*out\_size*).

Dentro do modelo apresentado, a definição das camadas da rede é realizada de forma organizada e sequencial utilizando a classe *nn.Sequential*. Essa arquitetura é composta por três blocos convolucionais, em que cada bloco inclui uma camada de convolução, normalização (*Batch Normalization*), funções de ativação e amostragem (*pooling*), seguido da tradicional rede totalmente conectada, sendo que os valores dos parâmetros são apresentados no Algoritmo 50.

---

**Algorithm 49** `network(nn.Module)`


---

```

1: function __INIT__(input_size, hidden_size, out_size)
2:   super(network,self).__init__() ▷ Construtor da classe
3:   self.features ← nn.Sequential(
4: ▷ 1º Bloco Convolucional
5:     nn.Conv2d(in_channels=1, out_channels=6, kernel_size=5),
6:     nn.BatchNorm2d(num_features=6),
7:     nn.Tanh(),
8:     nn.AvgPool2d(kernel_size=2),
9:
10: ▷ 2º Bloco Convolucional
11:     nn.Conv2d(in_channels=6, out_channels=16, kernel_size=5),
12:     nn.BatchNorm2d(num_features=16),
13:     nn.Tanh(),
14:     nn.AvgPool2d(kernel_size=2),
15:
16: ▷ 3º Bloco Convolucional
17:     nn.Conv2d(in_channels=16, out_channels=120, kernel_size=5),
18:     nn.BatchNorm2d(num_features=120),
19:     nn.Tanh(),
20:     nn.Flatten(),
21:
22: ▷ Camadas Totalmente Conectadas
23:     nn.Linear(in_features=120, out_features=84, bias=True),
24:     nn.Tanh(),
25:     nn.Linear(in_features=84, out_features=self.out_size, bias=True), )
26: end function
27:
28: function FORWARD(X)
29: ▷ Método para a frente
30:    $X \leftarrow X.view(X.size(0), -1)$  ▷ Lineariza o X
31:   feature ← self.features(X)
32:   output ← self.softmax(self.out(feature))
33:   return output
34: end function

```

---

O Algoritmo 50 trata do módulo `__net_parameter(.)` que requisita o módulo `network(nn.Module)` e estabelece os critérios para calcular o erro, o otimizador empregado e a otimização dos hiperparâmetros.

---

**Algorithm 50** `__net_parameter__()`


---

```

1: function __NET_PARAMETER__(self, lr, weight_decay, class_weights)
2:   import torch
3:   from torch import nn, optim
4:   net ← network(self.input_size, self.hidden_size, self.out_size)
5:   criterion ← nn.CrossEntropyLoss(weight=torch.Tensor(class_weights))
6:   optimizer ← optim.Adam(net.parameters(), lr=lr, weight_decay=weight_decay)
7:   return (net, criterion, optimizer) ▷ Arquitetura RNC, função custo e otimizador
8: end function

```

---

Por fim, são tratados os módulos responsáveis pelas etapas de treinamento, teste e avaliação das métricas para o modelo RNC. Estes módulos (`__train__()`, `__test__()` e *Métricas de Avaliação*) seguem a mesma estrutura de código e lógica de implementação à da Rede Neural Multilayer Perceptron (RNMP), uma vez que utilizam o *PyTorch* como ferramenta de desenvolvimento. Assim, os Algoritmos 34, 35 e 36 delineiam minuciosamente a etapa de cálculo dos pesos e bias no fluxo computacional do modelo de Rede Neural Convolutiva (RNC), conforme apresentado na Figura 77. Estes algoritmos encapsulam o processo iterativo de ajuste dos parâmetros da rede neural durante o treinamento, a avaliação do desempenho do modelo sobre o conjunto de teste e a computação das métricas de avaliação para aferir a qualidade da classificação realizada pela RNC.

Para informações desses três módulos, recomenda-se ver os Algoritmos 33, 34 e 35.

## 6 Análise de Resultados

Nesta seção, são apresentados e discutidos os resultados obtidos com as modelagens computacionais RNMP e RNC, aplicadas aos conjuntos de dados de cavitação em rotores hidráulicos e falhas em rolamentos mecânicos, captados por acelerômetros instalados nos mancais do motor elétrico. Para ambas as bases de dados, as modelagens computacionais descritas nas Seções 4 e 5 foram mantidas inalteradas, evidenciando a robustez dos modelos de aprendizado de máquina e a metodologia empregada.

Esta seção está organizada por conjuntos de dados, começando com os dados de rolamento e, em seguida, abordando os dados de cavitação. Para cada conjunto, foi realizada uma análise exploratória para definir o valor limiar e o tamanho da janela TFCT. O valor limiar é determinado com base na estatística descritiva dos sinais, enquanto o tamanho da janela é estabelecido por meio de uma análise de sinais (TRF). Em seguida, são apresentados os resultados obtidos pelos modelos RNMP e RNC, abrangendo a fase de preparação dos dados, extração de características, classificação por meio de matriz de confusão e métricas de desempenho. Por fim, esses resultados são comparados com o artigo de referência por meio de testes de hipóteses, como: análise de variância combinada com Tukey e análise de Kruskal Wallis e Dunn.

### 6.1 Análise Exploratória dos Dados de Rolamento

#### 6.1.1 Definição dos Rótulos

Os dados estão organizados conforme o tipo, a intensidade de falha e a carga aplicada ao motor elétrico, resultando em 40 rótulos. No entanto, é possível agrupar esses rótulos de forma a abstrair a carga do motor, permitindo observar apenas o tipo e a intensidade de falha, conforme (ZHANG et al., 2020). Nesta seção, propõe-se uma análise estatística sobre o valor médio do sinal para a constatação ou criação novos rótulos.

Os rótulos dos conjuntos de dados foram definidos com base no valor médio do sinal referente a cada condição de falha. Essa rotulagem dos dados incluiu várias análises estatísticas, tais como: histogramas e gráficos  $Q - Q$  para uma análise qualitativas, testes de Shapiro-Wilk e kolmogorov-Smirnov para a verificação da normalidade, o teste de Levene no caso variância, t de Student, t de Welch, Mann-Whitney U, e, por fim, a análise de variância (ANOVA) combinada com o teste de Tukey, além dos testes de Kruskal-Wallis e Dunn, com o intuito de verificar a igualdade entre médias.

Percebe-se que existem testes de hipóteses nas situações em que o conjunto de dados apresentam um comportamento normal e na situação em que não apresentam um

comportamento normal. Isso foi necessário, pois na maioria das vezes as informações eram inconclusíveis ou contraditórias, especialmente na condição da normalidade dos dados, ou seja os testes Shapiro-Wilk e kolmogorov-Smirnov rejeitavam a hipótese nula enquanto que os histogramas e gráficos  $Q - Q$  ilustravam um comportamento de normalidade, independente do tamanho da amostra. Por tanto, todas as análises foram realizadas com o intuito de verificar a consistência das informações contidas na base de dados.

Esses testes também foram realizados com diferentes tamanhos de amostras -  $N$ ,  $N/10$ ,  $N/100$  e  $N/1000$  - para diagnosticar a sensibilidade da análise em relação ao conjunto em estudo, em que  $N$  é o tamanho do conjunto de dados. Na Tabela 30 são apresentados o números de amostras contidas em cada ensaio.

Tabela 30 – Tamanhos das amostras para o estudo estatístico.

| Carga      | <i>NRM</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>3HP</i> | 485.643    | 488.545    | 486.804    | 486.804    | 485.643     | 485.063     | 489.125     | 484.483     | 488.545     | 486.224     |
| <i>2HP</i> | 485.063    | 486.804    | 487.384    | 487.384    | 485.643     | 487.964     | 491.446     | 483.323     | 486.804     | 486.804     |
| <i>1HP</i> | 483.903    | 487.384    | 486.224    | 486.804    | 486.224     | 48.915      | 485.063     | 482.742     | 484.483     | 486.224     |
| <i>0HP</i> | 243.938    | 244.739    | 249.146    | 243.938    | 243.938     | 63.788      | 244.339     | 129.969     | 245.140     | 130.549     |

Dentro desse contexto de estudos, foram selecionados os tamanhos de amostra  $n = N/100$  e  $n = N/1.000$  para a escolha dos grupos que formam os rótulos, em que  $N$  é o número de amostras. Esses tamanhos de amostra estão em conformidade com os requisitos dos métodos estatísticos que são utilizados na biblioteca *statsmodels* na linguagem de programação *Python*, que recomenda amostras menores que 5.000 ( $n < 5.000$ ).

A seguir são apresentados os testes de normalidade, variância e os de hipóteses, especificamente, a análise de variância combinada com o teste de Tukey e o teste de Kruskal-Wallis combinado com o teste de Dunn.

- Teste de Shapiro-Wilk: O  $p$  – valor destes testes são apresentados nas Tabelas 31 e 32 para as quantidades  $N/100$  e  $N/1000$ , respectivamente. Como esses valores são menores que 0,05, rejeita-se a hipótese nula, pois os dados não seguem uma distribuição normal.

Tabela 31 – Valores –  $p$  do teste de Shapiro-Wilk para tamanho  $N/100$ .

| Carga      | <i>NRM</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>3HP</i> | 1,5E-03    | 9,7E-02    | 2,4E-36    | 3,4E-01    | 1,5E-40     | 3,6E-68     | 1,1E-05     | 7,7E-51     | 5,1E-09     | 5,8E-70     |
| <i>2HP</i> | 2,2E-11    | 8,6E-06    | 2,9E-66    | 1,5E-01    | 6,8E-42     | 8,7E-59     | 3,1E-18     | 4,5E-50     | 6,3E-05     | 1,1E-73     |
| <i>1HP</i> | 2,9E-11    | 3,9E-01    | 1,1E-43    | 3,5E-03    | 3,4E-43     | 6,6E-60     | 1,2E-04     | 1,8E-53     | 1,3E-03     | 1,7E-67     |
| <i>0HP</i> | 1,1E-02    | 6,8E-03    | 6,6E-18    | 8,9E-03    | 4,1E-26     | 6,6E-16     | 1,9E-11     | 4,3E-01     | 8,2E-03     | 5,2E-40     |

Nota: valor –  $p < 0,05$  - Rejeita  $H_0$ , os modelos não seguem uma distribuição normal

Tabela 32 – *Valores – p* do teste de Shapiro-Wilk para tamanho  $N/1.000$ .

| Carga      | <i>NRM</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>3HP</i> | 2,7E-01    | 2,8E-01    | 2,2E-01    | 7,4E-01    | 6,3E-14     | 2,9E-04     | 1,2E-01     | 2,2E-01     | 7,0E-04     | 5,3E-02     |
| <i>2HP</i> | 3,5E-01    | 8,8E-04    | 2,0E-16    | 9,6E-02    | 1,1E-16     | 1,6E-06     | 9,6E-06     | 3,3E-13     | 3,1E-07     | 3,0E-04     |
| <i>1HP</i> | 4,6E-06    | 7,9E-03    | 2,5E-07    | 1,0E-01    | 4,6E-12     | 6,7E-08     | 8,3E-02     | 1,9E-11     | 2,3E-05     | 1,5E-03     |
| <i>0HP</i> | 4,1E-03    | 4,2E-01    | 2,4E-02    | 2,1E-01    | 1,9E-09     | 1,7E-03     | 7,7E-02     | 3,4E-03     | 1,0E-02     | 3,8E-01     |

Nota: *valor – p* < 0,05 - Rejeita  $H_0$ , os modelos não seguem um distribuição normal

- Teste de Kolmogorov-Smirnov: O *p – valor* destes testes são apresentados nas Tabelas 33 e 34 para as quantidades  $N/100$  e  $N/1000$ , respectivamente. Como esses valores são menores que 0,05, rejeita-se a hipótese nula, pois os dados não seguem uma distribuição normal.

Tabela 33 – *Valores – p* do teste de Kolmogorov-Smirnov para tamanho  $N/100$ .

| Carga      | <i>NRM</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>3HP</i> | 0E+00      | 0E+00      | 0E+00      | 0E+00      | 0E+00       | 0E+00       | 4E-189      | 0E+00       | 0E+00       | 0E+00       |
| <i>2HP</i> | 0E+00      | 0E+00      | 0E+00      | 0E+00      | 0E+00       | 0E+00       | 1,5E-44     | 0E+00       | 0E+00       | 0E+00       |
| <i>1HP</i> | 0E+00      | 0E+00      | 0E+00      | 0E+00      | 0E+00       | 0E+00       | 1,5E-70     | 0E+00       | 0E+00       | 0E+00       |
| <i>0HP</i> | 0E+00      | 2E-296     | 0E+00      | 3E-276     | 2,6E-82     | 6,1E-74     | 1,2E-34     | 3E-164      | 0E+00       | 1E-114      |

Nota: *valor – p* < 0,05 - Rejeita  $H_0$ , os modelos não seguem um distribuição normal

Tabela 34 – *Valores – p* do teste de Kolmogorov-Smirnov para tamanho  $N/1.000$ .

| Carga      | <i>NRM</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>3HP</i> | 1,5E-85    | 1,7E-63    | 4,4E-61    | 2,9E-52    | 1,7E-38     | 4,4E-84     | 5,8E-20     | 2,2E-51     | 9,1E-62     | 8,8E-61     |
| <i>2HP</i> | 3,5E-87    | 2,3E-58    | 1,9E-50    | 7,6E-56    | 1,9E-39     | 1,5E-52     | 3,7E-02     | 8,8E-40     | 3,0E-57     | 4,3E-71     |
| <i>1HP</i> | 1,7E-86    | 3,3E-74    | 9,8E-53    | 1,9E-70    | 5,5E-38     | 1,0E-34     | 1,7E-09     | 2,6E-19     | 4,3E-77     | 1,0E-40     |
| <i>0HP</i> | 1,3E-39    | 7,5E-33    | 8,9E-42    | 1,8E-33    | 1,0E-10     | 1,6E-11     | 5,1E-01     | 5,9E-21     | 7,2E-79     | 3,0E-25     |

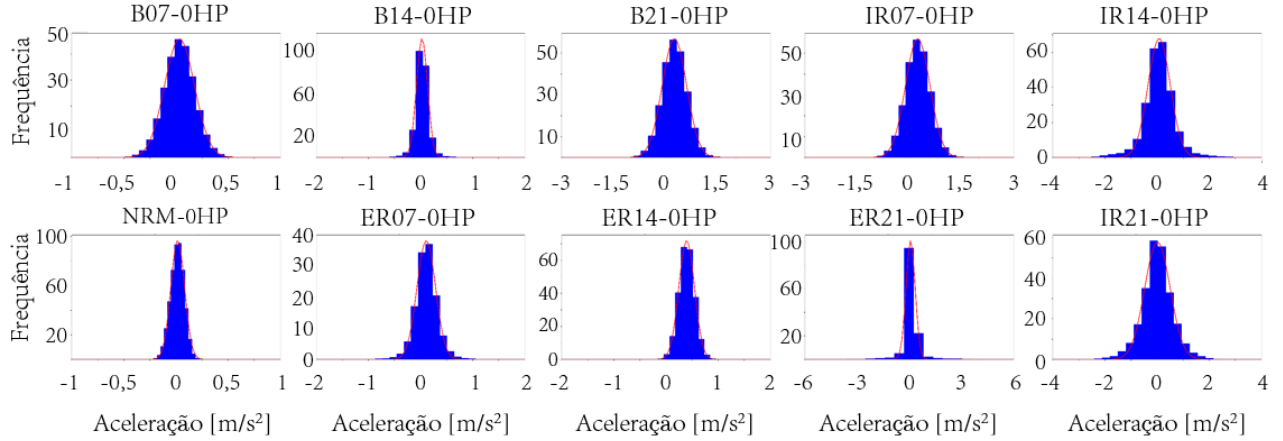
Nota: *valor – p* < 0,05 - Rejeita  $H_0$ , os modelos não seguem um distribuição normal

- Histograma e gráficos  $Q-Q$ : Para entender a distribuição dos dados são utilizados os histogramas e os gráficos  $Q-Q$  (*Quantile-Quantile*). No histograma busca visualizar simetrias, tendência central, dispersão (largura dos intervalos) e outliers. No gráficos  $Q-Q$  deseja observar se os resíduos estão conforme uma distribuição normal.
  - **Linha reta**: Segue uma distribuição normal.
  - **Curva Côncava**: Sugere uma distribuição leptocúrtica.
  - **Curva Convexa**: Sugere uma distribuição platicúrtica.

Na Figura 79 são apresentados os histogramas das falhas com potência de eixo de *0HP*. É possível observar uma tendência de comportamento normal para todas as condições, que se torna mais evidente com o aumento no número de amostras.

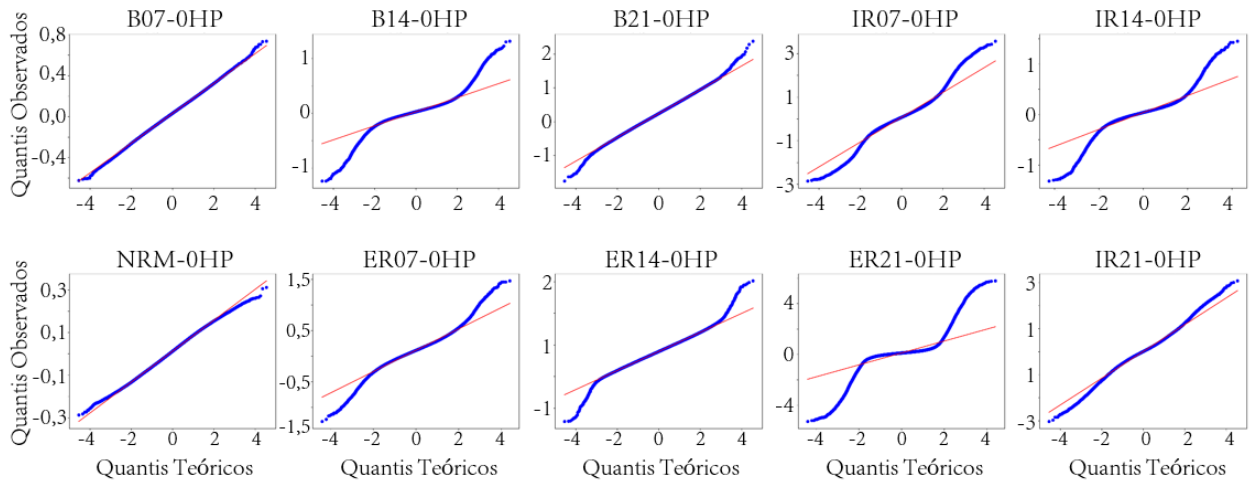
No entanto, nos casos de *B14* e *ER21*, essa distribuição apresentaram as maiores distorções em comparação com os demais.

Figura 79 – Histogramas dos dados de potência com 0HP com  $N/100$ .



Na Figura 80 são apresentados os gráficos  $Q - Q$  das falhas com potência de eixo de 0HP, que correspondem por sua vez aos histogramas da Figura 79. É possível observar uma tendência de comportamento normal para todas as condições, que se torna mais evidente com o aumento no número de amostras. No entanto, nos casos de *IR14* e *ER121*, essa distribuição apresentaram as maiores distorções em comparação com os demais.

Figura 80 – Gráficos  $Q - Q$  para a potência de eixo de 0HP com  $N/100$ .



- Teste de Levene: Com esse teste busca verificar se duas séries temporais possuem variâncias iguais. Por possuir várias séries temporais foi realizado esse teste de forma pareada ao ponto de cobrir todas combinações sem repetição. Os resultados são apresentados na Tabela 35 por meio do  $p$ -valor. Como estes valores são menores que 0.05, rejeita-se a hipótese nula de que as variâncias são iguais para todas as condições

operativas. Isto indica que as variâncias entre os grupos não são homogêneas, o que pode aumentar a eficácia dos modelos de classificação.

Tabela 35 – *Valor – p* do teste de Levene com tamanho  $N/100$ .

| Carga          | <i>NRM</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> |
|----------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>0HP 1HP</i> | 0E+00      | 5E-97      | 2E-173     | 0E+00      | 0E+00       | 2E-312      | 3E-177      | 0E+00       | 0E+00       | 2E-05       |
| <i>0HP 2HP</i> | 0E+00      | 3E-22      | 0E+00      | 0E+00      | 0E+00       | 6E-137      | 0E+00       | 1E-211      | 1E-102      | 6E-101      |
| <i>0HP 3HP</i> | 0E+00      | 1E-24      | 0E+00      | 0E+00      | 0E+00       | 0E+00       | 0E+00       | 6E-111      | 5E-13       | 0E+00       |
| <i>1HP 2HP</i> | 2E-89      | 2E-34      | 3E-201     | 0E+00      | 3E-20       | 0E+00       | 0E+00       | 3E-66       | 3E-283      | 2E-136      |
| <i>1HP 3HP</i> | 6E-08      | 2E-32      | 0E+00      | 0E+00      | 3E-60       | 0E+00       | 0E+00       | 1E-252      | 2E-80       | 0E+00       |
| <i>2HP 3HP</i> | 6E-08      | 5E-1       | 2E-152     | 1E-7       | 1E-12       | 0E+00       | 0E+00       | 2E-60       | 1E-274      | 2E-211      |

Nota: *valor – p* < 0,05 - Rejeita  $H_0$ , os dados não possuem as mesmas variâncias.

- Análise de variância e teste de Tukey: Foi realizada uma análise de variância seguida do teste de Tukey na base de dados para comparar as médias entre os grupos de carga *0HP*, *1HP*, *2HP* e *3HP*, e identificar diferenças significativas. Esses testes assumem que os dados seguem uma distribuição normal, conforme indicado pelos histogramas e gráficos  $Q - Q$  apresentados nas Figuras 79 e 80.

Na Tabela 36 são apresentados os resultados dessa análise que fornece a rejeição do  $H_0$  a um nível de significância de 5%, enquanto que nas Tabelas 37, 38 e 39 fornecem uma interpretação resumida das demais condições operativas.

Tabela 36 – Análise de variância e teste Tukey para *NRM* com  $\alpha=0.05$ 

| Grupo 1          | Grupo 2          | Diferença<br>entre médias | <i>p – valor</i> | Inferior | Superior | Rejeitar $H_0$ |
|------------------|------------------|---------------------------|------------------|----------|----------|----------------|
| <i>NRM – 0HP</i> | <i>NRM – 1HP</i> | -0.0031                   | 0.2112           | -0.0073  | 0.001    | FALSE          |
| <i>NRM – 0HP</i> | <i>NRM – 2HP</i> | 0.0019                    | 0.6262           | -0.0022  | 0.0061   | FALSE          |
| <i>NRM – 0HP</i> | <i>NRM – 3HP</i> | 0.0004                    | 0.9928           | -0.0037  | 0.0046   | FALSE          |
| <i>NRM – 1HP</i> | <i>NRM – 2HP</i> | 0.0051                    | 0.0007           | 0.0017   | 0.0085   | TRUE           |
| <i>NRM – 1HP</i> | <i>NRM – 3HP</i> | 0.0036                    | 0.0342           | 0.0002   | 0.007    | TRUE           |
| <i>NRM – 2HP</i> | <i>NRM – 3HP</i> | -0.0015                   | 0.6693           | -0.0049  | 0.0019   | FALSE          |

Tabela 37 – Análise de variância e teste Tukey para *B07*, *B14* e *B21* com  $\alpha=0.05$ 

|            | <i>B07</i> |            |            | <i>B14</i> |            |            | <i>B21</i> |            |            |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|            | <i>1HP</i> | <i>2HP</i> | <i>3HP</i> | <i>1HP</i> | <i>2HP</i> | <i>3HP</i> | <i>1HP</i> | <i>2HP</i> | <i>3HP</i> |
| <i>0HP</i> | TRUE       | TRUE       | TRUE       | TRUE       | TRUE       | TRUE       | TRUE       | TRUE       | TRUE       |
| <i>1HP</i> |            | FALSE      | FALSE      |            | FALSE      | FALSE      |            | FALSE      | FALSE      |
| <i>2HP</i> |            |            | FALSE      |            |            | FALSE      |            |            | FALSE      |

- Testes Kruskal-Wallis e Dunn: Foi realizada os testes de hipóteses nas condições em que os dados não apresentam uma distribuição de normalidade, conforme constatado

Tabela 38 – Análise de variância e teste Tukey para *IR07*, *IR14* e *IR21* com  $\alpha=0.05$ 

|     | IR07 |       |       | IR14  |       |       | IR21  |       |       |
|-----|------|-------|-------|-------|-------|-------|-------|-------|-------|
|     | 1HP  | 2HP   | 3HP   | 1HP   | 2HP   | 3HP   | 1HP   | 2HP   | 3HP   |
| 0HP | TRUE | TRUE  | TRUE  | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| 1HP |      | FALSE | FALSE |       | FALSE | FALSE |       | FALSE | FALSE |
| 2HP |      |       | FALSE |       |       | FALSE |       |       | FALSE |

Tabela 39 – Análise de variância e teste Tukey para *ER07*, *ER14* e *ER21* com  $\alpha=0.05$ 

|     | ER07 |       |       | ER14 |       |       | ER21 |       |       |
|-----|------|-------|-------|------|-------|-------|------|-------|-------|
|     | 1HP  | 2HP   | 3HP   | 1HP  | 2HP   | 3HP   | 1HP  | 2HP   | 3HP   |
| 0HP | TRUE | TRUE  | TRUE  | TRUE | TRUE  | TRUE  | TRUE | TRUE  | TRUE  |
| 1HP |      | FALSE | FALSE |      | FALSE | FALSE |      | FALSE | FALSE |
| 2HP |      |       | FALSE |      |       | FALSE |      |       | FALSE |

nas Tabelas 31 e 33. O teste de Kruskal-Wallis e Dunn foi aplicado sobre a hipótese de que médias dos grupos *0HP*, *1HP*, *2HP* e *3HP* são iguais. Nas Tabelas 40, 41 e 42, são apresentados os resultados dessas análises utilizando a biblioteca *statsmodels* do Python, que fornece o *p* – valor de cada testes a um nível de significância de 5%.

Tabela 40 – Testes Kruskal-Wallis e Dunn para *B07*, *B14* e *B21* com  $\alpha=5\%$ 

|     | B07     |         |         | B14     |         |         | B21    |        |         |
|-----|---------|---------|---------|---------|---------|---------|--------|--------|---------|
|     | 1HP     | 2HP     | 3HP     | 1HP     | 2HP     | 3HP     | 1HP    | 2HP    | 3HP     |
| 0HP | 2,2E-04 | 2,4E-07 | 1,1E-07 | 1,0E-12 | 3,3E-12 | 1,5E-14 | 0,0E00 | 0,0E00 | 0,0E00  |
| 1HP |         | 5,7E-01 | 4,1E-01 |         | 1,0E00  | 1,0E00  |        | 1,0E00 | 3,8E-01 |
| 2HP |         |         | 1,0E00  |         |         | 1,0E00  |        |        | 9,7E-01 |

Tabela 41 – Testes Kruskal-Wallis e Dunn para *IR07*, *IR14* e *IR21* com  $\alpha=5\%$ 

|     | IR07    |         |         | IR14    |         |         | IR21   |        |        |
|-----|---------|---------|---------|---------|---------|---------|--------|--------|--------|
|     | 1HP     | 2HP     | 3HP     | 1HP     | 2HP     | 3HP     | 1HP    | 2HP    | 3HP    |
| 0HP | 1,5E-13 | 5,2E-15 | 4,9E-18 | 5,5E-04 | 7,5E-04 | 1,2E-04 | 1,0E00 | 1,0E00 | 1,0E00 |
| 1HP |         | 1,0E00  | 7,7E-01 |         | 1,0E00  | 1,0E00  |        | 1,0E00 | 1,0E00 |
| 2HP |         |         | 1,0E00  |         |         | 1,0E00  |        |        | 1,0E00 |

Tabela 42 – Testes Kruskal-Wallis e Dunn para *ER07*, *ER14* e *ER21* com  $\alpha=5\%$ 

|     | ER07    |         |         | ER14   |        |        | ER21    |         |         |
|-----|---------|---------|---------|--------|--------|--------|---------|---------|---------|
|     | 1HP     | 2HP     | 3HP     | 1HP    | 2HP    | 3HP    | 1HP     | 2HP     | 3HP     |
| 0HP | 8,5E-66 | 2,0E-67 | 8,6E-67 | 0,0E00 | 0,0E00 | 0,0E00 | 1,5E-46 | 2,2E-51 | 9,2E-49 |
| 1HP |         | 1,0E00  | 1,0E00  |        | 1,0E00 | 1,0E00 |         | 1,0E00  | 1,0E00  |
| 2HP |         |         | 1,0E00  |        |        | 1,0E00 |         |         | 1,0E00  |

Ao final dessas análises, são construídos diagramas de Venn para observar os grupos com médias estatisticamente iguais. As formações dos grupos para a análise de variâncias

com o teste de Tukey são apresentadas nas Figuras 81, enquanto que teste Kruskal-Wallis com Dunn são apresentados na Figura 82. Em ambos os casos são utilizados amostras de tamanho  $N/100$  e  $N/1000$ .

Figura 81 – Rótulos considerando a normalidade dos dados: (a)  $N/100$ ; (b)  $N/1000$ .

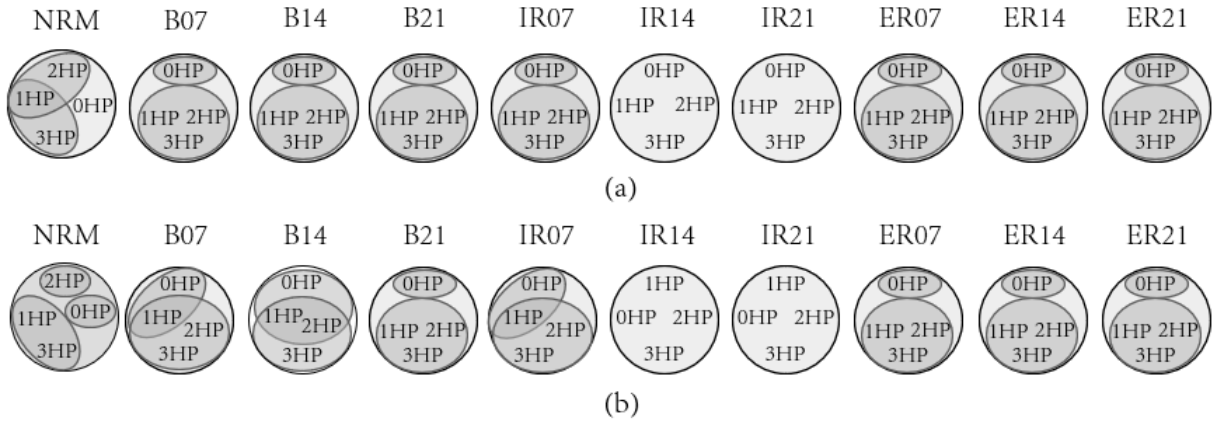
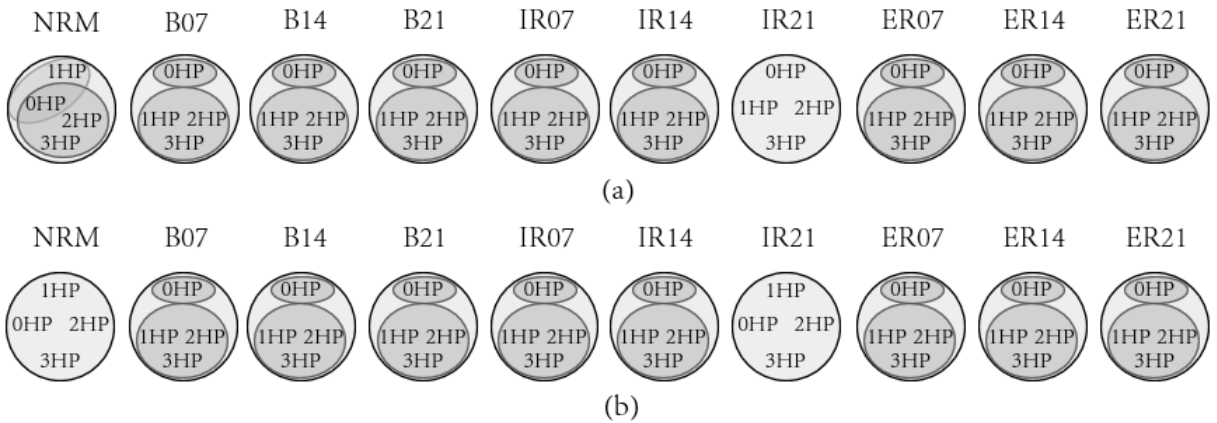


Figura 82 – Rótulos desconsiderando a normalidade dos dados: (a)  $N/100$ ; (b)  $N/1000$ .



Conclui-se que os grupos de cada classe também podem ser organizados de acordo com a potência mecânica no eixo. Na maioria dos casos, observou-se a distinção de dois grupos: um com carga no eixo e outro sem carga no eixo. Para as condições *NRM* e *IR21*, pode-se criar um único grupo para representar essa condição de falha. Portanto, a análise estatística demonstrou que, em condições normais, pode haver um único grupo, enquanto que, nas condições de falha, podem ser subdivididos em dois novos grupos: um com carga no eixo e outro sem carga, resultando em um total de 19 rótulos ao invés dos 40 originais. Outro ponto importante é que a média é uma variável crucial na parametrização dos modelos de aprendizado de máquina.

Entretanto, para fins de comparação com o trabalho de referência, optou-se por manter as mesmas classes, ou seja, o tipo e a intensidade da falha, resultando em 10 rótulos (ZHANG et al., 2020).

### 6.1.2 Definição do Valor Limiar

Para estimar o valor do limiar utilizado na técnica de contagem de *hits*, foi realizada uma análise estatística descritiva dos dados. Essa análise visou encontrar um valor que abrangesse uma quantidade significativa de dados, na tentativa de obter uma classificação adequada sem incluir todos os dados ou causar uma filtragem completa do sinal. As métricas estatísticas utilizadas incluíram a média, o desvio padrão, a curtose e a assimetria, com o objetivo de compreender o comportamento da base de dados e identificar tendências de simetria e tipos de distribuição dos dados.

Na Tabela 43 são apresentados os valores estatísticos encontrados sobre a base de dados que está organizada de acordo com o tipo de falha (classe) e condição operativa (potência). Os valores médios dos sinais de falhas permaneceram constantes para as condições *NRM* e *IR21*. Nas demais condições, houve uma variação no valor médio, porém, não apresentaram diferenças significativas quando alterada a carga no eixo, conforme tratado na Seção 6.1.1.

Com relação ao desvio padrão, a classe *NRM* apresentou um sinal mais estável em comparação aos seus pares. Isso sugere que, ao normalizar o sinal, as amplitudes foram as menores, confirmando a hipótese de que valores de limiar maiores reduzem o número de ocorrências dos eventos na classe *NRM* antes das demais classes. Além disso, conforme a intensidade da falha, aumenta em cada classe ( $7\mu''$ ,  $14\mu''$  e  $21\mu''$ ), o desvio padrão é alterado, o que na prática corresponde a um sinal mais ou menos ruidoso.

Tabela 43 – Estatística descritiva dos dados brutos

|             | Média |       |       | Desvio Padrão |       |       | Curtose |        |        | Assimetria |        |        |
|-------------|-------|-------|-------|---------------|-------|-------|---------|--------|--------|------------|--------|--------|
| Classe      | 3HP   | 2HP   | 0HP   | 3HP           | 2HP   | 0HP   | 3HP     | 2HP    | 0HP    | 3HP        | 2HP    | 0HP    |
| <i>NRM</i>  | 0,012 | 0,013 | 0,013 | 0,065         | 0,065 | 0,073 | -0,043  | -0,069 | -0,236 | -0,128     | -0,173 | -0,035 |
| <i>B07</i>  | 0,015 | 0,016 | 0,032 | 0,142         | 0,143 | 0,146 | 0,041   | 0,087  | 0,149  | 0,007      | 0,008  | -0,022 |
| <i>B14</i>  | 0,011 | 0,011 | 0,030 | 0,165         | 0,171 | 0,133 | 14,08   | 34,67  | 7,235  | 0,260      | 0,127  | -0,012 |
| <i>B21</i>  | 0,012 | 0,012 | 0,241 | 0,322         | 0,333 | 0,353 | 15,59   | 13,76  | 0,217  | 0,040      | 0,047  | -0,004 |
| <i>IR07</i> | 0,015 | 0,017 | 0,077 | 0,279         | 0,280 | 0,583 | 3,828   | 4,369  | 3,027  | -0,157     | -0,092 | 0,01   |
| <i>IR14</i> | 0,018 | 0,017 | 0,038 | 0,268         | 0,237 | 0,178 | 11,94   | 16,31  | 8,669  | 0,225      | 0,125  | -0,091 |
| <i>IR21</i> | 0,011 | 0,011 | 0,011 | 0,763         | 0,763 | 0,585 | 0,652   | 0,937  | 1,168  | 0,049      | 0,049  | -0,031 |
| <i>ER07</i> | 0,012 | 0,012 | 0,113 | 0,244         | 0,263 | 0,211 | 7,149   | 9,394  | 2,494  | 0,023      | 0,042  | 0,026  |
| <i>ER14</i> | 0,011 | 0,011 | 0,396 | 0,151         | 0,146 | 0,151 | -0,001  | 0,678  | 0,768  | 0,012      | -0,006 | 0,014  |
| <i>ER21</i> | 0,018 | 0,019 | 0,112 | 0,406         | 0,475 | 0,594 | 18,096  | 19,919 | 22,642 | 0,153      | 0,169  | 0,154  |

Na análise da curtose, todos os índices foram positivos e superiores ao valor crítico, o que significa que há uma quantidade de dados nas caudas das distribuições maior do que na distribuição Gaussiana. Por fim, o coeficiente de assimetria de Pearson ficou entre o intervalo de mais ou menos uma unidade, o que implica em distribuições simétricas.

Para estimiar um valor de limiar adequado aos dados, optou-se por utilizar os histogramas de frequência com um nível de significância de 68,2% o que corresponde a  $1 \times \sigma$ . Assim valores superiores a esses limites são registrados e caracterizam a forma da onda. Nas Figuras 83 e 84, são ilustradas as distribuições dos dados referente as classes *NRM*, *B14*, *IR14* e *ER21*, para as condições com e sem carga mecânica no eixo, respectivamente.

Figura 83 – Histogramas dos dados com carga: (a) *NRM*; (b) *B07*; (c) *IR07*; (d) *ER21*.

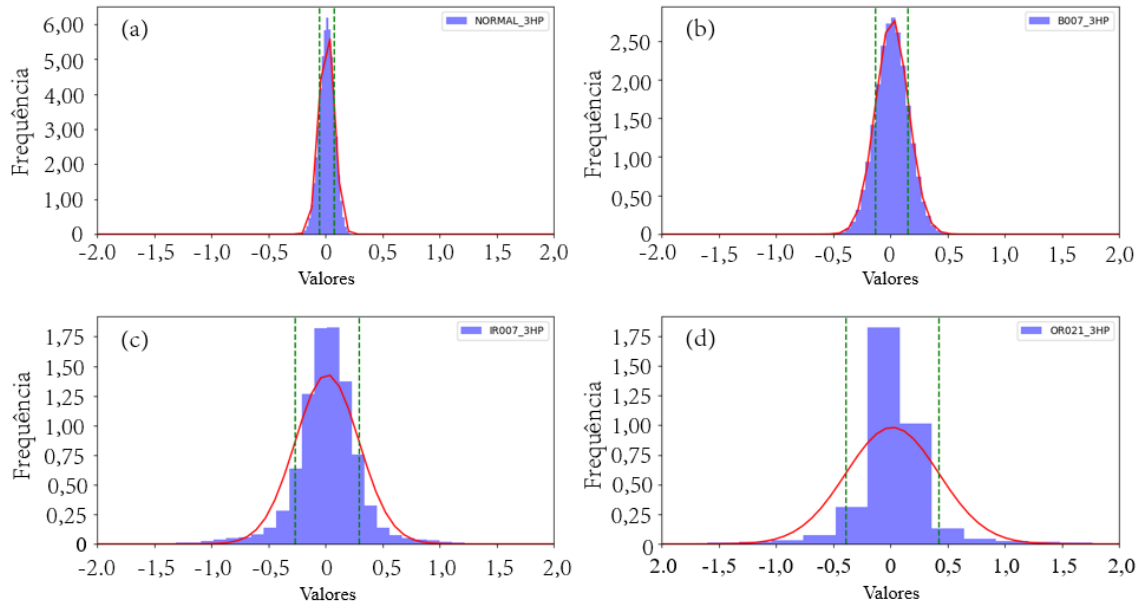
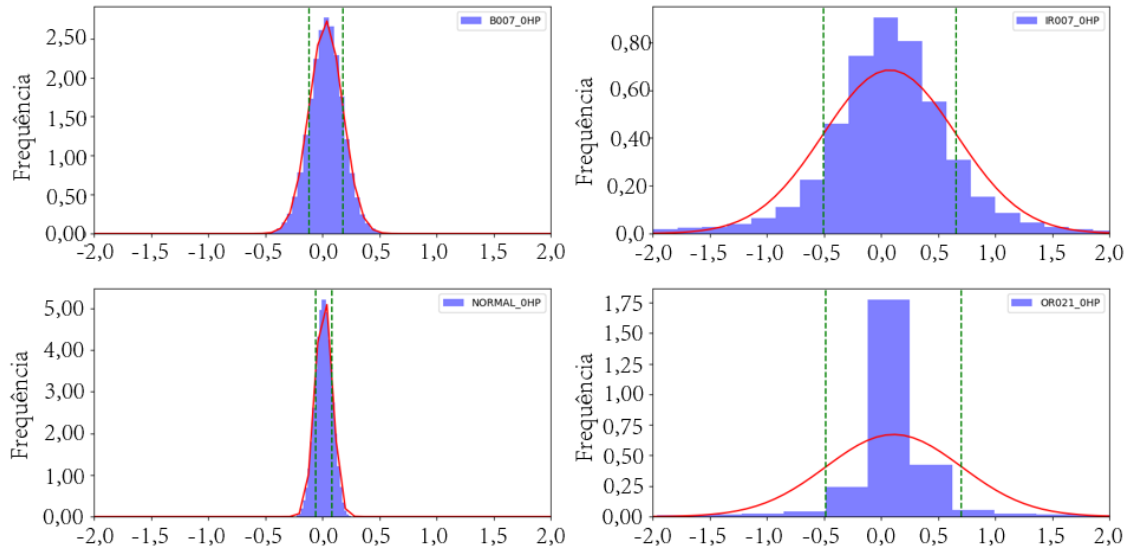
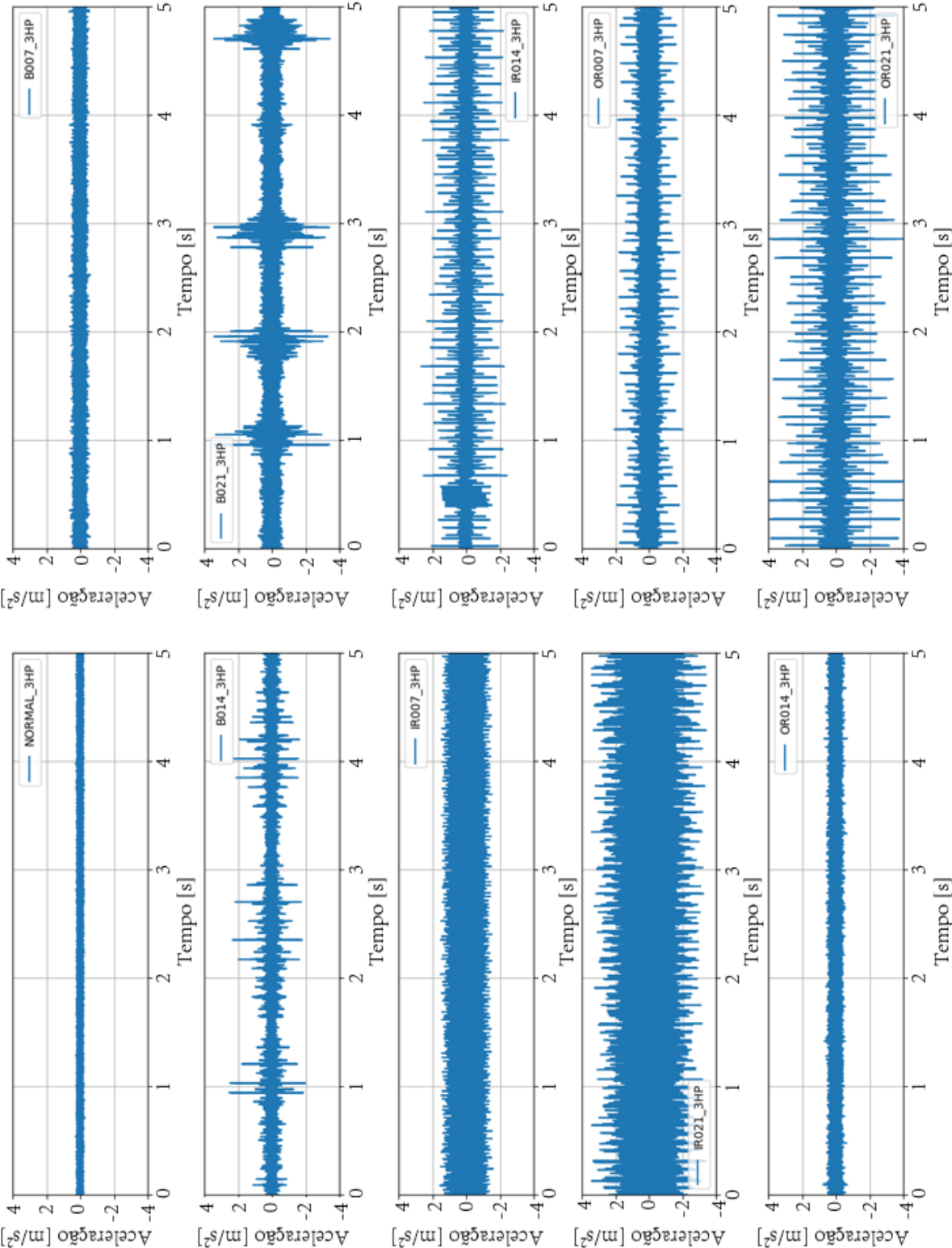


Figura 84 – Histogramas dos dados sem carga: (a) *NRM*; (b) *B07*; (c) *IR07*; (d) *ER21*.



Na Figura 85 são apresentados os sinais da base de dados para todos os tipos e intensidades de falhas no rolamentos, com uma potência mecânica no eixo de 3HP. Observa-se que a condição *NRM* possui o menor nível de ruído, enquanto as demais falhas exibem características distintas. Além disso, conforme a intensidade da falha aumenta, o ruído também aumenta.

Figura 85 – Sinal Bruto dos dados de rolamento para a carga 3HP.



Na Tabela 44, são apresentados os intervalos de confiança 68,2% para todas as condições operativas e estruturais. Observa-se que uma estimativa do valor máximo encontrado para um grau de confiança de 68,26% é de  $0,05m/s^2$ , o que significa que 78% dos picos dos dados estão abaixo desse limiar. Por outro lado, ao utilizar um grau de confiança de 99,74%, obtém-se um valor de limiar de  $0,52m/s^2$ , correspondendo a 54% da base de dados.

Tabela 44 – Valores dos intervalos de confiança para 68,2% e 99,74%.

| Classe      | 3HP             | 2HP             | 1HP             | 0HP             | Vlr <sub>68</sub> | Vlr <sub>99</sub> |
|-------------|-----------------|-----------------|-----------------|-----------------|-------------------|-------------------|
| <i>NRM</i>  | $[-0,05; 0,08]$ | $[-0,05; 0,08]$ | $[-0,05; 0,08]$ | $[-0,06; 0,09]$ | <b>0,05</b>       | 0,18              |
| <i>B07</i>  | $[-0,13; 0,16]$ | $[-0,13; 0,16]$ | $[-0,12; 0,16]$ | $[-0,11; 0,18]$ | 0,11              | 0,56              |
| <i>B14</i>  | $[-0,15; 0,18]$ | $[-0,16; 0,18]$ | $[-0,14; 0,16]$ | $[-0,10; 0,16]$ | 0,10              | 2,00              |
| <i>B21</i>  | $[-0,31; 0,33]$ | $[-0,32; 0,35]$ | $[-0,20; 0,23]$ | $[-0,11; 0,59]$ | 0,11              | 1,28              |
| <i>IR07</i> | $[-0,26; 0,08]$ | $[-0,26; 0,30]$ | $[-0,26; 0,30]$ | $[-0,51; 0,66]$ | 0,26              | 1,80              |
| <i>IR14</i> | $[-0,25; 0,08]$ | $[-0,22; 0,25]$ | $[-0,16; 0,23]$ | $[-0,14; 0,22]$ | 0,14              | 0,81              |
| <i>IR21</i> | $[-0,75; 0,08]$ | $[-0,75; 0,77]$ | $[-0,60; 0,62]$ | $[-0,57; 0,60]$ | 0,57              | 2,25              |
| <i>ER07</i> | $[-0,23; 0,08]$ | $[-0,25; 0,28]$ | $[-0,26; 0,29]$ | $[-0,10; 0,32]$ | 0,10              | 3,74              |
| <i>ER14</i> | $[-0,25; 0,08]$ | $[-0,13; 0,16]$ | $[-0,13; 0,15]$ | $[-0,25; 0,55]$ | 0,13              | 0,84              |
| <i>ER21</i> | $[-0,39; 0,08]$ | $[-0,46; 0,49]$ | $[-0,53; 0,57]$ | $[-0,48; 0,71]$ | 0,39              | <b>0,52</b>       |

Fonte: Adaptado de [Center West Reserve University \(2018\)](#).

Assim, foi definido um intervalo para o valor do limiar na base de dados de rolamento, variando de  $0,05m/s^2$  a  $0,5m/s^2$ . Com isso, torna-se necessária uma análise de sensibilidade desse hiperparâmetro, como apresentado na Tabela 45. Esta tabela fornece um estudo comparativo entre parâmetros do ensaio, incluindo o número de classes e o valor do limiar, conforme tratada na análise estatística das Seções 6.1.1 e 6.1.2, respectivamente. Além disso, são incluídos as abordagens de tratamento de dados: dados brutos, DVS, DVS-SMOTE e DVS-SMOTE-ENN, abordadas na Seção 6.2.

Tabela 45 – Delineamento experimental incompleto para os dados de rolamento.

| Hiperparâmetros |               |        | Tratamento de dados |     |        |            |
|-----------------|---------------|--------|---------------------|-----|--------|------------|
| Regra Hit       | Nº de rótulos | Limiar | Brutos              | DVS | +SMOTE | +SMOTE-ENN |
| 1               | 10            | 0,05   |                     |     |        | X          |
| 1               | 10            | 0,25   | X                   | X   | X      | X          |
| 1               | 10            | 0,50   | X                   | X   | X      | X          |
| 1               | 40            | 0,05   |                     |     |        | X          |
| 1               | 40            | 0,25   |                     |     |        | X          |
| 1               | 40            | 0,50   | X                   | X   | X      | X          |

Fonte: Autor.

Cada marcação na Tabela 45 representa a criação de um modelo, abrangendo a otimização dos hiperparâmetros da rede por meio do método de Bayes à obtenção das

métricas de desempenho - acurácia, precisão, sensibilidade, F1-score - para a fase de teste. Com base nesse estudo comparativo, é tratado nos próximos capítulos dessa dissertação somente a condição 10 rótulos e limiar  $0,25m/s^2$ , pois apresentou o melhor resultado médio quando submetidos os dados a fase dos testes.

### 6.1.3 Análise dos parâmetros da Contagem de *Hits*

Visando avaliar a eficácia dos parâmetros de *hits* derivados do valor limiar de  $0,25mm/s^2$ , aplicou-se um método de contagem de ocorrência, resultando em tabelas de frequências cujos os resultados são discutidos nesta seção.

Essa análise abrangeu todas as características, incluindo duração, tempo de subida, número de picos, número de picos máximos, amplitude e energia, com o objetivo de identificar padrões nos dados organizados com e sem as técnicas de transformação *DVS*. No caso da variável de duração, utilizou-se a regra de *Sturges*, por se tratar de uma variável quantitativa contínua, estabelecendo intervalos uniformes.

Observa-se na Tabela 46 que a técnica revela um aumento na duração dos Hits conforme a intensidade da falha aumenta, ou seja as maiores durações estão para as maiores intensidades de falha conforme observa-se para as condições *IR* e *ER*.

Tabela 46 – Duração por rótulo com dados brutos.

| Duração<br>( $\times 10^{-2}$ ) | Rótulo     |            |            |            |             |             |             |             |             |             |
|---------------------------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                                 | <i>NRM</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> |
| (0 ; 0,0159]                    | 8          | 8          | 0          | 270        | 2568        | 3542        | 24          | 10          | 42          | 32          |
| (0,0159 ; 0,116]                | 26         | 7022       | 14968      | 18042      | 45706       | 20418       | 3108        | 20990       | 9536        | 16806       |
| (0,116 ; 0,241]                 | 0          | 6942       | 2158       | 4826       | 5860        | 1406        | 2154        | 2620        | 2062        | 4816        |
| (0,241 ; 0,37]                  | 0          | 1796       | 316        | 1212       | 822         | 532         | 1222        | 288         | 658         | 2102        |
| (0,37 ; 0,473]                  | 0          | 412        | 28         | 642        | 180         | 198         | 468         | 12          | 80          | 504         |
| (0,473 ; 0,597]                 | 0          | 4046       | 18         | 982        | 44          | 316         | 458         | 4           | 38          | 316         |
| (0,597 ; 0,745]                 | 0          | 1406       | 6          | 414        | 4           | 94          | 684         | 0           | 4           | 230         |
| (0,745 ; 1,03]                  | 0          | 488        | 8          | 154        | 0           | 4           | 896         | 0           | 2           | 268         |
| (1,03 ; 1,39]                   | 0          | 6          | 10         | 10         | 0           | 0           | 1196        | 0           | 0           | 164         |
| (1,39 ; 113,3]                  | 0          | 22         | 68         | 0          | 0           | 2           | 1150        | 0           | 0           | 272         |
| Total                           | 34         | 22148      | 17580      | 26552      | 55184       | 26512       | 11360       | 23924       | 12422       | 25510       |

Fonte: Autor.

Nesse contexto, a condição normal apresentou as menores durações de *hits*, o que pode indicar um estado de operação mais estável e com menor intensidade de falhas detectadas. Esse comportamento de curta duração se reflete diretamente no número de picos e na energia, uma vez que esses eventos tem uma correlação direta com a duração do hit. Esse padrão sugere que, em condições normais, o sistema opera com menor intensidade de vibrações, o que é coerente com um estado de funcionamento sem falhas significativas.

Além disso, observa-se que o número do picos máximos, em cada hit também são afetados, evidenciando uma correlação entre a estabilidade estrutural da condição normal e a intensidade dos parâmetros avaliados. A menor duração dos hits implica uma redução tanto no tempo necessário para atingir o pico máximo quanto na quantidade de picos máximos registrados, conforme mostrado na Tabela 47. Contudo, a condição de falha da esfera *B* apresentou um comportamento inesperado, com a condição *B14* registrando a maior frequência de picos máximos.

Tabela 47 – Número de pico máximo por rótulos com dados brutos.

| Duração   | Rótulo     |            |            |            |             |             |             |             |             |             |
|-----------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
|           | <i>NRM</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> |
| (0 ; 1]   | 34         | 4.652      | 10.428     | 14.050     | 37.176      | 19.512      | 2.566       | 15.680      | 7.202       | 15.338      |
| (1 ; 2]   | 0          | 13.676     | 6416       | 8.938      | 13.712      | 5.620       | 2.928       | 7.566       | 4.648       | 6.218       |
| (2 ; 4]   | 0          | 3.238      | 528        | 1.664      | 3.270       | 1.210       | 2.126       | 566         | 432         | 2.148       |
| (4 ; 5]   | 0          | 164        | 56         | 916        | 488         | 136         | 528         | 74          | 88          | 512         |
| (5 ; 6]   | 0          | 96         | 38         | 658        | 162         | 14          | 240         | 22          | 32          | 330         |
| (6 ; 7]   | 0          | 190        | 38         | 248        | 240         | 18          | 302         | 12          | 16          | 296         |
| (7 ; 9]   | 0          | 46         | 4          | 16         | 92          | 0           | 84          | 2           | 2           | 70          |
| (9 ; 10]  | 0          | 76         | 8          | 46         | 24          | 0           | 432         | 2           | 2           | 164         |
| (10 ; 13] | 0          | 10         | 8          | 6          | 20          | 0           | 384         | 0           | 0           | 126         |
| (13 ; 17] | 0          | 0          | 14         | 10         | 0           | 0           | 1670        | 0           | 0           | 232         |
| (17 ; 37] | 0          | 0          | 18         | 0          | 0           | 0           | 78          | 0           | 0           | 48          |
| (37 ; 57] | 0          | 0          | 8          | 0          | 0           | 0           | 16          | 0           | 0           | 22          |
| Total     | 34         | 22.148     | 17.582     | 26.552     | 55.184      | 26.512      | 11.360      | 23.924      | 12.422      | 25.510      |

Fonte: Autor.

Ao aplicar a *DVS* sobre os dados que definem a forma da onda (matriz de características), esse evento é superado, reforçando a hipótese de que o estado normal se caracteriza por variáveis de resposta mais controladas e menos intensas, contrastando com condições anômalas conforme observa-se as condições *IR* e *ER*.

De maneira semelhante, foi realizada uma análise de frequência da variável amplitude para investigar a relação entre o valor máximo observado e a intensidade das falhas. A Tabela 48 apresenta as contagens dessa variável para cada condição operacional. Observou-se que a condição *NRM* se concentra em uma faixa estreita de amplitudes, indicando uma menor variabilidade dos eventos associados ao estado normal.

Esse comportamento sugere que, sob condições normais, os eventos têm uma amplitude mais uniforme. No entanto, embora a concentração da condição *NRM* em uma faixa restrita seja coerente com o esperado, era previsto que esses valores de amplitude estivessem posicionados nas menores faixas da tabela, refletindo que a condição normal tenderia a apresentar as menores amplitudes. Em contrapartida, observou-se que as condições *IR14* e *ER14*, associadas a falhas mais significativas, exibiram as maiores amplitudes

e uma maior quantidade de eventos. Isso contradiz a expectativa inicial de que falhas mais severas resultariam automaticamente em sinais de maior intensidade.

Tabela 48 – Amplitude por rótulo com dados brutos.

| Duração          | Rótulo     |            |            |            |             |             |             |             |             |             |
|------------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                  | <i>NRM</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> |
| (0,0141 ; 1,308] | 0          | 13.414     | 12         | 21.244     | 27.354      | 9.356       | 4.848       | 0           | 0           | 14.918      |
| (1,308 ; 1,901]  | 0          | 540        | 5.654      | 1.942      | 13.828      | 9.156       | 1.416       | 8.176       | 5.476       | 6.656       |
| (1,901 ; 2,264]  | 0          | 600        | 5.696      | 480        | 2.786       | 2.790       | 986         | 7.512       | 2.528       | 1.642       |
| (2,264 ; 2,694]  | 0          | 890        | 3.830      | 386        | 1.952       | 2.028       | 1.266       | 4.904       | 1.560       | 964         |
| (2,694 ; 3,269]  | 0          | 1.398      | 1.832      | 366        | 2.612       | 1.610       | 1.366       | 2.694       | 1.184       | 518         |
| (3,269 ; 3,598]  | 12         | 972        | 320        | 194        | 1.572       | 396         | 630         | 398         | 366         | 120         |
| (3,598 ; 3,928]  | 12         | 1.190      | 142        | 124        | 1.480       | 238         | 474         | 150         | 254         | 116         |
| (3,928 ; 4,282]  | 6          | 1.480      | 46         | 158        | 1.400       | 178         | 270         | 58          | 190         | 88          |
| (4,282 ; 4,639]  | 4          | 1.244      | 18         | 182        | 1.044       | 90          | 68          | 28          | 176         | 68          |
| (4,639 ; 5,37]   | 0          | 420        | 6          | 320        | 1.088       | 160         | 36          | 4           | 236         | 124         |
| (5,37 ; 6,63]    | 0          | 0          | 2          | 594        | 68          | 216         | 0           | 0           | 200         | 118         |
| (6,63 ; 8,136]   | 0          | 0          | 12         | 352        | 0           | 194         | 0           | 0           | 132         | 88          |
| (8,136 ; 19,81]  | 0          | 0          | 12         | 210        | 0           | 100         | 0           | 0           | 118         | 90          |
| Total            | 34         | 22.148     | 17.582     | 26.552     | 55.184      | 26.512      | 11.360      | 23.924      | 12.422      | 25.510      |

Para a variável amplitude, aplicou-se a transformação dos dados utilizando as técnicas *DVS*, com o objetivo de observar uma tendência equivalente às variáveis de duração, tempo de subida, número de picos, número do pico máximo e energia. Os resultados estão resumidos na Tabela 49, na qual nota-se que a condição normal apresentou as menores amplitudes com menor variação. Além disso, as falhas de maior intensidade apresentaram um volume maior nos picos, e o oposto também foi observado: menor variação em relação à amplitude e menores valores.

Tabela 49 – Amplitude por rótulo combinado ao *DVS*.

| Duração           | Rótulo     |            |            |            |             |             |             |             |             |             |
|-------------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                   | <i>NRM</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> |
| (-1,165 ; -0,086] | 34         | 11.516     | 11.282     | 6.824      | 38.030      | 18.330      | 1.494       | 16.984      | 7.844       | 4.880       |
| (-0,086 ; 0,37]   | 0          | 9.250      | 5.438      | 17.798     | 17.054      | 7.112       | 5.164       | 6.294       | 3.292       | 18.018      |
| (0,37 ; 0,913]    | 0          | 1.374      | 644        | 446        | 100         | 310         | 1.240       | 540         | 700         | 1.310       |
| (0,913 ; 2,382]   | 0          | 8          | 146        | 896        | 0           | 294         | 1.886       | 104         | 452         | 702         |
| (2,382 ; 3,513]   | 0          | 0          | 22         | 510        | 0           | 248         | 1.000       | 0           | 72          | 214         |
| (3,513 ; 5,153]   | 0          | 0          | 22         | 78         | 0           | 180         | 492         | 0           | 38          | 184         |
| (5,153 ; 7,614]   | 0          | 0          | 12         | 0          | 0           | 36          | 68          | 0           | 18          | 126         |
| (7,614 ; 10,86]   | 0          | 0          | 4          | 0          | 0           | 0           | 16          | 0           | 4           | 38          |
| (10,86 ; 50,44]   | 0          | 0          | 12         | 0          | 0           | 2           | 0           | 2           | 2           | 38          |
| Total             | 34         | 22.148     | 17.582     | 26.552     | 55.184      | 26.512      | 11.360      | 23.924      | 12.422      | 25.510      |

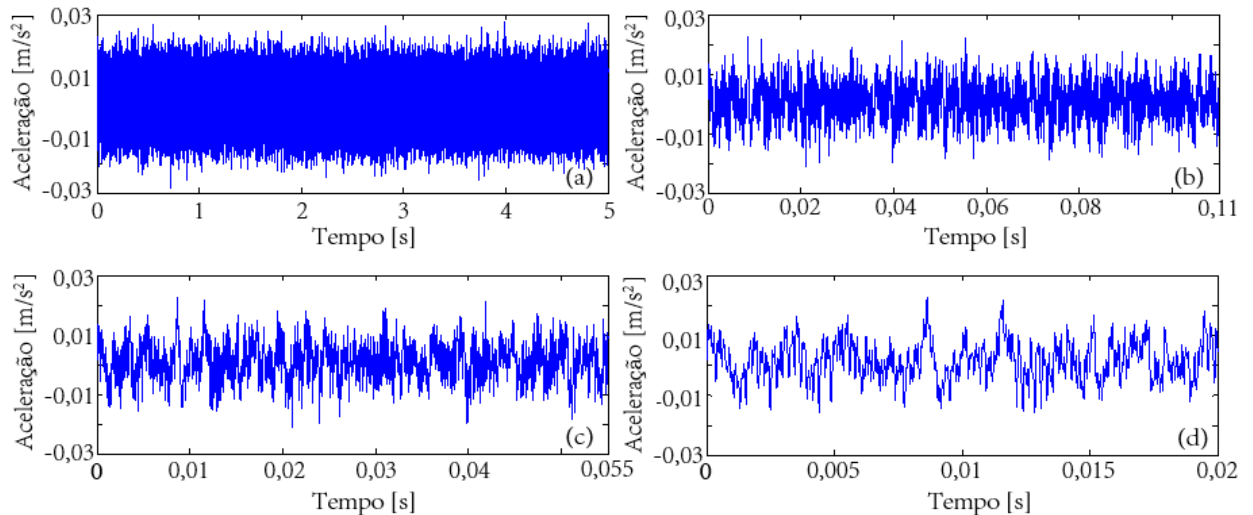
### 6.1.4 Análise de Frequência do Sinal

Objetivando ampliar a base de dados (espectros de frequência) para o modelo RNC, realizou-se uma subamostragem dos dados usando janelas retangulares. Assim, são apresentados nessa seção, os critérios aplicados sobre os espectros de frequências para determinar o tamanho da janela retangular e, conseqüentemente, o novo número de amostras fornecidas ao modelo RNC.

A escolha do tamanho dessa janela retangular deve considerar um aumento expressivo de amostras sem prejudicar os valores das frequências de oscilação, o que tem um impacto positivo na definição das frequências e na criação dos respectivos espectros. No entanto, um número menor de amostras prejudica o desempenho do modelo RNC. Portanto, é crucial encontrar um equilíbrio, entre o número de amostras criadas e o valores das frequências.

Para balizar o tamanho da janela retangular optou-se pelo menor valor de frequência de oscilação encontrada nos ensaios (maior período de onda), que foi de  $180\text{ Hz}$  na condição  $NRM - 2HP$ , o que corresponde a um período de onda de  $0,0055\text{s}$ . Com base nesse valor, definiu-se três tamanhos de janelas, sendo 4, 10 e 20 vezes esse período, resultando nas janelas de tamanho  $0,02\text{s}$ ,  $0,055\text{s}$  e  $0,11\text{s}$ , respectivamente. Na Figura 86 são ilustrados os sinais  $NRM - 2HP$  completo e com utilização das três janelas.

Figura 86 – Sinal janelado de  $NRM - 2HP$ : (a) 5s; (b) 0,11s; (c) 0,055s e (d) 0,02s.



Após a definição do tamanho da janela é realizada a transformação do sinal do domínio do tempo para o domínio da frequência por meio da TRF para todas as classes da base de dados. Este procedimento viabiliza a obtenção dos valores de frequência e quantidade de amostras de acordo com a classe e tamanho da janela. Na Tabela 50 são apresentados os valores de frequências encontrados ao utilizar o sinal completo e as janelas retangulares, expressos em percentuais de variação. Observou-se que à medida

que o tamanho da janela diminui, o desvio relativo das frequências aumenta, chegando ao ponto em que certas frequências deixam de ser observadas nos espectros, como evidenciado nas classes *B07*, *IR14* e *ER21* para o tamanho de janela 0,02s. Além disso, nota-se que à medida que o tamanho da janela diminui, a quantidade de espectros aumenta, conforme apresentado na Tabela 51.

Tabela 50 – Principais frequências dos sinais de acordo com o janelamento

| Falha       | 5s                |      |      |      |      |      |   |      | 0,11s  |      | 0,055s |      | 0,02s  |       |
|-------------|-------------------|------|------|------|------|------|---|------|--------|------|--------|------|--------|-------|
|             | Frequências [kHz] |      |      |      |      |      |   | Qdt. | Desvio | Qdt. | Desvio | Qdt. | Desvio | Qdt.  |
| <i>NRM</i>  | 0,36              | 0,62 | 4,15 | 4,26 | 8,41 | -    | 7 |      | 0,2%   | 321  | 2,8%   | 642  | 3,2%   | 1.768 |
| <i>B07</i>  | 0,36              | 0,72 | 1,28 | 2,89 | 3,38 | 8,39 | 4 |      | 1,0%   | 181  | 5,9%   | 363  | (*)    | 1.000 |
| <i>B14</i>  | 0,36              | 1,38 | 2,48 | 2,76 | 8,4  | -    | 4 |      | 4,7%   | 181  | 4,7%   | 363  | 5,0%   | 1.000 |
| <i>B21</i>  | 1,39              | 2,48 | 2,77 | 3,29 | -    | -    | 4 |      | 1,2%   | 181  | 0,7%   | 363  | 0,8%   | 1.000 |
| <i>IR07</i> | 0,59              | 1,42 | 2,51 | 2,82 | 4,05 | 8,4  | 4 |      | 0,0%   | 181  | 1,7%   | 363  | 23%    | 1.000 |
| <i>IR14</i> | 1,38              | 2,52 | 2,83 | 3,28 | -    | -    | 5 |      | 2,4%   | 268  | 3,2%   | 536  | (*)    | 1.476 |
| <i>IR21</i> | 2,52              | 2,83 | -    | -    | -    | -    | 7 |      | 0,0%   | 323  | 1,1%   | 347  | 1,1%   | 1.781 |
| <i>ER07</i> | 1,38              | 2,58 | 2,78 | 3,27 | -    | -    | 6 |      | 3,5%   | 298  | 4,0%   | 597  | 1,5%   | 1.644 |
| <i>ER14</i> | 1,38              | 2,9  | 3,28 | -    | -    | -    | 7 |      | 0,7%   | 322  | 10%    | 645  | 3,9%   | 1.774 |
| <i>ER21</i> | 0,57              | 1,38 | 2,73 | 3,32 | -    | -    | 6 |      | 1,4%   | 301  | 1,8%   | 602  | (*)    | 1.658 |

Nota: (\*) - Baixa resolução da frequência do sinal ao ponto de prejudicar a precisão do espectro.

Tabela 51 – Quantidade total de amostras de acordo com o tamanho da janela

| Falha | 5s | 0,11s | 0,055s | 0,02s  |
|-------|----|-------|--------|--------|
| Total | 54 | 2.557 | 5.121  | 15.464 |

Com essa análise, foi observado que existe um tamanho de janela que não afeta o espectro de frequência, viabilizando assim a subamostragem. Este fenômeno é evidenciado pelas janelas de tamanho 0,11 segundos, conforme mostrado nas Figuras 87, 88, 89 e 90. Além disso, observa-se que à medida que o tamanho da janela diminui, as amplitudes são modificadas e frequências anteriormente imperceptíveis tornam-se irrelevantes, resultando em um espalhamento das frequências. Esse fenômeno é notado com um número reduzido de amostras, como pode ser visto nas Figuras 87, 88, 89 e 90.

Outro ponto crucial destacado por esta análise é que o aumento na potência mecânica (carga no motor) alterou os valores de frequência, especialmente os valores de baixa intensidade, como evidenciado na condição *NRM* da Figura 87. Além disso, foi constatado que a intensidade de falha altera os espectrogramas, conforme observado nas Figuras 88, 89, 90. Isso viabiliza a categorização em grupos com base na intensidade falhas e carga no motor, conforme constatado na Seção 6.1.1.

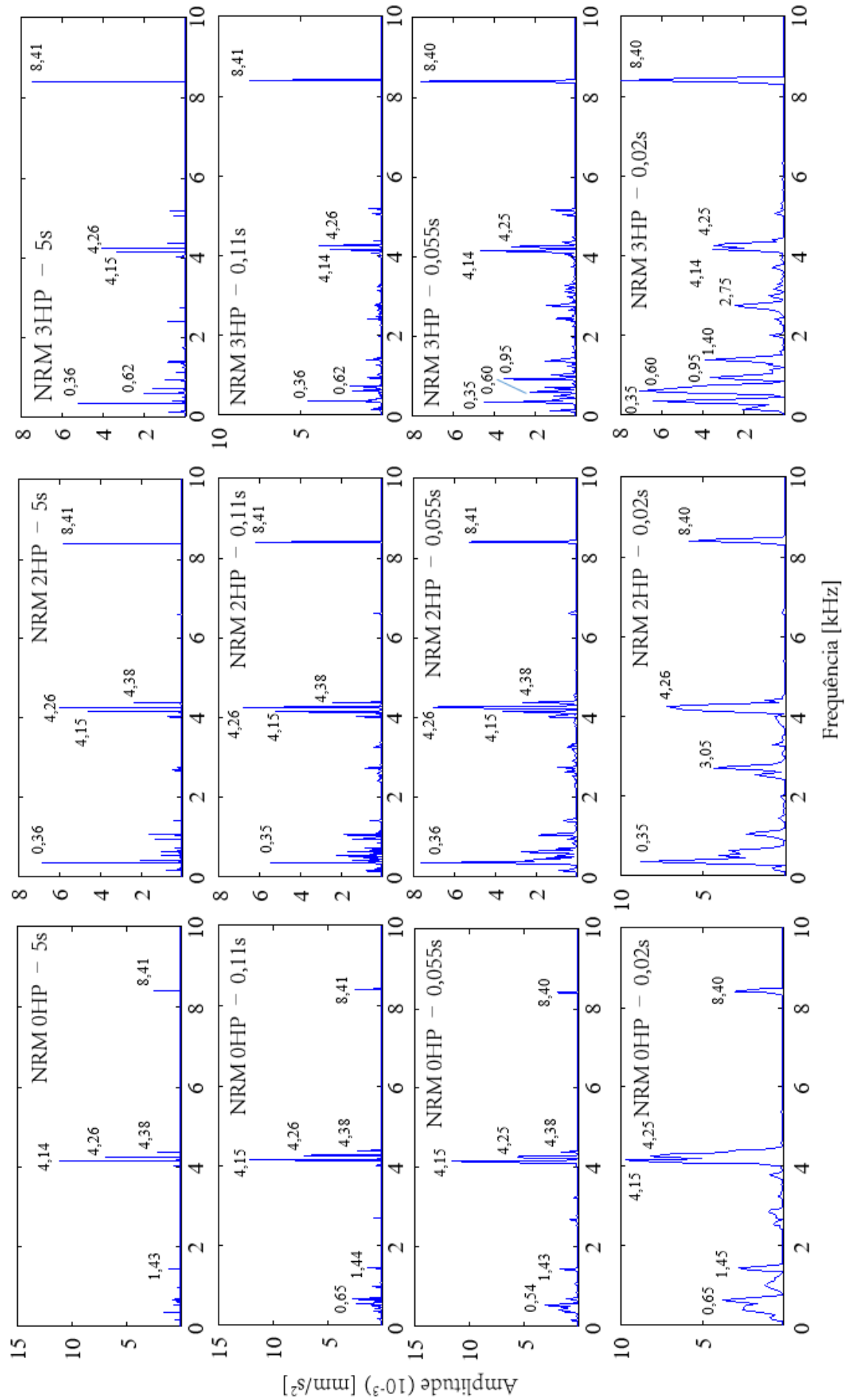
Figura 87 – Análise de sinais da condição normal (*NRM*).

Figura 88 – Análise de sinais da condição de falha na esfera do rolamento (B07, B14 e B21).

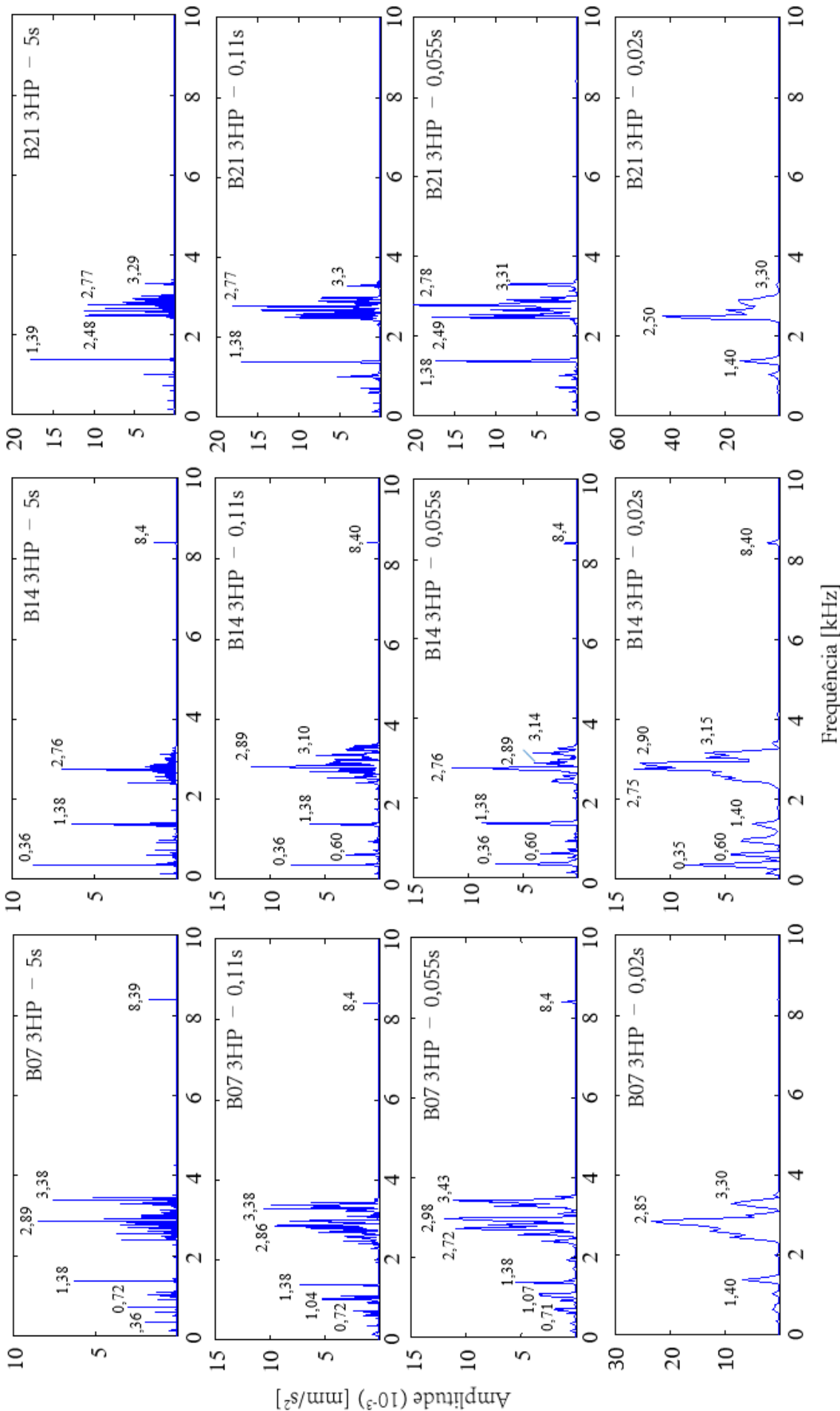


Figura 89 – Análise de sinais da condição de falha na pista interna do rolamento (*IR07*, *IR14* e *IR21*).

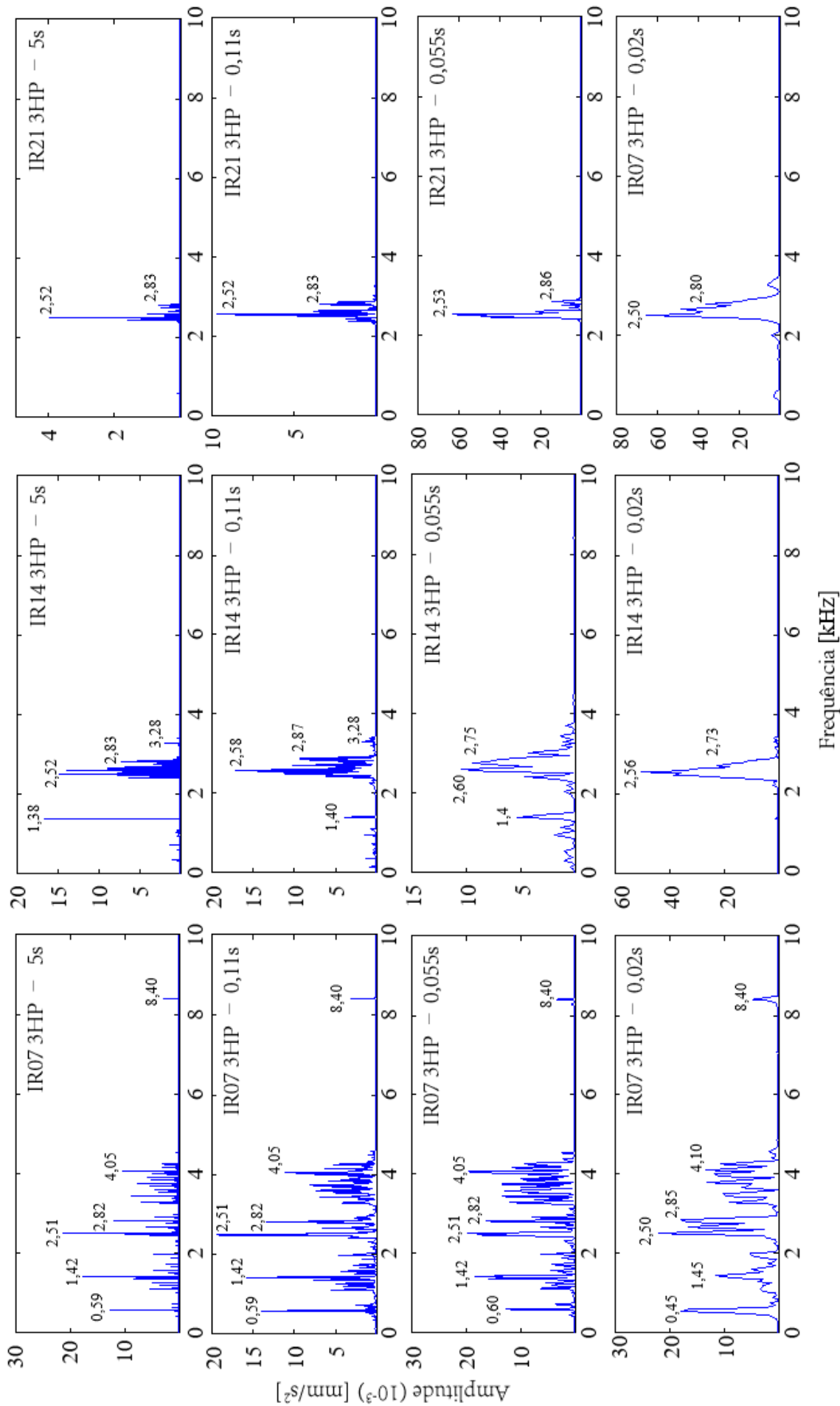
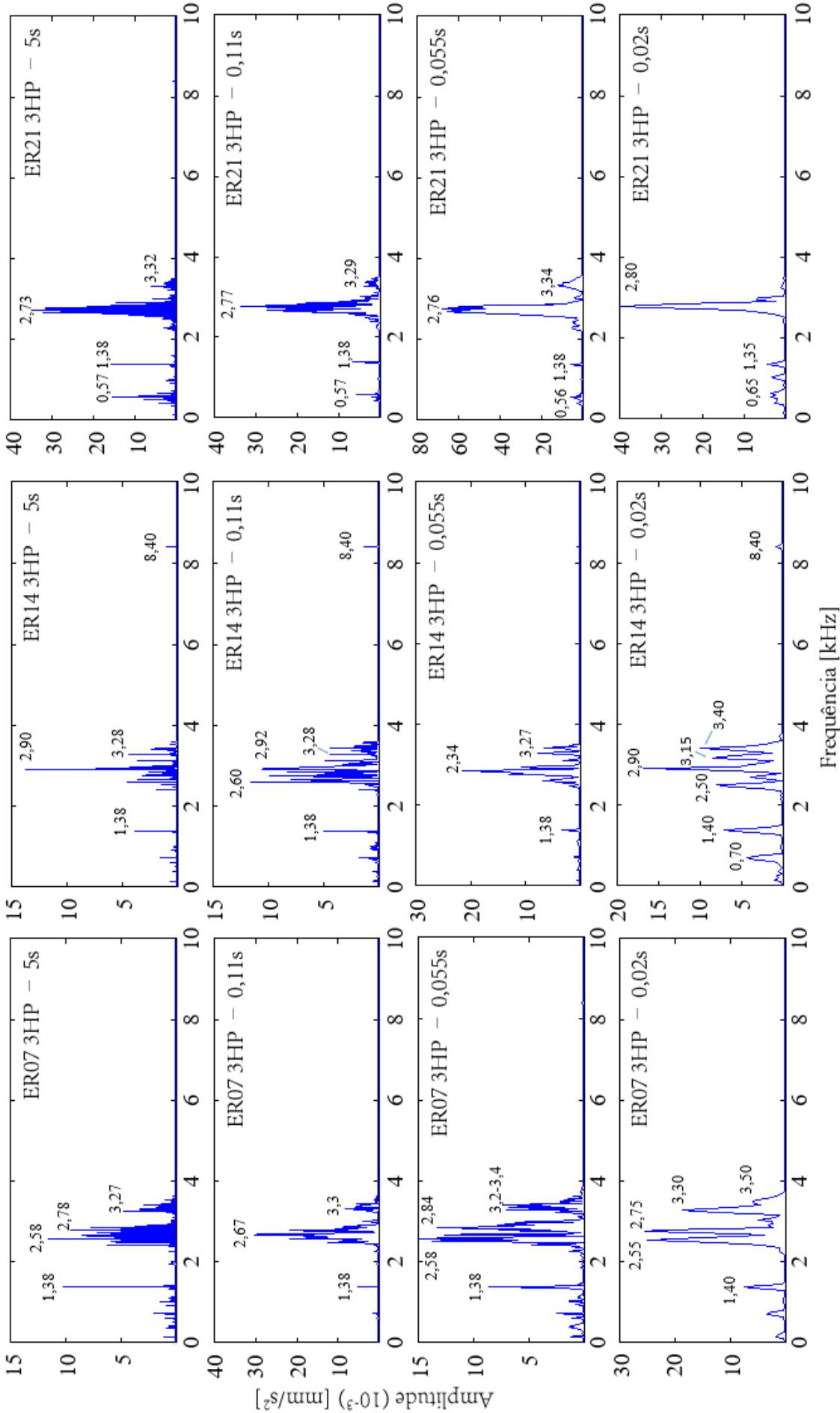


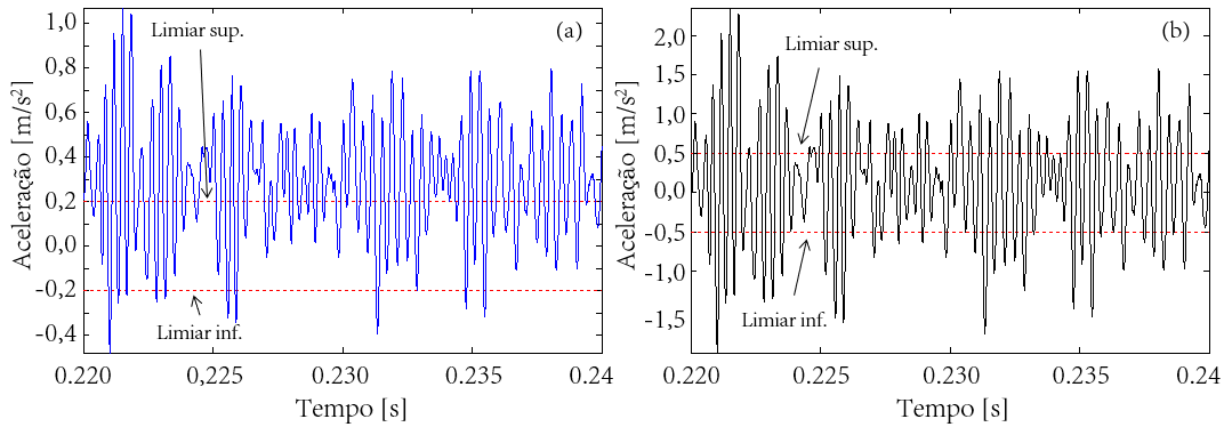
Figura 90 – Análise de sinais da condição de falha na pista externa do rolamento (ER07, ER14 e ER21).



## 6.2 Modelo RNMP

O modelo RNMP foi configurado para o rolamento “DE”, com uma taxa de aquisição de dados de  $48\text{ kHz}$ . Durante a preparação dos dados, houve uma expansão da base de dados para  $N_{aug} = 2^1 \times N$ , onde  $N$  representa o número de amostras do ensaio. A série foi então normalizada para corrigir tendências e contagens equivocadas. No caso presente, os picos da série foram ajustados próximo ao limiar superior, enquanto os vales foram ajustados próximo ao limiar inferior, como ilustrado na Figura 91.

Figura 91 – Classe  $B21$  0,22s a 0,24s: (a) dados brutos; (b) dados normalizados.



Essas etapas foram repetidas para cada classe do conjunto de dados, resultando nos valores médios e desvio padrão necessários para definir o valor limiar, conforme tratado na análise exploratória dos dados (Tabela 43). Enquanto que o número de amostras antes e depois do aumento de dados (interpolação dos dados) é detalhado na Tabela 52.

Tabela 52 – Número de amostras: Aplicação do *data augmentation* (Algoritmo 8).

| Falha       | $N$ [-] |         |         |         | $N_{aug}$ [-] |         |         |         |
|-------------|---------|---------|---------|---------|---------------|---------|---------|---------|
|             | 3HP     | 2HP     | 1HP     | 0HP     | 3HP           | 2HP     | 1HP     | 0HP     |
| <i>NRM</i>  | 485.643 | 485.063 | 483.903 | 243938  | 971284        | 970.126 | 967804  | 487874  |
| <i>B07</i>  | 488.545 | 486.804 | 487.384 | 244.739 | 977.088       | 973.606 | 974.766 | 489.476 |
| <i>B14</i>  | 486.804 | 487.384 | 486.224 | 249.146 | 973.606       | 974.766 | 972.446 | 498.290 |
| <i>B21</i>  | 486.804 | 487.384 | 486.804 | 243.938 | 973.606       | 974.766 | 973.606 | 487.874 |
| <i>IR07</i> | 485.643 | 485.643 | 486.224 | 243.938 | 971.284       | 971.284 | 972.446 | 487.874 |
| <i>IR14</i> | 485.063 | 487.964 | 489.125 | 63.788  | 970.124       | 975.926 | 978.248 | 127.574 |
| <i>IR21</i> | 491.446 | 491.446 | 485.063 | 244.339 | 982.890       | 982.890 | 970.124 | 488.676 |
| <i>ER07</i> | 484.483 | 483.323 | 482.742 | 129.969 | 968.964       | 966.644 | 965.482 | 259.936 |
| <i>ER14</i> | 488.545 | 486.804 | 484.483 | 245.140 | 977.088       | 973.606 | 968.964 | 490.278 |
| <i>ER21</i> | 486.224 | 486.804 | 486.224 | 130.549 | 972.446       | 973.606 | 972.446 | 261.096 |

Fonte: Adaptado de [Center West Reserve University \(2018\)](#).

A etapa de extração de características inicia com a determinação do valor de limiar do sinal, obtido por meio da análise exploratória dos dados. Esse valor foi estabelecido

com base no valor médio e desvio padrão do sinal, resultando em  $0,25 \text{ m/s}^2$ . Ao final desta etapa, os dados são estruturados para o modelo RNMP, totalizando 221.228 eventos de *hits*, incluindo informações, como: duração, tempo de subida, contagem, amplitude, contagem de pico e energia. A Tabela 53 apresenta dez eventos, um de cada classe, e a Tabela 54 apresenta o número de ocorrências por classe. Essa técnica de extração de características resulta em desbalanceamento na base de dados, afetando negativamente a classe com menor número de amostras, como a condição NRM.

Tabela 53 – Dados organizados para o modelo RNMP.

| N    | Duração  | Tempo subida | Amplitude | Contagens | Energia   | Rótulo      |
|------|----------|--------------|-----------|-----------|-----------|-------------|
| 7046 | 0,000270 | 0,000021     | 4,118901  | 1/1       | 0,000463  | <i>NRM</i>  |
| 97   | 0,000656 | 0,000323     | 1,914729  | 2/2       | 0,000167  | <i>B07</i>  |
| 659  | 0,001093 | 0,000385     | 2,226965  | 2/3       | 0,000471  | <i>B14</i>  |
| 226  | 0,005072 | 0,001531     | 4,113059  | 6/15      | -0,000241 | <i>B21</i>  |
| 56   | 0,001343 | 0,000521     | 1,538612  | 1/4       | 0,000195  | <i>IR07</i> |
| 192  | 0,002604 | 0,000375     | 3,237083  | 2/7       | 0,000031  | <i>IR14</i> |
| 348  | 0,012083 | 0,008042     | 1,930882  | 22/31     | -0,000032 | <i>IR21</i> |
| 124  | 0,001854 | 0,000979     | 0,795245  | 4/6       | -0,000027 | <i>ER07</i> |
| 123  | 0,001822 | 0,000760     | 2,672477  | 3/5       | 0,000154  | <i>ER14</i> |
| 367  | 0,005854 | 0,001146     | 5,866345  | 4/17      | 0,000024  | <i>ER21</i> |

Nota: Contagens - Contagem à pico / Contagem.

Tabela 54 – Quantidade de amostras por classe para o modelo RNMP.

|       | <i>NRM</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> |
|-------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Qntd. | 34         | 22148      | 17582      | 26552      | 55184       | 26512       | 11360       | 23924       | 12422       | 25510       |
|       | 0,02%      | 10,0%      | 7,9%       | 12,0%      | 24,9%       | 11,9%       | 5,1%        | 10,8%       | 5,6%        | 11,5%       |

Nota: As porcentagens estão aproximadas.

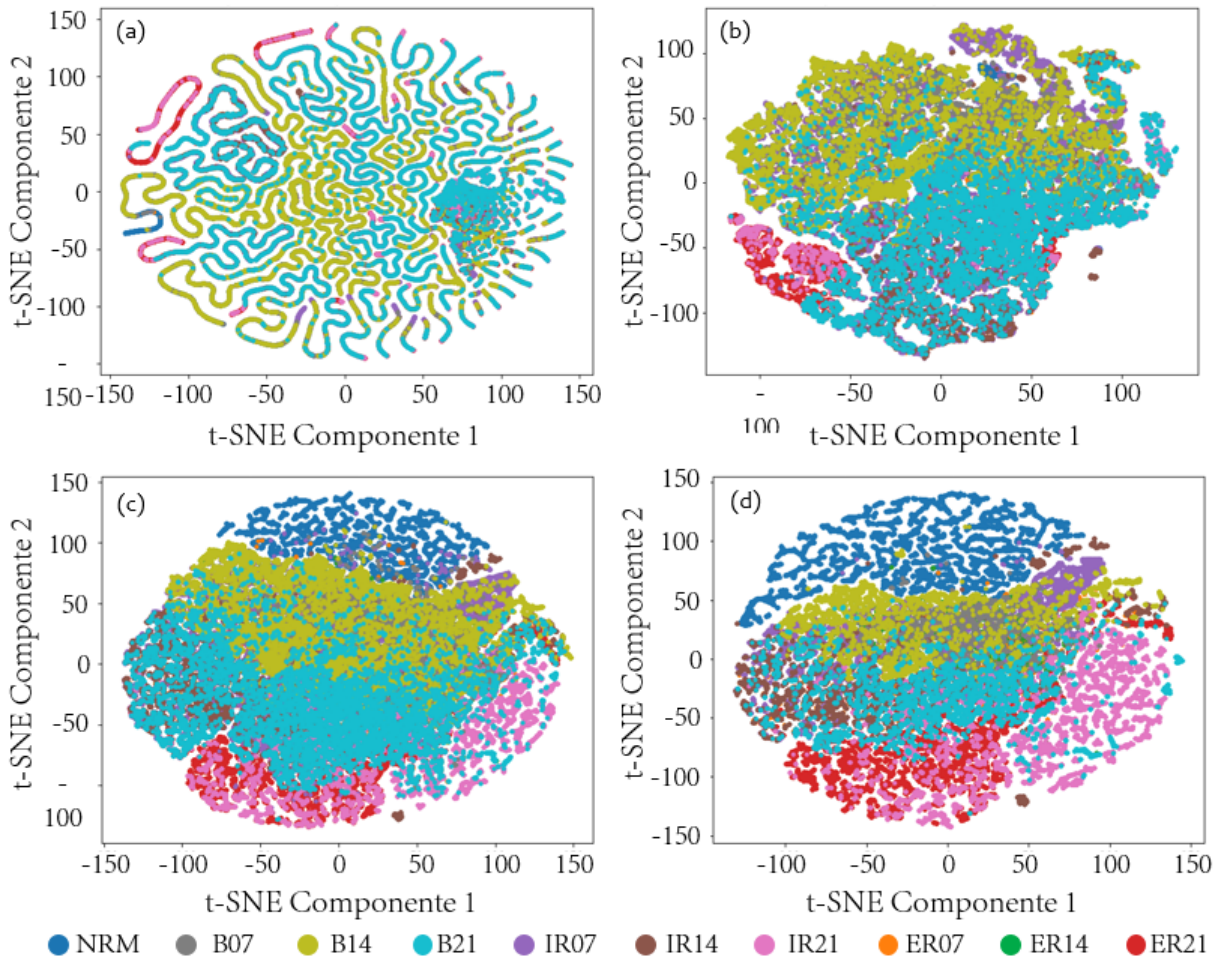
Ao empregar a DVS sobre os dados organizados, chega-se a uma estrutura com 7 componentes principais que representam a forma da onda. Após a definição das componentes principais é realizado o balanceamento da base de dados por meio do modelo SMOTE-ENN. Esse modelo proporcionou a criação de dados sintéticos das classes minoritárias e, em seguida, removeu amostras com base em sua vizinhança. Na Tabela 55, são apresentadas as quantidades de amostras de origem do ensaio, sintéticas, excluídas e os resultados desses tratamentos dos dados.

Tabela 55 – Quantidade de amostras do ensaio ao modelo RNMP.

| Ensaio  | Sintéticos | Excluídos | Técnica |            |         |         |
|---------|------------|-----------|---------|------------|---------|---------|
|         |            |           | Brutos  | <i>DVS</i> | SMOTE   | ENN     |
| 221.228 | 264.548    | -48.049   | 221.228 | 221.228    | 485.776 | 437.727 |

Na Figura 92, apresenta-se a modelagem t-SNE dos conjuntos de dados brutos, seguida pela implementação das técnicas de transformação dos dados (*DVS*), balanceamento *SMOTE*, e a exclusão dos dados pela técnica *ENN*. Na Figura 92-(a), observa-se a dificuldade inicial em distinguir as classes, o que é mitigado após a aplicação da transformação dos dados (*DVS*), como evidenciado na Figura 92-(b) e (c). Esse avanço é ainda mais notável na Figura 92-(d), onde os *clusters* se tornam mais distintos.

Figura 92 – Visualização dos dados separados por classe com a técnica t-SNE: (a) brutos; (b) com *DVS*; (c) com *DVS* e *SMOTE* e (d) *DVS* e *SMOTE-ENN*.



Essa abordagem tem como objetivo aprimorar a qualidade dos dados de entrada para o modelo RNMP. Dependendo dos resultados, pode sinalizar a necessidade de melhorias na fase de coleta, como a eliminação de ruídos e incertezas do ensaio, bem como na definição adequada das classes, que no atual caso, está de acordo com o artigo de referência (ZHANG et al., 2020). Além disso, pode-se melhorar a precisão da contagem de *hits* por meio da utilização de novos parâmetros. Essas medidas mencionadas, apenas contribuem e facilitam a classificação da rede (transformações lineares) para o melhor desempenho de classificação. É importante destacar que a abordagem da *DVS* não visa fornecer uma explicação física dos dados, mas sim uma interpretação estritamente matemática. Isso fica evidente ao analisar os dados organizados na Tabela 56, especialmente

quando se observa as características de tempo de subida, contagem e contagem até o pico, as quais apreensetam valores negativos ou não inteiros.

Tabela 56 – Dados organizados para o modelo RNMP.

| N   | Duração   | Tempo subida | Amplitude             | Contagens | Energia   | Rótulo      |
|-----|-----------|--------------|-----------------------|-----------|-----------|-------------|
| 2   | -0,168943 | 0,030514     | -0,150065 / 1,617862  | -0,616791 | 0,318098  | <i>NRM</i>  |
| 97  | -0,051931 | 0,071444     | -0,038447 / 0,476762  | 0,082412  | -0,039543 | <i>B07</i>  |
| 659 | -0,172497 | -0,150529    | -0,057121 / 0,259459  | -0,228011 | 0,055596  | <i>B14</i>  |
| 226 | 2,992845  | 7,366324     | 1,412158 / -5,854308  | 1,543998  | 3,573145  | <i>B21</i>  |
| 56  | -0,175163 | -0,286365    | 0,010801 / -0,759745  | 0,033390  | -0,038240 | <i>IR07</i> |
| 192 | -0,173191 | -0,185879    | -0,038936 / -0,005782 | -0,100729 | -0,105542 | <i>IR14</i> |
| 348 | 0,065768  | 0,147414     | 0,059494 / -0,401427  | 0,090292  | 0,059798  | <i>IR21</i> |
| 124 | 0,787542  | -0,109990    | 0,020896 / 1,655456   | 0,528156  | -0,405228 | <i>ER07</i> |
| 123 | 0,533688  | 2,075462     | 0,347362 / -1,968583  | 0,459447  | -2,059467 | <i>ER14</i> |
| 367 | 0,622834  | -0,017067    | 0,062424 / 0,870009   | 0,254927  | -0,349653 | <i>ER21</i> |

Nota: Contagens - Contagem à pico / Contagem.

A Tabela 57 contempla as novas quantidade de dados por classe (balanceamento com dados sintéticos). Observa-se que não ocorreu um balanceamento completo, inerente da técnica ENN, sendo ajustados apenas 20% do total da matriz *design*. Esse desbalanceamento foi corrigido no modelo neural por meio de ponderações em cada classe.

Tabela 57 – Quantidade de amostras por classe para o modelo RNMP.

| Método     | <i>NRM</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR021</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> |
|------------|------------|------------|------------|------------|-------------|-------------|--------------|-------------|-------------|-------------|
| +SMOTE     | 44.161     | 48.650     | 47.651     | 49.486     | 55.184      | 49.367      | 46.444       | 48.931      | 46.676      | 49.226      |
|            | 9.09%      | 10.01%     | 9.81%      | 10.19%     | 11.36%      | 10.16%      | 9.56%        | 10.07%      | 9.61%       | 10.13%      |
| +SMOTE-ENN | 44.161     | 47.579     | 41.606     | 44.815     | 40.349      | 40.950      | 45.006       | 48.931      | 42.152      | 42.178      |
|            | 10,09%     | 10,87%     | 9,51%      | 10,24%     | 9,22%       | 9,36%       | 10,28%       | 11,18%      | 9,63%       | 9,64%       |

Nota: As porcentagens são aproximadas.

Com base nos dados organizados e nos hiperparâmetros definidos na Tabela 2.3 da Seção 2.3, foi conduzida a otimização dos hiperparâmetros do modelo RNMP utilizando a técnica de otimização Bayesiana. Os hiperparâmetros otimizados incluíram o número de camadas, o número de neurônios em cada camada, as funções de ativação, a taxa de aprendizado, o número de épocas, e a função de regularização do tipo  $L_2$  (decaimento de pesos), enquanto os hiperparâmetros fixos consistiram no número de neurônios de entrada, na utilização do *batch normalization* entre o neurônio, no tamanho dos *batches*, no otimizador e na relação entre os conjuntos de treinamento e teste.

Para verificar o desempenho dos processos de transformação dos dados aplicou-se essa otimização para as quatro base de dados apresentadas na Figura 92. O resumo dos resultados estão presentes na Tabela 58.

Tabela 58 – Resultados dos conjuntos de dados de origem do ensaio e suas variações.

| Base de dados | Neurônios por camada           | Épocas | $\alpha$ | $lr$    | Acurácia |        |
|---------------|--------------------------------|--------|----------|---------|----------|--------|
|               |                                |        |          |         | treino   | teste  |
| Dados brutos  | 379/483/32/508/328/326/35/43   | 109    | 6,31E-5  | 3,35E-5 | 50,27%   | 48,89% |
| DVS           | 136/185/321/487/44/502/210/135 | 284    | 4,00E-5  | 9,98E-5 | 70,64%   | 69,44% |
| + SMOTE       | 301/216/498/439/378/145/155/51 | 188    | 5,14E-5  | 1,66E-5 | 77,75%   | 69,46% |
| + SMOTE-ENN   | 484/302/217/40/143/148/360/325 | 178    | 2,22E-5  | 6,81E-5 | 81,29%   | 69,50% |

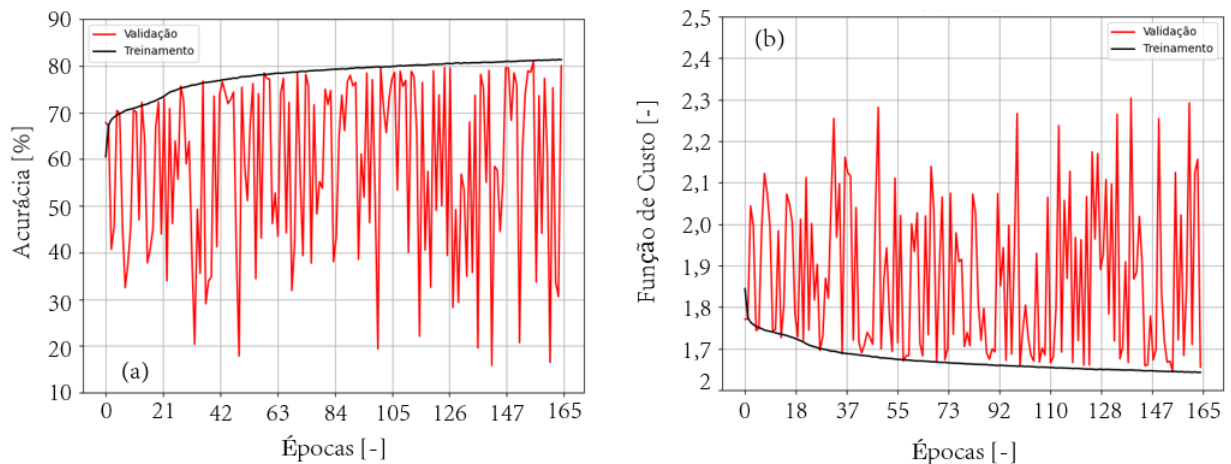
O conjunto de dados *DVS + SMOTE – ENN* apresentou o melhor resultado na fase de teste, alcançando uma acurácia de 69,50%. Em seguida, são mostrados os intervalos de busca dos hiperparâmetros e os resultados obtidos na otimização (Tabela 59).

Tabela 59 – Exploração do espaço e seleção dos hiperparâmetros

| Hiperparâmetro                 | Intervalo de busca | Valores encontrados            |
|--------------------------------|--------------------|--------------------------------|
| Número de camadas              | 2 à 8              | 8                              |
| Número de neurônios por camada | 32 à 512           | 484/302/217/40/143/148/360/325 |
| Função de ativação             | ReLu ou Tanh       | ReLu                           |
| Número de épocas               | 100 à 300          | 178                            |
| Taxa de aprendizado            | 1E-5 à 1E-3        | 6,81E-5                        |
| Decaimento de pesos            | 1E-5 à 1E-4        | 2,22E-5                        |

Após identificar os hiperparâmetros que resultam em uma acurácia ótima local, é essencial aprofundar a análise do modelo RNMP, realizando a execução das fases de treinamento, validação e teste. A Figura 93 ilustra os gráficos de convergência da função de custo e da acurácia tanto para as fases de treino quanto para validação. Os valores de convergência para a acurácia e o custo foram de 81,29% e 1,64 respectivamente, durante o treinamento. Enquanto para a validação, os valores atingiram 80% e 1,76.

Figura 93 – Desempenho geral da RNMP nas fase de treino: (a) acurácia; (b) custo.



Para avaliar a acurácia e precisão por classe, foram criadas matrizes de confusão para as fases de treinamento, validação e teste da rede, abrangendo todos os rótulos. No Tabela 60, é exibida a matriz de confusão apenas para a fase de teste. Nesta matriz os rótulos de origem do ensaio possuem o índice  $r$  enquanto que os valores preditos pelo modelo não possuem índices. Nesses resultados ocorreram erros dos tipos  $I$  e  $II$ , por exemplo, na classe  $NRM$ , o modelo previu 13 ocorrências, das quais 1 e 4 são das classes  $IR07$  e  $IR14$ , respectivamente. Além disso, não houve amostra que tenha sido erroneamente classificadas em outra classe. Enquanto o pior resultado do modelo RNMP foi para a condição  $ER14$ , pois somente 1.318 das 2.651 predições estão corretas, sendo que 2.165 foram classificadas em outros rótulos, e portanto necessário uma investigação e tratamento de dados sobre esse conjunto em específico.

Tabela 60 – Matriz confusão do modelo RNMP com dados de rolamento da fase de teste.

|          | $NRM$    | $ER07$      | $ER14$      | $ER21$      | $IR07$      | $IR14$      | $IR21$      | $B07$       | $B14$       | $B21$       |
|----------|----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| $NRM_r$  | <b>8</b> | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |
| $ER07_r$ | 0        | <b>3972</b> | 151         | 0           | 0           | 340         |             | 0           | 0           | 0           |
| $ER14_r$ | 0        | 0           | <b>1318</b> | 2           | 4           | 22          | 2           | 1563        | 548         | 24          |
| $IR07_r$ | 0        | 112         | 8           | <b>4251</b> | 53          | 15          | 582         | 7           | 10          | 261         |
| $IR07_r$ | 1        | 45          | 43          | 449         | <b>6556</b> | 902         | 105         | 385         | 728         | 1327        |
| $IR14_r$ | 4        | 0           | 26          | 38          | 371         | <b>3543</b> | 33          | 269         | 194         | 1144        |
| $IR21_r$ | 0        | 13          | 0           | 199         | 9           | 5           | <b>1958</b> | 0           | 4           | 93          |
| $B07_r$  | 0        | 4           | 972         | 6           | 11          | 25          | 3           | <b>3039</b> | 690         | 13          |
| $B14_r$  | 0        | 0           | 255         | 1           | 15          | 45          | 2           | 404         | <b>1760</b> | 28          |
| $B21_r$  | 0        | 15          | 29          | 148         | 80          | 240         | 187         | 28          | 137         | <b>4188</b> |

Nota: r - rótulo.

Com relação aos resultados gerais do modelo RNMP, obteve-se na fase de testes 69,50% de acurácia, 73,50% de sensibilidade, 67,07% de precisão e 68,63% no  $F1-score$ . Esses resultados foram calculados com base na matriz de confusão e na utilização das Equações 6.1, 6.2, 6.3 e 6.4.

Além disso, foram feitos os cálculos de acurácia, precisão, sensibilidade e F1-score por classe, sendo sistematizada nas Tabelas 62, 61 e 63.

$$\text{Acurácia} = \frac{\text{Verdadeiros Positivos}}{\text{Total de Instâncias da Classe}} \quad (6.1)$$

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Positivos}} \quad (6.2)$$

$$\text{Recall} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Negativos}} \quad (6.3)$$

$$F1-score = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (6.4)$$

Tabela 61 – Sensibilidade do modelo RNMP para os dados de rolamento.

| Fase             | <i>NRM</i> | <i>ER07</i> | <i>ER14</i> | <i>OR21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | Global       |
|------------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|--------------|
| <i>Train</i> [%] | 99.99      | 96.40       | 42.98       | 89.25       | 81.05       | 72.74       | 89.42       | 74.24      | 74.46      | 87.03      | <b>80,76</b> |
| <i>Valid</i> [%] | 100.0      | 91.55       | 45.40       | 86.31       | 74.33       | 69.10       | 91.70       | 67.41      | 78.83      | 89.58      | <b>79,42</b> |
| <i>Test</i> [%]  | 100.0      | 89.00       | 37.84       | 80.22       | 62.20       | 63.09       | 85.84       | 63.80      | 70.12      | 82.90      | <b>73,50</b> |

Fonte: Autor.

Tabela 62 – Precisão do modelo RNMP para os dados de rolamento.

| Fase             | <i>NRM</i> | <i>ER07</i> | <i>ER14</i> | <i>OR21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | Global       |
|------------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|--------------|
| <i>Train</i> [%] | 99.77      | 95.56       | 61.44       | 88.05       | 90.63       | 83.90       | 90.65       | 56.37      | 74.72      | 76.70      | <b>81,78</b> |
| <i>Valid</i> [%] | 99.81      | 97.55       | 53.45       | 76.84       | 78.82       | 75.54       | 86.00       | 52.55      | 69.03      | 67.93      | <b>80,65</b> |
| <i>Test</i> [%]  | 61.54      | 95.46       | 49.72       | 81.05       | 92.35       | 73.91       | 60.96       | 53.36      | 43.23      | 59.17      | <b>67,07</b> |

Fonte: Autor.

Tabela 63 – *F1-score* do modelo RNMP para os dados de rolamento.

| Fase             | <i>NRM</i> | <i>ER07</i> | <i>ER14</i> | <i>OR21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | Global       |
|------------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|--------------|
| <i>Train</i> [%] | 99.88      | 95.98       | 50.58       | 88.65       | 85.57       | 77.92       | 90.03       | 64.08      | 74.59      | 81.54      | <b>80,88</b> |
| <i>Valid</i> [%] | 99.93      | 94.44       | 51.45       | 86.89       | 81.96       | 75.85       | 87.95       | 61.42      | 74.41      | 81.96      | <b>79,63</b> |
| <i>Test</i> [%]  | 76.19      | 92.12       | 42.97       | 80.63       | 74.33       | 68.07       | 71.29       | 58.12      | 53.49      | 69.05      | <b>68,86</b> |

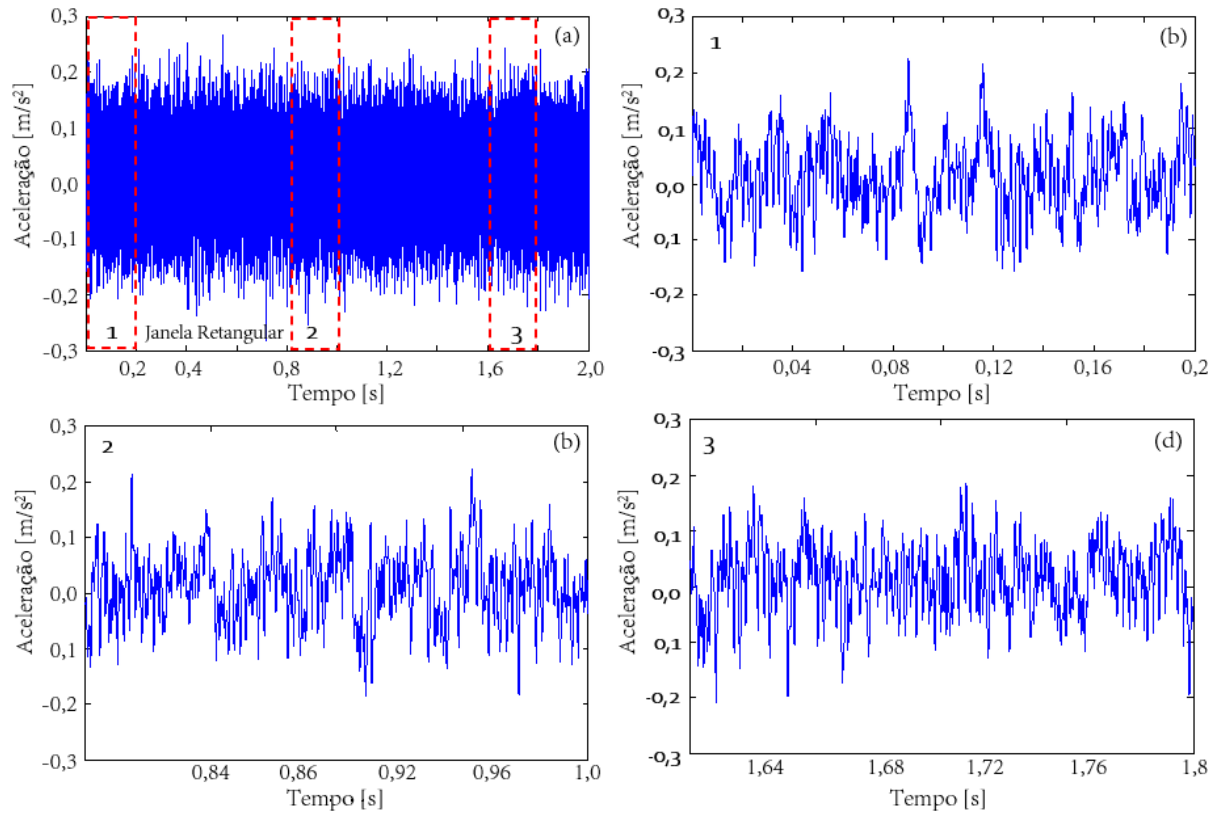
Fonte: Autor.

## 6.3 Modelo RNC

O modelo RNC foi ajustado especificamente para o rolamento “DE” com uma taxa de aquisição de dados de 48 *kHz*, visando a construção de um modelo convolucional para o diagnóstico de falhas em rolamentos. Inicialmente, os dados foram lidos e normalizados pelo cálculo *doz-score*, garantindo que os espectros de frequência apresentem intensidades comparáveis, independentemente de suas ordens de grandeza.

Para expandir a base de dados, são realizadas técnicas de subamostragem e aplicação da TFCT, utilizando uma janela retangular com duração de 0,02s. A escolha da janela retangular é importante para manter a integridade dos dados temporais durante a transformação. Este procedimento permite capturar variações de frequência em intervalos de tempo específicos, melhorando a capacidade do modelo de generalizar falhas. A Figura 94 ilustra este procedimento para a condição *NRM*, mostrando 3 das 1.768 janelas utilizadas.

Figura 94 – Aplicação das janelas retangulares: (a) sinal completo; (b) janela do instante 0 a 0,2s; (c) janela do instante 0,8 a 1s; (d) janela 1,6 a 1,8s.



A escolha do valor de 0,02s foi determinada com base na conservação das frequências naturais do sinal, mantendo um volume de dados adequado para a RNC. Os valores de frequência estão apresentados na Tabela 64, com uma tolerância de variação de até 5%, conforme discutido na Seção 6.1.4. Dessa forma, 36 arquivos organizados no ensaio com durações de 5 e 10s, apresentados nos Quadros 6 e 7, foram transformados em 15.464 novas séries de 0,02s cada. Na Tabela 64, são apresentadas as frequências encontradas em cada janela, onde se percebe que algumas frequências são atenuadas enquanto outras são destacadas em cada espectro.

Tabela 64 – Valores frequência do sinal na classe *ONCAV – E4.Q108*.

| Tamanho Janela | Frequências [Hz] |      |      |      |      |  |
|----------------|------------------|------|------|------|------|--|
| Janela 5s      | 0,36             | 0,62 | 4,15 | 4,26 | 8,41 |  |
| Janela 0,11    | 0,35             | -    | 3,05 | 4,26 | 8,40 |  |
| Janela 0,055   | 0,33             | 0,6  | -    | 4,27 | 8,41 |  |
| Janela 0,02    | 0,37             | 0,61 | 3,05 | 4,25 | 8,40 |  |

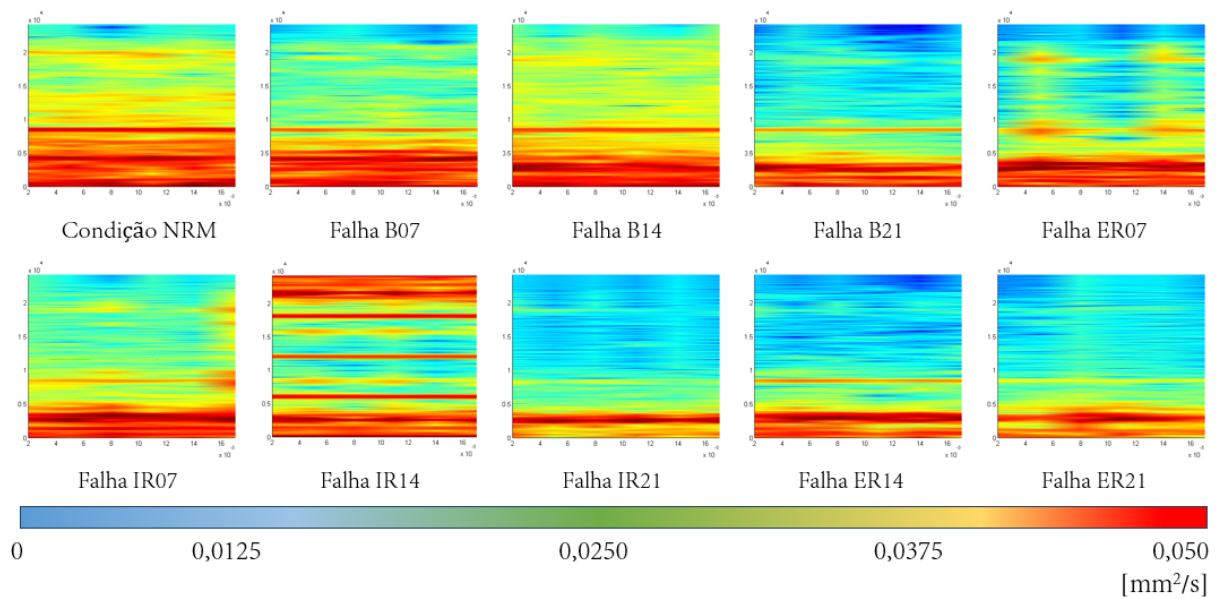
Dando sequência à etapa de preparação dos dados, foi realizada a Transformada de Fourier de Curto Tempo (TFCT) por meio do Algoritmo 76, conforme apresentado na Seção 5.2.2. Esse procedimento inclui os valores do tamanho do *overlap*, tipo e tamanho da janela, e frequência de amostragem. Esses parâmetros, em conjunto com os dados dos

ensaios, geraram um total de 15.464 espectrogramas, cujas proporções estão detalhadas na Tabela 65. A Figura 95 ilustra um dos espectros de frequência de cada classe.

Tabela 65 – Quantidade de dados

|             | 3HP | 2HP | 1HP | 0HP | Total |
|-------------|-----|-----|-----|-----|-------|
| <i>B07</i>  | 509 | 507 | 508 | 255 | 1779  |
| <i>B14</i>  | 507 | 508 | 506 | 260 | 1780  |
| <i>B21</i>  | 507 | 508 | 507 | 254 | 1776  |
| <i>IR07</i> | 506 | 506 | 507 | 254 | 1773  |
| <i>IR14</i> | 506 | 508 | 398 | 87  | 1498  |
| <i>IR21</i> | 510 | 512 | 505 | 255 | 1781  |
| <i>ER07</i> | 505 | 503 | 503 | 135 | 1645  |
| <i>ER14</i> | 509 | 507 | 505 | 255 | 1775  |
| <i>ER21</i> | 507 | 507 | 507 | 136 | 1656  |

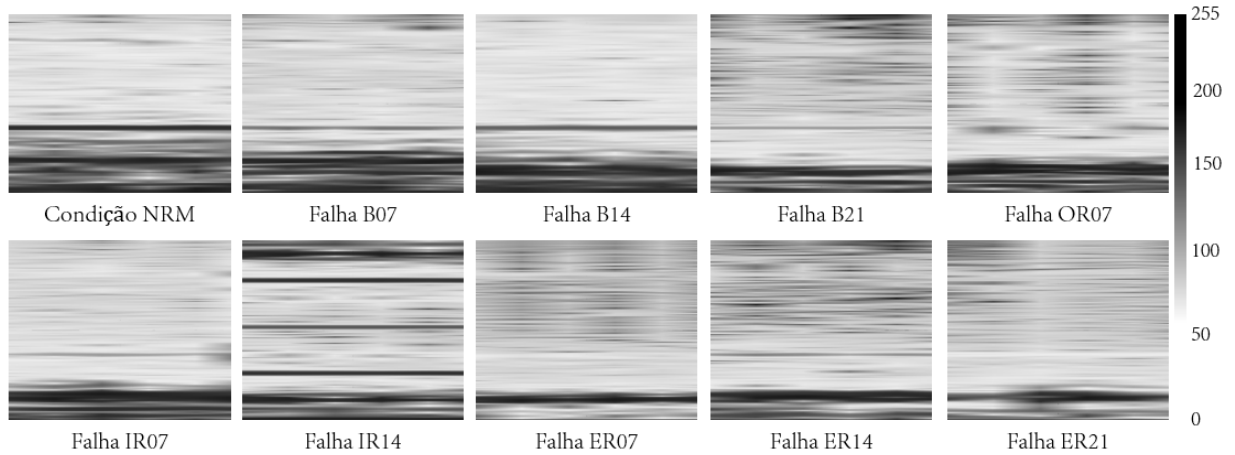
Figura 95 – Espectrogramas das condições da integridade estrutural dos rolamentos (TFCT).



O próximo passo consiste na transformação dos espectros de frequência em imagens. Esse processo envolve a conversão dos valores de frequência em uma escala de 0 a 255, o redimensionamento das imagens para um formato padrão, a remoção de bordas indesejadas, o registro das imagens com seus respectivos rótulos e, por fim, a criação de uma base de dados formatada adequadamente para o modelo RNC, que é então armazenada em um diretório específico.

A Figura 96 apresenta os padrões presentes nas imagens, incluindo detalhes visuais característicos que ilustram cada um dos rótulos.

Figura 96 – Imagens das condições da integridade estrutural dos rolamentos (cinza).



Fonte: Autor.

A base de dados para a RNC é composta por imagens em tons de cinza, com dimensões de  $32 \times 32$  pixels, e um arquivo “.csv” que contém metadados detalhados sobre cada imagem. Estes metadados incluem informações como o tipo de falha, a velocidade de rotação, o intervalo de tempo do sinal e a técnica utilizada para a aquisição das imagens. A Figura 97 ilustra a estrutura da base de dados, proporcionando uma visão clara da organização e das informações associadas a cada imagem.

Figura 97 – Base de dados de rolamento para o modelo RNC.



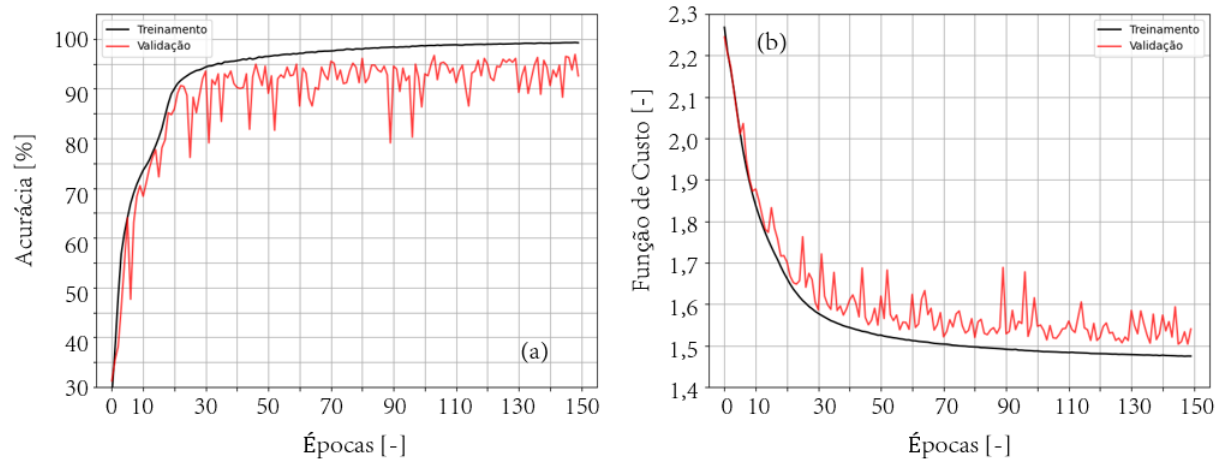
Esse arquivo “.csv” contém o nome de cada imagem e informações detalhadas, como tipo de falha, velocidade de rotação, intervalo de tempo do sinal e técnica utilizada, organizadas no formato “crop\_falha\_velocidade de rotação\_intervalo de tempo do sinal\_técnica utilizada”, conforme apresentada na Tabela 66 o respectivo conteúdo.

Tabela 66 – Dados estruturados para a base de rolamento.

| N  | Nome do arquivo                  | Rótulo      |
|----|----------------------------------|-------------|
| 1  | CROP_B007_HP0_DE_0DS-A-2DS_STFT  | <i>B07</i>  |
| 2  | CROP_B014_HP1_DE_0DS-A-2DS_STFT  | <i>B14</i>  |
| 3  | CROP_B021_HP2_DE_0DS-A-2DS_STFT  | <i>B21</i>  |
| 6  | CROP_IR007_HP1_DE_0DS-A-2DS_STFT | <i>IR07</i> |
| 7  | CROP_IR014_HP0_DE_0DS-A-2DS_STFT | <i>IR14</i> |
| 8  | CROP_IR021_HP1_DE_0DS-A-2DS_STFT | <i>IR21</i> |
| 9  | CROP_OR007_HP3_DE_0DS-A-2DS_STFT | <i>ER07</i> |
| 10 | CROP_OR014_HP2_DE_0DS-A-2DS_STFT | <i>ER14</i> |
| 11 | CROP_OR021_HP0_DE_0DS-A-2DS_STFT | <i>ER21</i> |

Apartir da base de dados e os hiperparâmetros definidos na Tabela 28 da Seção 4, realiza-se as fases de treinamento, validação e teste do modelo RNC. A Figura 98 ilustra os gráficos de convergência da função de custo e da acurácia média referente a cada época tanto para as fases de treino quanto para validação. Os valores de convergência para a acurácia e o custo foram de 99,23% e 1,49 respectivamente, durante o treinamento. Enquanto que para a validação, os valores atingiram 94,2% e 1,55.

Figura 98 – Desempenho da RNC na fase de treino: (a) acurácia; (b) função de custo.



Para avaliar a acurácia e precisão por classe, foram criadas matrizes de confusão para as fases de treinamento, validação e teste da rede, abrangendo todos os rótulos. Na Tabela 67, é exibida a matriz de confusão apenas para a fase de teste, mas as acurácias individuais para cada rótulo são fornecidas na Tabela 68 para as três fases. Nesta matriz os rótulos de origem do ensaio possui o índice  $r$  enquanto que os valores preditos pelo modelo não possuem índices. Nesses resultados, também ocorreram erros dos tipos  $I$  e  $II$ , por exemplo, na classe *ER07*, o modelo previu 274 ocorrências, das quais 34 pertencem às classes *ER21* e *IR14*, *B14* e *B21* respectivamente. Além disso, houve 3 amostras que pertencem à classe *ER07* que foram erroneamente classificadas em outras classes.

Tabela 67 – Matriz confusão do modelo RNC com dados da fase de teste.

|                         | <i>NRM</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> |
|-------------------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|
| <i>NRM<sub>r</sub></i>  | <b>265</b> | 0           | 0           | 0           | 0           | 0           | 0           | 0          | 0          | 0          |
| <i>ER07<sub>r</sub></i> | 0          | <b>240</b>  | 0           | 1           | 0           | 0           | 0           | 1          | 0          | 1          |
| <i>ER14<sub>r</sub></i> | 0          | 0           | <b>240</b>  | 0           | 0           | 0           | 0           | 16         | 0          | 0          |
| <i>ER21<sub>r</sub></i> | 0          | 3           | 0           | <b>223</b>  | 0           | 0           | 0           | 2          | 1          | 0          |
| <i>IR07<sub>r</sub></i> | 0          | 0           | 0           | 0           | <b>161</b>  | 0           | 0           | 1          | 0          | 0          |
| <i>IR14<sub>r</sub></i> | 0          | 27          | 0           | 7           | 0           | <b>132</b>  | 5           | 1          | 12         | 27         |
| <i>IR21<sub>r</sub></i> | 0          | 0           | 0           | 0           | 0           | 0           | <b>266</b>  | 0          | 0          | 0          |
| <i>B07<sub>r</sub></i>  | 0          | 0           | 1           | 0           | 0           | 0           | 0           | <b>135</b> | 0          | 0          |
| <i>B14<sub>r</sub></i>  | 0          | 2           | 0           | 4           | 0           | 0           | 0           | 4          | <b>129</b> | 4          |
| <i>21<sub>r</sub></i>   | 0          | 2           | 0           | 0           | 0           | 0           | 0           | 1          | 2          | <b>132</b> |

Com base na constatação dos erros dos tipos *I* e *II* presentes na matriz de confusão, realizou-se os cálculos da acurácia, precisão, sensibilidade e *F1-score* global e por classe (*One vs All*), apresentadas nas Tabelas 69, 68 e 70.

Tabela 68 – Precisão do modelo RNC para os dados de rolamento.

| Fase             | <i>NRM</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | Global       |
|------------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|--------------|
| <i>Train</i> [%] | 100,00     | 98,34       | 99,72       | 98,89       | 100,00      | 98,54       | 99,79       | 99,14      | 98,74      | 98,86      | <b>99,20</b> |
| <i>Valid</i> [%] | 100,00     | 94,30       | 99,46       | 96,43       | 100,00      | 99,76       | 98,94       | 91,45      | 93,00      | 89,44      | <b>96,28</b> |
| <i>Test</i> [%]  | 100,00     | 87,59       | 99,59       | 94,89       | 100,00      | 100,00      | 98,15       | 83,85      | 89,58      | 80,49      | <b>93,41</b> |

Tabela 69 – Sensibilidade do modelo RNC para os dados de rolamento.

| Fase             | <i>NRM</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | Global       |
|------------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|--------------|
| <i>Train</i> [%] | 100,00     | 97,89       | 99,37       | 98,09       | 100,00      | 96,54       | 99,72       | 98,65      | 95,96      | 97,26      | <b>98,35</b> |
| <i>Valid</i> [%] | 100        | 93,24       | 95,6        | 94,12       | 99,45       | 85,32       | 99,05       | 89,79      | 86,9       | 87,34      | <b>38,08</b> |
| <i>Test</i> [%]  | 100,00     | 98,77       | 93,39       | 97,38       | 99,38       | 62,56       | 100         | 99,26      | 90,21      | 96,35      | <b>93,73</b> |

Tabela 70 – *F1-score* do modelo RNC para os dados de rolamento.

| Fase             | <i>NRM</i> | <i>ER07</i> | <i>ER14</i> | <i>ER21</i> | <i>IR07</i> | <i>IR14</i> | <i>IR21</i> | <i>B07</i> | <i>B14</i> | <i>B21</i> | Global       |
|------------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|------------|------------|--------------|
| <i>Train</i> [%] | 100,00     | 98,11       | 99,54       | 98,49       | 100,00      | 97,53       | 99,75       | 98,90      | 97,33      | 98,06      | <b>98,85</b> |
| <i>Valid</i> [%] | 100,00     | 93,77       | 97,49       | 95,26       | 99,72       | 91,98       | 98,99       | 90,61      | 89,84      | 88,38      | <b>94,60</b> |
| <i>Test</i> [%]  | 100,00     | 92,84       | 96,58       | 96,12       | 99,69       | 76,97       | 99,07       | 90,91      | 89,90      | 87,71      | <b>92,98</b> |

Com relação aos resultados globais do modelo RNC, obteve-se 93,41% de acurácia; 93,77% sensibilidade, 93,41% de precisão e 92,98% no *F1-score*. Esses resultados foram calculados com base na matriz de confusão e na utilização das Equações 6.1, 6.2, 6.3 e 6.4. Como o modelo *LeNet* – 5 apresentou excelentes resultados, não houve necessidade de calcular novos hiperparâmetros por meio da otimização bayesiana.

## 6.4 Desempenho dos modelos RNMP e RNC

Esta seção tem como objetivo realizar uma análise comparativa do desempenho dos modelos RNMP e RNC no contexto do diagnóstico de falhas em rolamentos, baseada nos resultados obtidos por (ZHANG et al., 2020). A comparação é realizada por meio de uma validação cruzada em quatro subconjuntos de dados incluindo o cálculo da média e do desvio padrão desses resultados. Por conveniência, realizou a mesma validação cruzada apresentada em (ZHANG et al., 2020).

Considerando que o pré-processamento dos modelos RNC e RNMP segue critérios discutidos nessa pesquisa, a quantidade de dados utilizada em cada modelo pode diferir inclusive com (ZHANG et al., 2020). No entanto, o número de conjuntos de amostras permanece constante, com quatro conjuntos sendo aplicados em cada modelo. Na Tabela 71, são apresentadas as quantidades de dados dos conjuntos de amostras *A*, *B*, *C*, *D* e *E*.

Tabela 71 – Dados para a validação cruzada dos modelos DFCNN, MLP, RNMP e RNC.

| Categoria   | Zhang et al. (2020) |          |                |          | Autor.         |          |                |          |
|-------------|---------------------|----------|----------------|----------|----------------|----------|----------------|----------|
|             | MLP                 |          | DFCNN          |          | RNMP           |          | RNC            |          |
|             | <i>A/B/C/D</i>      | <i>E</i> | <i>A/B/C/D</i> | <i>E</i> | <i>A/B/C/D</i> | <i>E</i> | <i>A/B/C/D</i> | <i>E</i> |
| <i>NRM</i>  | 50                  | 200      | 50             | 200      | 4.287          | 17.148   | 442            | 1768     |
| <i>B07</i>  | 50                  | 200      | 50             | 200      | 1.150          | 4.601    | 444            | 1779     |
| <i>B14</i>  | 50                  | 200      | 50             | 200      | 1.982          | 7.929    | 445            | 1780     |
| <i>B21</i>  | 50                  | 200      | 50             | 200      | 2.070          | 8.281    | 444            | 1776     |
| <i>IR07</i> | 50                  | 200      | 50             | 200      | 1.566          | 6.265    | 443            | 1773     |
| <i>IR14</i> | 50                  | 200      | 50             | 200      | 2.556          | 10.227   | 374            | 1498     |
| <i>IR21</i> | 50                  | 200      | 50             | 200      | 4.012          | 16.049   | 445            | 1781     |
| <i>ER07</i> | 50                  | 200      | 50             | 200      | 1.169          | 4.679    | 411            | 1645     |
| <i>ER14</i> | 50                  | 200      | 50             | 200      | 1.555          | 6.220    | 443            | 1775     |
| <i>ER21</i> | 50                  | 200      | 50             | 200      | 3.300          | 13.201   | 414            | 1656     |

Inicialmente, os modelos foram treinados individualmente para cada subconjunto de amostras. Em seguida, foi realizada a validação cruzada, em que cada conjunto foi testado com o conjuntos vizinhos. Por exemplo, o modelo treinado com o conjunto *A* foi testado nos conjuntos *B*, *C* e *D*, e assim sucessivamente. Espera-se que em ambas as validações cruzadas apresentem uma acurácia superior na condição de treinamento quando comparado aos testes realizados em seus pares, proporcionando um indicativo de que o modelo conseguiu classificação generalista, ou seja, sem ocorrência de *overfitting*.

Nas Tabelas 72 e 73 são apresentados os resultados das validações cruzadas para os modelos RNMP e RNC, respectivamente. Percebe-se que não foi realizado o teste com o conjunto de dados da base *E*, uma vez que esses resultados foram apresentados nas respectivas seções de cada modelo.

Tabela 72 – Validação cruzada do modelo RNMP para os dados de rolamento.

| $\begin{matrix} \text{Test} \\ \text{Train} \end{matrix}$ | $A$           | $B$           | $C$           | $D$           | Média  |
|-----------------------------------------------------------|---------------|---------------|---------------|---------------|--------|
| $A$                                                       | <b>81,41%</b> | 80,71%        | 80,14%        | 79,49%        | 80,11% |
| $B$                                                       | 79,82%        | <b>81,50%</b> | 80,82%        | 80,59%        | 80,41% |
| $C$                                                       | 78,21%        | 79,32%        | <b>80,40%</b> | 79,68%        | 79,07% |
| $D$                                                       | 79,30%        | 80,00%        | 79,95%        | <b>80,76%</b> | 79,75% |

Tabela 73 – Validação cruzada do modelo RNC para os dados de rolamento.

| $\begin{matrix} \text{Test} \\ \text{Train} \end{matrix}$ | $A$           | $B$           | $C$           | $D$           | Média  |
|-----------------------------------------------------------|---------------|---------------|---------------|---------------|--------|
| $A$                                                       | <b>99,64%</b> | 94,98%        | 94,32%        | 94,44%        | 94,58% |
| $B$                                                       | 88,70%        | <b>99,55%</b> | 87,68%        | 88,04%        | 88,14% |
| $C$                                                       | 92,07%        | 92,37%        | <b>99,31%</b> | 92,25%        | 92,23% |
| $D$                                                       | 95,05%        | 94,95%        | 94,32%        | <b>99,43%</b> | 94,77% |

Percebe-se que em ambas as validações cruzadas a acurácia do valor de treinamento mantiveram maiores que os testes cruzados, e a média global é inferior aos modelos com a base  $E$ . No caso do modelo RNMP o desvio padrão dos valores da média foi de 0,23%, enquanto que no modelo RNC esse valor foi de 2,95%. Esse desvio foi devido a base  $B$  apresentar uma acurácia inferior a 90%. Apartir das validações cruzadas dos modelos RNMP e RNC, resgatou os resultados aprendizados por [Zhang et al. \(2020\)](#) e montou a mesma tabela e o gráfico de desempenho da validação cruzada, conforme apresentadas na Tabela 74 e Figura 99.

Tabela 74 – Desempenho entre diferentes modelos de aprendizado de máquina.

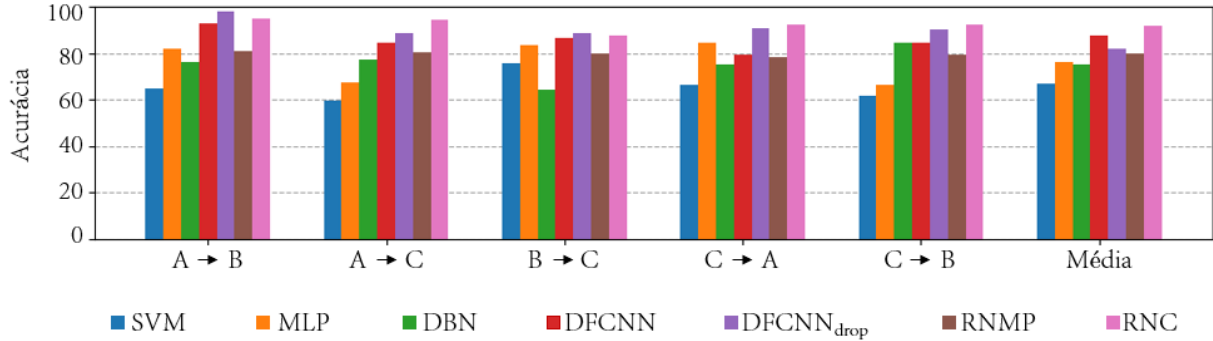
| Modelos               | $A - B$ | $A - C$ | $B - A$ | $B - C$ | $C - A$ | $C - B$ | Média       |
|-----------------------|---------|---------|---------|---------|---------|---------|-------------|
| SVM                   | 64,62%  | 59,56%  | 72,33%  | 75,67%  | 66,33%  | 61,32%  | 66,64±6,27% |
| MLP                   | 82,06%  | 67,53%  | 71,61%  | 83,39%  | 84,22%  | 66,33%  | 75,86±8,28% |
| DBN                   | 76,05%  | 77,31%  | 72,10%  | 64,40%  | 75,11%  | 84,45%  | 74,97±6,55% |
| DFCNN                 | 92,67%  | 84,45%  | 97,61%  | 86,32%  | 79,27%  | 84,63%  | 87,49±6,56% |
| DFCNN <sub>drop</sub> | 97,97%  | 88,74%  | 90,54%  | 90,38%  | 82,06%  | 84,92%  | 90,60±6,97% |
| RNMP                  | 80,71%  | 80,14%  | 79,82%  | 80,82%  | 78,21%  | 79,32%  | 79,84±0,89% |
| RNC                   | 94,98%  | 94,32%  | 88,70%  | 87,68%  | 92,07%  | 92,37%  | 91,68±2,95% |

Fonte: Adaptado de [Zhang et al. \(2020\)](#)

Os resultados apresentados na Tabela 74, mostram que os modelos RNMP e RNC obtiveram valores de desvio padrão menores em comparação com os resultados obtidos por [Zhang et al. \(2020\)](#). Em termos de valores médios, os modelos baseados em redes neurais convolucionais se destacaram pela maior acurácia, com especial ênfase para o modelo

RNC. A Figura 99 apresenta os mesmos resultados de acurácia dos modelos, porém com a inclusão dos modelos RNMP e RNC.

Figura 99 – Desempenho entre diferentes modelos de aprendizado de máquina.



Fonte: Adaptado de Zhang et al. (2020)

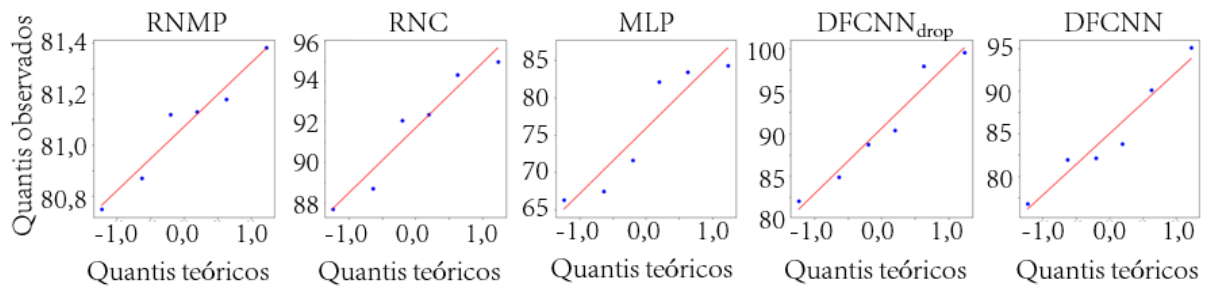
Para aplicar uma análise de hipótese, deve-se verificar se os dados apresentam um comportamento de normalidade. Por este motivo, aplicou-se os testes Shapiro-Wilk, Kolmogorov-Smirnov e o gráfico  $Q - Q$ , para os valores de acurácias dos modelos MLP, RNMP, DFCNN, DFCNN<sub>drop</sub> e RNC. Os resultados dos  $p - \text{valores}$  obtidos são apresentados na Tabela 75, enquanto que na Figura 100 são apresentados os gráficos  $Q - Q$ .

Tabela 75 – Testes de normalidade dos dados.

| Modelos               | Shapiro-Wilk | Kolmogorov-Smirnov | $\alpha$ | Hipótese $H_0$    |
|-----------------------|--------------|--------------------|----------|-------------------|
| MLP                   | 0,11         | 0,29               | 0,05     | Não rejeita $H_0$ |
| DFCNN                 | 0,67         | 0,24               | 0,05     | Não rejeita $H_0$ |
| DFCNN <sub>drop</sub> | 0,59         | 0,91               | 0,05     | Não rejeita $H_0$ |
| RNMP                  | 0,71         | 0,73               | 0,05     | Não rejeita $H_0$ |
| RNC                   | 0,47         | 0,87               | 0,05     | Não rejeita $H_0$ |

Nota:  $H_0$  - Os resultados de acurácias dos modelos seguem uma distribuição normal.

Figura 100 – Gráficos  $Q - Q$ .



Com base nos testes estatísticos e na análise gráfica  $Q - Q$ , conclui-se que não há evidências para rejeitar a normalidade dos dados e portanto os dados seguem uma distribuição normal.

Para determinar qual das redes neurais com a mesma estrutura - RNMP com MLP, RNC com DFCNN e DFCNN<sub>drop</sub> - apresenta as melhores acurácias de forma significativa, emprega-se testes de hipóteses. Utilizando o teste  $t$  de Student para verificar se as médias são iguais, e os testes Mann-Whitney  $U$  e Wilcoxon Signed-Rank para verificar se as distribuições das amostras são iguais. Esses testes são realizados com base nos dados apresentados na Tabela 74.

Para o caso dos modelos RNMP e MLP, obteve-se os valores das estatística  $t$ ,  $U$  e  $W$  foram de 1,54, 21 e 6, respectivamente enquanto que os  $p$  - valores foram de 0,15, 0,35 e 0,22. Ao aplicar uma condição 95% de grau de confiança ( $\alpha=5\%$ ), não se rejeita a hipótese nula para nenhum dos testes, pois não existe uma diferença significativa entre as acurácias médias entre as redes neurais e nem suas distribuições.

Para os casos dos modelos DFCNN, DFCNN<sub>drop</sub> e RNC, foram realizados duas análises de hipótese a primeira foi a RNC com a DFCNN, enquanto que a segunda foi da RNC com a DFCNN<sub>drop</sub>. Em ambas as comparações, os valores estatísticos foram os seguintes: para o teste  $t$ , 1,45 e 0,45; para  $U$ , 21 e 16; e para  $W$ , 4 e 9. Os  $p$  - valores correspondentes para os testes  $t$  foram 0,73 e 0,17, para  $U$  foram 0,11 e 0,40, e para  $W$  foram 0,22 e 0,84, respectivamente. Com um nível de significância de 5%, não foi possível rejeitar a hipótese nula em nenhum dos testes, indicando que não há diferença significativa entre as acurácias dos modelos ou suas distribuições. Na Tabela 76 é apresentado o resumo dos testes de hipótese.

Tabela 76 – Comparação de desempenho entre diferentes modelos

| Teste        | Modelo                     | Estatística | $p$ - valor | $\alpha$ | hipótese $H_0$        |
|--------------|----------------------------|-------------|-------------|----------|-----------------------|
| t-Student    | RNMP-MLP                   | 1,54        | 0,15        | 0,05     | Não rejeita $H_{0-t}$ |
|              | RNC-DFCNN                  | 1,45        | 0,17        | 0,05     | Não rejeita $H_{0-t}$ |
|              | RNC- DFCNN <sub>drop</sub> | 0,45        | 0,73        | 0,05     | Não rejeita $H_{0-t}$ |
| Mann-Whitney | RNMP-MLP                   | 21,0        | 0,35        | 0,05     | Não rejeita $H_{0-U}$ |
|              | RNC-DFCNN                  | 10,0        | 0,11        | 0,05     | Não rejeita $H_{0-U}$ |
|              | RNC- DFCNN <sub>drop</sub> | 16,0        | 0,40        | 0,05     | Não rejeita $H_{0-U}$ |
| Wilcoxon     | RNMP-MLP                   | 6,0         | 0,22        | 0,05     | Não rejeita $H_{0-W}$ |
|              | RNC-DFCNN                  | 4,0         | 0,22        | 0,05     | Não rejeita $H_{0-W}$ |
|              | RNC- DFCNN <sub>drop</sub> | 9,0         | 0,84        | 0,05     | Não rejeita $H_{0-W}$ |

*Nota:*  $H_{0-t}$  - As acurácias médias entre as redes são iguais.

$H_{0-U}$  e  $H_{0-W}$  - As distribuições das acurácias são iguais

Uma análise de variância (ANOVA) é realizada sobre as acurácias originadas pela validação cruzada apresentada na Tabela 74. Além dessa análise é realizada um teste de normalidade dos resíduos por meio do Teste de Shapiro-Wilk e o gráfico  $Q-Q$ , o testes de

homogeneidade das variâncias utilizando o teste de Levene e o box-plot com a finalidade de visualização dos dados, identificação de métricas estatísticas.

- Teste de Shapiro-Wilk: O  $p$  – valor deste teste é 0.2709. Como este valor é maior que 0,05, não rejeita-se a hipótese nula, pois os resíduos seguem uma distribuição normal.
- Teste de Levene: O  $p$  – valor deste teste é 0.00308. Como este valor é menor que 0.05, rejeita-se a hipótese nula de que as variâncias são iguais. Isto indica que as variâncias entre os grupos não são homogêneas.
- *Boxplot*: Ao observar o gráfico do *boxplot* que as distribuições são não simétrica, que os modelos MLP, CNN e DFCNN<sub>drop</sub> são os que possuem as maiores variâncias e amplitude interquartil, além de não existirem *outlayers*..
- Gráfico  $Q - Q$  dos Resíduos: Percebe-se nesse gráfico que os resíduos formam uma tendência de reta, indicando que os resíduos são aproximadamente normais.

Figura 101 – (a) *Boxplot*; (b) Gráfico  $Q - Q$ .

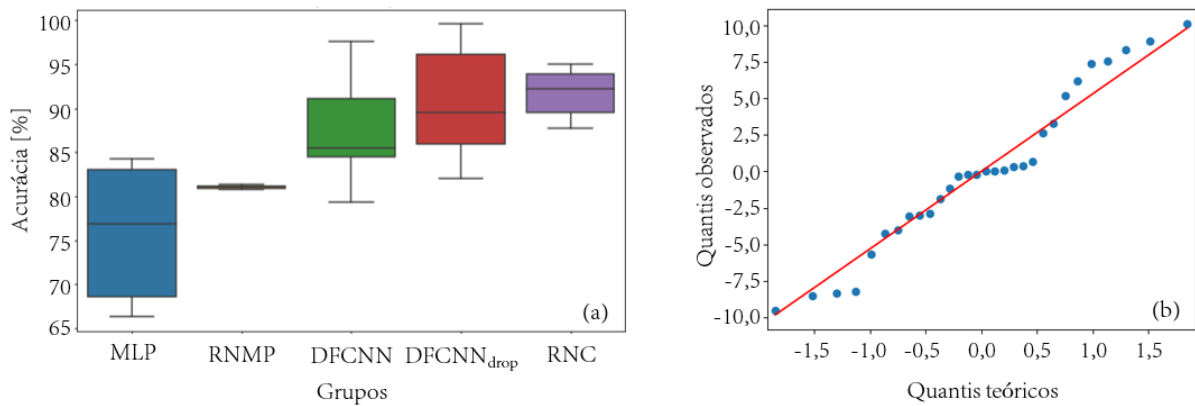


Tabela 77 – Comparação de desempenho entre diferentes modelos

| Fonte de variação | Soma dos quadrados | Graus de liberdade | Quadrado médio | $F$      |
|-------------------|--------------------|--------------------|----------------|----------|
| Regressão         | 1084,48            | 4                  | 8.01682        | 0.000265 |
| Resíduos          | 845,47             | 25                 |                |          |
| Total             | 1929,95            | 29                 |                |          |

Ao considerar uma nível de significância de 5%, ou seja  $p$  – valor = 0,05, rejeita-se a hipótese nula, uma vez que o fator  $F$  é menor que  $p$  – valor. Isso indica que há uma diferença estatisticamente significativa entre as médias dos cinco grupos (MLP, RNMP, CNN, DFCNN<sub>drop</sub>, RNC), ou seja pelo menos uma das média é diferente das demais.

## 6.5 Dados de Cavitação

### 6.5.1 Análise Exploratória dos Dados

Os dados estão organizados conforme a condição operativa da bomba, vazão e o número do ensaio (altura manométrica), que impacta no  $NPSH$  e consequentemente na presença ou não do fenômeno de cavitação, resultando 20 rótulos. No entanto, é possível agrupar esses rótulos de forma a abstrair a condição operativa, permitindo observar apenas o fenômeno de cavitação. Nesta seção, utilizou a análise estatística sobre o valor médio do sinal para a criação de novos rótulos além dos definidos pelas variável  $NPSH_{\text{ration}}$ .

Essa rotulagem incluiu várias análises estatísticas, como histogramas, gráficos  $Q - Q$  para análises qualitativas, testes de Shapiro-Wilk e Kolmogorov-Smirnov para verificação da normalidade, teste de Levene para variância, t de Student, t de Welch, Mann-Whitney U, análise de variância (ANOVA) combinada com o teste de Tukey, e os testes de Kruskal-Wallis e Dunn para verificar a igualdade entre médias.

Houve a necessidade de testes de hipóteses tanto para dados com comportamento normal quanto para aqueles sem, devido à frequente contradição nas informações, especialmente em relação à normalidade. Enquanto os testes de Shapiro-Wilk e Kolmogorov-Smirnov frequentemente rejeitavam a hipótese nula, histogramas e gráficos Q-Q sugeriam normalidade, independentemente do tamanho da amostra. Portanto, todas as análises foram realizadas para verificar a consistência das informações na base de dados.

Esses testes também foram realizados com diferentes tamanhos de amostras -  $N$ ,  $N/10$ ,  $N/100$  e  $N/1000$  - para diagnosticar a sensibilidade da análise em relação ao conjunto em estudo, em que  $N$  é o tamanho do conjunto de dados. Na Tabela 78 são apresentados o números de amostras contida em cada ensaio.

Tabela 78 – Tamanhos das amostras dos dados de cavitação para o estudo estatístico.

| Classe        | Ensaio    | Vazão [ $m^3/h$ ] |         |         |         |
|---------------|-----------|-------------------|---------|---------|---------|
|               |           | Q55               | Q108    | Q161    | Q193    |
| <i>OFFCAV</i> | <i>E0</i> | 327.600           | 327.600 | 327.600 | 327.600 |
| <i>OFFCAV</i> | <i>E1</i> | 327.600           | 327.600 | 327.600 | 327.600 |
| <i>OFFCAV</i> | <i>E2</i> | 327.600           | 327.600 | 327.600 | 327.600 |
| <i>OFFCAV</i> | <i>E3</i> | 327.600           | 327.600 | 327.600 | 327.600 |
| <i>ONCAV</i>  | <i>E4</i> | 327.600           | 327.600 | 327.600 | 327.600 |

Dentro desse contexto, foi selecionado o tamanhos de amostra  $n = N/100$  para formar os grupos que constituem os rótulos. Esses tamanhos estão em conformidade com os requisitos da biblioteca *statsmodels* na linguagem de programação *Python*, que recomenda amostras menores que  $n < 5.000$ . A seguir, são apresentados os testes de normalidade,

variância e hipóteses, especificamente a análise de variância combinada com o teste de Tukey e o teste de Kruskal-Wallis com o teste de Dunn.

- Teste de Shapiro-Wilk: O  $p$  – valor destes testes são apresentados na Tabela 79 para as condições com e sem cavitação. Por tanto, segundo esse teste as condições  $E1.Q55$ ,  $E2.Q55$ ,  $E3.Q108$ ,  $E1.Q161$ ,  $E2.Q161$  e  $E3.Q161$  não seguem, seguem uma distribuição Gaussiana, enquanto que as demais seguem esse tipo de distribuição.

Tabela 79 – Valores –  $p$  do teste Shapiro-Wilk para os dados de cavitação com  $N/100$ .

| Classe        | Ensaio    | Vazão [ $m^3/s$ ] |         |         |         |
|---------------|-----------|-------------------|---------|---------|---------|
|               |           | $Q.55$            | $Q.108$ | $Q.161$ | $Q.193$ |
| <i>OFFCAV</i> | <i>E0</i> | 1E-1              | 5E-1    | 2E-1    | 3E-1    |
| <i>OFFCAV</i> | <i>E1</i> | 5E-4              | 6E-1    | 4E-2    | 6E-1    |
| <i>OFFCAV</i> | <i>E2</i> | 3E-5              | 5E-1    | 3E-3    | 1E-1    |
| <i>OFFCAV</i> | <i>E3</i> | 5E-1              | 1E-4    | 2E-6    | 4E-1    |
| <i>ONCAV</i>  | <i>E4</i> | 2,1E-1            | 4,4E-1  | 9,1E-1  | 5,7E-1  |

- Teste de Kolmogorov-Smirnov: Todos os  $p$  – valores deste teste foram iguais a zero, rejeitando a hipótese nula, pois os dados não seguem uma distribuição normal.
- Histograma e gráficos  $Q - Q$ : Para entender a distribuição dos dados, utilizou histogramas e gráficos  $Q - Q$  (*Quantile-Quantile*). Nas Figuras 102 e 103 são apresentados os histogramas e gráficos  $Q - Q$  para todas as condições com a presença do fenômeno e para quatro condições sem cavitação. Em todos os gráficos, observa-se uma tendência de comportamento normal para todas as condições, divergindo de alguns dos resultados obtidos nos testes de Kolmogorov-Smirnov e Shapiro-Wilk.

Figura 102 – Histogramas para as condições com e sem cavitação com  $N/100$ .

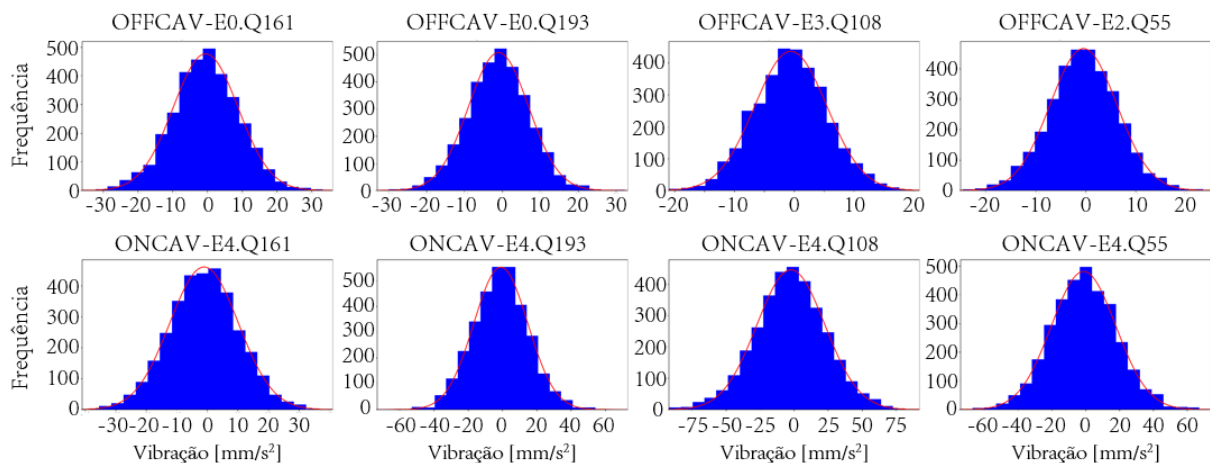
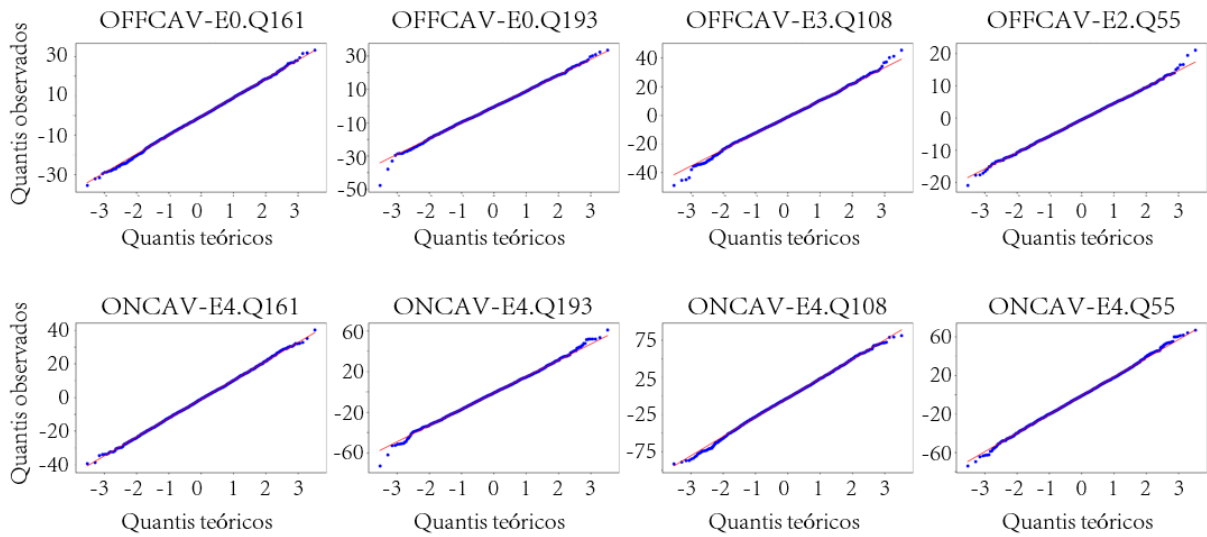


Figura 103 – Gráficos  $Q - Q$  para as condições com e sem cavitação com  $N/100$ .

- Teste de Levene: Com esse teste, busca verificar se duas séries temporais possuem variâncias iguais. Por possuir várias séries temporais foi realizado esse teste de forma pareada ao ponto de cobrir todas combinações sem repetição. Os resultados diferentes de zero são apresentados na Tabela 80 por meio do *valor - p*. Como estes valores são menores que 0.05, rejeita-se a hipótese nula de que as variâncias são iguais para

Tabela 80 – *Valor - p* do teste de Levene com tamanho  $N/100$ .

| Condição Operativa      |                         |  | Valor-p | Hipótese          |
|-------------------------|-------------------------|--|---------|-------------------|
| <i>OFFCAV - E0.Q161</i> | <i>OFFCAV - E1.Q161</i> |  | 2E-48   | Rejeita $H_0$     |
| <i>OFFCAV - E2.Q161</i> | <i>OFFCAV - E1.Q193</i> |  | 9E-01   | Não Rejeita $H_0$ |
| <i>OFFCAV - E2.Q193</i> | <i>OFFCAV - E3.Q193</i> |  | 1E-21   | Rejeita $H_0$     |

Nota: *valor - p* < 0,05 - Rejeita  $H_0$ , os dados não possuem as mesmas variâncias

- Análise de variância e teste de Tukey: Foi realizada uma análise de variância seguida do teste de Tukey para comparar as médias entre os grupos com e sem cavitação. Na Tabela 81 são apresentados os resultados para as condições com cavitação.

Tabela 81 – Análise de variância e teste Tukey para *ONCAV* com  $\alpha=0.05$ .

| Grupo 1        | Grupo 2        | Diferença Médias | p-valor | Inferior | Superior | Rejeitar $H_0$ |
|----------------|----------------|------------------|---------|----------|----------|----------------|
| <i>E4.Q193</i> | <i>E4.108</i>  | -1,6309          | 0,0027  | -2,8315  | -0,4303  | TRUE           |
| <i>E4.Q193</i> | <i>E4.Q55</i>  | 0,0238           | 1       | -1,1768  | 1,2244   | FALSE          |
| <i>E4.Q193</i> | <i>E4.Q161</i> | 0,0016           | 1       | -1,199   | 1,2022   | FALSE          |
| <i>E4.Q108</i> | <i>E4.Q55</i>  | 1,6547           | 0,0023  | 0,4541   | 2,8553   | TRUE           |
| <i>E4.Q108</i> | <i>E4.Q161</i> | 1,6325           | 0,0027  | 0,4319   | 2,8331   | TRUE           |
| <i>E4.Q55</i>  | <i>E4.Q161</i> | -0,0221          | 1       | -1,2228  | 1,1785   | FALSE          |

Já na Tabela 82, fornece uma interpretação resumida para a condição sem cavitação, utilizando a biblioteca statsmodels do Python, que fornece a rejeição do  $H_0$  a um nível de significância de 5%. Esses testes assumem que os dados seguem uma distribuição normal, conforme indicado pelos histogramas e gráficos  $Q-Q$  apresentados nas Figuras 102 e 103.

Tabela 82 – Análise de variância e teste Tukey para *OFFCAV* com  $\alpha=0.05$ .

| Ensaio  | Q161 |    |    |    | Q193 |    |    |    | Q108 |    |    |    | Q55 |    |    |
|---------|------|----|----|----|------|----|----|----|------|----|----|----|-----|----|----|
|         | E0   | E1 | E2 | E3 | E0   | E1 | E2 | E3 | E0   | E1 | E2 | E3 | E0  | E1 | E2 |
| E0.Q161 | T    | F  | F  | F  | F    | F  | F  | F  | F    | F  | T  | F  | F   | F  | F  |
| E1.Q161 |      | T  | F  | F  | F    | F  | F  | F  | F    | F  | T  | F  | F   | F  | F  |
| E2.Q161 |      |    | T  | F  | F    | F  | F  | F  | F    | F  | T  | F  | F   | F  | F  |
| E3.Q161 |      |    |    | T  | F    | F  | F  | F  | F    | F  | T  | F  | F   | F  | F  |
| E0.Q193 |      |    |    |    | T    | F  | F  | F  | F    | F  | T  | F  | F   | F  | F  |
| E1.Q193 |      |    |    |    |      | T  | F  | F  | F    | F  | T  | F  | F   | F  | F  |
| E2.Q193 |      |    |    |    |      |    | T  | F  | F    | F  | T  | F  | F   | F  | F  |
| E3.Q193 |      |    |    |    |      |    |    | T  | F    | F  | T  | F  | F   | F  | F  |
| E0.Q108 |      |    |    |    |      |    |    |    | T    | F  | T  | F  | F   | F  | F  |
| E1.Q108 |      |    |    |    |      |    |    |    |      | T  | T  | F  | F   | F  | F  |
| E2.Q108 |      |    |    |    |      |    |    |    |      |    | T  | F  | F   | F  | F  |
| E3.Q108 |      |    |    |    |      |    |    |    |      |    |    | T  | T   | T  | T  |
| E0.Q55  |      |    |    |    |      |    |    |    |      |    |    |    | T   | F  | F  |
| E1.Q55  |      |    |    |    |      |    |    |    |      |    |    |    |     | T  | F  |
| E2.Q55  |      |    |    |    |      |    |    |    |      |    |    |    |     |    | T  |

- Testes Kruskal-Wallis e Dunn: Foram realizados testes de hipóteses para dados que não apresentaram distribuição normal, conforme a Tabela 79. O teste de Kruskal-Wallis e Dunn foi aplicado para verificar se as médias dos grupos com e sem cavitação são iguais. Os resultados dessas análises, com um nível de significância de 5%, estão nas Tabelas 83 e 84.

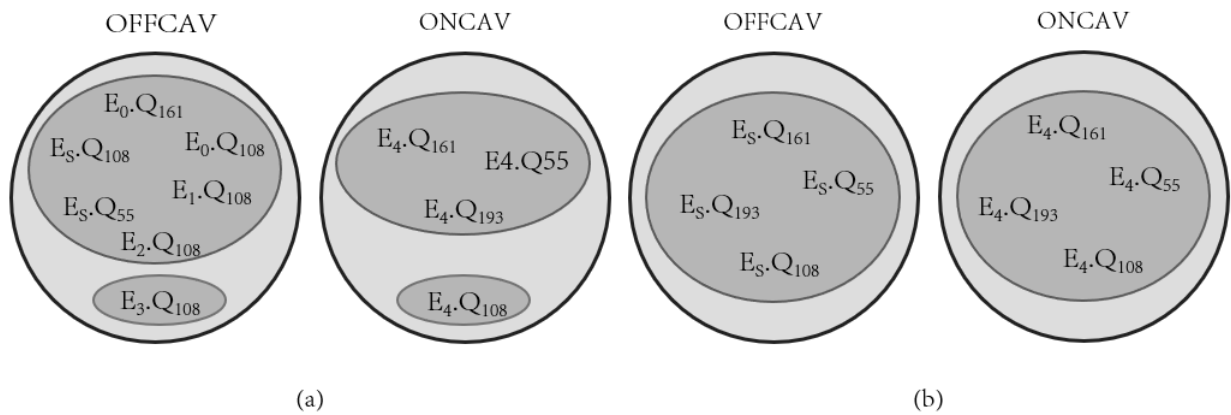
Tabela 83 – Testes Kruskal-Wallis e Dunn para *ONCAV* com  $\alpha=0.05$ .

| Ensaio                        | <i>ONCAV</i> – <i>E4</i> . |        |        |        |
|-------------------------------|----------------------------|--------|--------|--------|
|                               | Q161                       | Q193   | Q108   | Q55    |
| <i>ONCAV</i> – <i>E4.Q161</i> | 1                          | 1      | 2,1E-2 | 1      |
| <i>ONCAV</i> – <i>E4.Q193</i> | 1                          | 1      | 2,5E-2 | 1      |
| <i>ONCAV</i> – <i>E4.Q108</i> | 2,1E-2                     | 2,5E-2 | 1      | 3,3E-2 |
| <i>ONCAV</i> – <i>E4.Q55</i>  | 1                          | 1      | 3,4E-2 | 1      |

Tabela 84 – Testes Kruskal-Wallis e Dunn para OFFCAV com  $\alpha=0.05$ .

| Ensaio  | Q161 |    |    |    | Q193 |    |    |    | Q108 |    |    |      | Q55  |      |      |      |
|---------|------|----|----|----|------|----|----|----|------|----|----|------|------|------|------|------|
|         | E0   | E1 | E2 | E3 | E0   | E1 | E2 | E3 | E0   | E1 | E2 | E3   | E0   | E1   | E2   | E3   |
| E0.Q161 | 1    | 1  | 1  | 1  | 1    | 1  | 1  | 1  | 1    | 1  | 1  | 7E-9 | 1    | 1    | 1    | 1    |
| E1.Q161 |      | 1  | 1  | 1  | 1    | 1  | 1  | 1  | 1    | 1  | 1  | 2E-8 | 1    | 1    | 1    | 1    |
| E2.Q161 |      |    | 1  | 1  | 1    | 1  | 1  | 1  | 1    | 1  | 1  | 4E-3 | 8E-1 | 1    | 1    | 8E-1 |
| E3.Q161 |      |    |    | 1  | 1    | 1  | 1  | 1  | 1    | 1  | 1  | 3E-3 | 9E-1 | 1    | 1    | 1    |
| E0.Q193 |      |    |    |    | 1    | 1  | 1  | 1  | 1    | 1  | 1  | 4E-3 | 8E-1 | 1    | 1    | 9E-1 |
| E1.Q193 |      |    |    |    |      | 1  | 1  | 1  | 1    | 1  | 1  | 3E-3 | 1    | 1    | 1    | 1    |
| E2.Q193 |      |    |    |    |      |    | 1  | 1  | 1    | 1  | 1  | 3E-3 | 9E-1 | 1    | 1    | 1    |
| E3.Q193 |      |    |    |    |      |    |    | 1  | 1    | 1  | 1  | 2E-3 | 1    | 1    | 1    | 1    |
| E0.Q108 |      |    |    |    |      |    |    |    | 1    | 1  | 1  | 2E-9 | 1    | 1    | 1    | 1    |
| E1.Q108 |      |    |    |    |      |    |    |    |      | 1  | 1  | 2E-8 | 1    | 1    | 1    | 1    |
| E2.Q108 |      |    |    |    |      |    |    |    |      |    | 1  | 1E-8 | 1    | 1    | 1    | 1    |
| E3.Q108 |      |    |    |    |      |    |    |    |      |    |    | 1    | 2E-9 | 3E-3 | 1E-9 | 2E-9 |
| E0.Q55  |      |    |    |    |      |    |    |    |      |    |    |      | 1    | 1    | 1    | 1    |
| E1.Q55  |      |    |    |    |      |    |    |    |      |    |    |      |      | 1    | 1    | 1    |
| E2.Q55  |      |    |    |    |      |    |    |    |      |    |    |      |      |      | 1    | 1    |
| E3.Q55  |      |    |    |    |      |    |    |    |      |    |    |      |      |      |      | 1    |

Ao final dessas análises, são construídos diagramas de Venn para observar os grupos com médias estatisticamente iguais. As formações dos grupos para a análise de variâncias com o teste de Tukey e teste Kruskal-Wallis com Dunn são apresentadas na Figura 104.

Figura 104 – Rótulos dos dados de cavitação: (a)  $N/100$ ; (b)  $N/1000$ .

Conclui-se que os grupos de cada classe podem ser organizados de acordo com a intensidade da cavitação. Para a condição com  $N/100$ , houve a classificação de duas categorias para cada condição de cavitação, ou seja quatro grupos. A criação dessas duas categorias possivelmente está relacionada às diferentes intensidades

de cavitação dentro das condições operativas. A condição *OFFCAV* – *E3.Q108* pode ser tratada como uma condição iminente do fenômeno, enquanto a condição *OFFCAV* – *E4.Q108* pode representar uma condição severa de cavitação. Ao tratar os dados com  $N/1000$ , constatou que os dados podem ser organizados apenas em dois grupos: com e sem o fenômeno de cavitação. Portanto, ao utilizar amostras menores, como no caso de TFTC, dois grupos são suficientes para que o modelo de aprendizado de máquina faça a classificação.

## 6.5.2 Definição do Valor Limiar

Para estimar o valor do limiar utilizado na técnica de contagem de hits, foi realizada uma análise estatística descritiva dos dados. As métricas estatísticas utilizadas incluíram a média e o desvio padrão para compreender o comportamento da base de dados e identificar tendências das distribuição dos dados com e sem a presença do fenômeno de cavitação.

Na Tabela 85 são apresentados os valores estatísticos encontrados sobre a base de dados que está organizada de acordo com a condição de operação, representada por vazão, número do ensaio e o fenômeno de cavitação.

Tabela 85 – Estatística descritiva dos dados de cavitação.

| Classe        | Ensaio    | Média [ $mm/s^2$ ] |         |         |         | Desvio padrão [ $mm^2/s$ ] |         |         |         |
|---------------|-----------|--------------------|---------|---------|---------|----------------------------|---------|---------|---------|
|               |           | Q55                | Q108    | Q161    | Q193    | Q55                        | Q108    | Q161    | Q193    |
| <i>OFFCAV</i> | <i>E0</i> | -0,5418            | -0,5410 | -0,5402 | -1,0798 | 3,6210                     | 4,9732  | 9,1547  | 8,2663  |
| <i>OFFCAV</i> | <i>E1</i> | -0,5418            | -0,5428 | -0,5415 | -1,0818 | 4,5789                     | 6,3869  | 9,4443  | 10,9914 |
| <i>OFFCAV</i> | <i>E2</i> | -1,0846            | -0,5424 | -1,0841 | -1,0822 | 12,500                     | 7,0185  | 11,0112 | 11,9371 |
| <i>OFFCAV</i> | <i>E3</i> | -0,5418            | -2,7128 | -1,0794 | -1,0804 | 4,3133                     | 18,2349 | 12,8053 | 12,5279 |
| <i>ONCAV</i>  | <i>E4</i> | -1,0865            | -2,7078 | -1,0846 | -1,0840 | 25,735                     | 20,345  | 10,852  | 16,301  |

Os valores médios desses sinais demonstram que a condição operativa é influenciada pela vazão volumétrica e pelo valor do NPSH<sub>req</sub>. Com relação ao desvio padrão, as condições *ONCAV* apresentaram os maiores valores quando comparado as condições *OFFCAV*.

Para estimar um valor de limiar adequado aos dados, optou-se por utilizar os histogramas de frequência com um nível de significancia de 99,72% e 68,2%, o que corresponde a  $\sigma$  e  $3 \times \sigma$ . Assim valores superiores a esses limites são registrados e caracterizam a forma da onda. Nas Figuras 105 e 106, são ilustradas as distribuições dos dados referente as classes *ONCAV* e *OFFCAV*, respectivamente. Esses histogramas tem um comportamento Gaussiano e que os valores limiares devem estar em um intervalo de 0 à 60 [ $mm/s^2$ ].

Figura 105 – Histogramas dos dados com cavitação: (a) E4.Q55; (b) E4.Q108; (c) E4.Q161 e (d) E4.Q193.

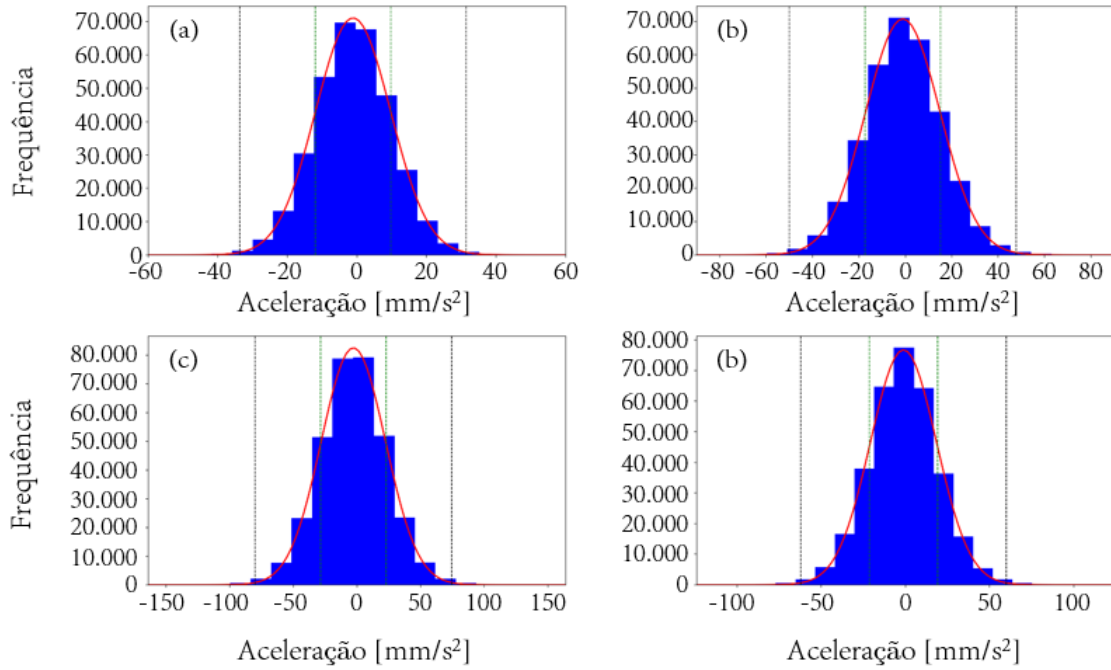
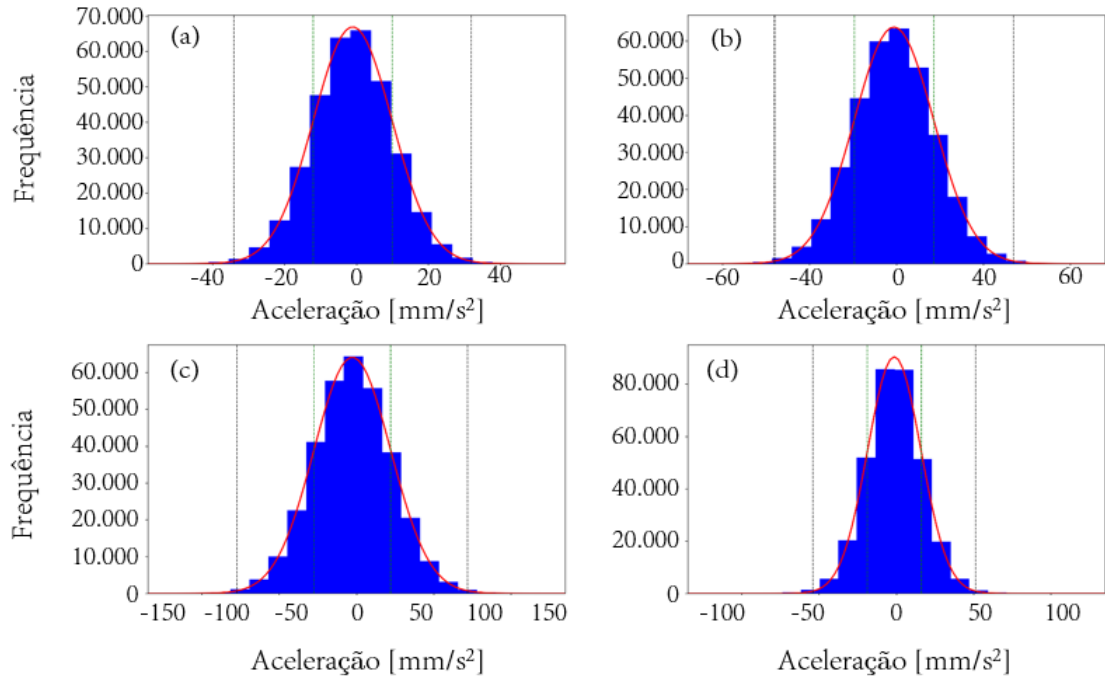
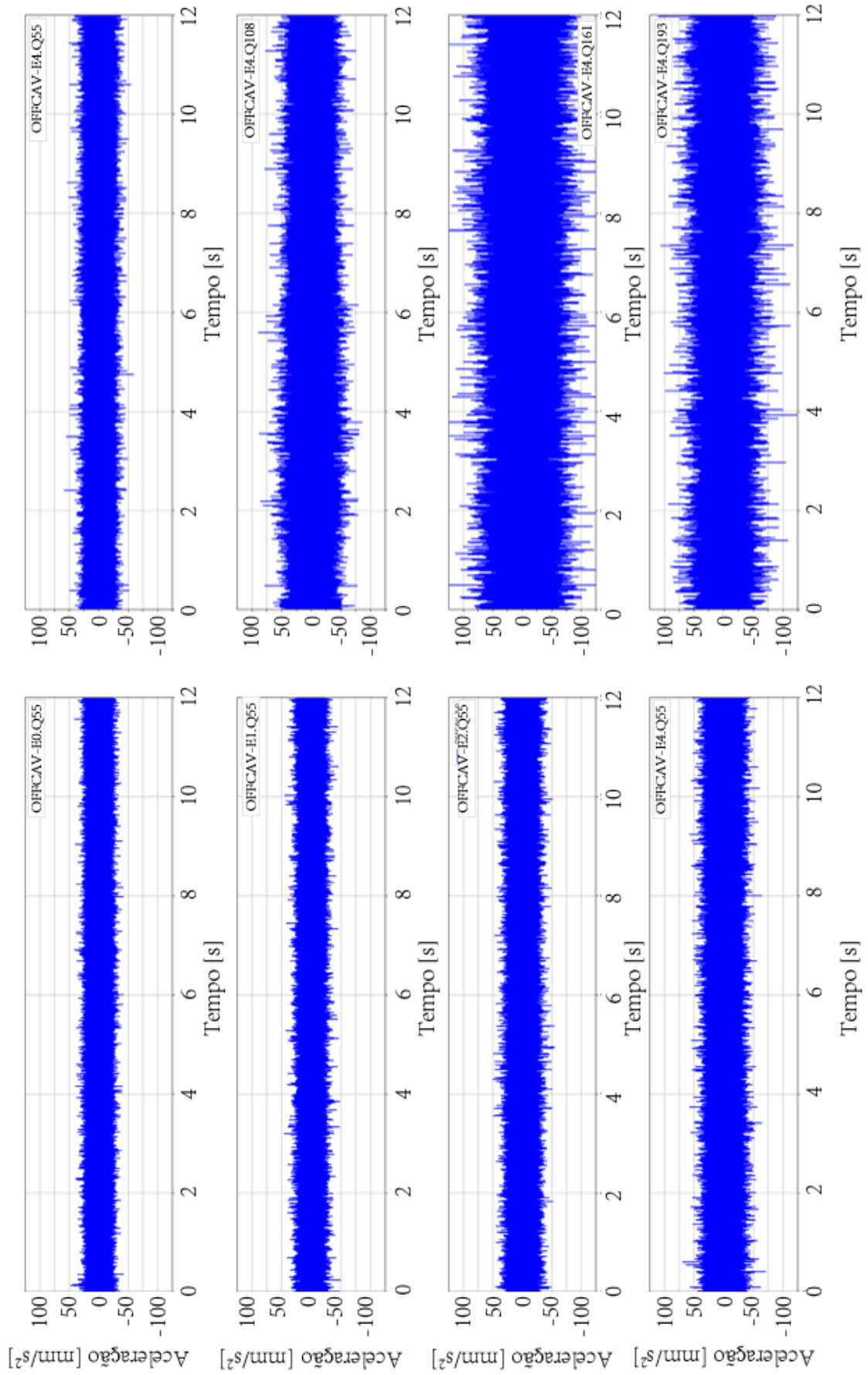


Figura 106 – Histogramas dos dados sem cavitação: (a) E1.Q55; (b) E1.Q108; (c) E1.Q161 e (d) E1.Q193.



Na Figura 107 são apresentados os sinais da base de dados para a condição de vazão  $Q = 161[m^3/s]$  e  $NPSH_r = 1$ . Observa-se que para as condições sem cavitação (*OFFCAV*), ou seja condições em  $NPSH_r \neq 1$ , os desvio padrão do sinal é maior conforme menor é o  $NPSH_r$ , enquanto que nas condições *ONCAV* esse mesmo efeito é ocasionada pela vazão.

Figura 107 – Sinal Bruto.



Na Tabela 86, são apresentados os intervalos de confiança 68,2% para todas as condições operativas e estruturais. Observa-se que uma estimativa do valor máximo encontrado para um grau de confiança de 68,26% é de 4,43 [ $m/s^2$ ], o que significa que 78% dos picos estão abaixo desse limiar. Por outro lado, ao utilizar um grau de confiança de 99,74%, obtém-se um valor de limiar de 20,51 [ $m/s^2$ ], correspondendo a 54% do número de picos da base de dados.

Tabela 86 – Valores dos intervalos de confiança para 68,2% 99,74%.

| Classe        | Ensaio    | Q55             | Q108             | Q161             | Q193             | Vlr <sub>68</sub> | Vlr <sub>99</sub> |
|---------------|-----------|-----------------|------------------|------------------|------------------|-------------------|-------------------|
| <i>OFFCAV</i> | <i>E0</i> | [8, 61; 26, 92] | [8, 90; 27, 79]  | [9, 92; 31, 95]  | [11, 72; 37, 33] | 8, 61             | 26, 92            |
| <i>OFFCAV</i> | <i>E1</i> | [7, 18; 23, 72] | [9, 91; 31, 89]  | [10, 85; 34, 73] | [11, 45; 36, 50] | 7, 18             | 23, 72            |
| <i>OFFCAV</i> | <i>E2</i> | [4, 43; 14, 37] | [5, 84; 18, 62]  | [15, 52; 51, 99] | [6, 48; 20, 51]  | 4, 43             | <b>20,51</b>      |
| <i>OFFCAV</i> | <i>E3</i> | [3, 08; 10, 32] | [4, 03; 13, 19]  | [11, 42; 36, 42] | [3, 78; 12, 40]  | 3, 08             | 26, 92            |
| <i>ONCAV</i>  | <i>E4</i> | [9, 79; 31, 47] | [15, 21; 47, 82] | [23, 03; 74, 50] | [19, 26; 59, 95] | <b>9,79</b>       | 31, 47            |

Desta forma, foi estabelecido um intervalo para o valor do limiar na base de dados de cavitação, variando de 9,79 [ $mm/s^2$ ] a 20,51 [ $mm/s^2$ ]. Como ocorre na base de dados de rolamento, é necessária uma análise de sensibilidade desse hiperparâmetro, conforme apresentado na Tabela 87. Esta tabela fornece um estudo comparativo dos parâmetros de ensaio, incluindo o número de classes e o valor do limiar, de acordo com a análise estatística nas Seções 6.5.1 e 6.5.2. Também são considerados diferentes tratamentos de dados dados brutos, DVS, DVS-SMOTE e DVS-SMOTE-ENN descritos na Seção 6.6.

Tabela 87 – Delineamento experimental incompleto para os dados de cavitação.

| Hiperparâmetros |               |        | Tratamento de dados |     |        |            |
|-----------------|---------------|--------|---------------------|-----|--------|------------|
| Regra Hit       | Nº de rótulos | Limiar | Brutos              | DVS | +SMOTE | +SMOTE-ENN |
| 1               | 10            | 10     |                     |     |        | X          |
| 1               | 10            | 15     |                     |     |        | X          |
| 1               | 10            | 20     | X                   | X   | X      | X          |
| 1               | 20            | 10     |                     |     |        | X          |
| 1               | 20            | 15     |                     |     |        | X          |
| 1               | 20            | 20     | X                   | X   | X      | X          |

Fonte: Autor.

Cada marcação na Tabela 87 representa um estudo desde a otimização dos hiperparâmetros da rede por meio do método de Bayes a obtenção das métricas de desempenho para a fase de teste. Com base nesse estudo comparativo, é tratado nos próximos capítulos dessa dissertação somente a condição 2 rótulos e limiar 20 $mm/s^2$ , pois apresentou o melhor resultado médio quando submetidos os dados a fase dos testes.

### 6.5.3 Análise dos parâmetros da Contagem de *Hits*

Nessa seção, avaliou a eficácia dos parâmetros de hits derivados de um valor limiar de  $20 [m/s^2]$ , utilizando o método de contagem, resultando em tabelas de frequências. Esta análise incluiu características como duração, tempo de subida, número de picos, número de picos máximos, amplitude e energia, com o objetivo de identificar padrões nos dados, tanto com quanto sem o uso das técnicas DVS e SMOTE-ENN. No caso da variável de duração, foi aplicada a regra de Sturges para definir intervalos uniformes.

A Tabela 88 mostra que a técnica revela um aumento na duração dos hits à medida que a intensidade da cavitação aumenta, ou seja, as maiores durações correspondem ao fenômeno de cavitação. Nessa análise, a condição sem cavitação apresentou as menores durações de Hits. Esse comportamento também é observado no número de picos e na energia, devido à sua relação intrínseca com a duração do evento. Além disso, o fenômeno é igualmente observado nas variáveis tempo de subida e número de picos máximos conforme apresentado na Tabela 88.

Tabela 88 – Duração e número de pico máximo por rótulo com dados brutos.

| Duração<br>( $\times 10^{-2}$ ) | Rótulo        |              | Pico máximo | Rótulo        |              |
|---------------------------------|---------------|--------------|-------------|---------------|--------------|
|                                 | <i>OFFCAV</i> | <i>ONCAV</i> |             | <i>OFFCAV</i> | <i>ONCAV</i> |
| (0, 0.0465]                     | 37.042        | 51.900       | (1, 2]      | 37.950        | 53.544       |
| (0.0465, 0.132]                 | 4.482         | 11.154       | (2, 3]      | 2.948         | 7.684        |
| (0.132, 0.213]                  | 166           | 912          | (3, 4]      | 588           | 1.808        |
| (0.213, 0.294]                  | 14            | 120          | (4, 5]      | 138           | 670          |
| (0.294, 0.361]                  | 2             | 20           | (5, 6]      | 44            | 98           |
| (0.361, 0.442]                  | 0             | 6            | (6, 7]      | 20            | 28           |
| (0.442, 0.524]                  | 0             | 6            | (7, 11]     | 10            | 28           |
| (0.524, 0.728]                  | 0             | 6            | (11, 17]    | 8             | 8            |
| Total                           | 41.706        | 64.124       |             | 41.706        | 64.116       |

De maneira semelhante, realizou-se uma análise de frequência da variável amplitude para investigar a relação entre o valor máximo. A Tabela 89 apresenta os valores de contagem dessa variável, em que foi possível notar que a condição *OFFCAV* e *ONCAV* tiveram a maior concentração dos dados na região de menor amplitude. Diferentemente o que foi observado nas outras variáveis.

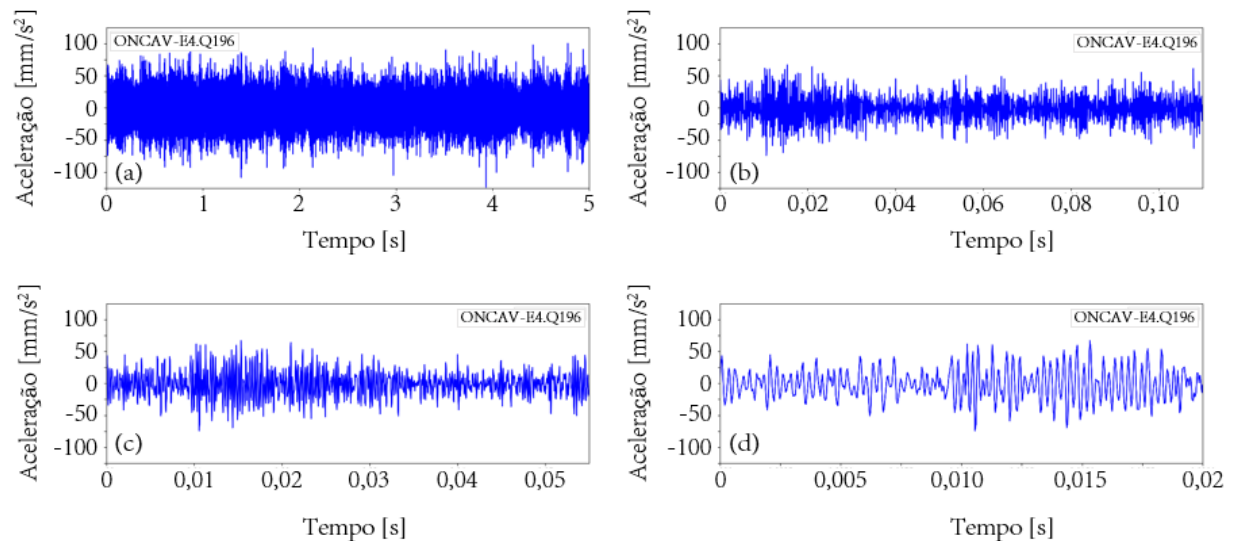
Para a variável amplitude, foram aplicadas as técnicas *DVS* e *SMOTE* para transformar os dados para observar tendências similares às das variáveis de duração, tempo de subida, número de picos, número de picos máximos e energia. Os resultados estão resumidos na Tabela 89, que mostra a condição *OFFCAV* com uma relevância no número de eventos para menores amplitude enquanto que as condições *ONCAV* passou a ter um maior número de eventos com amplitudes maiores, conforme percebe-se na Tabela 89.

Tabela 89 – Amplitude por rótulo com dados brutos e com DVS.

| Amplitude       | Rótulo        |              | Amplitude DVS    | Rótulo        |              |
|-----------------|---------------|--------------|------------------|---------------|--------------|
|                 | <i>OFFCAV</i> | <i>ONCAV</i> |                  | <i>OFFCAV</i> | <i>ONCAV</i> |
| (0,943 ; 1,703] | 6822          | 38790        | (-5.583, -0.289] | 31970         | 9180         |
| (1,703 ; 2,304] | 22528         | 18582        | (-0.289, 3.577]  | 31806         | 32016        |
| (2,304 ; 3,3]   | 11080         | 6280         | (3.577, 8,960]   | 318           | 466          |
| (3,300 ; 4,523] | 1114          | 450          | (8,960, 13.159]  | 24            | 30           |
| (4,523 ; 5,171] | 70            | 6            | (13.159, 18,703] | 4             | 10           |
| (5,171 ; 7,589] | 22            | 2            | (4.783, 7.095]   | 0             | 4            |
| Total           | 41.706        | 64.124       |                  | 41.706        | 64.116       |

### 6.5.4 Análise de Frequência do Sinal

Para expandir a base de dados (espectros de frequência) destinada ao modelo RNC, foi realizada uma subamostragem dos dados utilizando janelas retangulares. Para definir o tamanho da janela, utilizou-se o menor valor de frequência de 180 Hz (período de 0,0055 s), resultando em janelas de 0,02s, 0,055s e 0,11s. A Figura 108 mostra os sinais *ONCAV* – *E4.Q196* com essas janelas.

Figura 108 – Sinal janelado *ONCAV* – *E4.Q196*: (a) 12s; (b) 0,11s; (c) 0,055s e (d) 0,02s.

Após definir o tamanho da janela, o sinal é transformado do domínio do tempo para o domínio da frequência usando a TRF para todas as classes da base de dados. Isso permite obter valores de frequência e quantidade de amostras de acordo com a classe e o tamanho da janela. A Tabela 90 mostra a variação percentual das frequências de referência com o sinal completo e as janelas retangulares. Observou-se que, à medida que o tamanho da janela diminui, o desvio relativo das frequências aumenta e certas frequências podem desaparecer dos espectros, enquanto que a quantidade de espectros aumenta conforme o tamanho da janela diminui, como apresentado na Tabela 91.

Tabela 90 – Principais frequências dos sinais de cavitação acordo com o janelamento.

| Falha         | 12s               |      |      |      |      |   | 0,11s |        | 0,055s |        | 0,02s |        |
|---------------|-------------------|------|------|------|------|---|-------|--------|--------|--------|-------|--------|
|               | Frequências [kHz] |      |      |      |      |   | Qdt.  | Desvio | Qdt.   | Desvio | Qdt.  | Desvio |
| OFF – E0.Q55  | 0,10              | 1,30 | 3,43 | 4,43 | 5,04 | 4 |       | 1,8%   | 476    | 7,4%   | 952   | 9,8%   |
| OFF – E0.Q108 | 0,10              | 1,31 | 3,41 | 4,49 | 5,03 | 4 |       | 3,0%   | 476    | 4,5%   | 952   | 7,2%   |
| OFF – E0.Q161 | 0,10              | 1,32 | 3,43 | 4,48 | 4,95 | 4 |       | 0,4%   | 476    | 3,8%   | 952   | 5,5%   |
| OFF – E0.Q193 | 0,10              | 1,31 | 3,46 | 4,43 | 5,04 | 4 |       | 2,9%   | 476    | 7,6%   | 952   | (*)    |
| ON – E4.Q108  | 0,10              | 1,31 | 3,43 | 6,02 | 7,17 | 4 |       | 0,7%   | 476    | 7%     | 952   | 15%    |

Nota: As 5 frequências de maiores amplitudes.

Tabela 91 – Quantidade de amostras de cavitação de acordo com a janela.

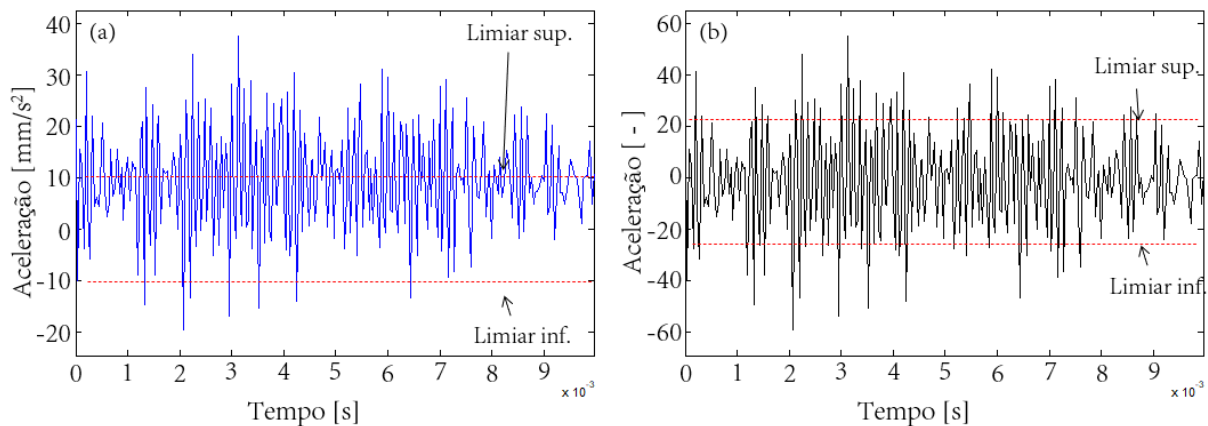
| Falha | 5s | 0,11s | 0,055s | 0,02s  |
|-------|----|-------|--------|--------|
| Total | 20 | 2.380 | 4.760  | 13.100 |

Com essa análise, foi observado que existe um tamanho de janela que não afeta o espectro de frequência, permitindo assim a subamostragem. Este fenômeno é evidenciado pelas janelas de tamanho 0,11 segundos, porém optou-se em utilizar o tamanho 0,02s para verificar a qualidade dos resultados e robustez da solução.

## 6.6 Modelo RNMP

O modelo RNMP foi configurado para a bomba centrífuga 1 e o acelerômetro horizontal (sensor 1) e uma taxa sistema de aquisição de dados de  $25\text{kHz}$ . Durante a preparação dos dados, houve uma expansão da base de dados para  $N_{aug} = 2^1 \times N$ , onde  $N$  representa o número de amostras devido ao ensaio. A série foi então normalizada para corrigir tendências e contagens equivocadas. No presente caso, os picos da série foram ajustados próximos ao limiar superior, enquanto os vales foram ajustados próximos ao limiar inferior, como ilustrado na Figura 109.

Figura 109 – Classe OFFCAV – E0.Q55: (a) dados brutos; (b) dados normalizados.



Essas etapas foram repetidas para todas classes do conjunto de dados, resultando nos valores médios e nos desvios padrão necessários para definir o valor limiar, conforme apresentado na análise exploratória dos dados (Tabela 86). O número de amostras antes e depois do aumento de dados (*data augmentation*) é detalhada na Tabela 92.

Tabela 92 – Aplicação do *data augmentation* aos dados de cavitação (Algoritmo 8).

| Classe        | Ensaio    | N       |         |         |         | N <sub>aug</sub> |         |         |         |
|---------------|-----------|---------|---------|---------|---------|------------------|---------|---------|---------|
|               |           | Q55     | Q108    | Q161    | Q193    | Q55              | Q108    | Q161    | Q193    |
| <i>OFFCAV</i> | <i>E0</i> | 327.680 | 327.680 | 327.680 | 327.680 | 655.358          | 655.358 | 655.358 | 655.358 |
| <i>OFFCAV</i> | <i>E1</i> | 327.680 | 327.680 | 327.680 | 327.680 | 655.358          | 655.358 | 655.358 | 655.358 |
| <i>OFFCAV</i> | <i>E2</i> | 327.680 | 327.680 | 327.680 | 327.608 | 655.358          | 655.358 | 655.358 | 655.214 |
| <i>OFFCAV</i> | <i>E3</i> | 327.680 | 327.680 | 327.680 | 327.680 | 655.358          | 655.358 | 655.358 | 655.358 |
| <i>ONCAV</i>  | <i>E4</i> | 327680  | 327680  | 327680  | 327613  | 655358           | 655358  | 655358  | 655224  |

A etapa de extração de características começa com a determinação do valor de limiar do sinal, obtido por meio da análise exploratória dos dados. Esse valor foi estabelecido com base no valor médio e desvio padrão do sinal, resultando em  $20[mm/s^2]$ . Ao final desta etapa, os dados são estruturados para o modelo RNMP, totalizando **105.825 eventos de hits**. A Tabela 93 apresenta cinco eventos de cada classe, enquanto na Tabela 94 é tratada o número de ocorrências por classe. Essa técnica de extração de características resulta em desbalanceamento na base de dados, afetando negativamente a classe com menor número de amostras, como a condição sem cavitação.

Tabela 93 – Dados brutos organizados da base de cavitação para o modelo RNMP.

| N     | Duração | Tempo subida | Amplitude | Contagens | Energia   | Rótulo        |
|-------|---------|--------------|-----------|-----------|-----------|---------------|
| 156   | 0,00018 | 2,00E-05     | 2,0178292 | 1/1       | 2,10E-05  | <i>OFFCAV</i> |
| 157   | 0,00046 | 0,00026      | 2,3124566 | 2/2       | -3,14E-05 | <i>OFFCAV</i> |
| 508   | 0,00076 | 0,00052      | 2,9258387 | 3/3       | 6,74E-05  | <i>OFFCAV</i> |
| 33846 | 0,00068 | 4,00E-05     | 1,9228535 | 1/3       | -5,07E-06 | <i>OFFCAV</i> |
| 77073 | 0,00022 | 2,00E-05     | 1,0664552 | 1/1       | 1,25E-05  | <i>ONCAV</i>  |
| 77101 | 0,0017  | 0,00118      | 1,5032843 | 5/6       | 2,55E-05  | <i>ONCAV</i>  |
| 77191 | 0,0029  | 0,00234      | 2,2303264 | 9/10      | 7,81E-05  | <i>ONCAV</i>  |
| 77192 | 0,00458 | 0,00358      | 2,947841  | 13/16     | 9,72E-05  | <i>ONCAV</i>  |

Nota: Contagens - Contagem à pico / Contagem.

Tabela 94 – Quantidade de amostras de cavitação para o modelo RNMP.

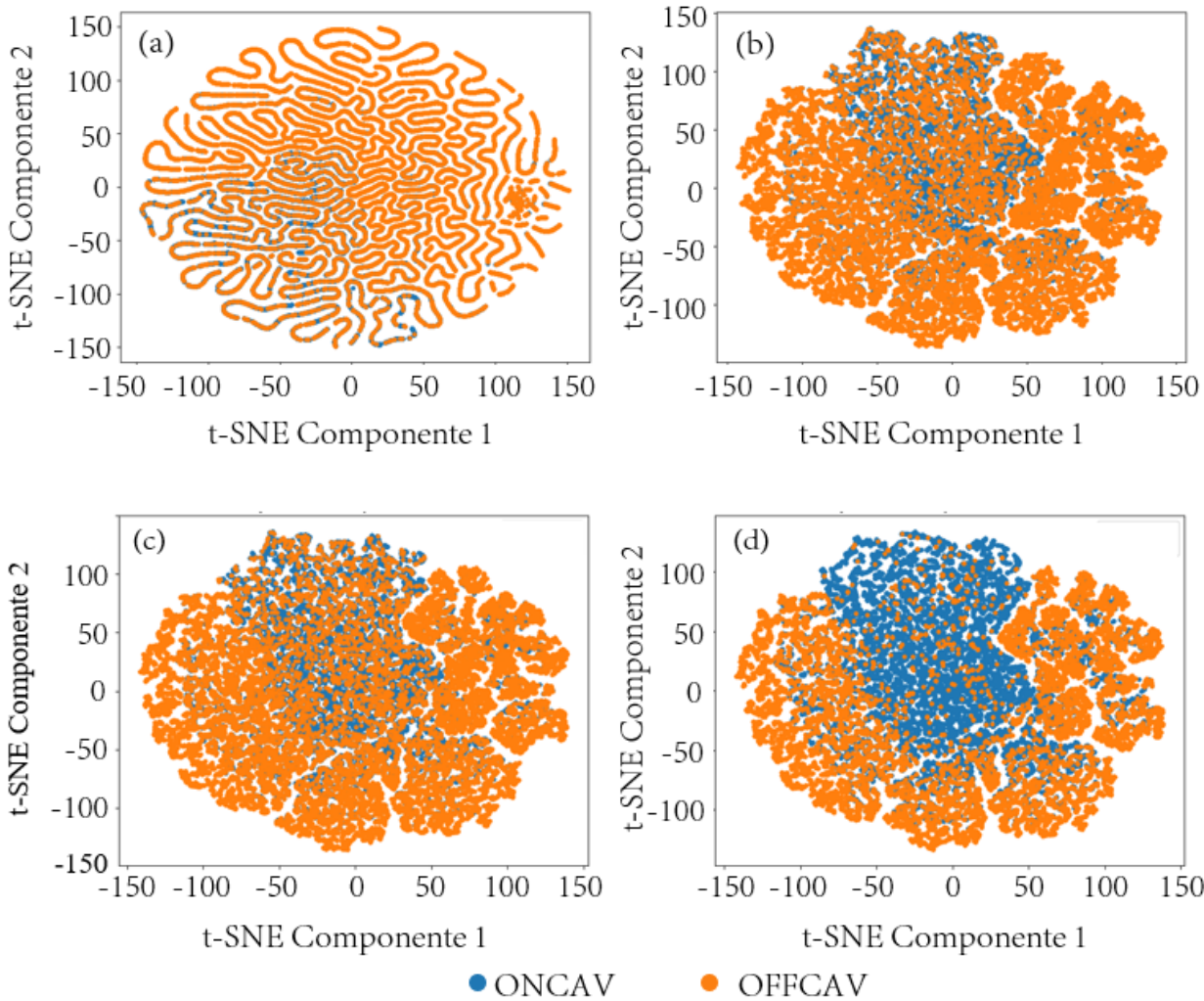
|       | <i>OFFCAV</i> | <i>ONCAV</i> |
|-------|---------------|--------------|
| Qntd. | 41.706        | 64.124       |
|       | 39,41%        | 60,59%       |

Ao aplicar a DVS à matriz dos dados, obtêm-se 7 componentes principais que capturam a forma da onda, permitindo uma classificação eficiente pelo modelo de aprendizado de máquina. O balanceamento da base é realizado com o modelo SMOTE-ENN, que gera dados sintéticos para classes minoritárias e remove amostras com base na vizinhança. A Tabela ?? detalha as amostras originais, sintéticas e excluídas. A Figura 110 ilustra a modelagem  $t-SNE$  do conjunto de dados, evidenciando a melhoria na separação das classes após a transformação DVS, o balanceamento SMOTE e a exclusão pelo ENN.

Tabela 95 – Amostras de cavitação utilizadas na modelagem RNMP.

| Ensaio  | Sintéticos | Excluídos | Técnicas |         |         |         |
|---------|------------|-----------|----------|---------|---------|---------|
|         |            |           | Brutos   | DVS     | SMOTE   | ENN     |
| 105.825 | 18.049     | -15.840   | 105.825  | 105.825 | 123.874 | 108.034 |

Figura 110 – Vizualização dos dados de cavitação separados por classe com a técnica t-SNE: (a) brutos; (b) com DVS; (c) com DVS e SMOTE e (d) DVS e SMOTE-ENN.



Fonte: Autor.

Esta abordagem visa melhorar a qualidade dos dados de entrada para a RNMP. Dependendo dos resultados obtidos, pode indicar a necessidade de melhorias na fase de coleta, como a redução de ruídos e incertezas do ensaio, além de aprimorar a precisão da contagem de hits através da introdução de novos parâmetros.

Assim como ocorre com os dados de rolamento, esta abordagem não tem como objetivo fornecer uma explicação física dos dados, mas sim uma interpretação estritamente matemática. Isso fica evidente ao examinar os dados organizados na Tabela 96, especialmente ao observar características como tempo de subida, contagem e contagem até o pico, que podem apresentar valores negativos ou não inteiros.

Tabela 96 – Dados organizados de cavitação para o modelo RNMP.

| N     | Duração      | Tempo subida | Amplitude    | Contagens       | Energia      | Rótulo |
|-------|--------------|--------------|--------------|-----------------|--------------|--------|
| 156   | -0.051303683 | 0.770631871  | -0.057826491 | -0.4010/0.0095  | 0.979416944  | 0      |
| 508   | -0.252185479 | 0.509431997  | 0.057500468  | 0.3055/0.0080   | 0.449290188  | 0      |
| 33845 | -0.049330452 | 0.773197533  | -0.058959345 | -0.1063/0.0099  | 0.233306977  | 0      |
| 33846 | -0.33092564  | 0.407048911  | 0.102705411  | 0.8416/0.0063   | -0.458140203 | 0      |
| 39474 | 0.775258513  | -0.855834048 | 1.088970083  | -0.7326/0.0022  | 0.095522708  | 1      |
| 72090 | 0.297236947  | -1.477389081 | 1.363404093  | -1.1076/0.0078  | 1.192840064  | 1      |
| 63200 | 0.201785013  | 1.099714289  | -0.203125747 | 0.5604/0.130    | -0.3767936   | 1      |
| 68220 | 1.643916111  | -2.427564614 | 2.211599412  | -1.2342/-0.0147 | -0.343079323 | 1      |

Nota: Contagens - Contagem à pico / Contagem.

Na Tabela 97 contempla as novas quantidade de dados por classe (balanceamento com dados sintéticos). Observa-se que não ocorreu um balanceamento completo, inerente da técnica ENN, sendo ajustados apenas 20% total da matriz *design*. Esse desbalanceamento foi corrigido no modelo neural por meio de ponderações em cada classe durante as fases de treinamento.

Tabela 97 – Quantidade de amostras de cavitação por classe para o modelo RNMP

| Método     | <i>OFFCAV</i> | <i>ONCAV</i> |
|------------|---------------|--------------|
| +SMOTE     | 59,750        | 64,124       |
|            | 48,23%        | 51,76%       |
| +SMOTE-ENN | 59,750        | 48,284       |
|            | 55,31%        | 44,69%       |

Nota: As porcentagens são aproximadas.

Com base nos dados organizados e nos hiperparâmetros listados na Tabela 2.3 da Seção 2, foi realizada a otimização dos parâmetros do modelo RNMP utilizando a técnica de otimização Bayesiana. Esse processo visou identificar a combinação ideal para maximizar o desempenho do modelo, considerando hiperparâmetros ajustáveis, como: número de camadas ocultas, número de neurônios por camada, funções de ativação, taxa de aprendizado, número de épocas e a penalização de regularização  $L_2$ .

Por outro lado, alguns hiperparâmetros foram mantidos fixos durante a otimização, incluindo: número de neurônios na camada de entrada, aplicação de *batch normalization* entre camadas, tamanho dos *batches*, tipo de otimizador e a proporção de divisão entre os conjuntos de treinamento e teste. Essa abordagem balanceou a flexibilidade para explorar configurações complexas e a estabilidade proporcionada pelos parâmetros fixos.

Esta otimização foi aplicada às quatro bases de dados apresentadas na Figura 110 para avaliar o desempenho dos processos de transformação dos dados. Um resumo dos resultados pode ser encontrado na Tabela 98.

Tabela 98 – Resultados dos conjuntos de dados de cavitação.

| Base de dados | Neurônios por camada           | Épocas | $\alpha$ | $lr$     | Acurácia |        |
|---------------|--------------------------------|--------|----------|----------|----------|--------|
|               |                                |        |          |          | treino   | teste  |
| Dados brutos  | 301/216/498/439/378/145/155/51 | 188    | 5,13E-5  | 5,14E-5  | 76,73%   | 75,38% |
| DVS-TRUNC     | 193/245/32/32/512/67/512/151   | 300    | 1,00E-4  | 1,00E-5  | 78,98%   | 77,48% |
| + SMOTE       | 301/216/498/439/378/145/155/51 | 188    | 5,13E-5  | 5,14E-5  | 79,82%   | 77,90% |
| + SMOTE-ENN   | 112/512/488/260/512/301/32/314 | 300    | 2,00E-4  | 2,003E-5 | 87,83%   | 53,52% |

Assim como na base de dados dos rolamentos, o conjunto de dados *DVS-SMOTE* apresentou o melhor desempenho em termos de acurácia, atingindo 77,90% na fase de testes. Em comparação, os conjuntos *DVS + SMOTE - ENN* obtiveram acurácias de aproximadamente 87,83% e 53,52%, respectivamente, o que levanta a necessidade de uma investigação mais aprofundada sobre o impacto da técnica ENN nesse caso.

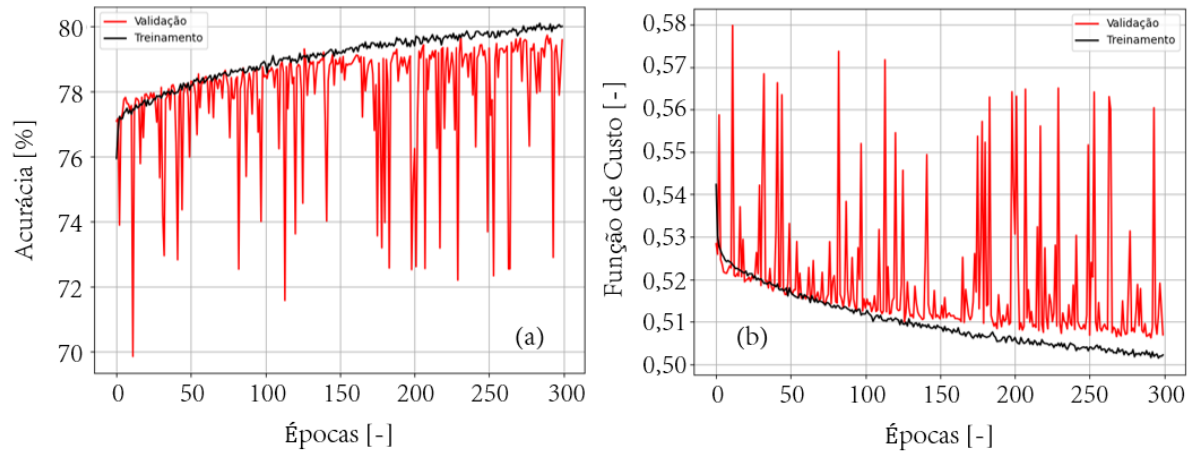
Na Tabela 99, são detalhados os intervalos de busca dos hiperparâmetros e os resultados do processo de otimização de Bayes aplicado aos dados transformados pela técnica *DVS - SMOTE*.

Tabela 99 – Exploração do espaço e seleção dos hiperparâmetros.

| Hiperparâmetro                 | Intervalo de busca         | Valores encontrados            |
|--------------------------------|----------------------------|--------------------------------|
| Número de camadas              | 2 à 8                      | 8                              |
| Número de neurônios por camada | 32 à 512                   | 301/216/498/439/378/145/155/51 |
| Função de ativação             | <i>ReLU</i> ou <i>Tanh</i> | <i>ReLU</i>                    |
| Número de épocas               | 100 à 300                  | 188                            |
| Taxa de aprendizado            | 1E-5 à 1E-3                | 5,13E-5                        |
| Decaimento de pesos            | 1E-5 à 1E-3                | 5,14E-5                        |

Após identificar os hiperparâmetros é essencial aprofundar a análise do modelo RNMP, realizando a execução das fases de treinamento, validação e teste. A Figura 111 ilustra os gráficos de convergência da função de custo e da acurácia tanto para as fases de treino quanto para validação.

Figura 111 – Desempenho geral da RNMP das fase de treino: (a) acurácia ; (b) custo.



Fonte: Autor

Durante o treinamento, os valores de convergência para a acurácia e o custo foram de 79,82% e 0,51, enquanto que para a validação, os valores alcançaram 79,54% e 0,50.

Para avaliar a acurácia e precisão por classe, foram criadas matrizes de confusão para as fases de treinamento, validação e teste da rede, abrangendo todos os rótulos. Na Tabela 100, é exibida as matrizes de confusão para as fase de treino e teste, e as acurácias individuais para cada rótulo são fornecidas na Tabela 101 para as três etapas fases. Nestas matrizes os rótulos de origem do ensaio possui o índice  $r$  enquanto que os valores preditos pelo modelo não possuem índices. Nos resultados dos testes ocorreram erros dos tipos I e II, por exemplo, na classe *OFFCAV*, o modelo previu 7.122 ocorrências, das quais 1.715 são da classe *ONCAV* enquanto que 2.924 amostras não foram contabilizadas, pois eram *OFFCAV* e foram classificadas em *ONCAV*.

Tabela 100 – Matriz de confusão do modelo RNMP com dados da fase de teste.

|                           | Treino        |              | Teste         |              |
|---------------------------|---------------|--------------|---------------|--------------|
|                           | <i>OFFCAV</i> | <i>ONCAV</i> | <i>OFFCAV</i> | <i>ONCAV</i> |
| <i>OFFCAV<sub>r</sub></i> | 34.150        | 6.784        | 5.407         | 2.924        |
| <i>ONCAV<sub>r</sub></i>  | 9.714         | 31.272       | 1.715         | 10.946       |

Tabela 101 – Resultados de precisão, sensibilidade e *F1-score* do modelo RNMP

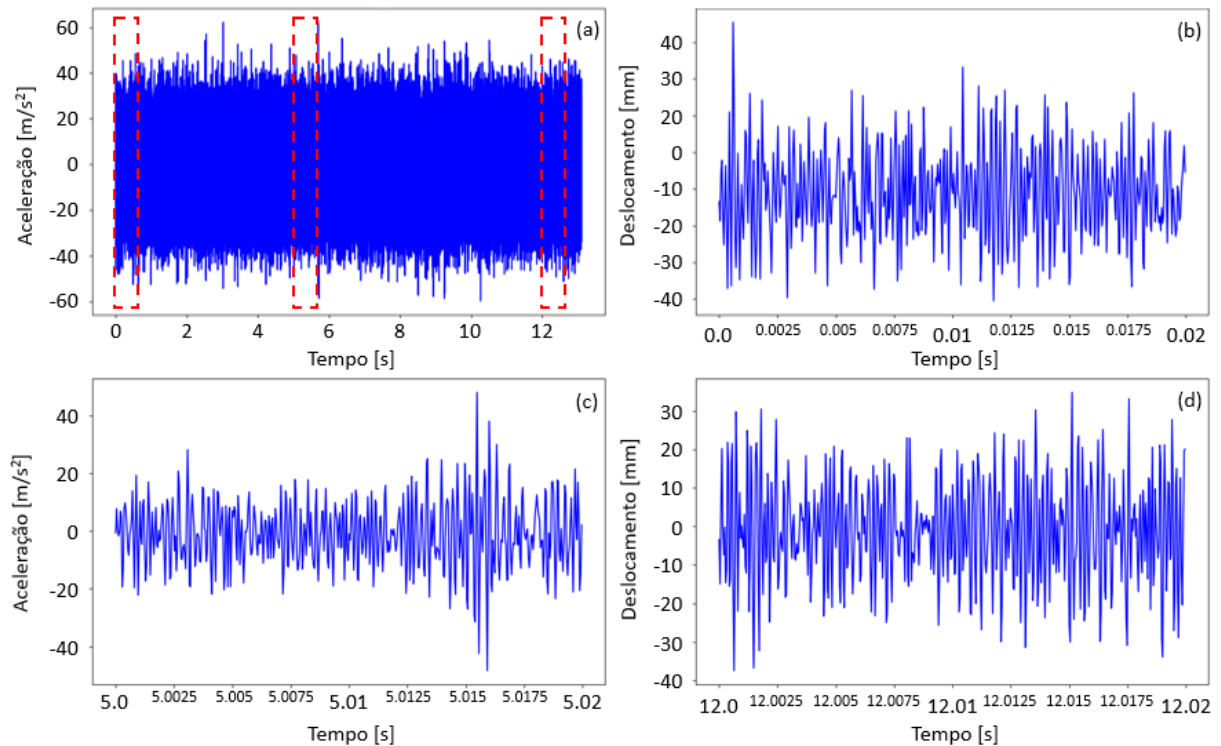
| Fase             | Precisão      |              |              | Sensibilidade |              |              | <i>F1 – Score</i> |              |              |
|------------------|---------------|--------------|--------------|---------------|--------------|--------------|-------------------|--------------|--------------|
|                  | <i>OFFCAV</i> | <i>ONCAV</i> | Global       | <i>OFFCAV</i> | <i>ONCAV</i> | Global       | <i>OFFCAV</i>     | <i>ONCAV</i> | Global       |
| <i>Train</i> [%] | 77,85         | 82,17        | <b>80,01</b> | 83,43         | 76,30        | <b>79,86</b> | 80,54             | 79,13        | <b>79,84</b> |
| <i>Valid</i> [%] | 77,08         | 82,25        | <b>79,67</b> | 83,96         | 74,86        | <b>79,41</b> | 80,37             | 78,38        | <b>79,38</b> |
| <i>Test</i> [%]  | 78,92         | 78,92        | <b>75,68</b> | 64,90         | 86,45        | <b>77,42</b> | 69,98             | 82,51        | <b>76,25</b> |

Com relação aos resultados gerais do modelo RNMP, obteve-se na fase de teste 77,90% de acurácia, 77,42% de sensibilidade, 75,68% de precisão e 76,25% no *F1-score*.

## 6.7 Modelo RNC

O modelo RNC foi ajustado para o rolamento bomba centrífuga 1 com o sensor 1, com uma taxa de aquisição de dados de  $25kHz$ , para realizar o treinamento do modelo destinado ao diagnóstico em cavitação. Para expandir a base de dados, são realizadas subamostragem e aplicação da TFCT utilizando uma janela retangular com duração de  $0,02s$ . Esse procedimento é ilustrado na Figura 112 para as condições *OFFCAV* – Q55, em que são exibidas 3 das 2.620 janelas utilizadas.

Figura 112 – Condição *OFFCAV* – Q55: (a) sinal completo; (b) janela do instante 0 a 0,2s; (c) janela do instante 5,0 a 5,02s; (d) janela 12 a 12,02s.



A escolha do valor de  $0,02s$  foi determinada com base na conservação das frequências naturais do sinal, mantendo um volume de dados adequado para o modelo RNC. Os valores de frequência estão apresentados na Tabela 102, onde se percebe que algumas frequências são atenuadas enquanto outras são destacadas em cada espectro, conforme discutido na Seção 6.5.4.

Dessa forma, 20 arquivos organizados no ensaio com durações de aproximadamente  $13s$ , que foram apresentados nos Quadros 6 e 7, foram transformados em 13.100 novas séries de  $0,02s$  cada.

Dando sequência à etapa de preparação dos dados, é realizada a TFCT por meio do Algoritmo 76, conforme apresentado na Seção 5.2.2. Esse procedimento inclui os valores do tamanho do *overlap*, tipo e tamanho da janela, e frequência de amostragem. Esses parâmetros, em conjunto com os dados dos ensaios, geraram um total de 13.100 espectro-

Tabela 102 – Valores frequência do sinal em cada classe.

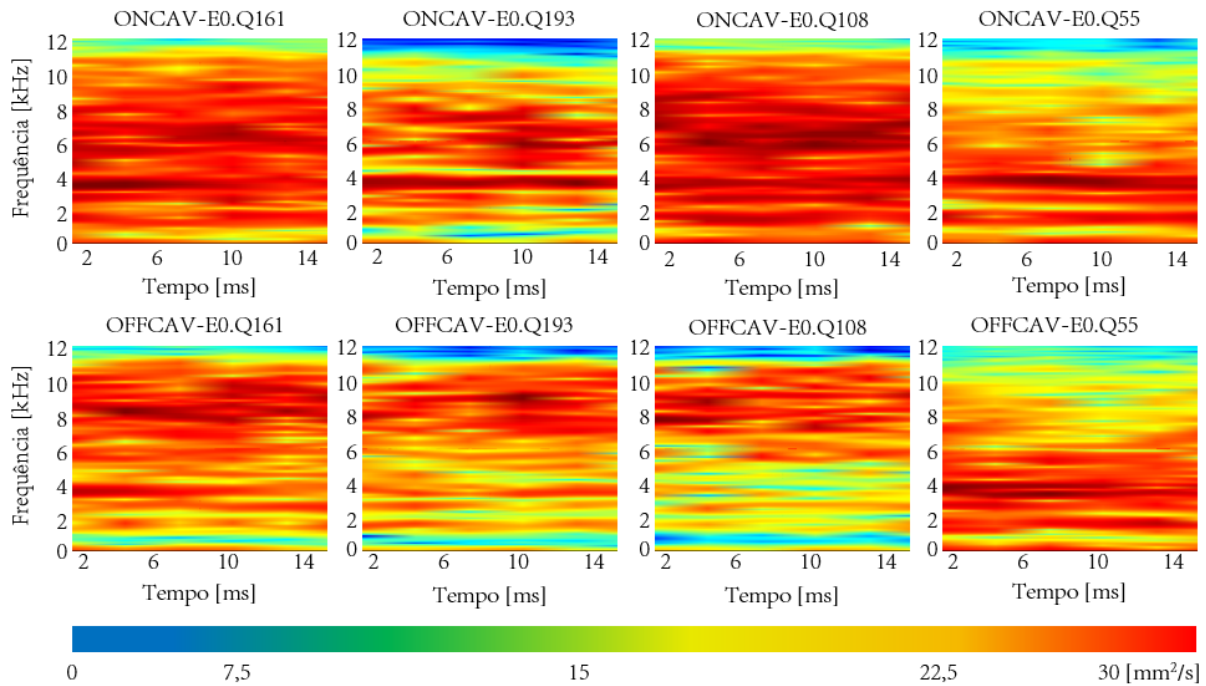
| Tamanho Janela |    |     | Frequências [Hz] |       |       |       |       |       |       |       |
|----------------|----|-----|------------------|-------|-------|-------|-------|-------|-------|-------|
| Janela 12s     | 49 | 100 | 1.307            | 1.563 | 2.706 | 3.427 | 3.693 | 5.779 | 6.016 | 7.175 |
| Janela 5s      | 49 | 100 | 1.307            | 1.560 | 2.706 | 3.433 | 3.721 | 5.819 | 6.014 | 7.173 |
| Janela 0,11s   | -  | -   | 1.300            | 1.573 | 2.727 | 3.391 | 3.663 | 5.409 | 6.027 | 7.136 |
| Janela 0,055s  | -  | -   | 1.327            | 1.600 | 2.745 | 3.436 | 3.709 | 5.418 | 6.027 | 7.182 |
| Janela 0,02s   | -  | -   | -                | 1.650 | -     | 3.450 | 5.800 | -     | 6.000 | 7.100 |

gramas, cujas proporções estão detalhadas na Tabela 103. A Figura 113 ilustra um dos espectros de frequência de cada classe.

Tabela 103 – Número de amostras: Aplicação do *data augmentation* (Algoritmo 38)

| Classe        | Ensaio    | Vazão [ $m^3/s$ ] |      |      |      |       |
|---------------|-----------|-------------------|------|------|------|-------|
|               |           | Q55               | Q108 | Q161 | Q193 | Total |
| <i>OFFCAV</i> | <i>E0</i> | 655               | 655  | 655  | 655  | 2.620 |
| <i>OFFCAV</i> | <i>E1</i> | 655               | 655  | 655  | 655  | 2.620 |
| <i>OFFCAV</i> | <i>E2</i> | 655               | 655  | 655  | 655  | 2.620 |
| <i>OFFCAV</i> | <i>E3</i> | 655               | 655  | 655  | 655  | 2.620 |
| <i>ONCAV</i>  | <i>E4</i> | 655               | 655  | 655  | 655  | 2.620 |

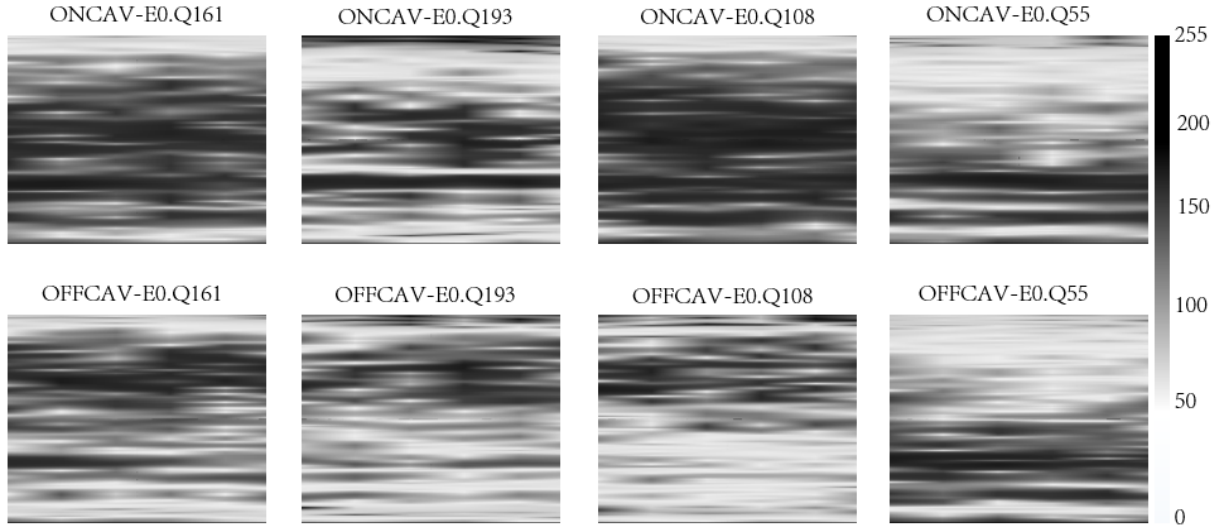
Figura 113 – Espectrogramas das condições de cavitação (TFCT).



O próximo passo envolve a transformação dos espectros de frequência em imagens. Esse processo inclui a conversão dos valores de frequência em um vetor de escala de 0 a 255, o redimensionamento das imagens, a eliminação das bordas, o registro das imagens com os rótulos correspondentes e, por fim, a criação da base de dados em um formato

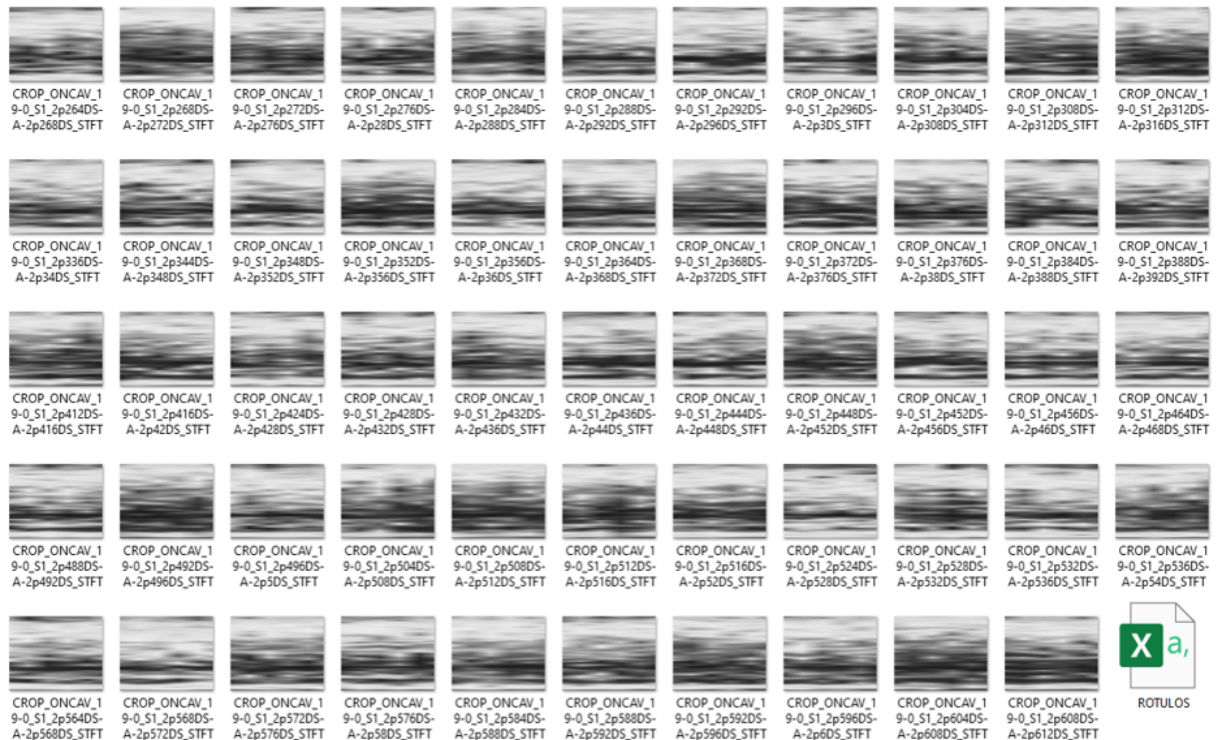
adequado para o modelo RNC, armazenada em um diretório. A Figura 114 ilustra os padrões presentes nas imagens, representando cada um dos rótulos.

Figura 114 – Imagens das condições de cavitação (Gray).



A base de dados da RNC é composta por imagens em tons de cinza com dimensões de  $32 \times 32$  pixels e um arquivo “.csv”, todos armazenados em um diretório específico conforme ilustrado na Figura 115 proporcionando uma visão da estrutura e das informações associadas a cada imagem.

Figura 115 – Base de dados de cavitação para o modelo RNC.



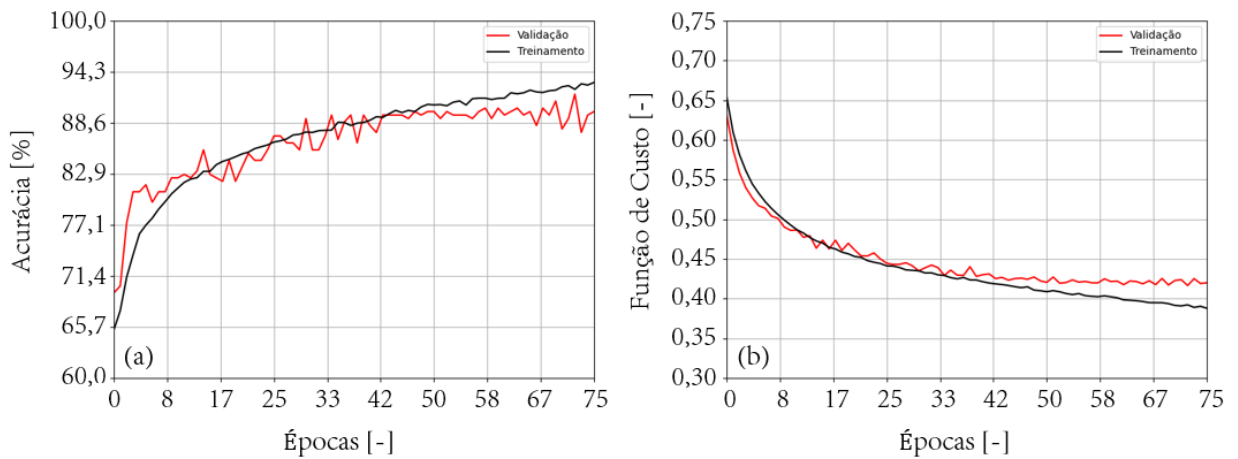
Esse arquivo “.csv” contém o nome de cada imagem e informações detalhadas, como tipo de falha, velocidade de rotação, intervalo de tempo do sinal e técnica utilizada, organizadas no formato “crop\_cavitação\_ensaio\_intervalo de tempo do sinal\_técnica utilizada”, conforme apresentada na Tabela 104.

Tabela 104 – Dados estruturados.

| N | Nome do arquivo                              | Rótulo        |
|---|----------------------------------------------|---------------|
| 1 | CROP_OFFCAV_E0.Q55_S1_1p632DS-A-1p636DS_STFT | <i>OFFCAV</i> |
| 2 | CROP_OFFCAV_E1.Q108_S1_1p656DS-A-1p66DS_STFT | <i>OFFCAV</i> |
| 3 | CROP_ONCAV_E4.Q161_S1_1p984DS-A-1p988DS_STFT | <i>ONCAV</i>  |
| 7 | CROP_ONCAV_E4.Q196_S1_2p616DS-A-2p62DS_STFT  | <i>ONCAV</i>  |

Apartir da base de dados e os hiperparâmetros definidos na Tabela 28 da Seção 2.3, executou-se as fases de treinamento, validação e teste do modelo RNC. A Figura 116 ilustra os gráficos de convergência da função de custo e da acurácia média referente a cada época tanto para as fases de treino quanto para validação. Os valores de convergência para a acurácia e o custo foram de 0,35 e 96,79% respectivamente, durante o treinamento. Enquanto que para a validação, os valores atingiram 0,425 e 90,77%.

Figura 116 – Desempenho da RNC na fase de treino: (a) acurácia; (b) função de custo



Na Tabela 105, é exibida a matriz de confusão para as fases de treino e teste, enquanto que as acurácias individuais de cada rótulo são fornecidas na Tabela 105 para as três fases. Nesta matriz os rótulos de origem do ensaio possuem o índice  $r$  enquanto que os valores preditos pelo modelo não possuem índices. Nesses resultados, também ocorreram erros dos tipos  $I$  e  $II$ ; por exemplo, na classe *OFFCAV*, o modelo previu 1.128 ocorrências, das quais 85 pertencem a classe *ONCAV* e 99 amostras foram classificadas erroneamente como *ONCAV*.

Com base na constatação dos erros dos tipos  $I$  e  $II$  presentes na matriz de confusão, realizou-se os cálculos da precisão, sensibilidade e  $F1 - score$ , chegando a Tabela 106.

Tabela 105 – Matriz de confusão do modelo RNMP com dados da fase de teste

|                           | Treino        |              | Teste         |              |
|---------------------------|---------------|--------------|---------------|--------------|
|                           | <i>OFFCAV</i> | <i>ONCAV</i> | <i>OFFCAV</i> | <i>ONCAV</i> |
| <i>OFFCAV<sub>r</sub></i> | 6.554         | 108          | 1.043         | 99           |
| <i>ONCAV<sub>r</sub></i>  | 219           | 3.318        | 85            | 473          |

Fonte: Autor.

Tabela 106 – Resultados de precisão, sensibilidade e *F1-score* do modelo RNMP

| Fase             | Precisão      |              |              | Sensibilidade |              |              | <i>Score-F1</i> |              |              |
|------------------|---------------|--------------|--------------|---------------|--------------|--------------|-----------------|--------------|--------------|
|                  | <i>OFFCAV</i> | <i>ONCAV</i> | Global       | <i>OFFCAV</i> | <i>ONCAV</i> | Global       | <i>OFFCAV</i>   | <i>ONCAV</i> | Global       |
| <i>Train</i> [%] | 96,77         | 96,85        | <b>96,80</b> | 98,34         | 93,81        | <b>96,09</b> | 97,57           | 95,30        | <b>96,43</b> |
| <i>Valid</i> [%] | 95,48         | 85,81        | <b>90,64</b> | 93,59         | 88,93        | <b>91,26</b> | 92,21           | 86,47        | <b>94,34</b> |
| <i>Test</i> [%]  | 92,46         | 82,69        | <b>87,58</b> | 91,33         | 84,77        | <b>88,05</b> | 91,89           | 83,71        | <b>87,80</b> |

Com relação aos resultados globais do modelo RNC aos dados de cavitação, obteve-se 89,17% de acurácia, 87,57 de sensibilidade, 88,05% de precisão e 87,80% no *score-F1*. Esses resultados foram calculados com base na matriz de confusão e Equações 6.1, 6.2, 6.3 e 6.4. Como o modelo LeNet-5 apresentou resultados satisfatórios, não houve necessidade de calcular novos hiperparâmetros por meio da otimização bayesiana.

Foi realizada uma validação cruzada com  $K = 4$  *fold* e cada conjunto foi testado com os conjuntos adjacentes como no modelo RNC, em que o conjunto  $A$  foi testado nos conjuntos  $B$ ,  $C$  e  $D$ , e assim sucessivamente. A Tabela 107 apresenta os resultados dessa validação cruzada para 150 épocas, e nela é possível observar que esses resultados apresenta uma consistência na acurácia dos testes cruzados e ainda menor que acurácia do treinamento.

Tabela 107 – Validação cruzada do modelo RNC

| Test \ Train | A             | B             | C             | D             | Estatística dos testes |
|--------------|---------------|---------------|---------------|---------------|------------------------|
| A            | <b>96,15%</b> | 86,38%        | 84,65%        | 85,94%        | $85,65 \pm 0,73\%$     |
| B            | 85,55%        | <b>96,76%</b> | 84,93%        | 86,05%        | $85,51 \pm 0,46\%$     |
| C            | 85,27%        | 85,6%         | <b>97,04%</b> | 85,32%        | $85,39 \pm 0,14\%$     |
| D            | 88,00%        | 86,94%        | 86,27%        | <b>96,60%</b> | $87,07 \pm 0,71\%$     |

Percebe-se que as acurácias médias são próximas a 85% e que o desvio padrão calculado máximo foi de 0,74%. Esses valores são indícios que o modelo está em condição *optimal fitting*, entretanto há espaço para melhorias para que a acurácia dos testes apresentem valores próximos ao treinamento.

## 7 Conclusão

O diagnóstico automatizado de falhas em componentes mecânicos e o fenômeno da cavitação em bombas centrífugas têm sido objeto de intensas pesquisas recentes, devido à importância da detecção precoce para viabilizar a implementação de medidas preditivas, evitando não apenas a degradação estrutural e custos elevados, mas também riscos à vida humana. Esse cenário motivou a pesquisa apresentada nesta dissertação de mestrado, que teve como objetivo desenvolver um modelo computacional em AM para a classificação de cavitação em bombas centrífugas e defeitos em rolamentos, utilizando dados de acelerômetros.

Esta articulação, focada nos diagnósticos de falhas em rolamentos e fenômenos de cavitação, utilizaram as bases de dados da *Case Western Reserve University* e *UWA System Health Lab Prognostics Data Library*, respectivamente, resultando em uma metodologia para rotulação dos dados por meio de um conjunto de métodos estatísticos. Os métodos utilizados incluíram testes de normalidade, variância e hipóteses de igualdade das médias. Essa abordagem mostrou coerência quando comparada aos métodos tradicionais de engenharia, como o cálculo do *NPSH* para o fenômeno de cavitação e a análise dos parâmetros das formas de onda em relação à intensidade e tipo de falha nos dados de rolamentos. Além disso, esses métodos proporcionaram uma visão clara da rotulagem, com base no valor médio dos dados, identificando níveis intermediários de cavitação e rotulando os níveis de carga mecânica resistiva.

Após a criação das matrizes de *design*, foi empregado o método DVS para reduzir sua dimensionalidade, mantendo as informações relevantes para as soluções. Essa técnica foi crucial na etapa de preparação dos dados que acabou proporcionando melhores resultados ao modelo de classificação RNMP. No caso dos dados de rolamento, a técnica DVS resultou em um ganho de acurácia de 42,03%, passando dos 48,89% com os dados brutos para 69,44% com dados DVS. Enquanto que para os dados de cavitação, a acurácia aumentou de 75,38% para 77,90%, representando um ganho de 3,35%. Esse estudo das variações dos modelos de classificação demonstraram que a aplicação dos métodos DVS, SMOTE e ENN foi benéfica, principalmente para os dados de rolamento.

A técnica SMOTE foi utilizada para mitigar o problema de desbalanceamento das classes, decorrente da técnica de contagem de *hits*. Esse modelo combinado a técnica ENN resultou em ganho para a acurácia de 0,1% aos dados de rolamento com DVS, enquanto que para os dados de cavitação foi de 0,5%, conforme as Tabelas 58 e 98, respectivamente.

Com relação aos dados de rolamento, os principais resultados para o modelo RNMP foram, respectivamente, acurácia de 81,29% e o custo de 1,64 para o treinamento; na va-

validação esses valores foram de 80,00% e 1,76, enquanto que no teste os valores foram 69,50% de acurácia com um custo de 1,81, respectivamente. Sua matriz de confusão indicou melhor resultado para a condição *ER07*, que obteve 3.972 acertos dos 4.161 previstos. Como resultado geral, obteve-se na fase de testes 67,07% de sensibilidade, 73,50% de precisão e 68.63% no *F1 – score*. Esses resultados indicam um desempenho consistente quando aplicado a novos dados. Enquanto que para os dados de cavitação, a acurácia de 79,82% e o custo de 0,502 para o treinamento; na validação esses valores foram de 79,54% e 0.508, respectivamente. Como resultado de teste do modelo, obteve-se 77,90% de acurácia, 77,42% de sensibilidade, 75,68% de precisão e 75.25% no *F1 – score*. Sua matriz de confusão indicou melhor resultado para a condição *ONCAV*, que obteve 10.946 acertos dos 13.870 previstos, resultando em 82,51% de acertos para a condição de cavitação.

Os dados de entrada do modelo RNC são imagens bidimensionais originadas de sinais temporais. Esses sinais foram subamostrados utilizando janelamento retangular, garantindo que as frequências relevantes mantivessem suas amplitudes significativas. Dessa forma, 54 sinais foram transformados em 15.464 subamostras, representando uma técnica de *data augmentation* no aprendizado de máquina.

Assim o modelo RNC produziu para os dados de rolamento 15.464 espectrogramas, por meio da TFCT, que foram convertidos em imagens com escalas cinza. Os valores de convergência à acurácia e custo foram, respectivamente, de 99.23% e 1.49. Na validação alcançou-se, respectivamente, 94.2% e 1.55. Aos estudos da acurácia e precisão por classe, foram geradas matrizes de confusão para treinamento, validação e teste, ocorrendo alguns erros dos tipos *I* e *II*, a exemplo da classe *ER07*, em que o modelo previu 274 ocorrências, mas 34 delas pertencem outras classes e, semelhantemente, 3 amostras da classe *ER07* foram erroneamente classificadas. Os resultados globais do modelo RNC foram de 93.90% em acurácia, 93.77% de sensibilidade, 93.41% em precisão e 92.98% no *F1 – score*. Além disso, foi realizada a validação cruzada com  $K = 4$  para verificar a capacidade de generalização dos modelos, obtendo resultados de *optimal – fitting*, com acurácias médias acima de 90% para toda a base cruzada, demonstrando um desempenho superior apenas aos dados utilizados no treinamento.

Enquanto que nos dados de cavitação, o modelo RNC produziu 13.100 espectrogramas que também foram convertidos em imagens de escala cinza. Os valores de convergência à acurácia e custo foram, respectivamente, de 96,79% e 0,39. Na validação alcançou-se, respectivamente, 90,77% e 0,41. Ao gerar as matrizes de confusão, também foram constatados erros do tipo *I* e *II*, para os dados *ONCAV* e *OFFCAV* em que refletiram em resultados globais para os dados de teste de 89,17% em acurácia, 87,57% de sensibilidade, 88,05% em precisão e 87,80% no *F1 – score*.

No caso dos dados de rolamento, foi realizada ainda uma análise comparativa dos desempenhos dos modelos RNMP e RNC com aqueles de (ZHANG et al., 2020). Os

modelos RNMP e RNC apresentaram menor valor médio no desvio padrão, 0,23% e 2,95%, respectivamente contra 6,97% e 8,28%. E ainda, os modelos RNMP e RNC apresentaram maior acurácia média em relação aos obtidos por (ZHANG et al., 2020). Nas análises das acurácias dos modelos utilizou-se uma ANOVA combinada com o teste de Tukey com nível de significância de 5%, que indicou diferenças significativas entre as médias dos grupos, a exceção de RNC e DFCNN<sub>drop</sub>.

Portanto, essa pesquisa resultou em uma metodologia para rotulação dos dados por meio de testes de hipóteses e estatística descritiva para a definição do valor limiar e uma técnica de amostragem a partir de análise de sinais. No caso da etapa de preparação de dos dados contou com os métodos *DVS*, *SMOTE*, *TRF* e *TFCT* para cada fins de aplicação. Com relação a etapa de extração de característica utilizou a técnica de contagem de *hits* que pode ser ainda aprimorada ao incluir novos parâmetros da forma da onda na matriz *design* que proporcionou uma melhor separação nos *cluster* dos dados de entrada. Para a classificação de padrões de falhas de rolamento e presença de cavitação, foram utilizados os modelos RNC e RNMP, e seus resultados foram avaliados usando matriz de confusão, além das métricas usuais de acurácia, precisão, *recall* e *F1-score*. Além disso, foram realizadas validações cruzadas nos dois modelos (RNMP e RNC) e nas duas bases de dados (rolamento e cavitação). Em todas as situações foram criados 4 grupos para treinamento e testes cruzados, obtendo em todos os casos desempenhos de generalização robusta, uma vez que os resultados com as bases cruzadas foram inferiores aos do treinamento, confirmando a capacidade dos modelos em generalizar novos dados. Por fim, no caso da base de rolamento foram realizadas testes de hipóteses para comparar o desempenho dos modelos de AM desta dissertação com (ZHANG et al., 2020).

No encerramento desta dissertação, identificam-se oportunidades para melhorias e avanços, tanto nos modelos computacionais quanto em suas aplicações. A seguir, são apresentados os principais trabalhos futuros:

- Aprimoramento na etapa de extração de características no modelo RNMP: Nesta etapa, sugere-se a utilização de parâmetros estatísticos para melhorar a extração de características no modelo RNMP, facilitando a separação dos dados de entrada.
- Balanceamento da base de dados para o modelo RNC: Assim como no modelo RNMP, propõe-se o desenvolvimento de um modelo *SMOTE* combinado com autoencoder para balancear os dados de classes minoritárias.
- Aplicação dos modelos RNC e RNMP à base de dados de cavitação e trincamento da Itaipu ParqueTec: Implementar um sistema de monitoramento que integre dados de falhas mecânicas e operacionais em uma única fonte de dados.
- Aplicação de outros modelos de AM aplicados a todos esses conjuntos de dados.

# Referências

AGGARWAL, C. C. et al. Neural networks and deep learning. *Springer*, Springer, v. 10, n. 978, p. 3, 2018. Citado 4 vezes nas páginas 79, 81, 85 e 86.

ALVAREZ, G. P. Real-time fault detection and diagnosis using intelligent monitoring and supervision systems. In: *Fault Detection, Diagnosis and Prognosis*. [S.l.]: IntechOpen, 2020. Citado na página 25.

BATISTA, G. E. A. P. A.; PRATI, R. C.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, ACM, v. 6, n. 1, p. 20–29, 2004. Citado 2 vezes nas páginas 89 e 90.

BISHOP, C. M.; NASRABADI, N. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006. v. 4. Citado na página 66.

BOMBAS, W. <https://wortecbombas.com.br/conceito-das-bombas-centrifugas-radiais/>. 2023. Disponível em: <<https://wortecbombas.com.br/conceito-das-bombas-centrifugas-radiais/>>. Citado na página 48.

BRUNET, F. Contributions to parametric image registration and 3d surface reconstruction. *European Ph. D. in Computer Vision, Université d'Auvergne, Clérmont-Ferrand, France, and Technische Universitat Munchen, Germany*, v. 234, 2010. Citado na página 65.

BURKOV, A. *Machine learning engineering*. [S.l.]: True Positive Incorporated Montreal, QC, Canada, 2020. v. 1. Citado 4 vezes nas páginas 61, 62, 63 e 65.

CANZIANI, A.; PASZKE, A.; CULURCIELLO, E. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016. Citado na página 83.

CENTER WEST RESERVE UNIVERSITY. *Download a Data File the Center West Reserve University (CWRU)*. 2018. Disponível em: <<https://engineering.case.edu/bearingdatacenter/download-data-file>>. Citado 5 vezes nas páginas 101, 102, 104, 164 e 174.

CHAO, Q. et al. Cavitation intensity recognition for high-speed axial piston pumps using 1-d convolutional neural networks with multi-channel inputs of vibration signals. *Alexandria Engineering Journal*, Elsevier, v. 59, n. 6, p. 4463–4473, 2020. Citado 4 vezes nas páginas 14, 30, 39 e 40.

CHAO, Q. et al. Identification of cavitation intensity for high-speed aviation hydraulic pumps using 2d convolutional neural networks with an input of rgb-based vibration data. *Measurement Science and Technology*, IOP Publishing, v. 31, n. 10, p. 105102, 2020. Citado 4 vezes nas páginas 14, 30, 38 e 39.

COANDĂ, P.; AVRAM, M.; CONSTANTIN, V. A state of the art of predictive maintenance techniques. In: IOP PUBLISHING. *IOP Conference Series: Materials Science and Engineering*. [S.l.], 2020. v. 997, n. 1, p. 012039. Citado na página 22.

- DENG, S. et al. Research on fault diagnosis of mud pump fluid end based on acoustic emission. *Advances in Mechanical Engineering*, SAGE Publications Sage UK: London, England, v. 9, n. 6, p. 1687814017711393, 2017. Citado 3 vezes nas páginas 14, 29 e 32.
- ESCALER, X. et al. Detection of cavitation in hydraulic turbines. mechanical systems. *Signal Processing*, v. 20, n. 4, p. 983–1007, 2006. Citado na página 49.
- FAYYAD, U.; PIATETSKY-SHAPIO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37–37, 1996. Citado na página 96.
- FERNÁNDEZ, A. et al. *Learning from imbalanced data sets*. [S.l.]: Springer, 2018. v. 10. Citado na página 89.
- FERNÁNDEZ-OSETE, I. et al. Review of the uses of acoustic emissions in monitoring cavitation erosion and crack propagation. *Foundations*, MDPI, v. 4, n. 1, p. 114–133, 2024. Citado 2 vezes nas páginas 24 e 51.
- FILIPPIN, C. G. et al. *Emissão acústica: conceitos e aplicações*. 1. ed. Curitiba: Instituto Lactec; Usina Elétrica a Gás de Araucária (UEGA); Moura & Fiorito Consultoria, 2017. EBook, Cod. 40,290, 128 p. : il. color. ; 10 MB. ISBN 9788569527022. Citado 2 vezes nas páginas 52 e 53.
- FOLLAND, G. B. *Fourier analysis and its applications*. [S.l.]: American Mathematical Soc., 2009. v. 4. Citado 3 vezes nas páginas 54, 55 e 56.
- GABOR, D. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-part III: radio and communication engineering*, IET, v. 93, n. 26, p. 429–441, 1946. Citado na página 56.
- GARNETT, R. *Bayesian optimization*. [S.l.]: Cambridge University Press, 2023. Citado 4 vezes nas páginas 92, 93, 94 e 95.
- GHOSH, K. et al. The class imbalance problem in deep learning. *Machine Learning*, Springer, p. 1–57, 2022. Citado na página 89.
- GOLUB, G.; KAHAN, W. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, SIAM, v. 2, n. 2, p. 205–224, 1965. Citado na página 90.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016. Citado 7 vezes nas páginas 60, 72, 80, 81, 82, 84 e 85.
- GROSSE, C. U. et al. *Acoustic emission testing: Basics for research–applications in engineering*. [S.l.]: Springer Nature, 2021. Citado na página 51.
- HARRIS, T. A.; KOTZALAS, M. N. *Advanced concepts of bearing technology: rolling bearing analysis*. [S.l.]: CRC press, 2006. Citado na página 22.
- HAYKIN, S. *Redes neurais: princípios e prática*. [S.l.]: Bookman Editora, 2001. Citado 4 vezes nas páginas 68, 69, 71 e 72.
- HE, D. et al. Gas–liquid two-phase flow pattern identification of a centrifugal pump based on smote and artificial neural network. *Micromachines*, MDPI, v. 13, n. 1, p. 2, 2021. Citado 5 vezes nas páginas 14, 30, 36, 37 e 50.

- HEATON, J. Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618. *Genetic programming and evolvable machines*, Springer, v. 19, n. 1, p. 305–307, 2018. Citado 3 vezes nas páginas 59, 60 e 87.
- HODKIEWICZ, M.; PAN, J. Identification of transient axial vibration on double-suction pumps during partial flow operation. *Acoustics Australia*, v. 32, n. 1, p. 25–32, 2004. Citado 3 vezes nas páginas 98, 99 e 100.
- INGRAM, G. *Basic concepts in turbomachinery*. [S.l.]: Bookboon, 2009. Citado na página 48.
- IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: PMLR. *International conference on machine learning*. [S.l.], 2015. p. 448–456. Citado na página 85.
- JEAN-PIERRE, F.; PIERRE, M. J. *Fundamentals of cavitation*. [S.l.]: Kluwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow, 2004. Citado na página 49.
- JIANG, H. *Machine learning fundamentals: A concise introduction*. [S.l.]: Cambridge University Press, 2021. Citado 13 vezes nas páginas 59, 61, 62, 63, 64, 65, 66, 78, 79, 80, 81, 82 e 83.
- KARASSIK, I. J. et al. *Pump handbook*. [S.l.]: McGraw-Hill Education, 2008. Citado 3 vezes nas páginas 22, 25 e 50.
- KIRSCHNER, O.; RIEDELBAUCH, S. et al. Cavitation detection in hydraulic machinery by analyzing acoustic emissions under strong domain shifts using neural networks. *Physics of Fluids*, AIP Publishing, v. 35, n. 2, 2023. Citado 2 vezes nas páginas 45 e 87.
- KOLLMANNNSBERGER, S. et al. *Deep learning in computational mechanics*. [S.l.]: Springer, 2021. Citado 3 vezes nas páginas 74, 87 e 88.
- KUMAR, A. et al. Improved deep convolution neural network (cnn) for the identification of defects in the centrifugal pump using acoustic images. *Applied Acoustics*, Elsevier, v. 167, p. 107399, 2020. Citado 4 vezes nas páginas 14, 29, 34 e 35.
- KUMAR, D. et al. Identification of inlet pipe blockage level in centrifugal pump over a range of speeds by deep learning algorithm using multi-source data. *Measurement*, Elsevier, v. 186, p. 110146, 2021. Citado 4 vezes nas páginas 14, 30, 41 e 42.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Ieee, v. 86, n. 11, p. 2278–2324, 1998. Citado 4 vezes nas páginas 43, 84, 115 e 116.
- LEI, Y. et al. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, Elsevier, v. 138, p. 106587, 2020. Citado 3 vezes nas páginas 23, 66 e 67.
- MACINTYRE, A. J. *Bombas e instalações de bombeamento*. [S.l.]: Guanabara Dois, 1980. Citado na página 50.
- MALLAT, S. *A wavelet tour of signal processing*. [S.l.]: Elsevier, 1999. Citado 2 vezes nas páginas 57 e 58.

- MARTINS, A. O. Redes neurais no ensino básico. *Mestrado Profissional em Matemática em Rede Nacional*, Universidade Federal de Alagoas, p. 131, 2021. Citado na página 77.
- MICROSOFT. *Aprendizado profundo x Aprendizado de máquina em Azure Machine Learning versus Machine Learning*. 2023. Disponível em: <<https://learn.microsoft.com/pt-br/azure/machine-learning/concept-deep-learning-vs-machine-learning?view=azureml-api-2>>. Citado 2 vezes nas páginas 60 e 67.
- PANDA, A. K.; RAPUR, J. S.; TIWARI, R. Prediction of flow blockages and impending cavitation in centrifugal pumps using support vector machine (svm) algorithms based on vibration measurements. *Measurement*, Elsevier, v. 130, p. 44–56, 2018. Citado 3 vezes nas páginas 30, 35 e 36.
- PARQUETEC. *Relatório de Aplicabilidade e Fundamentação das Técnicas de Monitoramento de Cavitação e Trincas em Turbinas Francis*. [S.l.]: ITAIPU PARQUETEC, 2023. Citado 2 vezes nas páginas 23 e 24.
- PENHA, R. M. L. da. *ANÁLISE DE SINAIS EM REGIME TRANSIENTE APLICANDO A TÉCNICA DE WA VELET*. Tese (Doutorado) — UNIVERSIDADE DE SÃO PAULO, 1999. Citado 4 vezes nas páginas 55, 56, 57 e 58.
- PETERS, T. *Data-driven science and engineering: machine learning, dynamical systems, and control: by SL Brunton and JN Kutz, 2019, Cambridge, Cambridge University Press, 472 pp., £ 49.99 (hardback), ISBN 9781108422093. Level: postgraduate. Scope: textbook*. [S.l.]: Taylor & Francis, 2019. v. 60. Citado 2 vezes nas páginas 91 e 92.
- PRESS, W. H. *The art of scientific computing*. [S.l.]: Cambridge university press, 1992. Citado na página 55.
- PYTORCH. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37–37, 1996. Citado 2 vezes nas páginas 112 e 113.
- RANAWAT, N. S.; KANKAR, P. K.; MIGLANI, A. Fault diagnosis in centrifugal pump using support vector machine and artificial neural network. *J. Eng. Res. EMSME Spec. Issue*, v. 99, p. 111, 2020. Citado 2 vezes nas páginas 29 e 35.
- RUSSELL, S. J. *Artificial intelligence a modern approach*. [S.l.]: Pearson Education, Inc., 2010. Citado na página 59.
- SEBASTIAN. *STAT 479: Machine Learning*. 2019. Disponível em: <[https://github.com/rasbt/stat479-machine-learning-fs19/blob/master/01\\_overview/01-ml-overview\\_\\_notes.pdf](https://github.com/rasbt/stat479-machine-learning-fs19/blob/master/01_overview/01-ml-overview__notes.pdf)>. Citado 2 vezes nas páginas 59 e 60.
- SHANTHAMALLU, U. S.; SPANIAS, A. Machine and deep learning algorithms and applications. *Synthesis Lectures on Signal Processing*, Morgan & Claypool Publishers, v. 12, n. 3, p. 1–123, 2021. Citado 2 vezes nas páginas 66 e 67.
- SHENDE, S. et al. Bayesian topology optimization for efficient design of origami folding structures. *Structural and Multidisciplinary Optimization*, Springer, v. 63, p. 1907–1926, 2021. Citado na página 92.
- SILVA, I. N. D. et al. *Artificial neural network architectures and training processes*. [S.l.]: Springer, 2017. Citado 8 vezes nas páginas 69, 70, 71, 73, 74, 75, 76 e 78.

- SOUZA, J. C. S. et al. Application of machine learning models in predictive maintenance of francis hydraulic turbines. *Available at SSRN 4019014*, 2022. Citado na página 22.
- STURGES, H. A. The choice of a class interval. *Journal of the american statistical association*, New York, v. 21, n. 153, p. 65–66, 1926. Citado na página 117.
- TANG, S.; ZHU, Y.; YUAN, S. An improved convolutional neural network with an adaptable learning rate towards multi-signal fault diagnosis of hydraulic piston pump. *Advanced Engineering Informatics*, Elsevier, v. 50, p. 101406, 2021. Citado 3 vezes nas páginas 14, 30 e 42.
- TANG, S.; ZHU, Y.; YUAN, S. A novel adaptive convolutional neural network for fault diagnosis of hydraulic piston pump with acoustic images. *Advanced Engineering Informatics*, Elsevier, v. 52, p. 101554, 2022. Citado 4 vezes nas páginas 14, 30, 43 e 44.
- TIWARI, R.; BORDOLOI, D.; DEWANGAN, A. Blockage and cavitation detection in centrifugal pumps from dynamic pressure signal using deep learning algorithm. *Measurement*, Elsevier, v. 173, p. 108676, 2021. Citado 5 vezes nas páginas 14, 30, 37, 38 e 41.
- VIDHYA, A. *Overcoming Class Imbalance using SMOTE Techniques*. 2020. Acesso em: 13 de maio de 2024. Disponível em: <<https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/>>. Citado 3 vezes nas páginas 89, 90 e 94.
- YAN, J. et al. Research on fault diagnosis of hydraulic pump using convolutional neural network. *Journal of Vibroengineering*, JVE International Ltd., v. 18, n. 8, p. 5141–5152, 2016. Citado 4 vezes nas páginas 14, 29, 31 e 32.
- ZHANG, J. et al. A new bearing fault diagnosis method based on modified convolutional neural networks. *Chinese Journal of Aeronautics*, Elsevier, v. 33, n. 2, p. 439–447, 2020. Citado 13 vezes nas páginas 14, 29, 33, 34, 100, 154, 160, 176, 186, 187, 188, 214 e 215.
- ZHANG, N. et al. Cavitating flow-induced unsteady pressure pulsations in a low specific speed centrifugal pump. *Royal society open science*, The Royal Society Publishing, v. 5, n. 7, p. 180408, 2018. Citado na página 51.