

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

GABRIEL CIANNI DE OLIVEIRA

CLASSIFICADOR INTELIGENTE PARA FALHAS MECÂNICAS EM  
MOTORES DE INDUÇÃO TRIFÁSICOS

Porto Alegre

2023

GABRIEL CIANNI DE OLIVEIRA

CLASSIFICADOR INTELIGENTE PARA FALHAS MECÂNICAS EM  
MOTORES DE INDUÇÃO TRIFÁSICOS

Trabalho de Conclusão de Curso apresentado ao  
Curso de Engenharia Elétrica, desenvolvido no  
Laboratório de Instrumentação Eletro Eletrônica & Inteligencia Artificial da Universidade  
Federal do Rio Grande do Sul como requisitos  
parcial à obtenção do grau de Bacharel em Engenharia Elétrica

Orientador: Prof. Dr. Alexandre Balbinot

Porto Alegre

2023

GABRIEL CIANNI DE OLIVEIRA

CLASSIFICADOR INTELIGENTE PARA FALHAS MECÂNICAS EM  
MOTORES DE INDUÇÃO TRIFÁSICOS

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica, desenvolvido no Laboratório de Instrumentação Eletro Eletrônica & Inteligencia Artificial da Universidade Federal do Rio Grande do Sul como requisitos parcial à obtenção do grau de Bacharel em Engenharia Elétrica

Aprovado em: 23/03/2023

BANCA EXAMINADORA

---

Prof. Dr. Alexandre Balbinot  
Universidade Federal do Rio Grande do Sul  
Nota atribuída:   A  

---

Prof. Dr. Tiago Oliveria Weber  
Universidade Federal do Rio Grande do Sul  
Nota atribuída:   A  

---

Prof. Dr. Juliano Costa Machado  
IFSUL  
Nota atribuída:   A

Este trabalho é inteiramente dedicado a todos que me acompanharam durante minha formação, em especial a minha companheira, Aline Nunes, a quem minha inspira e me apoia incondicionalmente, assim como meu pai Marcos, minha mãe Ana Paula e minha irmã Ana Júlia, que me forneceram a oportunidade de uma excelente formação tanto acadêmica quanto pessoal.

## **AGRADECIMENTOS**

Gostaria de agradecer ao meu orientador Alexandre Balbinot, por todo o conhecimento compartilhado, ao longo de toda minha trajetória dentro da universidade, sendo sempre disponível e disposto a ensinar, com certeza detém um papel fundamental dentro do meu amadurecimento acadêmico que está transmutado na forma dessa monografia.

Aos meus pais, Ana Paula Marques Cianni de Olivera e Marcos Antonio de Oliveira Cruz, que assim como minha irmã, Ana Júlia Cianni de Olivera, são minha pedra angular e tem um papel tão fundamental em minha vida que qualquer descrição em palavras é aquém da real dimensão da sua importância.

A Aline Nunes de Souza, minha companheira desde a época do colégio técnico e que sempre foi meu maior apoio, minha verdadeira inspiração. Quem divide o fardo nas horas difíceis e multiplica alegria nos bons momentos. Muito obrigado, minhas conquistas são divididas com você, te amo.

Agradeço também aos meus amigos de curso, que são verdadeiros companheiros ao longo dessa trajetória, pelos conselhos, ajudas e risadas.

Ática forma! Altivo porte! em tua trama  
Homens de mármore e mulheres emolduras  
Como galhos de floresta e palmilhada grama:  
Tu, forma silenciosa, a mente nos torturas  
Tal como a eternidade: Fria Pastoral!  
Quando a idade apagar toda a atual grandeza,  
Tu ficarás, em meio às dores dos demais,  
Amiga, a redizer o dístico imortal:  
"A beleza é a verdade, a verdade a beleza"  
— É tudo o que há para saber, e nada mais.  
(Ode a uma urna grega, John Keats)

## RESUMO

Os motores de indução são um dos equipamentos mais relevantes para a indústria moderna devido a sua alta aplicabilidade e seu elevado rendimento energético. Em condições inadequadas de operação, como temperatura ou pressão elevada, vibrações ininterruptas e variações de carga, os motores tendem a aumentar a suscetibilidade a falhas externas e internas, acarretando em uma contração na sustentabilidade econômica do processo fabril e tornando-se uma fonte de vibrações excessivas, o que para análise estrutural pode ser fontes de risco a segurança de todo o ambiente no qual é instalado e seus arredores. Este trabalho tem como objetivo elaborar um sistema de diagnóstico para falhas mecânicas de motores de indução trifásico, utilizando sinais de vibração do maquinário provenientes base de dados MaFaulDa, para produzir um estudo comparativo de algoritmos inteligentes, como redes neurais artificiais, Florestas aleatórias, Adaboost, regressão logística e XGBoost utilizando características extraídas das séries temporais: índices de entropia, curtose, amplitude média, magnitude espectral média, máxima, sua centróide, em um sistema de segmentação de 10 ms. Avaliando duas técnicas para a otimização de hiperparâmetros - Algoritmos genéticos e a pesquisa randômica. Obtendo um teor de generalização em sua topologia final, indicado pela métrica de *accuracy* acima de 98%, adequando-se a faixa de valores dos modelos referencias na área.

Palavras-chave: Modelos Inteligentes. Inteligencia Computacional. Motores de indução trifásicos. Manutenção preditiva.

## ABSTRACT

*Induction motors are one of the most relevant equipment in modern industry due to their high applicability and energy efficiency. Under inadequate operating conditions, such as high temperature or pressure, uninterrupted vibrations and load variations, motors tend to increase their susceptibility to external and internal failures, resulting in a contraction in the economic sustainability of the manufacturing process and becoming a source of excessive vibrations, which is for structural analysis a risk source to the entire environment safety. This work aims to develop a diagnostic system for mechanical failures of three-phase induction motors, using machinery vibration signals from the MaFaulDa database, to elaborate a comparative study of intelligent algorithms, such as artificial neural networks, Random Forests, Adaboost, logistic regression and XGBoost. Extracting features from the time series such as entropy, kurtosis, mean amplitude, average spectral magnitude, its maximum and its centroid, in a 10 ms segmentation system. Evaluating the best two techniques for hyperparameter optimization - Genetic Algorithms and Random Search. Obtaining a generalization content in its final topology, indicated by the accuracy metric above 98%, fitting in the range of values of the area's reference models.*

**Key-words:** Intelligent Models. Computational Intelligence. Three-phase induction motors. Predictive Maintenance.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Distribuição de falhas em máquinas rotativas. . . . .	18
Figura 2 – Mapa de Autores referenciados. . . . .	21
Figura 3 – Mapa de Artigos referenciados. . . . .	21
Figura 4 – Mapa de Artigos referenciados. . . . .	22
Figura 5 – Palavras chaves de publicação. . . . .	23
Figura 6 – <i>Zoom In</i> das palavras chaves de publicação. . . . .	23
Figura 7 – Esquema de montagem do experimento para captura dos dados. . . . .	33
Figura 8 – Série Temporal Operação Normal. . . . .	34
Figura 9 – Série Temporal Operação com Desalinhamento Horizontal. . . . .	35
Figura 10 – Série Temporal Operação com Desalinhamento Vertical. . . . .	35
Figura 11 – Série Temporal Operação com Desbalanço de Carga. . . . .	35
Figura 12 – Série Temporal Operação com Falha no Rolamento Dianteiro. . . . .	35
Figura 13 – Série Temporal Operação com Falha no Rolamento Posterior. . . . .	36
Figura 14 – Histograma Operação Normal. . . . .	37
Figura 15 – Histograma Operação com Desalinhamento Horizontal. . . . .	38
Figura 16 – Histograma Operação com Desalinhamento Vertical. . . . .	38
Figura 17 – Histograma Operação com Desbalanço de Carga. . . . .	39
Figura 18 – Histograma Operação com Falha no Rolamento Dianteiro. . . . .	39
Figura 19 – Histograma Operação com Falha no Rolamento Posterior. . . . .	40
Figura 20 – Espectograma Operação Normal. . . . .	41
Figura 21 – Espectograma Operação com Desalinhamento Horizontal. . . . .	41
Figura 22 – Espectograma Operação com Desalinhamento Vertical. . . . .	41
Figura 23 – Espectograma Operação com Desbalanço de Carga. . . . .	42
Figura 24 – Espectograma Operação com Falha no Rolamento Dianteiro. . . . .	42
Figura 25 – Exemplo de Segmentação de Sinal por janela deslizante. . . . .	44
Figura 26 – Correlação das Características Extraídas. . . . .	47
Figura 27 – Variância acumulada por número de componente. . . . .	48
Figura 28 – Correlação: PCA. . . . .	48
Figura 29 – Dendograma Agrupamento Hierárquico. . . . .	50
Figura 30 – Correlação: Agrupamento Hierárquico. . . . .	51
Figura 31 – Etapa de pré-processamento. . . . .	52

Figura 32 – AdaBoost Clusterização de Características com Algoritmo Genético. . . . .	57
Figura 33 – AdaBoost PCA com Algoritmo Genético. . . . .	57
Figura 34 – AdaBoost Clusterização de Características com Pesquisa Randômica. . . . .	58
Figura 35 – AdaBoost PCA com Pesquisa Randômica. . . . .	58
Figura 36 – Floresta Aleatórias Clusterização de Características com Algoritmo Genético.	60
Figura 37 – Floresta Aleatória PCA com Algoritmo Genético. . . . .	60
Figura 38 – Floresta Aleatória Clusterização de Características com Pesquisa Randômica.	61
Figura 39 – Floresta Aleatória PCA com Pesquisa Randômica. . . . .	61
Figura 40 – RNA MLP Clusterização de Características com Algoritmo Genérico. . . . .	63
Figura 41 – RNA MLP PCA com Algoritmo Genérico. . . . .	63
Figura 42 – RNA MLP Clusterização de Características com Pesquisa Randômica. . . . .	64
Figura 43 – RNA MLP PCA com Pesquisa Randômica. . . . .	64
Figura 44 – Regressão Logística Clusterização de Características com Algoritmo Genérico.	66
Figura 45 – Regressão Logística PCA com Algoritmo Genérico. . . . .	66
Figura 46 – Regressão Logística Clusterização de Características com Pesquisa Randômica. . . . .	67
Figura 47 – Regressão Logística PCA com Pesquisa Randômica. . . . .	67
Figura 48 – XGBOOST Clusterização de Características com Algoritmo Genérico. . . . .	69
Figura 49 – XGBOOST PCA com Algoritmo Genérico. . . . .	69
Figura 50 – XGBOOST Clusterização de Características com Pesquisa Randômica. . . . .	70
Figura 51 – XGBOOST PCA com Pesquisa Randômica. . . . .	70
Figura 52 – Resíduos - Modelo <i>Accuracy</i> . . . . .	73
Figura 53 – Resíduos - Modelo Custo Computacional. . . . .	74
Figura 54 – Gráfico de Efeitos - Desempenho. . . . .	74
Figura 55 – Gráfico de Efeitos - Custo Computacional. . . . .	75
Figura 56 – Estator de um motor de indução trifásico. . . . .	83
Figura 57 – Comparativos entre tipos de rotores. . . . .	86
Figura 58 – Topologia Neurônio MCP. . . . .	87
Figura 59 – Exemplo de Rede Neural MLP. . . . .	88
Figura 60 – Topologia de Floresta Aleatória. . . . .	90
Figura 61 – Topologia do Adaboost. . . . .	92
Figura 62 – Fluxograma do Algoritmo Genético. . . . .	95

## LISTA DE TABELAS

Tabela 1 – Métricas em Li, Ding e Sun (2018). . . . .	26
Tabela 2 – Métricas em Wen, Gao e Li (2017). . . . .	27
Tabela 3 – Métricas em Glowacz et al. (2018). . . . .	28
Tabela 4 – <i>Accuracy</i> em Jia et al. (2018). . . . .	29
Tabela 5 – Experimentos em Jiang et al. (2018). . . . .	30
Tabela 6 – Métricas em Jiang et al. (2018). . . . .	30
Tabela 7 – Métricas em Shao et al. (2018). . . . .	31
Tabela 8 – Tempo de Treinamento em Shao et al. (2018). . . . .	31
Tabela 9 – Resultados Médios destacados pela pesquisa bibliométrica. . . . .	31
Tabela 10 – Valores máximos de oscilações. . . . .	36
Tabela 11 – Valores mínimos de oscilações. . . . .	36
Tabela 12 – Valores RMS de oscilações. . . . .	37
Tabela 13 – Valores RMS médio e desvio padrão das oscilações segmentadas. . . . .	37
Tabela 14 – Curtose da distribuição das oscilações. . . . .	40
Tabela 15 – Distribuição categóricas . . . . .	43
Tabela 16 – Distribuição categóricas pós balanceamentos. . . . .	43
Tabela 17 – Vetor de Características. . . . .	46
Tabela 18 – <i>Accuracy</i> Médio Adaboost [%]. . . . .	59
Tabela 19 – Custo Computacional Médio Adaboost [s]. . . . .	59
Tabela 20 – Configurações de hiperparâmetros de Adaboost destacadas em relação aos possíveis cenários. . . . .	59
Tabela 21 – <i>Accuracy</i> Médio Floresta Aleatória [%]. . . . .	62
Tabela 22 – Custo Computacional Médio Floresta Aleatória [s]. . . . .	62
Tabela 23 – Configurações de hiperparâmetros de Florestas Aleatórias destacadas em relação aos possíveis cenários. . . . .	62
Tabela 24 – <i>Accuracy</i> Médio Redes Neurais MLP[%]. . . . .	65
Tabela 25 – Custo Computacional Médio Redes Neurais MLP [s]. . . . .	65
Tabela 26 – Configurações de hiperparâmetros de Redes Neurais MLP destacadas em relação aos possíveis cenários. . . . .	65
Tabela 27 – <i>Accuracy</i> Médio Regressão Logística[%]. . . . .	68
Tabela 28 – Custo Computacional Médio Regressão Logística [s]. . . . .	68

Tabela 29 – Configurações de hiperparâmetros de Regressão Logística destacadas em relação aos possíveis cenários. . . . .	68
Tabela 30 – Accuracy Médio XGBOOST[%]. . . . .	71
Tabela 31 – Custo Computacional Médio XGBOOST [s]. . . . .	71
Tabela 32 – Configurações de hiperparâmetros de XGBOOST destacadas em relação aos possíveis cenários. . . . .	71
Tabela 33 – Comparação métricas. . . . .	72
Tabela 34 – Resultados da Análise de Variância do DoE proposto à desempenho dos modelos. . . . .	72
Tabela 35 – Resultados da Análise de Variância do DoE proposto ao custo dos modelos .	73
Tabela 36 – Resultados Modelos referenciados. . . . .	75

## LISTA DE ABREVIATURAS E SIGLAS

Adaboost	<i>Adaptive Boosting</i>
ANFIS	<i>Adaptative Neuro-Fuzzy Inference System</i>
CNN	<i>Convolutional Neural Network</i>
C-LSTM	<i>Convolutional Long Short-Term Memory</i>
DBN	<i>Deep Belief Network</i>
DRBMNN	<i>Dynamic Routing-Based Multimodal Neural Network</i>
DWT	<i>Discret Wavelet Transform</i>
ELM	<i>Extreme Learning Machine</i>
GAN	<i>Generative Adversarial Network</i>
KNN	<i>K-Nearest Neighbour</i>
LSTM	<i>Long Short-Term Memory</i>
MDL	<i>Multimodal Deep Learning</i>
RNA MLP	Redes Neurais Artificiais Multi Layer Perceptron
RNN	<i>Recurrent Neural Neural</i>
STFT	<i>Short Time Fourier Transformer</i>
SVM	<i>Suport Vector Machine</i>
XGBoost	<i>Extreme Gradient Boost</i>

## LISTA DE SÍMBOLOS

$A$	Ampére
$I$	Intensidade de Corrente Elétrica
$I_p$	Intensidade máxima de Corrente Elétrica
$\omega$	Frequência Angular
$\vec{H}$	Vetor de Campo Magnético
$H_p$	Intensidade máxima do Campo Magnético
$\vec{B}$	Vetor de Indução Magnética
$B_p$	Intensidade máxima de Indução Magnética
$\mu$	Permeabilidade Magnética
$T$	Tesla
$\vec{E}$	Campo Elétrico
$\nabla$	Operação matemática de Gradiente

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>18</b>
2.1	Falhas em máquinas rotativas	18
2.2	Pesquisa Bibliométrica	20
<b>2.2.1</b>	<b>Trabalhos Relacionados</b>	<b>24</b>
2.2.1.1	Deep learning for smart manufacturing: Methods and applications	24
2.2.1.2	Remaining Useful Life Estimation in Prognostics Using Deep Convolution Neural Networks	25
2.2.1.3	A New Deep Transfer Learning Based on Sparse Auto-Encoder for Fault Diagnosis	26
2.2.1.4	Early fault diagnosis of bearing and stator faults of the single-phase induction motor using acoustic signals	27
2.2.1.5	Deep normalized convolutional neural network for imbalanced fault classifi- cation of machinery and its understanding via visualization	28
2.2.1.6	Multiscale Convolutional Neural Networks for Fault Diagnosis of Wind Tur- bine Gearbox	29
2.2.1.7	Highly-Accurate Machine Fault Diagnosis Using Deep Transfer Learning	30
2.2.1.8	Síntese de pesquisa biométrica	31
<b>3</b>	<b>METODOLOGIA</b>	<b>32</b>
3.1	Banco de Dados	32
3.2	Análise Exploratória dos dados	34
3.3	Engenharia de Característica	42
<b>3.3.1</b>	<b>Redução de Dimensionalidade</b>	<b>46</b>
<b>3.3.2</b>	<b>Normalização e Divisão dos Conjuntos de Dados</b>	<b>51</b>
3.4	Métodos Inteligentes e Otimização de Hiperparâmetros	53
<b>3.4.1</b>	<b>Métodos Inteligentes</b>	<b>53</b>
<b>3.4.2</b>	<b>Métodos de Otimização de Hiperparâmetros</b>	<b>55</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>56</b>
<b>4.0.1</b>	<b>Adaboost</b>	<b>56</b>
4.1	Florestas Aleatórias	59
4.2	Redes Neurais Artificiais MLP	62

4.3	Regressão Logística . . . . .	65
4.4	XGBOOST . . . . .	68
4.5	Comparações entre os modelos e Testes Estatísticos . . . . .	71
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>76</b>
<b>6</b>	<b>Trabalhos Futuros . . . . .</b>	<b>78</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>79</b>
.1	Motor de Indução Trifásico . . . . .	83
.2	Redes Neurais Artificiais . . . . .	87
.3	Árvore de Decisão e Florestas Aleatórias . . . . .	89
.4	Regressão Logística . . . . .	91
.5	AdaBoost . . . . .	91
.6	XGBoost . . . . .	92
.7	Pesquisa Extensiva e Pesquisa Randômica . . . . .	94
.8	Algoritmos Genéticos . . . . .	94



## 1 INTRODUÇÃO

O rápido desenvolvimento da ciência e tecnologia e sua integração com os processos industriais são os principais agentes motrizes de uma significativa mudança de paradigma produtivo. Marr (2018), avalia essa transição como um ícone de uma nova revolução industrial denominada como Indústria 4.0, a qual se baseia na incorporação de sistemas inteligentes nos meios fabris como forma de ampliar a autonomia e eficiência tanto de máquinas quanto processos (CARDOSO, 2016). Dessa forma, os motores de indução são um dos equipamentos mais relevantes para a indústria moderna, segundo Pinheiro, Maitelli e Salazar (2011) devido a sua alta aplicabilidade e seu rendimento energético elevado em circunstâncias apropriadas.

Motores em condições inadequadas de operação, como temperatura e pressão elevada, vibrações ininterruptas e variações de carga, tendem a aumentar a suscetibilidade a falhas externas e internas (PINHEIRO; MAITELLI; SALAZAR, 2011), acarretando em uma contração na sustentabilidade econômica do processo fabril. Ribeiro (2016) descreve o atual paradigma dos setores industriais como fonte de um aumento acelerado no grau de competição e rigidez operacional, acarretando baixas margens de erros para minimizar gastos com manutenção e interrupções não planejadas, assegurando maior controle sobre a perda de produção e consequentemente sobre a receita financeira do empreendimento.

As indústrias desenvolveram estratégias avançadas de monitoramento de condição para maquinário que auxiliam na predição de suas falhas (BEN-DAYA; KUMAR; MURTHY, 2016), visando a redução dos custos globais da operação, descrito por JANIER (2011) através da redução de ocorrência de paralisação da linha de produção e incremento da eficiência industrial.

Uma segunda abordagem para a justificativa do controle e predição de falhas em motores elétricos é elaborada a partir da ótica da saúde estrutural, Beards (1996) define dois principais fatores controláveis da amplitude e a frequência da vibração estrutural, a excitação aplicada e a resposta da estrutura a essa excitação particular. Essas excitações são produzidas, segundo Beards (1996), por fontes externas como vibrações do solo ou da fundação, mas também podem ser geradas internamente à estruturas, como cargas móveis e máquinas rotativas. As forças excitatórias podem ser periódicas, harmônicas ou aleatórias relativo à sua natureza, como, oscilações causada por cargas de choque ou impulsivas por exemplo.

Excesso de vibrações em pontos sensíveis da estrutura podem afetar sua integridade, acarretando colapso estrutural. O motor elétrico em operação não-normalizada é considerado uma fonte de vibração excessiva e, portanto, classificado por Farrar e Worden (2007) como pos-

sível provedor de oscilações ressonantes à estrutura na qual é instalado, acarretando riscos não só aos operadores, mas relativo ao caso, a todos presentes na estrutura em questão e arredores.

As falhas provenientes de motores de indução são enquadradas principalmente em três grupos referente ao seu local de ocorrência (SIDDIQUE; YADAVA; SINGH, 2005), estes sendo Rolamento, Estator e Rotor. Seu percentual de probabilidade de ocorrência, segundo a IEEE Motor Reliability Working Group (1985), é de, respectivamente, 41%, 37% e 10%. Os 12% restantes são formados por diversos outros grupos de falhas. Percebe-se o impacto das falhas de motores no ambiente fabril, tornando necessário a utilização de análises que as preveja e assim possam impedi-las, antes mesmo que as falhas ocorram (SPAMER, 2009).

Em função das colocações anteriores este trabalho tem por objetivo principal desenvolver um estudo comparativo de diferentes técnicas de aprendizagem de máquina, que através dos sinais de vibração provenientes de acelerômetros instalados em um motor de indução trifásico, adquiridos pela base de dados MaFaulda, consiga identificar, além da operação normalizada, as principais falhas operacionais mecânicas, essas sendo, o desbalanço de carga, desalinhamento vertical e horizontal no eixo, falhas nos rolamentos posteriores e inferiores da transmissão, tanto no anel externo e interno dos rolamentos quanto nas suas esferas.

Os objetivos secundários são:

- avaliar e identificar através de análises exploratória informações a respeito das assinaturas vibracionais dos três canais de vibração captados;
- analisar método de segmentação dos sinais para a extração de características;
- propor um algoritmo para redução da dimensionalidade do espaço de características;
- analisar desempenho de métodos de classificação inteligentes: Redes Neurais Artificiais Multi Layer Perceptron (RNA MLP), Regressão Logística, Random Forest, Apative Boosting (AdaBoost) e Extreme Gradient Boost (XGBoost);
- avaliar diferentes metodologias para o ajuste fino dos hiperparametrização do classificador através de pesquisa randômica e meta-heurística de algoritmos genéticos.

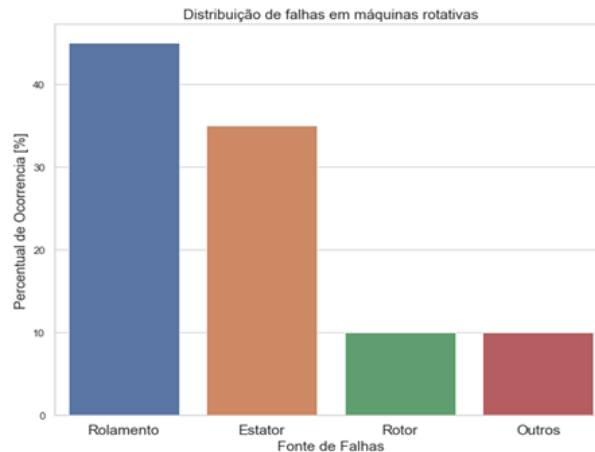
## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta de forma resumida tópicos referentes as principais falhas em máquinas rotativas, acompanhando um estudo das principais pesquisas publicadas dentro do escopo desta monografia. O funcionamento e modelo de um motor de indução trifásico estão descritos devidamente detalhados nos apêndice da monografia, assim como os principais modelos inteligentes utilizados e métodos de otimização de hiperparâmetros.

### 2.1 FALHAS EM MÁQUINAS ROTATIVAS

Os principais agrupamentos de irregularidades em máquinas rotativas são descritos quanto ao seu tipo - elétrico ou mecânico - ou quanto ao local de ocorrência, estator ou rotor. A distribuição do percentual tende a convergir em diversas pesquisas segundo Bazurto, Quispe e Mendoza (2016) para 45% de falhas mecânicas relacionadas ao rolamento, 35% irregularidades operacionais do estator, 10% falhas nos rotores e 10% restantes distribuídos em uma gama de outras categorias minoritárias, como a Figura 1 compara.

Figura 1 – Distribuição de falhas em máquinas rotativas.



Fonte: Adaptado de Bazurto, Quispe e Mendoza (2016).

As Falhas de rolamentos são compostas por inoperatividades no rolamento da máquina, geralmente são compostos por dois anéis concêntricos cujo, em seu interior, esferas são armazenadas como maneira de promover o movimento apenas de um dos anéis ao serem girados. As falhas podem ser causadas por pontos específicos que foram lascados, trincados e afundados ou distribuídas ao longo do elemento, como desalinhamentos, ondulações e maior índice de rugosidades (BAZURTO; QUISPE; MENDOZA, 2016). Causadas principalmente por problemas de

lubrificação insuficiente ou excessiva, desalinhamento, vibrações excessivas, superaquecimento e sobrecarga no eixo (ABB (EUA), 2015)

Falhas no estator são normalmente associadas às deformidades em sua isolação elétrica (BAZURTO; QUISPE; MENDOZA, 2016) causadas, segundo relatório da ABB (EUA) (2015), principalmente por superaquecimento ou sobrecarga operativa, ocasionando curto-circuitos entre os enrolamentos, falhas entre fases, circuito aberto entre bobinas ou terra. Influenciando a Condição de Operação devido ao alto consumo de corrente elétrica e causando degradações no maquinário pois as falhas são acumulativas (BINDU; THOMAS, 2014). Para Bhowmik, Pradhan e Prakash (2013) às falhas nos estatores podem afetar majoritariamente três aspectos da operação da máquina:

- frame: ocasionar problemas relacionados a vibração, correntes circulantes, falhas de aterramento e de resfriamentos;
- laminação do núcleo: reduz a rigidez do material e aumento de sua temperatura;
- enrolamentos: causados pelo enfraquecimento da isolação entre espiras ou danos localizados no condutor, gerando deslocamento de acionadores, contaminação por umidade, óleo ou outro elemento.

As falhas provenientes do rotor ocorrem, segundo Costa, Kashiwagi e Mathias (2015) em grande maioria devido à quebra das barras da gaiola de esquilo, através de vibrações excessivas, oscilações de torque e mudanças bruscas na dinâmica de partidas ou na temperatura, ocorrendo majoritariamente com motores de maior porte de potência, causa redução no tempo de serviço, torque médio e produção constantes de vibrações são produtos dos erros no rotor do maquinário e segundo Matic et al. (2012), são caracterizados como uma falha cumulativa, podendo destruir a máquina.

As falhas mecânicas resultam no desalinhamento do centro do eixo do rotor em relação ao estator, descrito muitas vezes como falhas de excentricidade. Causam empenamento do eixo e danos ao rolamento, resultantes de erro na fabricação e montagens incorretas, comprometendo o desempenho da máquina devido ao desgaste dos rolamentos (NANDI; TOLIYAT; LI, 2005). Bazurto, Quispe e Mendoza (2016) associa as falhas em máquinas rotativas com má utilização, estabelecendo a etapa de operação como principal ponto a se observar em um diagnóstico através da captura de sinais como temperatura, vibração e distorção de corrente elétrica.

Para determinar as falhas, utiliza-se sistemas detectores que podem ser classificados através de sua sistemática, sendo geralmente definidos como sistemas baseados em modelos ou

baseados em sinais.

A metodologia de diagnósticos de falhas baseadas em modelos, utiliza equações matemáticas computacionais para descrever o sistema dinâmico analisado, obtendo referências dos pontos de operação desejado (LIBONI, 2017). No entanto esse método é restrito ao grau de precisão do modelo produzido e da capacidade computacional de processamento devido a sua alta complexidade (LIBONI, 2017).

Sistemas de predição baseados em sinais são referência para a temática de detecção de falhas em máquinas rotativas, principalmente pela complexidade dos modelos de sistemas eletromecânicos (GHANBARI, 2016). Consistindo na análise de variáveis de motores através da extração características que possam apontar eventuais anomalias, através dessas características, sistemas de inferência são ajustados de forma a garantir a detecção de falhas de forma automática (LIBONI, 2017).

## 2.2 PESQUISA BIBLIOMÉTRICA

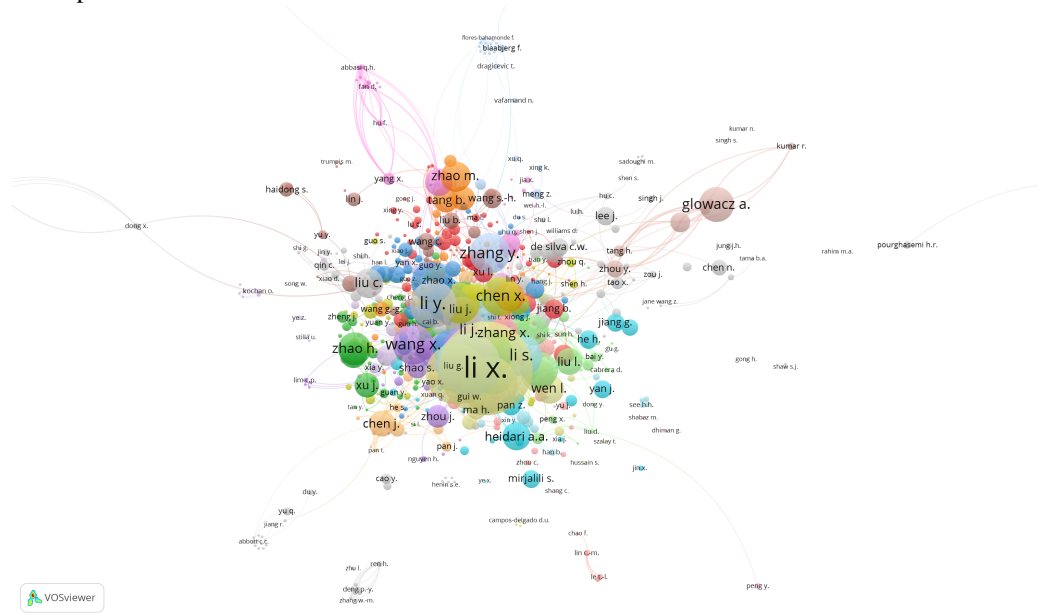
Com intuito de avaliar as publicações referência envolvendo predição de falha de motores elétricos através de metodologias inteligentes, utilizou-se o repositório da base de dados Scopus como fonte dos artigos escritos nos últimos 5 anos na área vigente. A pesquisa realizada foi feita através da seguinte *query*:

```
((("induction"OR "electrical") AND "motor"AND ("Fault"OR "Failure"OR "Malfunction") AND ("Detection"OR "Prediction") AND ("Machine Learning"OR ("Artificial"OR "Computational") AND "Intelligence") AND (limit-to (DOCTYPE, "ar")) AND (limit-to (PUBYEAR, 2022) OR limit-to (PUBYEAR, 2021) OR limit-to (PUBYEAR, 2020) OR limit-to (PUBYEAR, 2019) OR limit-to (PUBYEAR, 2018))
```

Resultado em 5423 artigos exportados em formato csv para análise bibliométrica no software *VOSviewer* versão 1.6.17. O principal objetivo dessa pesquisa era determinar as principais metodologias utilizadas através de um mapa clusterizado de palavras chaves mais relevantes, assim como, os principais artigos publicados, seus autores e países em número de publicações.

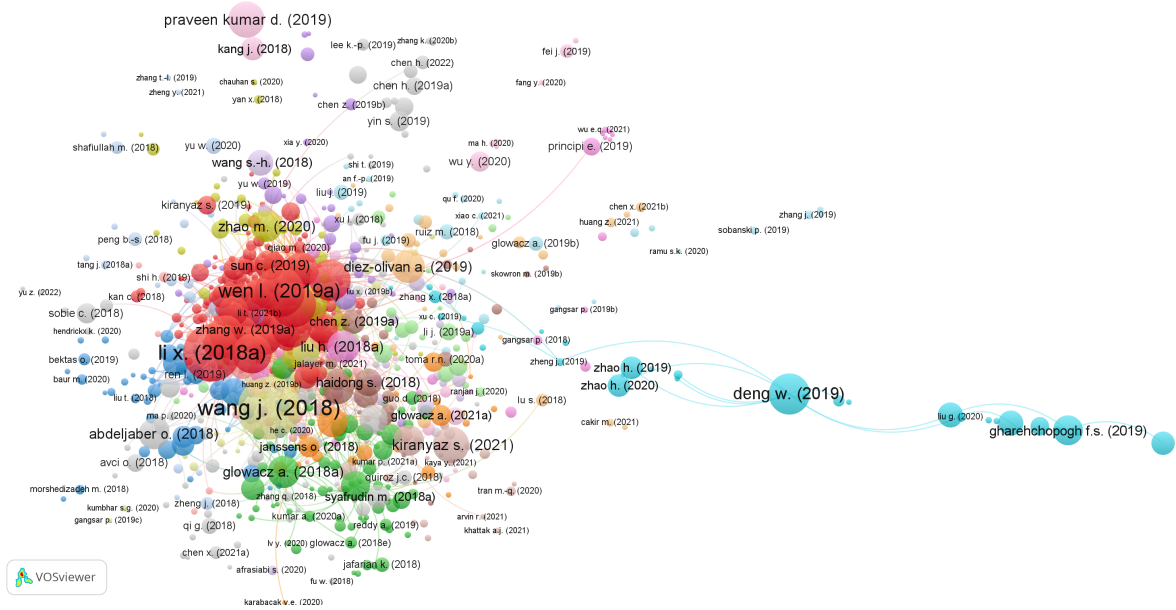
As Figuras 2, 3 e 4 ilustram os mapas de autores, artigos clusterizados e os principais países publicados, definindo o como critério de peso de interconexão como o número de citações elaborados para o autor e o artigo assim como o número de publicações para o país.

Figura 2 – Mapa de Autores referenciados.



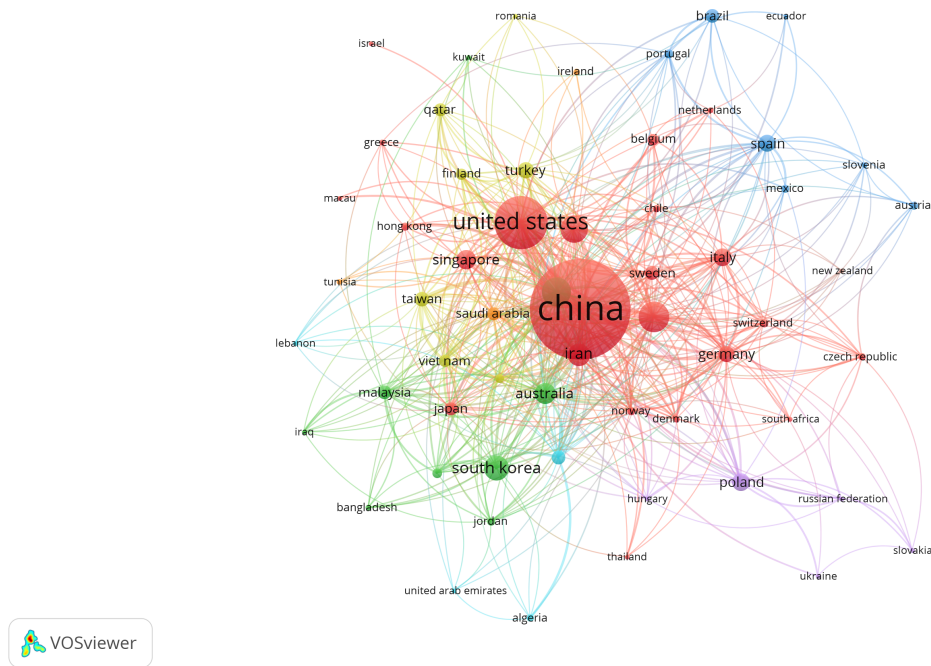
Fonte: O Autor (2023).

Figura 3 – Mapa de Artigos referenciados.



Fonte: O Autor (2023).

Figura 4 – Mapa de Artigos referenciados.



Fonte: O Autor (2023).

Sete destes artigos que mais destacaram-se devido ao maior número de citações tiveram sua resenha corteja a sub-sessão de trabalhos relacionados, dissecando os métodos utilizados e as métricas obtidas.

Visando determinar as principais tecnologias para inferência dos sistemas de detecção de falhas em motores elétricos, os dados foram clusterizados levando em conta as palavras-chaves das publicações, ilustradas na Figura 5 e 6.





(PCA), Redes Convolucionais (CNN), Máquina de Vetor de Suporte (SVM), Florestas Aleatórias, Redes Neurais Recorrentes (RNN), Auto Encoders e Adaboost.

### **2.2.1 Trabalhos Relacionados**

Esta subseção tem por objetivo descrever os principais artigos determinado pela pesquisa bibliométrica, detalhando as técnicas e métodos utilizados para as classificações de falhas em motores e seus respectivos resultados.

#### **2.2.1.1 Deep learning for smart manufacturing: Methods and applications**

Artigo publicado no Journal of Manufacturing Systems por Jinjiang Wang em 2018, foi estabelecido como o artigo mais referenciado entre as base SCOPUS a partir da pesquisa bibliométrica realizada. Seu principal objetivo é contrastar as principais diferenças entre os modelos de aprendizagem de máquinas tradicionais e profundos para aplicações industriais e elaborar através de uma revisão literária as um parecer entre as principais técnicas de deep learning utilizadas para inspeção de qualidade do maquinário, diagnóstico e análise preditiva de falhas em maquinários.

Wang et al. (2018) estabelece como principal diferença entre os modelos de aprendizagem de máquinas clássica e profunda em relação a presença integral da etapa de engenharia de características em conjunto a elaboração do modelo inteligente. Definindo os modelos clássicos como um processo de extração de características explicitado, na qual a determinação das grandezas relevantes é feita a partir de um domínio na área de conhecimento para posteriormente, utilizar essas características extraídas como entradas ao modelo inteligente, já os modelos profundos, segundo Wang et al. (2018) também aprendem uma maneira de representar as características mais relevantes dos dados em conjunto com a etapa de aprendizado do modelo como um todo.

Com o aumento crescente dos conjuntos de dados devido ao contexto de big data, os modelos de aprendizagem profunda estão se destacando por facilitar o processo de aprendizagem uma vez que removem a etapa de engenharia de características apesar de tornar o modelo em questão computacionalmente mais custoso.

A partir do contraste elaborado Wang et al. (2018) explicita as quatro principais técnicas de aprendizagem profundas aplicada no contexto industrial assim como fornece uma breve explicação de seu funcionamento. Os principais modelos são as Redes CNN, RBM, Auto Encoders e Redes Recorrentes LSTM

Com os modelos devidamente explicitado, Wang et al. (2018) estabelece as principais técnicas utilizadas ao longo de 40 artigos para inspeção de qualidade em maquinário, diagnóstico e análise preditiva de falhas em maquinários. Para esta monografia foi optado por reter atenção exclusivamente para as pesquisas de diagnóstico de falhas em maquinários devido uma maior similaridade na abordagem ao objeto de estudo.

Define-se o diagnóstico de falha em máquina rotativas a abordagem para monitorar as condições de maquinário e identificar falhas incipientes, diagnosticando as principais causas de origem para incorporar essas informações para o centro de controle e produção de manufatura.

Com uma vasta quantidade de técnicas de deep learning utilizadas para investigar o diagnóstico de falhas em máquina destacam as CNN inferindo imagens de espectrogramas dos canais de vibração de motores e com a maior métrica performática os Auto Encoders, utilizados como redutores de dimensionalidade de um sistema com características extraídas previamente, essas sendo, o valor médio, desvio padrão, raiz quadrada, curtose, fator de forma, fator de crista, fator de impulso e densidade espectral. Resultando no modelo como de melhor performance (99.68%) nas literaturas abordadas, concluindo que modelos de deep learning adquirem em média melhores resultados que modelos tradicionais de aprendizado de máquinas em termos de *accuracy* de classificação.

#### 2.2.1.2 Remaining Useful Life Estimation in Prognostics Using Deep Convolution Neural Networks

Publicado em 2018 pela Reliability Engineering and System Safety, Xiang Li, Qian Ding e Jian-Qiao Sun desenvolveram uma pesquisa para avaliar as técnicas de aprendizado de máquinas mais adequadas para predição da vida útil restante do motor aeronáutico, através da utilização da base de dados elaborada referente ao modelo de simulação de sistema de aeropropulsão modular comercial (C-MAPSS) publicada pela NASA.

A base de dados é subdivida em quatro datasets compostos de séries temporais multivariadas provenientes de 21 sensores. Cada dataset é dividido, por sua vez, em dois grandes conjuntos, segmentando os dados referentes à etapa de treinamento e da a etapa de teste dos modelos. Os datasets incluem amostras dos sensores de diversas turbinas operando até falha sob diferentes condições de operações e tipos de falhas obtidas

Li, Ding e Sun (2018) comparou as técnicas de redes neurais de camada únicas composta por 500 neurônios, redes neurais MLP com quatro camadas ocultas, com respectivamente, 500, 400, 300 e 100 neurônios. As Redes Recorrentes Neurais simples (RNN) foram pensadas como

detentoras de 5 camadas recorrentes, assim como as Redes Recorrentes LSTM. Já as Redes de Convolucionais profundas (DCNN) tiveram testadas a composição por apenas 1 camada convolucional até 7 camadas. Os modelos foram avaliados quanto ao valor de Raiz quadrada do erro-médio (RMSE) da estimativa de vida útil restante do maquinário utilizada como função custo do treinamento, e o tempo de treinamento obtido.

Para todos os modelos, os dados foram pré-processados utilizando a segmentação sem sobreposição por janela temporal para 30 amostras. Após a reestruturação dos dados, os mesmos passaram por uma normalização min-max tornando o sinal adequado para o treinamento dos modelos.

Os resultados dos experimentos indicaram o uso de redes convolucionais profundas como mais adequadas para a minimização do RSME, obtendo o valor médio de 17.73. Acompanhado pelo modelo RNN com 18.71 e o modelo LSTM com 18.89, de forma a sintetizar os valores obtidos a Tabela 1, explicita as métricas avaliadas pelos modelos testados para os 4 sub-datasets. Desta maneira o autor destaca o uso de redes DCNN para a predição do índice de vida útil restante de aero-turbinas, embora a carga computacional de modelos de aprendizagem profundos seja maior, as métricas performáticas para esses modelos são significativamente maiores.

Tabela 1 – Métricas em Li, Ding e Sun (2018).

	<b>RNA SLP</b>	<b>RNA MLP</b>	<b>RNN</b>	<b>LSTM</b>	<b>DCNN</b>
RSME	20.36	19.10	18.71	18.89	17.73
Tempo de Treinamento	133.92	178.57	647.32	2433.1	714.28

Fonte: Li, Ding e Sun (2018)

### 2.2.1.3 A New Deep Transfer Learning Based on Sparse Auto-Encoder for Fault Diagnosis

O Artigo publicado em 2017 no Journal IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS. Wen, Gao e Li (2017) propõe um novo modelo de aprendizado por transferência aplicados em Autoencoders utilizados em diagnósticos de falhas em motores de indução trifásicos, utilizando para aprendizado dados provenientes da base *Case Western Reserve University*.

Inicialmente Wen, Gao e Li (2017) explicita a utilização de métodos de aprendizagem profunda em detrimento do aprendizado de máquinas tradicional devido a maior dependência da performance do segundo método à etapa engenharia de características, o que acaba dificultando a elaboração do modelo tradicional.

É proposto assim um modelo de Auto Encoder Esparso (SAE) de três camadas em conjunto com um classificador *softmax*, baseado em aprendizado por transferência, ou seja, que o modelo base é pré-treinado em um conjunto de dados fonte e reutiliza-se os pesos estabelecidos para a inicialização dos mesmos em um segundo modelo que será treinado em um outro conjunto de dados pós-transferência.

Como o conjunto da base de dados pós-transferência, em questão não possui rótulos para treinamento, o modelo de *transfer learning* proposto é a partir de um SAE estabelecido para extrair as características latentes do conjunto pré e pós transferência, acompanhados por um ajuste fino feito em toda rede neural utilizando os rótulos da base de dados com a seguinte função de perda, descrita pela equação 2.1:

$$J(\theta) = Loss(Y_d, Y_o) + \mu MMD(\sigma_1, \sigma_2) \quad (2.1)$$

Onde o termo *Loss* é gerado pela diferença entre os valores obtidos na saída do classificador e os valores esperados e o segundo termo MMD produz a penalidade gerada pela discrepância entre as distribuições das características latentes da base de dados fonte e da base pós transferência. O MMD estima distancias não-paramétricas, evitando cálculos densidades intermediárias das distribuições normalizadas. Sua definição é dada pela Equação 2.2:

$$MMD(X_s, X_T) = \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} \phi(x_i^s) - \frac{1}{N_T} \sum_{i=1}^{N_T} \phi(x_i^t) \right\|_H \quad (2.2)$$

O sistema proposto foi comparado com técnicas tradicionais como DBN, SVM, ANN e metaclassificadores. Obtendo um resultado significativamente superior de accuracy, com a tabela abaixo ilustra. Indicando a prevalência de técnicas de aprendizado profundo devido as melhores métricas performáticas obtidas.

Tabela 2 – Métricas em Wen, Gao e Li (2017).

Métrica	Modelo Proposto	DBN	SVM	RNA MLP	Bayesiano	Bagging	Boosting
Accuracy	99.82%	87.45%	57.90%	67.70%	58.30%	46.65%	56.90%

Fonte: Wen, Gao e Li (2017)

#### 2.2.1.4 Early fault diagnosis of bearing and stator faults of the single-phase induction motor using acoustic signals

Publicada pela Measurement em 2017, Glowacz et al. (2018) descreve um sistema para diagnóstico de falhas principiantes em motores de indução monofásicos para três modos operacionais: normalizado, falha em rolamento e falha em rolamento em conjunto com curto-circuito

em bobinas do estator. Utilizando um método próprio de extração de características, denominado MSAF-20-MULTIEXPANDED (método de seleção das frequências de amplitudes - Multiexpandido), e comportamento o resultado de três modelos de aprendizado de máquinas independentes, classificadores Nearest Neighbour (NN), Classificadores Nearest Mean (NM) e modelos Gaussian Mixture (GMM).

As medidas utilizadas pelo sistema proposto foram sinais acústicos adquiridos por dois microfones omnidirecionais, relação sinal-ruído equivalente a 58dB e sensibilidade de 40dB e 34dB, respectivamente. Produzindo um arquivo wave pcm de resolução de 16-bits, 44.1KHz de frequência de amostragem e apenas um canal. Posteriormente esses dados foram processados via segmentação das 32768 amostras, equivalentes a 0.74s em 44100Hz e tiveram seu espectrograma extraído via FFT.

A partir dos espectrogramas, a etapa de extração de características seguiu o modelo proposto - MSAF-20-MULTIEXPANDED - o qual é formado por cinco etapas. a partir dos espectrogramas de cada estado do motor monofásico se calcula as diferença entre suas frequências, para definir seus valores absolutos utilizados na seleção das 20 maiores amplitudes e definindo as raias de frequência mais relevantes do conjunto de treinamento como valores a utilizar como característica para alimentar o modelo de aprendizado de máquinas. Com o vetor de características elaborado três técnicas de classificação inteligentes foram testadas, os modelos NN, NM e GMM, treinados com 10% dos conjunto de dados e os 90% restantes para testá-los, obtendo valores médios de *accuracy* sintetizados na tabela 3

Tabela 3 – Métricas em Glowacz et al. (2018).

<b>Métrica</b>	<b>Nearest Neighbour</b>	<b>Nearest Mean</b>	<b>Gaussian Mixture</b>
Accuracy Médio	86.96%	92.83%	74.00%

Fonte: Glowacz et al. (2018)

#### 2.2.1.5 Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization

Publicada pela Mechanical Systems and Signal Processing em 2018 por Jia et al. (2018) descreve um método para utilização de redes convolucionais profundas em problemas de classificação desbalanceadas para falhas de maquinários. Salienta-se que grande parte das publicações na área não consideram o balanceamento das bases de dados utilizadas, o que muitas vezes pode ser um fator falseador dos resultados obtidos. A DNN proposta é validada por três datasets com diferentes taxas de desbalanço (nula, baixa e alta) e comparado com modelos CNN

tradicionais

Por tratar-se de uma rede convolucional unidimensional o modelo é alimentado com sinais de vibração sem qualquer extração de característica, permitindo o modelo através das etapas de convolução de *pooling* definir as características mais significativas dos sinais analisados. O modelo proposto, ilustrado pela, é formado por dois intercalamentos de camadas convolucionais e poolings seguindo por uma rede MLP de três camadas. O classificador interno é formado por uma camada de saída *softmax* e duas camadas normalizadas com funções de ativação ReLU, escolhida para evitar desaparecimento de gradientes junto com a normalização de seus pesos aplicadas para melhorar a otimização dos parâmetros da DNCNN.

Para reduzir o impacto do desbalance do *dataset* utilizado no treinamento, optou-se por aumentar o peso do custo das classes minoritárias proporcionalmente ao seu grau de desbalanceamento, ressaltando seu efeito no processo de aprendizagem da Rede Convolucional.

Ao sistema proposto a Tabela 4 sintetiza os resultados obtidos para as bases de dados balanceadas (A), com baixo desbalanço (B) e altamente desbalanceada (C) para os modelos o modelo proposto (DCNN), modelos convolucionais sigmóide (S-CNN) e CNN utilizando ReLU (R-CNN), através da métrica de *Accuracy*.

Tabela 4 – *Accuracy* em Jia et al. (2018).

<b>Modelos</b>	<b>Dataset A</b>	<b>Dataset B</b>	<b>Dataset C</b>
S-CNN	96.48%	89.59%	74.46%
R-CNN	97.59%	95.52%	88.20%
DNCNN	99.22%	98.19%	95.52%

Fonte: Jia et al. (2018)

#### 2.2.1.6 Multiscale Convolutional Neural Networks for Fault Diagnosis of Wind Turbine Gearbox

Jiang et al. (2018) publicou na IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS um artigo que propôs um método de diagnóstico de falhas de transmissão de aeroturbinas através de redes convolucionais multiescalares (MSCNN). A validação do modelo foi feita por um experimento realizado pelos autores em ambiente de teste, descrevendo 8 modos de operações em 5 velocidades diferentes, com dois valores de cargas aplicadas. A Tabela 5 detalha os casos de testes.

O estudo buscou apreender automaticamente as features significativa ao processo de aprendizagem diretamente dos sinais de vibrações do maquinário, sem qualquer processamento anterior e por isso o método de aprendizagem de máquina optado foram as redes neurais convo-

Tabela 5 – Experimentos em Jiang et al. (2018).

<b>Rótulo</b>	<b>Descrição da Falha A</b>	<b>Velocidade do Motor(rpm)</b>	<b>Carga(KW)</b>
1	Normal	200/300/400/500/600	1.5/3
2	Engrenagem Quebrada	200/300/400/500/600	1.5/3
3	Engrenagem Lascada	200/300/400/500/600	1.5/3
4	Engrenagem Esburacada	200/300/400/500/600	1.5/3
5	Pista interna do rolamento defeituosa	200/300/400/500/600	1.5/3
6	Pista externa do rolamento defeituosa	200/300/400/500/600	1.5/3
7	Esfera de rolamento defeituosa	200/300/400/500/600	1.5/3
8	Desbalanço no eixo	200/300/400/500/600	1.5/3

Fonte: Jiang et al. (2018)

lucionais (CNN), no entanto para extrair assinatura das falhas e capturar características multi-escalares dos sinais foi utilizada uma arquitetura MSCNN para incorporar paralelamente essas características de diferentes escalas temporais automaticamente.

Os dados experimentais foram obtidos com frequência de aquisição equivalente a 10 kHz durante 2 segundos. As escalas utilizadas na MSCNN foram de: 1, 0.5 e 0.33. Equivalente à extração de características em janelas de 2000 ms, 1000 ms e 600 ms.

O Modelo proposto foi comparado com uma rede convolucional padrão (CNN) e três modelos de SVM referência, com extração via entropia multiescalar (MSE-SVM), decomposição de wavelet package (WPD-SVM) e decomposição empírica (EMD-SVM), com sua performance (F1-Score) sintetizada pela tabela 6.

Tabela 6 – Métricas em Jiang et al. (2018).

<b>Modelos</b>	<b>MSE-SVM</b>	<b>WPD-SVM</b>	<b>EMD-SVM</b>	<b>CNN</b>	<b>MSCNN</b>
F1-Score	82.88%	84.81%	72.71%	92.11%	98.53%

Fonte: Jiang et al. (2018)

### 2.2.1.7 Highly-Accurate Machine Fault Diagnosis Using Deep Transfer Learning

Publicado em 2018, Shao et al. (2018) propõe um modelo para diagnósticos de falhas em maquinário baseado em transferência de aprendizado de redes convolucionais baseado na rede VGG-16. O modelo produzido foi testado com três bases de dados, as quais armazenavam séries temporais das vibrações de motores de indução, caixas de transmissão e rolamentos, de tamanhos 6000, 9000 e 5000 amostras respectivamente, alcançando uma performance dentro do estado da arte de cada um dos datasets.

Os sinais de vibração foram pré-processados de forma a transformar o domínio temporal dos dados em um espectrograma. Gerando assim, uma imagem da evolução da distribuição das frequências ao longo do tempo de operação do maquinário. Após isso, os espectrogramas alimentavam a etapa de aprendizagem do modelo inteligente.

A rede neural composta por Shao et al. (2018) foi proposta seguindo a metodologia de aprendizagem por transferência, o que significa a reutilização dos parâmetros um modelo base, nesse caso VGG-16, como maneira de inicializar os do modelo alvo. Como o modelo pré-treinado (VGG-16) não foi pensado para utilização de análise espectrogramas, seus parâmetros foram preservados como valor inicial do modelo proposto, com apenas a fixação dos parâmetros das três primeiras camadas convolucionais, enquanto outros parâmetros teriam seu ajuste fino realizado ao longo do treinamento. Para os testes do modelo, comparou-se sua performance e custo computacional com um modelo de mesma arquitetura, mas que teve seu treinamento feito sem a utilização de um modelo pré-treinado como base. A Tabela 7 ilustra os resultados obtidos para cada um das bases de dados.

Tabela 7 – Métricas em Shao et al. (2018).

<b>Modelos</b>	<b>Dataset A</b>	<b>Dataset B</b>	<b>Dataset C</b>
CNN	98.00%	97.21%	97.50%
VGG-16+CNN	100%	99.78%	99.52%

Fonte: Shao et al. (2018)

Tabela 8 – Tempo de Treinamento em Shao et al. (2018).

<b>Modelos</b>	<b>Dataset A</b>	<b>Dataset B</b>	<b>Dataset C</b>
CNN	1317s	790s	1793s
VGG-16+CNN	397s	215s	536s

Fonte: Shao et al. (2018)

#### 2.2.1.8 Síntese de pesquisa biométrica

A partir das referencias estudadas pode-se perceber a predominância do uso de técnicas de aprendizado profundo para o diagnósticos de falhas em motores de indução, assim como a principal métrica de avaliação da performance do modelo proposto *accuracy* pertencendo a faixas de valores entre 92.83% e 99.82%, como exhibe a Tabela 9.

Tabela 9 – Resultados Médios destacados pela pesquisa bibliométrica.

<b>Artigos</b>	<b>Técnica de Aprendizado</b>	<b>Resultado</b>
Wang et al. (2018)	Auto Encoder Empilhado	<i>Accuracy</i> : 99.68%
Li, Ding e Sun (2018)	DCNN	RMSE:17.73
Wen, Gao e Li (2017)	Auto Encoder Esparso	<i>Accuracy</i> : 99.82%
Glowacz et al. (2018)	<i>Nearest Mean</i>	<i>Accuracy</i> :92.83%
Jia et al. (2018)	DCNN	<i>Accuracy</i> :99.22%
Jiang et al. (2018)	MSCNN	F1-Score:98.53%
Shao et al. (2018)	CNN	<i>Accuracy</i> :99.76%

Fonte: Shao et al. (2018)



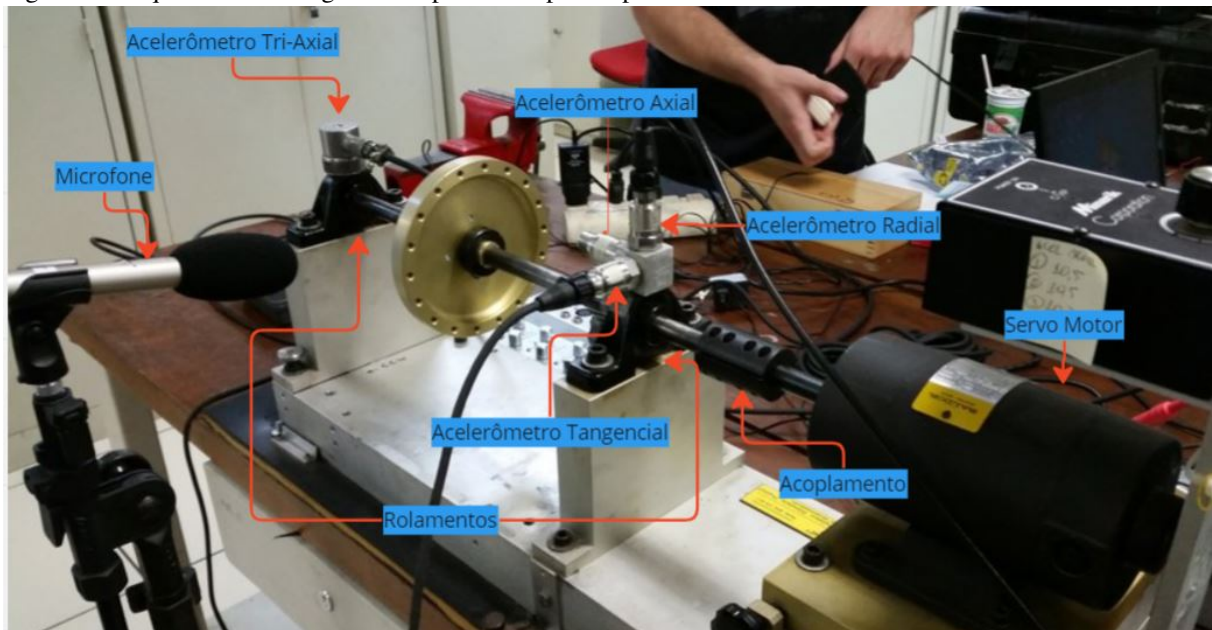
### 3 METODOLOGIA

#### 3.1 BANCO DE DADOS

Banco de dados utilizados para este trabalho - MaFaulDa - foi publicada em 2016 pelo Laboratório de Sinais, Multimídia e Telecomunicações pertencente a escola politécnica da Universidade Federal do Rio de Janeiro (UFRJ). Composto por séries temporais multivariáveis adquiridas por acelerômetros, microfones e tacômetros, dispostos em um simulador de falhas de maquinário tipo *alignment/balance vibration trainer* (ABVT) da fabricante americana *SpectraQuest*, responsável por simular a dinâmica de motores com dois rolamentos de acoplamento no eixo, permitindo o estudo de diversas falhas como desbalanço de cargas, desalinhamento de eixos e falhas relacionados ao rolamento de maneira controlada.

O Experimento montado para produção do banco de dados, descrito pela Figura 7, consiste na medida da vibração causada nos dois rolamentos de acoplamento do sistema, através de quatro acelerômetros: três mono-axial e um triaxial. Os acelerômetros mono-axiais – modelo 601A01 da IMI Sensor - foram dispostos de forma a capturar o sinal de vibração nas direções axiais, radiais e tangenciais de um rolamento enquanto o sensor triaxial - modelo 604B3 - garantia a medida das vibrações no outro rolamento. Foram medidos também a grandeza da velocidade angular da máquina através de um tacômetro e o ruído acústica através de um microfone. Os acelerômetros utilizados tem sua sensibilidade equivalente a  $100 \text{ mV/g} \pm 20\%$ , faixa de frequência como  $0.27\text{-}10.000 \text{ Hz} \pm 3 \text{ dB}$  para 601A01 e  $0.5 \text{ Hz a } 5.000 \text{ Hz} \pm 3 \text{ dB}$  para 604B3, ambos detém suas faixa mensurável como  $\pm 50 \text{ g} \pm 490 \frac{\text{m}}{\text{s}^2}$ .

Figura 7 – Esquema de montagem do experimento para captura dos dados.



Fonte: Adaptado livremente de Khan, Hwang e Kim (2021).

Os ensaios foram realizados com uma faixa de velocidade de operação de 737 a 3686 rpm com resolução de 60 rpm. Os 49 cenários experimentais envolveram as seguintes condições:

- *Operação Normal*: Essa classe representa o sistema sobre condição normal de operação, sem qualquer falha;
- *Desbalanço de Carga*: simula diferentes graus de desbalanço nas cargas aplicadas ao rotor dentro da faixa de 6 g; 10 g; 15 g; 20 g; 25 g; 30 g; e 35 g. No entanto para cargas a partir de 30g notou-se uma impraticabilidade na aquisição de frequências acima de 3300 rpm, limitando 44 cenários de ensaios. Ao total o banco de dados é formado por 333 cenários de desbalanço de carga;
- *Desalinhamento Horizontal*: essa falha foi realizada deslocando horizontalmente o eixo do motor 0.5mm, 1.0mm, 1.5mm e 2.0mm, resultando 197 diferentes cenários para essa classe de falha;
- *Desalinhamento Vertical*: falha gerada através do deslocamento vertical do eixo do motor nos valores de 0,51mm; 0,63mm; 1,27mm; 1,4 mm; 1,78mm e 1,9 mm. Resultando em 301 cenários de ensaios dentro das velocidades de operação;
- *Falhas de Rolamento*: segmentadas em em três tipos de falhas referente ao local de ocorrência, podendo ser no anel externo, interno ou nas esferas deslizantes. Os ensaios

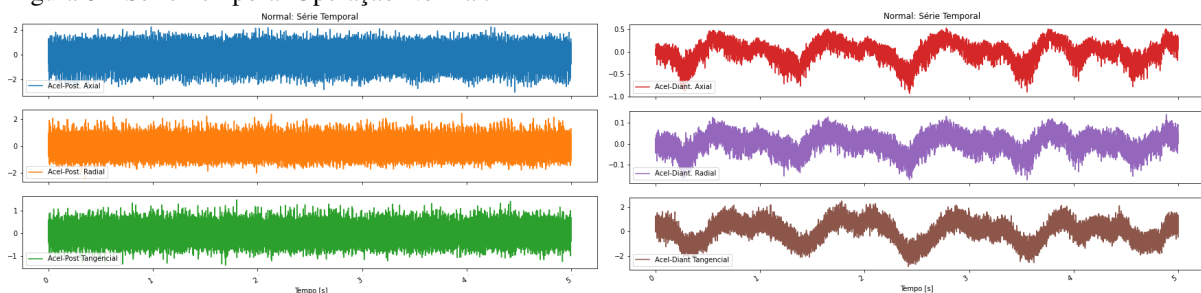
foram realizados para cada um dos dois rolamentos do eixo de transmissão. Resultando em 558 cenários para defeitos do rolamento posterior e 513 ensaios para o rolamento dianteiro.

Totalizando 1951 experimentos capturados através de dois módulos de aquisição analógica de 4 canais NI 9234 da *National Instruments* para cada um dos 8 sinais em uma frequência de aquisição de aproximadamente 50kHz durante um intervalo de 5 s, produzindo um *frame* de dados de 250000 elementos para cada ensaios.

### 3.2 ANÁLISE EXPLORATÓRIA DOS DADOS

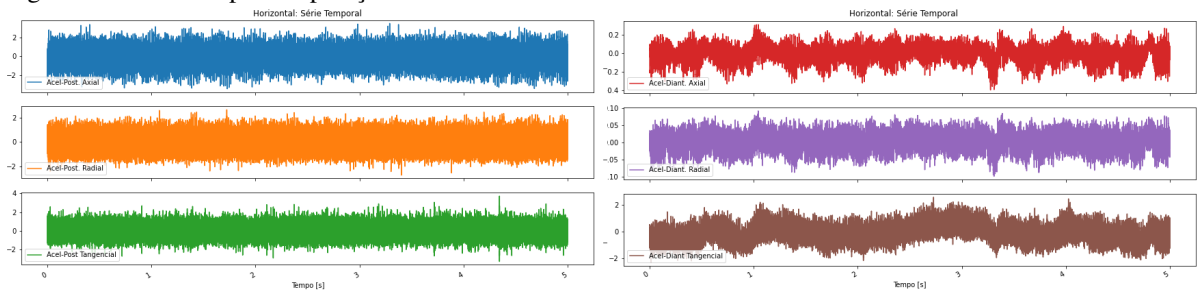
Este subcapítulo tem como função determinar as características que serão extraídas das séries temporais da base de dados e que serão utilizadas como entrada do modelo proposto. Essa definição é feita através da análise exploratória e testes estatísticos que levantarão possíveis hipóteses sobre as causas dos fenômenos observados e a definição das principais métricas avaliadas para a etapa da engenharia de característica. Para o banco de dados utilizado realizou-se uma análise exploratória inicial como forma de descrever e pontuar visualmente características dos ensaios. Optou-se por analisar os ensaios em velocidade operativa de 60 Hz, ao longo de todo os 5 segundo capturados. As Figuras 8 a 13 descrevem as curvas de tensão proveniente dos seis sinais de vibração capturados do rolamento dianteiro e posterior do esquemático exibido na figura 7, para cada uma dos modos de operação.

Figura 8 – Série Temporal Operação Normal.



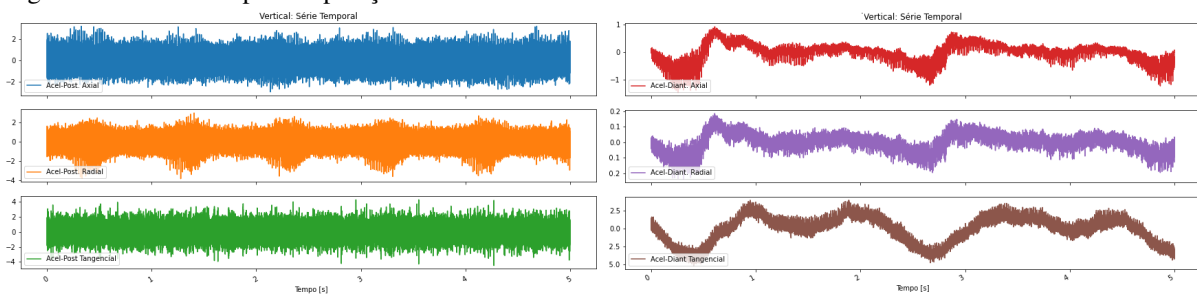
Fonte: O Autor (2023).

Figura 9 – Série Temporal Operação com Desalinhamento Horizontal.



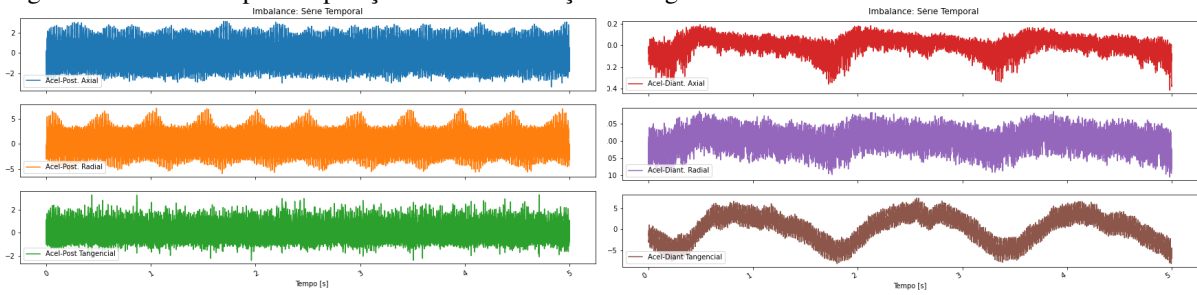
Fonte: O Autor (2023).

Figura 10 – Série Temporal Operação com Desalinhamento Vertical.



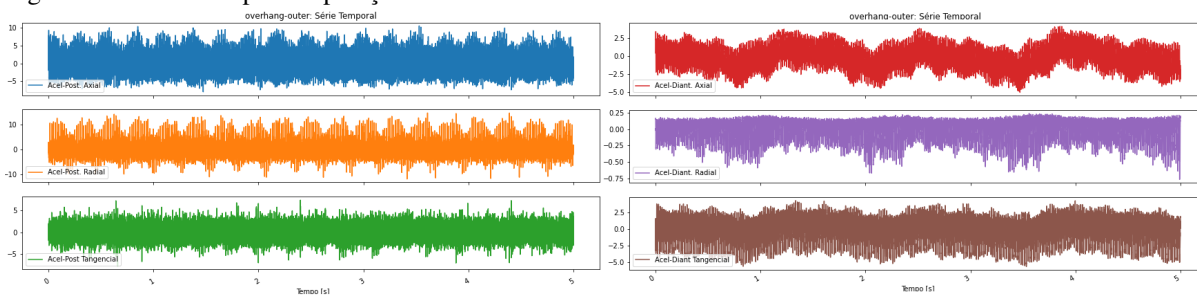
Fonte: O Autor (2023).

Figura 11 – Série Temporal Operação com Desbalanço de Carga.



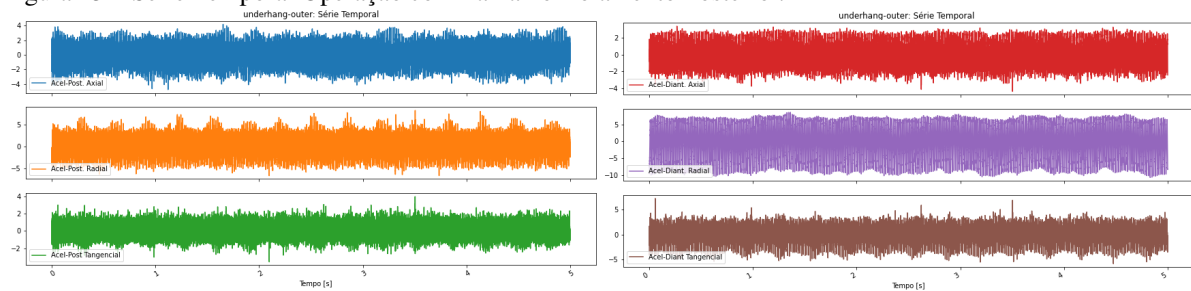
Fonte: O Autor (2023).

Figura 12 – Série Temporal Operação com Falha no Rolamento Dianteiro.



Fonte: O Autor (2023).

Figura 13 – Série Temporal Operação com Falha no Rolamento Posterior.



Fonte: O Autor (2023).

A partir das Figuras, nota-se uma maior segmentação comportamental entre as componentes oscilatórias provenientes do rolamento posterior ao dianteiro, destacando uma maior variabilidade nas dinâmicas adquiridas para o último caso, com componentes de frequências inferiores a 60 Hz. Oscilações Radiais dianteiras obtiveram uma amplitude dez vezes menores em relação as outras componentes. Como uma forma mais adequada de extrair informações de maneira mais adequada, optou-se por elaborar rotinas para determinar as métricas estatísticas de valores máximos, mínimos, RMS e RMS médio para o sinal segmentado em janelas deslizantes de 100 ms com *overlapping* em 10%, definidos através de testes preliminares, descritos pelas Tabelas 10, 11 12e 13, respectivamente.

Tabela 10 – Valores máximos de oscilações.

Modo de Operação	Vibração do Rolamento Posterior(V)			Vibração do Rolamento Dianteiro(V)		
	Axial	Radial	Tangencial	Axial	Radial	Tangencial
Normal	2,28	2,45	1,46	0,53	0,13	2,51
Desbalanço de Carga	3,09	7,218	3,30	0,19	0,08	7,39
Desalinhamento Vertical	3,20	2,995	4,265	0,914	0,18	3,90
Desalinhamento Horizontal	3,49	2,65	3,70	0,310	0,09	2,58
Falha de Rolamento Posterior	4,19	8,19	3,94	3,37	8,60	7,28
Falha de Rolamento Dianteiro	10,4	14,7	7,44	4,15	0,23	4,30

Fonte: O Autor (2023)

Tabela 11 – Valores mínimos de oscilações.

Modo de Operação	Vibração do Rolamento Posterior(V)			Vibração do Rolamento Dianteiro(V)		
	Axial	Radial	Tangencial	Axial	Radial	Tangencial
Normal	-3,04	-2,45	-1,39	-0,93	-0,17	-2,86
Desbalanço de Carga	-3,31	-5,98	-2,41	-0,41	-0,10	-8,12
Desalinhamento Vertical	-2,95	-3,83	-4,47	-1,46	-0,24	-5,24
Desalinhamento Horizontal	-3,44	-2,71	-3,25	-0,40	-0,10	-2,30
Falha de Rolamento Posterior	-4,77	-6,72	-3,54	-4,36	-10,7	-5,85
Falha de Rolamento Dianteiro	-7,95	-11,9	-7,882	-5,00	-0,77	-5,69

Fonte: O Autor (2023)

Tabela 12 – Valores RMS de oscilações.

Modo de Operação	Vibração do Rolamento Posterior(V)			Vibração do Rolamento Dianteiro(V)		
	Axial	Radial	Tangencial	Axial	Radial	Tangencial
Normal	2,28	2,45	1,46	0,53	0,13	2,51
Desbalanço de Carga	3,09	7,218	3,30	0,19	0,08	7,39
Desalinhamento Vertical	3,20	2,995	4,265	0,914	0,18	3,90
Desalinhamento Horizontal	3,49	2,65	3,70	0,310	0,09	2,58
Falha de Rolamento Posterior	4,19	8,19	3,94	3,37	8,60	7,28
Falha de Rolamento Dianteiro	10,4	14,7	7,44	4,15	0,23	4,30

Fonte: O Autor (2023)

Tabela 13 – Valores RMS médio e desvio padrão das oscilações segmentadas.

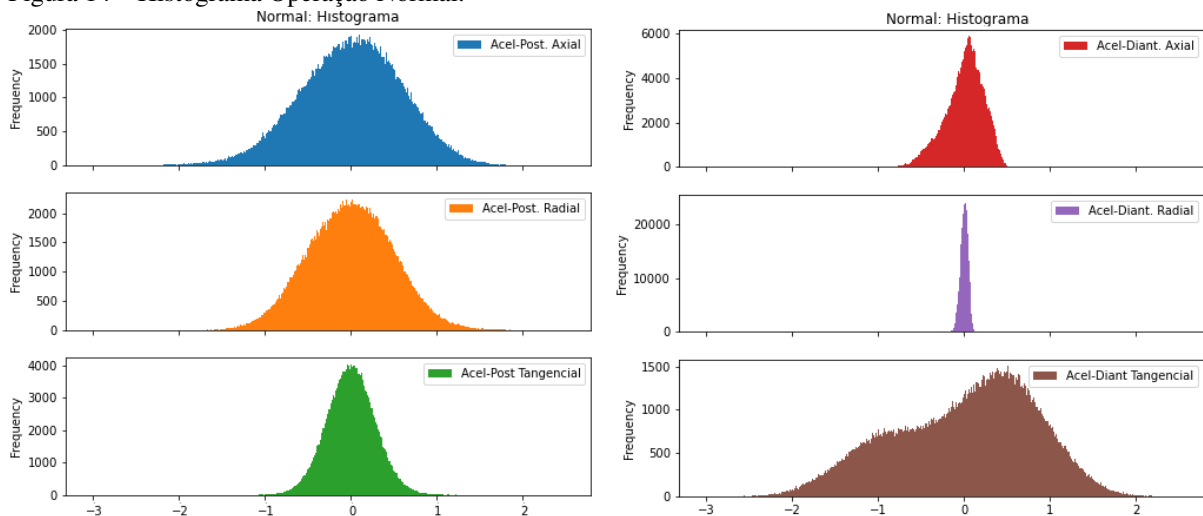
Modo de Operação	Vibração do Rolamento Posterior(V)			Vibração do Rolamento Dianteiro(V)		
	Axial	Radial	Tangencial	Axial	Radial	Tangencial
Normal	$0.59 \pm 0.05$	$0.50 \pm 0.05$	$0.28 \pm 0.03$	$0.20 \pm 0.10$	$0.04 \pm 0.02$	$0.75 \pm 0.34$
Desbalanço de Carga	$0.94 \pm 0.09$	$1.76 \pm 0.09$	$0.52 \pm 0.08$	$0.07 \pm 0.03$	$0.03 \pm 0.01$	$2.92 \pm 1.28$
Desalinhamento Vertical	$0.70 \pm 0.07$	$0.61 \pm 0.11$	$0.83 \pm 0.13$	$0.28 \pm 0.19$	$0.05 \pm 0.08$	$1.51 \pm 0.97$
Desalinhamento Horizontal	$0.77 \pm 0.08$	$0.57 \pm 0.03$	$0.57 \pm 0.06$	$0.08 \pm 0.02$	$0.03 \pm 0.01$	$0.61 \pm 0.14$
Falha de Rolamento Posterior	$1.12 \pm 0.12$	$1.86 \pm 0.12$	$0.68 \pm 0.08$	$1.38 \pm 0.14$	$5.35 \pm 0.43$	$1.48 \pm 0.14$
Falha de Rolamento Dianteiro	$2.45 \pm 0.54$	$2.34 \pm 0.28$	$1.23 \pm 0.14$	$1.49 \pm 0.33$	$0.15 \pm 0.04$	$1.67 \pm 0.31$

Fonte: O Autor (2023)

A partir das métricas torna-se claro uma intensificação das vibração dos rolamentos dianteiros para as Falhas de Rolamento. Desbalanço de Carga de maneira geral tende a intensificar a vibração geral do sistema quando sem falhas nos rolamentos e há um comportamento muito similar entre as falhas de desalinhamento vertical e horizontal em relação as amplitudes de intensidade vibracional.

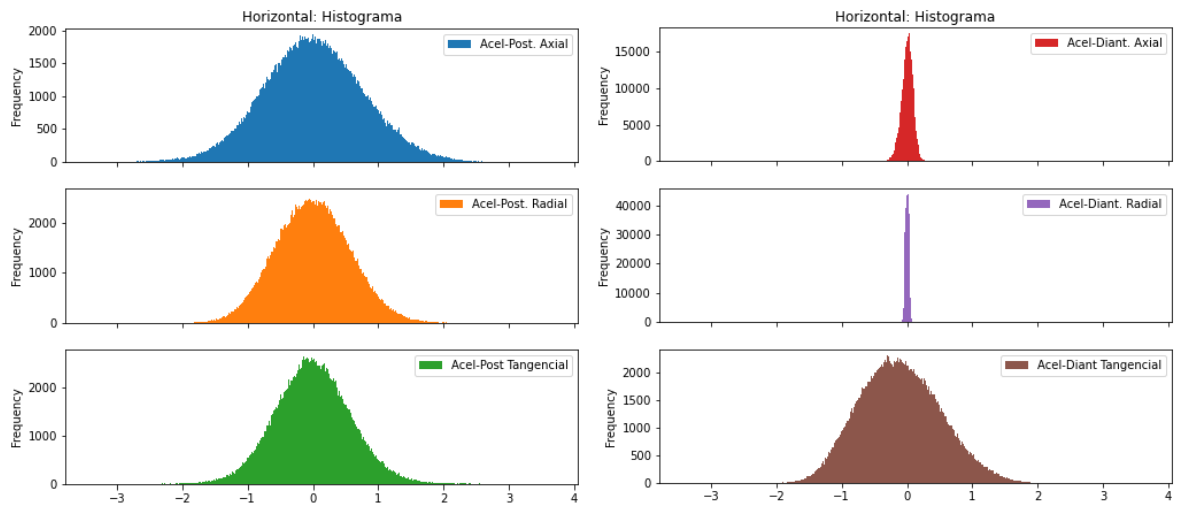
Ao visar analisar as curvas de vibração sob a ótica da dispersividade de sua faixa de valores, elaborou-se histogramas, definidos pelas Figuras (14) a (19).

Figura 14 – Histograma Operação Normal.



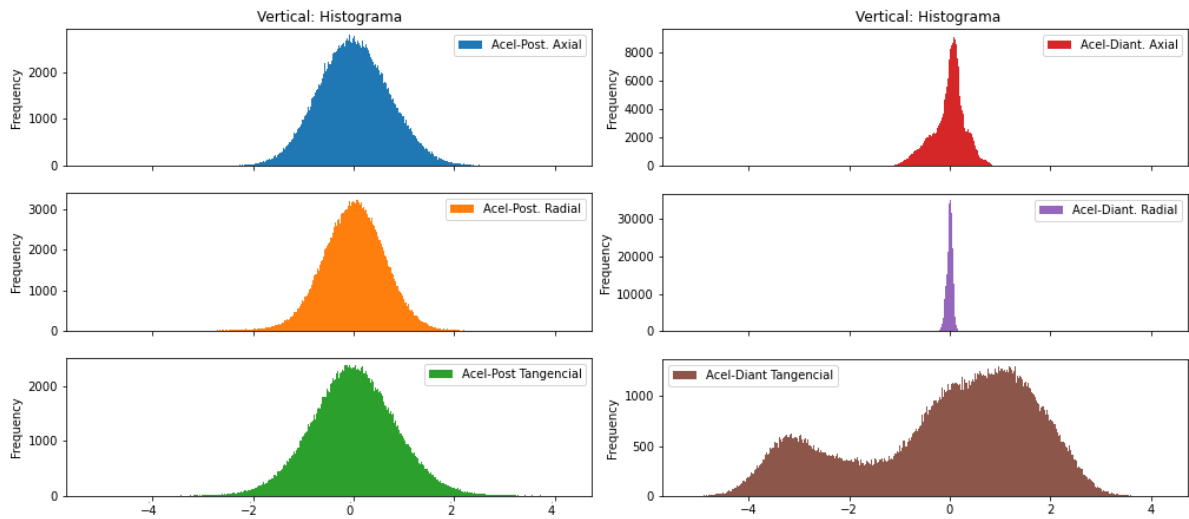
Fonte: O Autor (2023).

Figura 15 – Histograma Operação com Desalinhamento Horizontal.



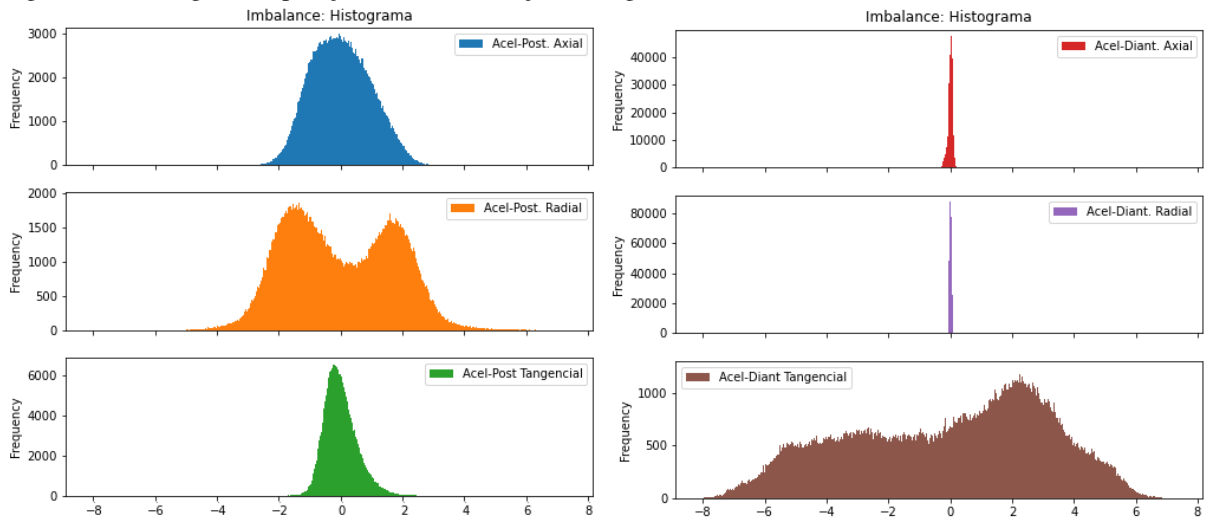
Fonte: O Autor (2023).

Figura 16 – Histograma Operação com Desalinhamento Vertical.



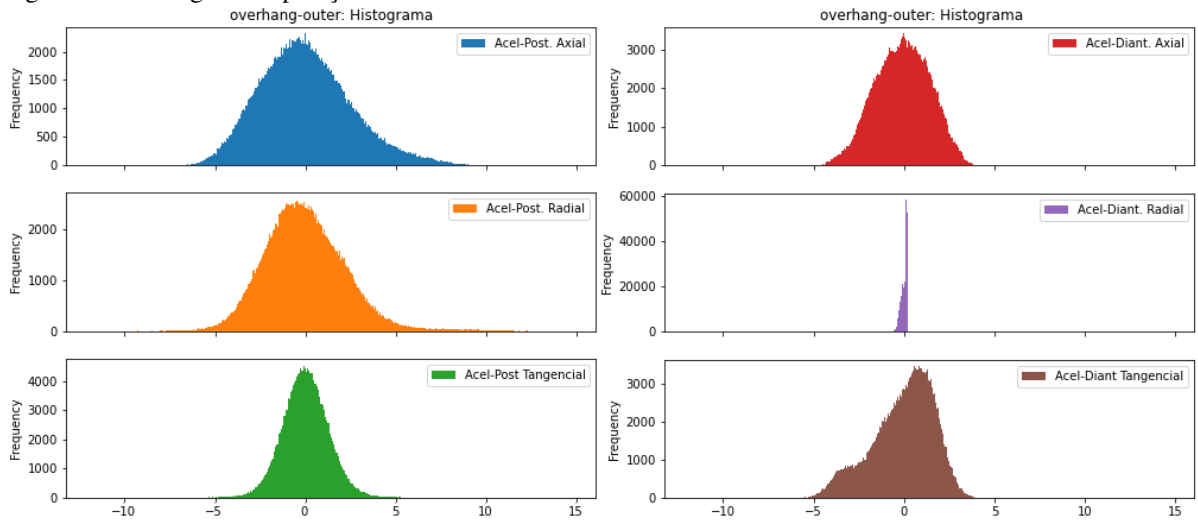
Fonte: O Autor (2023).

Figura 17 – Histograma Operação com Desbalanço de Carga.



Fonte: O Autor (2023).

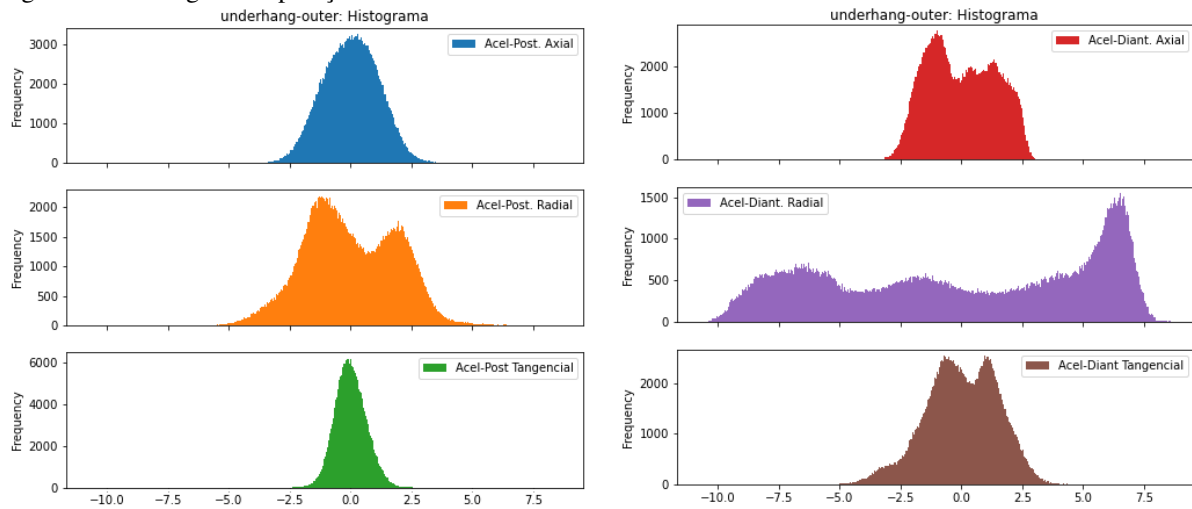
Figura 18 – Histograma Operação com Falha no Rolamento Dianteiro.



Fonte: O Autor (2023).



Figura 19 – Histograma Operação com Falha no Rolamento Posterior.



Fonte: O Autor (2023).

A partir das Figuras 14 a 19 infere-se uma tendência a distribuição normal para as vibrações no rolamento posterior, exceto a vibração radial em caso de desbalanço de carga e em falhas no rolamento posterior. As vibrações do rolamento dianteiro possuem uma distribuição mais concisa, trazendo uma curva mais aguda às vibrações axiais e radiais, exceto para casos de falhas alocadas nos rolamentos, sua componente tangencial apresentou uma distribuição esparsa, assim como as vibrações radiais para a falhas no rolamento posterior. A caracterização do grau de achatamento da curva é determinada ao medir a curtose no sinal, a Tabela 14 descreve a curtose para os 5 s dos sinais.

Tabela 14 – Curtose da distribuição das oscilações.

Modo de Operação	Vibração do Rolamento Posterior			Vibração do Rolamento Dianteiro		
	Axial	Radial	Tangencial	Axial	Radial	Tangencial
Normal	0,13	0,01	0,21	0,14	-0,10	-0,55
Desbalanço de Carga	-0,52	-0,79	0,81	1,31	-0,39	-0,94
Desalinhamento Vertical	0,02	0,69	0,30	0,578	0,17	-0,67
Desalinhamento Horizontal	0,19	-0,04	0,28	0,27	-0,59	-0,24
Falha de Rolamento Posterior	-0,33	-0,56	0,19	-1,0	-1,39	-0,16
Falha de Rolamento Dianteiro	0,16	2,2	0,50	-0,45	0,31	-0,20

Fonte: O Autor (2023)

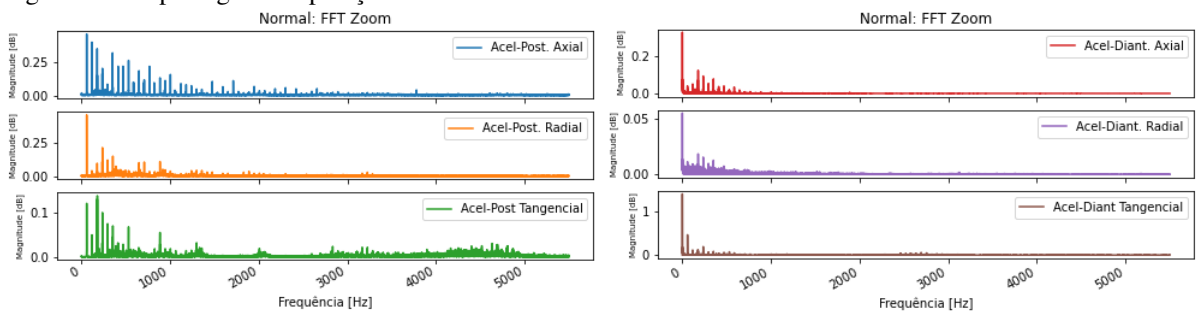
Os valores de curtose ilustram uma maior atenuação para operação normativa e com desalinhamento horizontal. Enquanto as Falhas em rolamentos aumentam as taxas de curtose, como era previsto através da visualização da distribuição dos dados.

Para estabelecer a presença de sazonalidade das séries temporais foi feito o teste estatístico de Dickey-Fuller, os resultados para todos os casos acataram a hipótese nula, assim todas as

curvas analisadas foram descritas com raiz unitárias, garantindo um comportamento estocástico para os sinais captados, necessidade da aplicação de modelos inteligentes para conseguir detectar padrões de forma mais robusta que métodos de classificação estatísticos tradicionais.

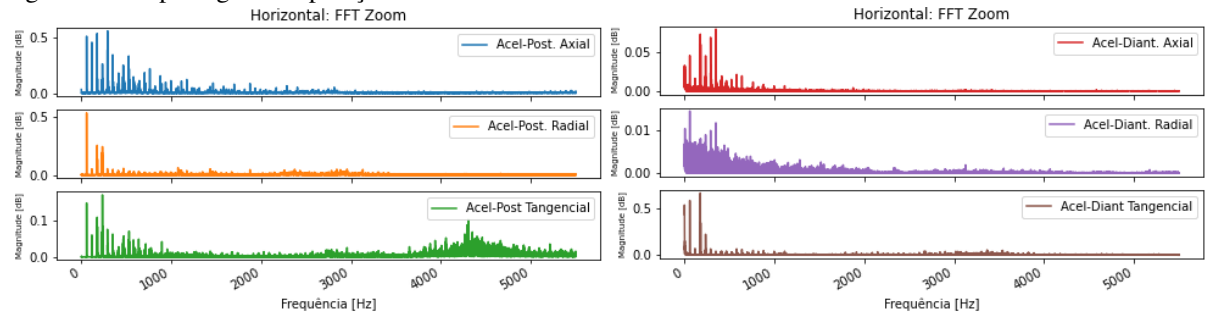
Uma segunda abordagem para avaliar os sinais captados é através da sua representação no domínio da frequência através do algoritmo de Transformada Rápida de Fourier (STFT), pode-se determinar os respectivos espectrogramas nas Figuras 20 a 24. Estes distribuem a magnitude das raia de frequência ao longo da faixa de até 5500 Hz, pois através de análises prévias percebeu-se essa como principal faixa com retenção de informação do sistema.

Figura 20 – Espectrograma Operação Normal.



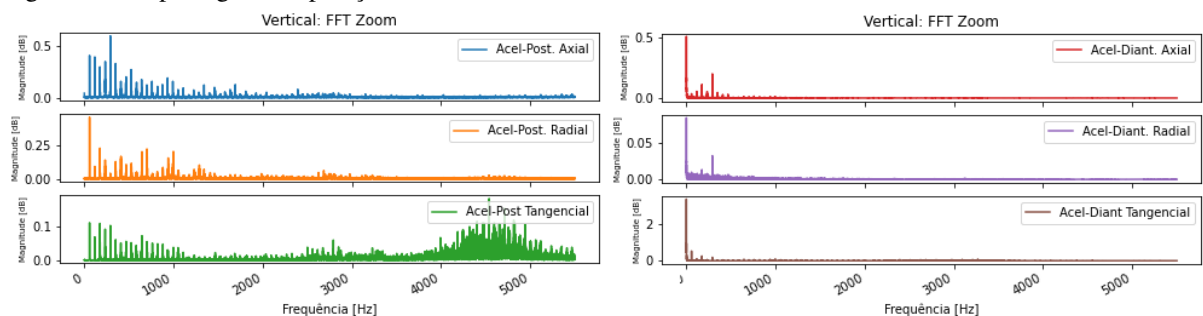
Fonte: O Autor (2023).

Figura 21 – Espectrograma Operação com Desalinhamento Horizontal.



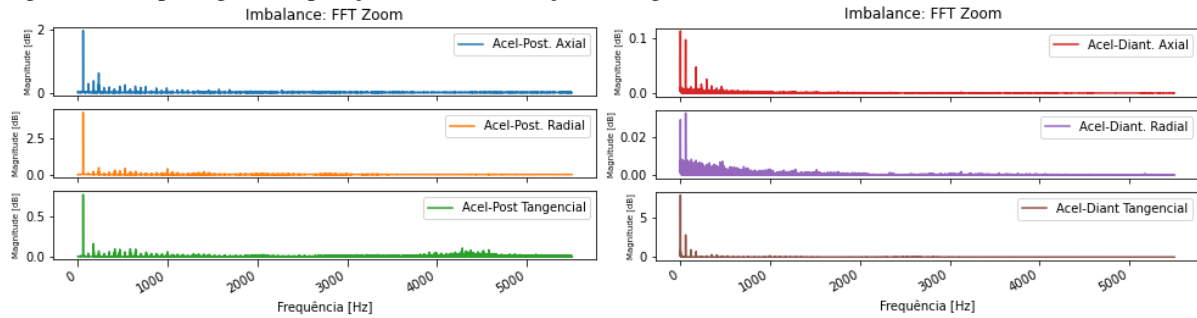
Fonte: O Autor (2023).

Figura 22 – Espectrograma Operação com Desalinhamento Vertical.



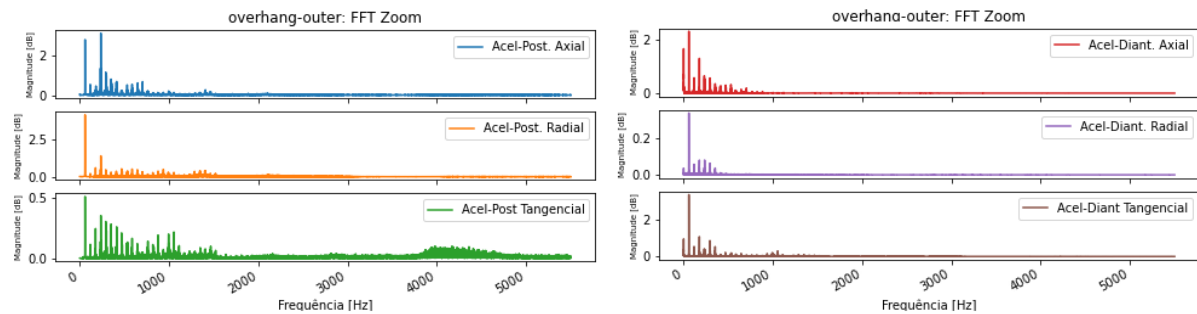
Fonte: O Autor (2023).

Figura 23 – Espectrograma Operação com Desbalanceamento de Carga.



Fonte: O Autor (2023).

Figura 24 – Espectrograma Operação com Falha no Rolamento Dianteiro.



Fonte: O Autor (2023).

Assim essa seção foi responsável por auxiliar, através de testes preliminares, a elaboração da etapa de engenharia de características para o modelo proposto, condicionando os métodos mais adequados para a extração das características significadas da base de dados utilizada.

### 3.3 ENGENHARIA DE CARACTERÍSTICA

Utilizou-se para a extração de características na etapa de *Feature Engineering* dados de 240 ensaios, referentes a todos tipos de falha descrito anteriormente nas faixas de 55 Hz e 60 Hz, uma vez que são os valores mais usuais para a operação do motor em questão. Através da análise exploratória dos dados, projetou-se um filtro digital passa-baixa de 6ª Ordem, tipo Butterworth responsável por atenuar frequências superiores a frequência de Nyquist, neste caso 25 kHz, para reduzir ruídos provenientes ao processo da aquisição de dados.

A Tabela 15 exhibe a relação da distribuição dos ensaios com o número total de amostras e modos de Falhas.

Sun, Wong e Kamel (2009) descreve a classificação de dados com distribuição de classe desbalanceado significativamente desvantajosa em relação ao desempenho da maioria dos al-

Tabela 15 – Distribuição categóricas

Modo de Operação	Número de Ensaios	Número de Amostras (Milhão)
Normal	8	2
Desbalanço de Carga	41	10.25
Desalinhamento Vertical	43	10.75
Desalinhamento Horizontal	27	6.75
Falha de Rolamento Posterior	22   24   21	5.5   6   5.25
Falha de Rolamento Dianteiro	10   22   21	2.5   5.5   5.25

Fonte: O Autor (2023)

goritmos de aprendizagem de máquina que assumem uma distribuição de classe relativamente balanceada, podendo também impactar de forma significativa na métrica de despenho do modelo inteligente.

Para balancear o número de amostras de classes, optou-se por reamostrar as classes desbalanceadas, equilibrando o número amostral positivamente (*upsampling*) através da metodologia *bootstrap* responsável por substituir os dados originais por uma maior amostragem de informações que simulam o comportamento da população original. Segundo Hung, Lee e Chuang (2011), essa metodologia apresenta propriedades robustas, motivando a utilização deste método de reamostragem de dados. Através da reamostragem foi possível obter um conjunto de dados totalmente balanceado, como a Tabela 16 ilustra.

Tabela 16 – Distribuição categóricas pós balanceamentos.

Modo de Operação	Número de Amostras Balanceadas (Milhão)
Normal	10.75
Desbalanço de Carga	10.75
Desalinhamento Vertical	10.75
Desalinhamento Horizontal	10.75
Falha de Rolamento Posterior	10.75   10.75   10.75
Falha de Rolamento Dianteiro	10.75   10.75   10.75

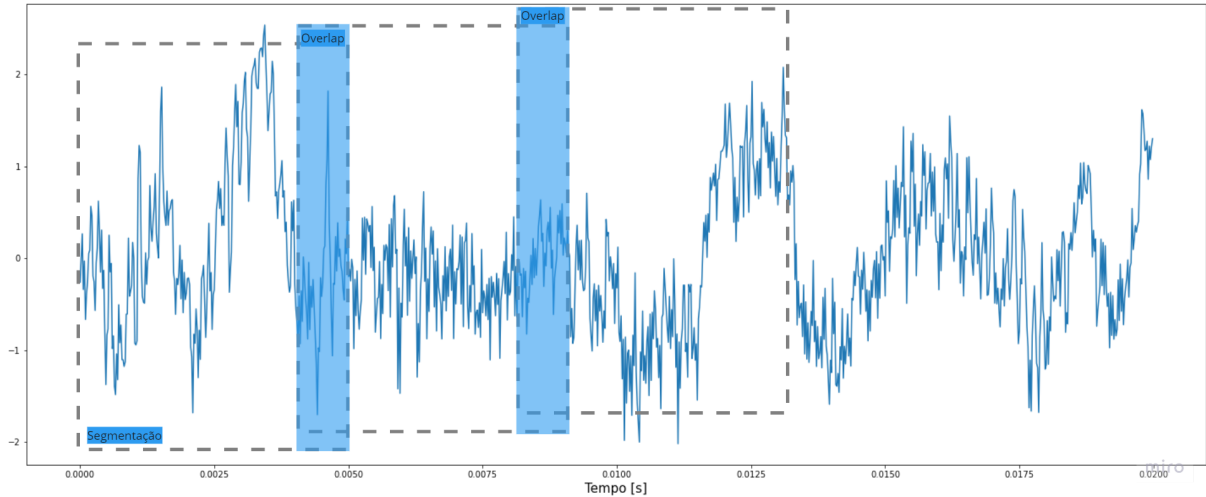
Fonte: O Autor (2023)

Para elaborar o balanceamento dos dados de natureza sequencial, inicia-se a etapa de segmentação do sinal, para isso utilizou-se a técnica da sobreposição de janelas deslizantes produzindo um vetor de dados pertencentes aos segmentos, prontos para alimentar os algoritmos responsáveis pela extração de características.

A janela deslizante é responsável por segmentar a informação em uma região pré-determinada  $T$ , com deslizamento ao longo da série de dados é feito por um passo  $p$ , caso  $p < T$  haverá uma sobreposição entre as faixas de dados dos segmentos, implicando em uma redundância de informação devido a compartilhamento dos dados em ambos segmentos, porém sua vantagem é a garantia da retenção de informação em transições de comportamentos do sinal

não contemplados em apenas um segmento. A Figura 25 ilustra o processo de segmentação por janela deslizante ao longo de uma série temporal, ressaltando as janelas, suas sobreposições e um exemplo de segmento de dado extraído do processo.

Figura 25 – Exemplo de Segmentação de Sinal por janela deslizante.



Fonte: O Autor (2023).

Através da análise prévia da autocorrelação dos sinais optou-se por segmentar o sinal em janelas de 10ms, totalizando quinhentas amostras, aproximando de faixas de autocorrelações próximas a unidade. Quanto ao passo da janela foi definido em 450 amostras, resultando em 10% de sobreposição, produzindo conjunto de dados com menos compartilhamento de informações entre janelas para validar uma maior faixa de generalização para o comportamento dos algoritmo de classificação.

Com o sinal devidamente segmentado cada conjunto de dados alimenta o processo de extração de características. Definido após a análise exploratória dos dados, determinou-se como relevantes na identificação dos modos de operação, as métricas avaliativas para a série temporal quanto sua magnitude, dispersividade e quantidade de informação, tanto sobre o domínio temporal quanto o espectral.

A partir do pacote *Scipy* pode-se transformar os dados segmentados em seu domínio espectral, definindo a Transformada de Fourier Discreta rápida (FFT) pela Equação 3.1, a qual estima a FFT  $\vec{X}(\omega)$  de um vetor temporal  $\vec{x}(t)$  descrito por n amostras em um segmento temporal de tamanho N.

$$\vec{X}(\omega) = \mathcal{F}(\vec{x}(t)) = \sum_{n=0}^{N-1} e^{-2\pi j \frac{\omega n}{N}} x[n] \quad (3.1)$$

A definição da Transformada de Fourier possibilita a descrição das características extraídas tanto sob domínio dinâmico  $\vec{x}(t)$  quanto espectral  $\vec{X}(\omega)$ . Referente a magnitude do sinal, utiliza-se quatro métricas: valor RMS, valor de pico, média e Centróide espectral. Medidas de dispersividade a característica obtida é o quarto momento distributivo, também chamado de curtose e a retenção de informações nas curvas é definido pela entropia de Shannon. A seguir discute-se a metodologia utilizada para a obtenção das métricas.

- **RMS:** o cálculo do Valor RMS ( $V_{rms}$ ) é definido pela Equação 3.2, que utiliza o vetor temporal  $\vec{x}$  de  $n$  elementos para cada uma das  $N$  janelas utilizadas para a excursão total sob o sinal. Estima-se a raiz quadrada da média dos quadrados das componentes dos vetores.

$$V_{rms} = \sqrt{\frac{\sum x[n]^2}{N}} \quad (3.2)$$

- **Máximo:** o valor de máximo é estimado pela função do pacote pandas `.max` utilizando apenas o domínio de frequência, determinando o maior valor das raias espectrais  $\vec{X}(\omega)$ .
- **Média:** o valor médio é estimado pela função do pacote pandas `.mean` utilizando apenas o domínio de frequência, determinando o valor médio da magnitudes das raias espectrais  $\vec{X}(\omega)$ .
- **Centróide Espectral:** o cálculo da centróide espectral ( $C$ ) é definido pela Equação 3.3 como média ponderada entre magnitude  $\vec{X}_\omega$  das raias espectrais e suas respectivas frequências ( $\omega$ ).

$$C = \frac{\sum \omega[n] \vec{X}_\omega[n]}{\sum \omega[n]} \quad (3.3)$$

- **Curtose:** a Equação 3.4 estima a curtose ( $k$ ) a partir da definição de Pearson (1905) como transformada em escala do quarto momento central de distribuição. Utilizando então da esperança da razão entre a diferença da distribuição dos dados ( $x$ ) e seu valor médio  $\mu$  com seu respectivo desvio padrão  $\sigma$

$$k = \frac{E(x - \mu)}{\sigma} \quad (3.4)$$

- **Entropia de Shannon:** para um conjunto de  $n$  eventos de uma variável  $X \in \mathbb{R}^n$  cujas probabilidades de ocorrência são descritas por  $p_i \in \mathbb{R}^n, i \in [0, n]$  de forma que a indicação  $i$  de  $p$  descreve a probabilidade da ocorrência da informação  $X_i$ . A equação 3.5 define a entropia  $H(P)$  como uma métrica da quantidade média de informação contida em uma

variável aleatória.

$$H(x) = \sum_{i=0}^N p_i(x) \cdot \log(p_i(x)), p_i > 0 \quad (3.5)$$

A Tabela 17 descreve as características extraídas para os seis canais, dentro do domínio temporal respectivamente: valor RMS, Curtose e Entropia e para o domínio espectral: valores máximos, médio e centroide espectral de seu espectrograma, assim como a curtose espectral. Resultando um vetor de 42 características devido aos 6 canais e 7 métricas estatísticas.

Tabela 17 – Vetor de Características.

	Magnitude	Dispersividade	Retenção de Informação
Domínio Temporal	RMS	Curtose	Entropia
Domínio Espectral	Máxima   Média   Centroide	Curtose	-

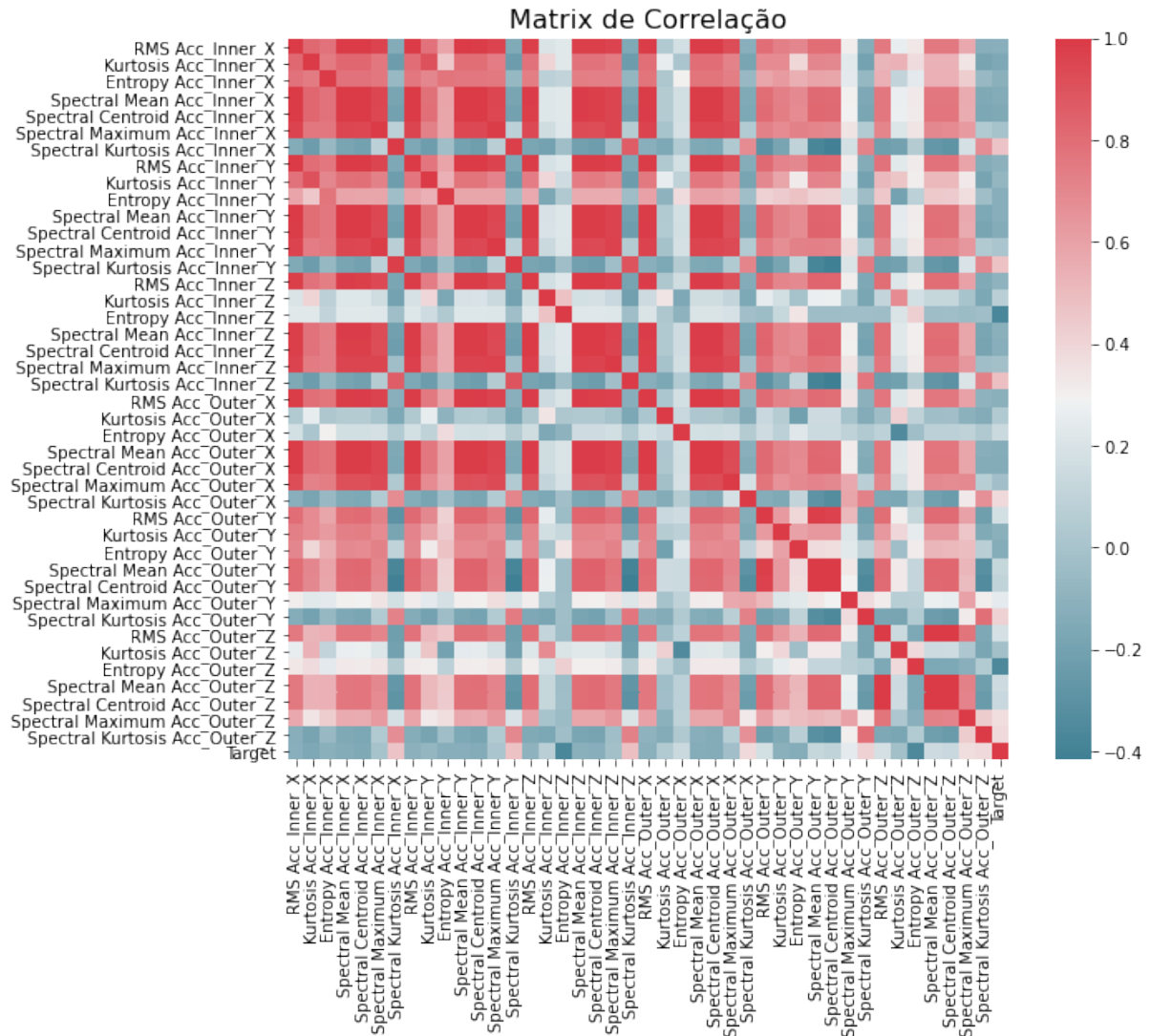
Fonte: O Autor (2023)

### 3.3.1 Redução de Dimensionalidade

Devido a grande dimensionalidade do conjunto de características extraídas formado na etapa anterior, observou-se através da Figura 26 suas correlações como meio de justificar e investigar a utilização de metodologias para redução de dimensionalidade.

Através da Figura 26 é possível ver repetidas taxas de correlação acima de 90% entre as características extraídas, para evitar excesso de carga computacional ao processar informações com alto teor de correlação, optou-se pela implementação de 2 modelos de redução de dimensionalidade: Análise das componentes principais, do inglês *Principal Component Analysis - PCA* e o agrupamento hierárquico (Clusterização das características).

Figura 26 – Correlação das Características Extraídas.



Fonte: O Autor (2023).

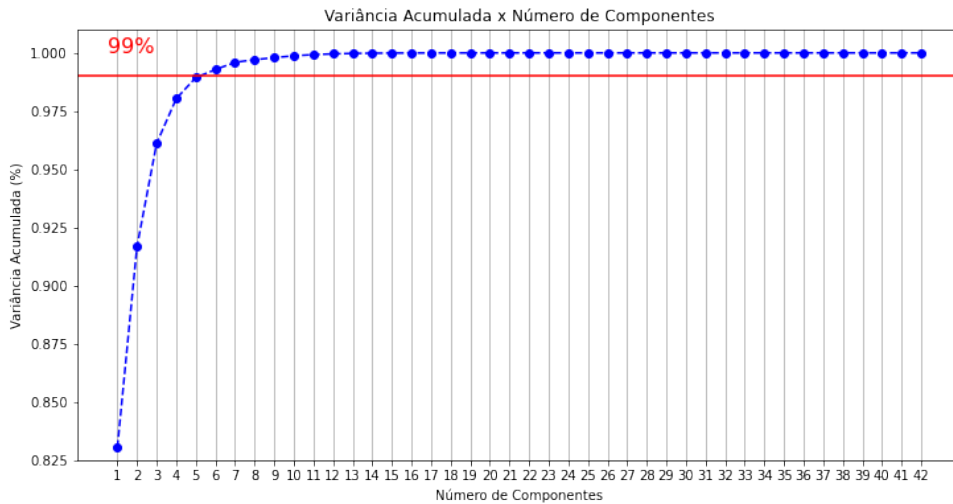
A análise das componentes principais, é uma transformação algébrica responsável por, através da ortogonalização de vetores, converter um conjunto de variáveis correlacionadas em um conjunto não-correlacionados que concentre a maior variação possível dos dados Grus (2019). Para a implementação do PCA utilizou-se o pacote *sklearn.decomposition*, o qual elaborase a redução de dimensionalidade linear através a Decomposição de Valor Singular (SVD) dos dados para projetá-lo em um espaço dimensional inferior, sendo posteriormente centralizados, mas não escalonado em cada uma das característica antes de aplicar o SVD completo através do pacote de álgebra linear LAPACK. Ao inferir o algoritmo de PCA, buscou-se inicialmente produzir a curva de variância acumulada para determinar o comportamento da análise, definindo a quantidade de componentes minimamente necessário para conter o patamar de 99% da



variância do sistema.

A curva é exibida pela Figura 27, na qual observa-se um comportamento exponencial no acúmulo de variância até aproximadamente 6 componentes, número esse responsável para ultrapassar o patamar desejado da variância sistêmica. Componentes adicionais acabam contendo faixas insignificantes de variâncias, sendo descartadas para alimentação dos algoritmos de classificação.

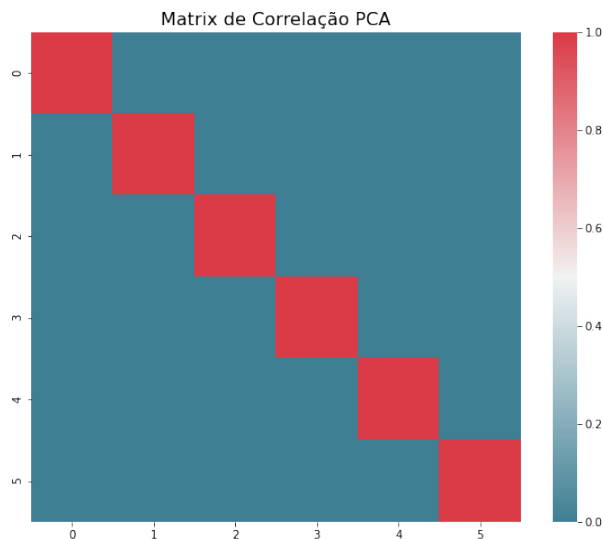
Figura 27 – Variância acumulada por número de componente.



Fonte: O Autor (2023).

Para avaliar a metodologia de implementação do PCA, buscou-se avaliar a correlação das seis principais componentes produzidas. Através da Figura 28 nota-se correlações nulas, como era esperado, indicando uma implementação adequada do algoritmo de PCA.

Figura 28 – Correlação: PCA.

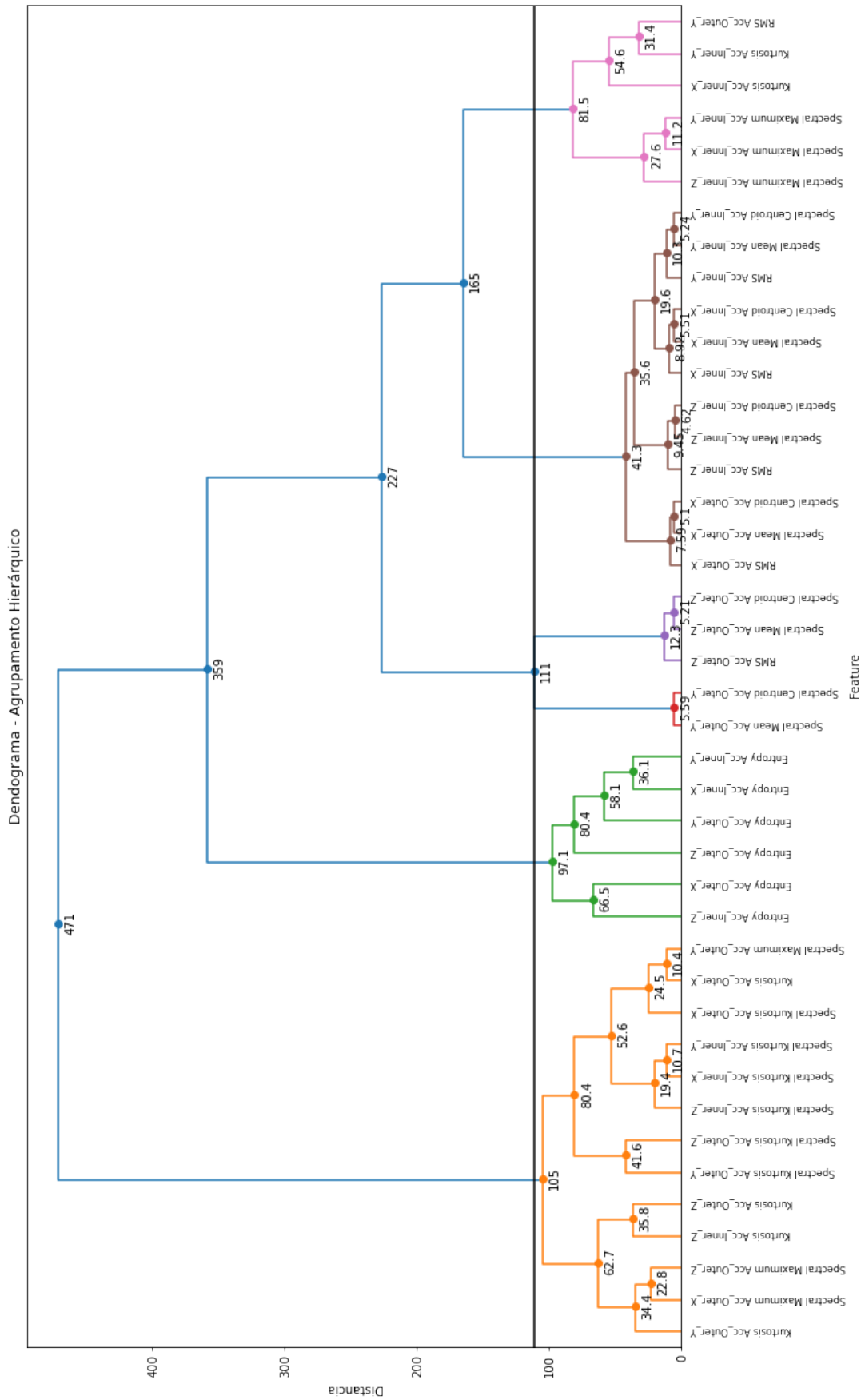


Fonte: O Autor (2023).

A utilização do agrupamento hierárquico para a seleção de característica é comumente elaborada, para formação de grupos compostos pela mescla de características extraídas de comportamentos similares em um *cluster* (PARK, 2013). O critério de estimativa de similaridade utilizado foi pelo método de Ward que minimiza a variância dos dados aglutinados.

A especificação do número de *clusters* formados foram equivalente ao número de componentes escolhidas pela análise do PCA, como forma de garantir uma comparação válida entre os métodos ao alimentar os dados formados ao modelo inteligente proposto. Os dados passaram pelo processo de normalização - método discutido a sessão subsequente - responsável por auxiliar a estimativa dos agrupamentos formados. A Figura 29 exhibe o dendograma formado pela Clusterização em conjunto com o separador dos seis grupos clusterizados, destacando a formação dos grupos provindos de magnitude, dispersividade distributiva e retenção de informação.

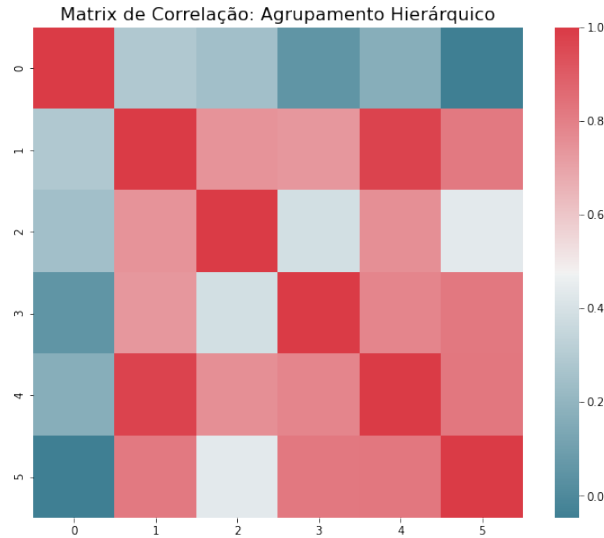
Figura 29 – Dendograma Agrupamento Hierárquico.



Fonte: O Autor (2023).

Diferentemente da produção das componentes pelo PCA, as quais tinham correlação nulas, para o agrupamento hierárquico os conjuntos de dados resultantes não necessitam da nulidade de suas correlações como a Figura 30 ilustra.

Figura 30 – Correlação: Agrupamento Hierárquico.



Fonte: O Autor (2023).

### 3.3.2 Normalização e Divisão dos Conjuntos de Dados

O objetivo da normalização é mudar os valores das colunas numéricas no conjunto de dados para usar uma escala comum, sem distorcer as diferenças nos intervalos de valores nem perder informações. A normalização também é necessária para alguns algoritmos para modelar os dados corretamente. A normalização é um processo necessário para aumentar a eficiência do processo de aprendizado dos algoritmos inteligentes, seu objetivo é alterar os valores para uma escala comum, sem distorcer as diferenças nos intervalos de valores, nem perder informações. Ao conjunto de dados produzidos pelos algoritmos de redução de dimensionalidade foi determinado sua normalização dentro do conjunto real  $C_{norm}$ , tal que  $C_{norm} \in [0, 1]^6 \subset \mathbb{R}^6$ . A transformação feita para a escala de cada uma das características é dada através da Equação 3.6, onde  $X$  é o dado de entrada proveniente do vetor de características,  $X_{max}$  o valor máximo da características em questão e  $X_{min}$  valores de mínimo da mesma.

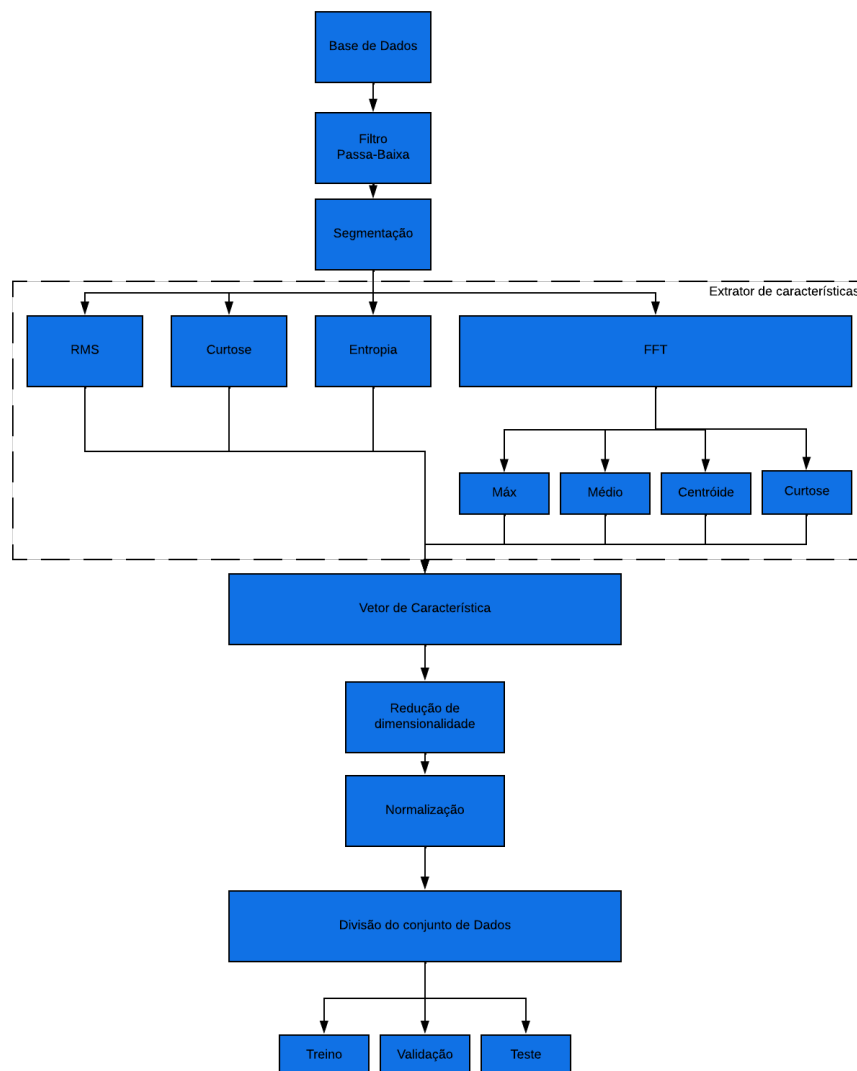
$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3.6)$$

Após a normalização dos dados, os mesmos foram divididos em três conjuntos, treinamento, validação e teste, cada um dos conjuntos são responsáveis por diferentes etapas da elaboração dos modelos inteligentes, tal que, o conjunto de treinamento é o agrupamento utili-

zado durante o processo de aprendizagem responsável para ajustar os parâmetros do modelo de aprendizado, o conjunto de validação é utilizado para comparar as diferentes concepções hiperparamétricas que o modelo é configurado durante a etapa de ajuste fino dos hiperparâmetros e o conjunto de teste é um conjunto de dados inéditos utilizado para verificar a capacidade de generalização do modelo determinado pela otimização de hiperparâmetros. As proporções entre a divisão de dados são relevantes a sua performance, assim optou-se pela proporção de 70%, 15% e 15%, para os conjuntos de treinamento, validação e testes, respectivamente, priorizando a etapa de aprendizado dos modelos.

Após a divisão dos dados a etapa de pré-processamento é encerrada, a Figura 31 exibe o fluxograma utilizado durante o processo descrito nesse capítulo, responsável pela formação dos conjuntos de dados que alimentarão os modelos propostos na seção seguinte.

Figura 31 – Etapa de pré-processamento.



### 3.4 MÉTODOS INTELIGENTES E OTIMIZAÇÃO DE HIPERPARÂMETROS

Essa seção visa discutir os modelos de aprendizados utilizados para a diagnóstico de falhas em máquinas rotativas através do vetor de característica produzido na etapa de engenharia de característica, discutido anteriormente, assim como, a elucidar os métodos de otimização do ajuste fino de hiperparâmetros utilizados.

Inicialmente a máquina responsável pelos testes tem as seguintes configurações:

- Unidade Central de Processamento (CPU): Intel Core i7-8565U @ 1.80GHz;
- Unidade de Processamento Gráfico (GPU): Nvidia MX110;
- Memória RAM: 8,00 GB (utilizável: 7,88 GB);
- Sistema Operacional: Windows 10 Home x64;
- Versão Interpretador Python: 3.8.5.

#### 3.4.1 Métodos Inteligentes

Os métodos utilizados para o processo de aprendizado foram: as Redes Nerais de Múltiplas Camadas de Perceptron (RNA MLP), Florestas Aleatórias (RF), *Boosting* Adaptativo (AdaBoost) e *Boosting* de Gradiente Extremo (xgboost). Para configurar os modelos é necessário especificar seus hiperparâmetros. como esse trabalho propõe otimizar os modelos a partir da utilização de algoritmos de metaheurística, define-se então os limites do universo de discurso dos hiperparâmetros para delimitar o espaço de pesquisa dos algoritmos de otimização.

Os hiperparâmetros do Modelo AdaBoost ditam a respeito dos estimadores bases, seu respectivo número e taxa de aprendizado. o espaços de pesquisa delimitados são:

- estimadores base: Bernoulli NB, Classificador SGD, Arvore de Decisão, Perceptron;
- número de estimadores: 10, 50, 100, 150 e 250;
- taxa de aprendizado: 0.001, 0.01, 0.1, 0.5, 1, 5 e 10.

Os hiperparâmetros das Florestas Aleatórias ditam a respeito do número de estimadores, máximo de profundidade, número mínimo de amostras necessárias para a divisão da estrutura, presença de *Bootstrap* e o critério de estimativa de complexidade para divisão o modelo. O espaços de pesquisa foi delimitados por:

- número de estimadores: 1, 2, 5, 10, 25, 20 e 30;

- profundidade máxima: 10, 15, 20, 30 e 50;
- número mínimo de amostra por divisão: 1, 2, 4, 8, 16 e 32;
- bootstrap: Verdadeiro ou Falso;
- critério: Gini ou Entropia;

Os hiperparâmetros das RNAs ditam a respeito do número de camadas ocultas, respectivo números de neurônios por camada, suas funções de ativação, o algoritmo de aprendizagem (otimizador), o método de agenda da taxa de aprendizado do modelo, a força do regulador L2. Portanto, o espaço de pesquisa foi delimitado por:

- número de camadas ocultas: 2, 3 e 4;
- número de neurônio por camadas oculta: 2, 4, 8, 32, 64, 128, 256;
- função de ativação: Identidade, Logística, Tangente Hiperbólica, ReLU;
- otimizador: Gradiente Descendente Estocástico (SGD), Adam;
- penalidade L2: 0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1;
- tipo de taxa de aprendizado: Constante, Adaptativo;

Os hiperparâmetros da Regressão Logística, são a penalidade L2, seu algoritmo de resolução (*solver*) e o parâmetro C. Seu respectivos espaços de pesquisa delimitados são:

- penalidade L2: Presente ou Ausente;
- solver: Newton-cg, lbfgs, sag e saga;
- C: 0.01, 0.05, 0.1, 0.5 e 1, 2;

Os hiperparâmetros do XGBOOST especifica sua profundidade máxima, o número de estimadores, sua taxa de aprendizado e ditam a respeito dos estimadores bases, seu respectivo número e taxa de aprendizado e a taxa de subamostragem por estimadores. o espaços de pesquisa delimitados são:

- profundidade máxima: 2, 3, 4, 5, 6 e 10;
- número de estimadores: 8, 32, 64, 128, 256, 512 e 1024;
- taxa de aprendizado: 0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.5 e 1;
- taxa de subamostragem: 0.1, 0.3, 0.7, 0.9 e 1;

### 3.4.2 Métodos de Otimização de Hiperparâmetros

Os algoritmos de aprendizagem de máquina têm sua topologia definida através da configuração de hiperparâmetros. Estes são fatores responsáveis por estipular o comportamento de aspectos do modelo proposto. Desta forma é desejável determinar uma topologia através da busca da minimização da função custo do modelo. Essa busca pode ser feita manualmente ou de forma automática através de algoritmos para sua otimização. Desta forma este trabalho buscou comparar duas técnicas de hiperparametrização, busca randômica e algoritmos genéticos, para os modelos de aprendizagem propostos, validando seu desempenho pela métrica de *accuracy* e sua carga computacional através do tempo de treinamento do modelo.

- Pesquisa Randômica: implementa configurações aleatória sobre arranjos de hiperparâmetros pertencentes ao universo de discurso definidos. Apesar de não garantir o minimização máxima da função custo, tem um custo computacional significativamente reduzido, em relação as pesquisas de extensiva e devido a distribuição probabilística da alocação de combinações ao longo do espaço hiperparamétrico é comum sua combinação resultante apresentar métricas satisfatórias na implementação do modelo. A pesquisa randômica foi elaborada através do *framework sci-kit learning*, acatando suas sugestões de parametrização da pesquisa randômica, produzindo 10 amostras ao longo do dicionário hiperparamétrico.

- Algoritmo Genético:

metodologia de busca e otimização baseada nos processos descritos pela biologia evolutiva, com o objetivo de elaborar uma população de diferentes representações de modelos e através da evolução da mesma ao longo de gerações, tenda a estipular um modelo mais adequado com a função de aptidão estabelecida inicialmente como premissa do sistema. O algoritmo genético elaborado para este trabalho baseou-se no *framework DEAP*, produzindo um processo de tamanho inicial da população com 10 indivíduos, probabilidade de mutação de 10%, probabilidade de cruzamento de 50%, número máximo de 50 gerações e o tamanho do torneio em 3 indivíduos, fatores atribuídos através de testes preliminares. Utilizando a métrica de *accuracy* como a função de aptidão a ser maximizada.



## 4 RESULTADOS E DISCUSSÕES

Esse Capítulo tem por função exibir as métricas dos modelos ao longo do processo de ajustes de hiperparâmetros através do conjunto de validação, ilustrando-os por meio de gráficos e avaliando o grau de generalização dos mesmos utilizando um conjunto de dados inéditos, distinguindo sua performance através dos métodos de ajuste fino dos hiperparâmetros dos classificadores utilizado e método de redução de dimensionalidade. Definido como métricas a *accuracy* do conjunto de teste e o custo computacional do processo de treinamento, como discutido na seção de metodologia.

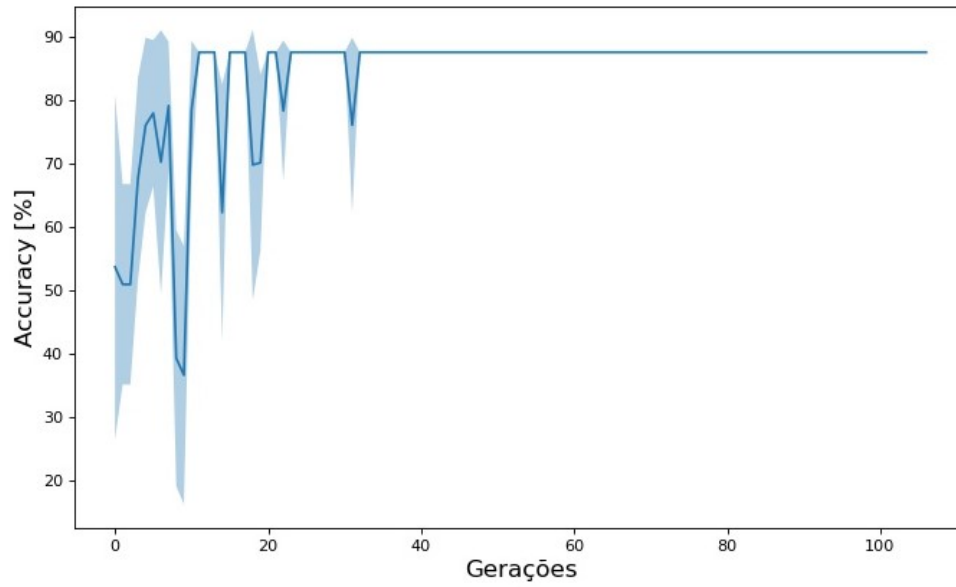
É possível desenvolver três principais cenários para avaliar os modelos: sistemas que priorizam a alta taxa de performance, sistemas que necessitam de modelos com baixo custo computacional de treinamento e sistemas que visam modelos com maior *accuracy* porém levando em conta o menor custo computacional possível.

Os três cenários, requerem a distinção entre métricas avaliativas aos modelos a fim de proporcionar a maior aptidão com o objetivo do mesmo. Ao priorizar a maximização da métrica de *accuracy* o modelo mais correspondente satisfaz a condição do primeiro cenário, assim com a minimização do custo computacional distingue o modelo mais apto para o segundo cenário e por fim ao avaliar o ponto otimizado entre *accuracy* e custo computacional é proposto a métrica razão entre performance (representado pela *accuracy*) e tempo. A maximização desse indicador determina o modelo mais adequado para o cenário abordado.

### 4.0.1 Adaboost

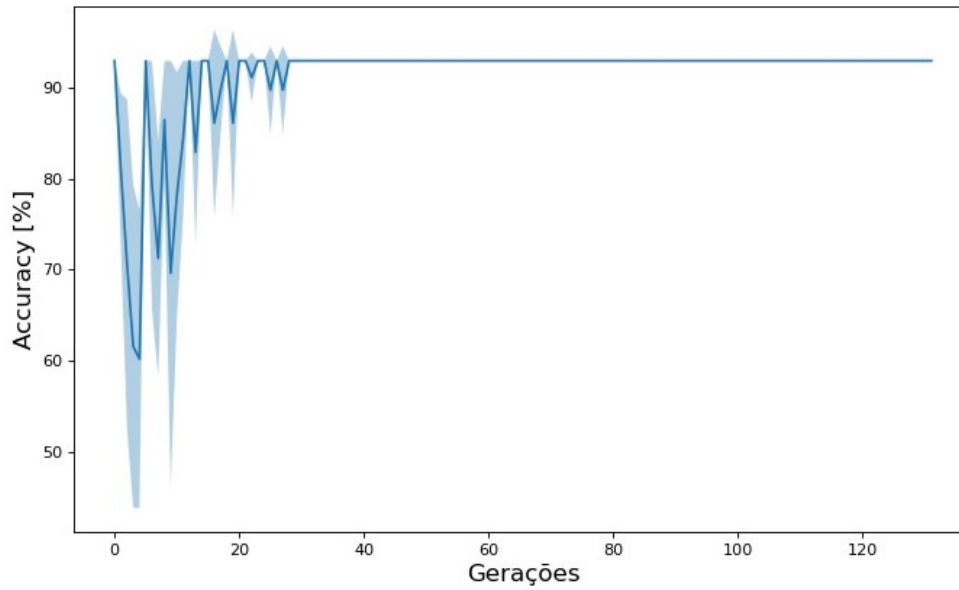
Ao implementar os sistemas propostos no capítulo anterior para o modelo AdaBoost, foi possível verificar resultados adequados à implementação através de cinco rodadas de repetições entre as combinações dos dois métodos de redução de dimensionalidade e as duas metodologias de ajuste fino de hiperparâmetros. As curvas de *accuracy* médias entre as rodadas de repetições, ao longo do processo de configuração de hiperparâmetros, são exibidos pelas Figuras 32, 33, 34, 35, para o conjunto de validação dos dados, sintetizando seu valores médio pela curva destacada e o respectivo desvio padrão de cada época através do destaque da sua região.

Figura 32 – AdaBoost Clusterização de Características com Algoritmo Genético.



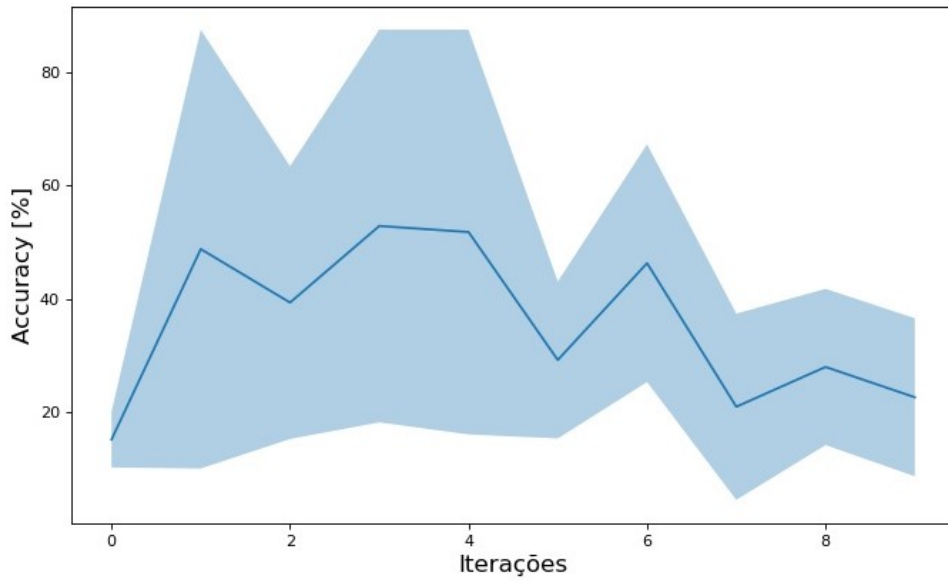
Fonte: O Autor.

Figura 33 – AdaBoost PCA com Algoritmo Genético.



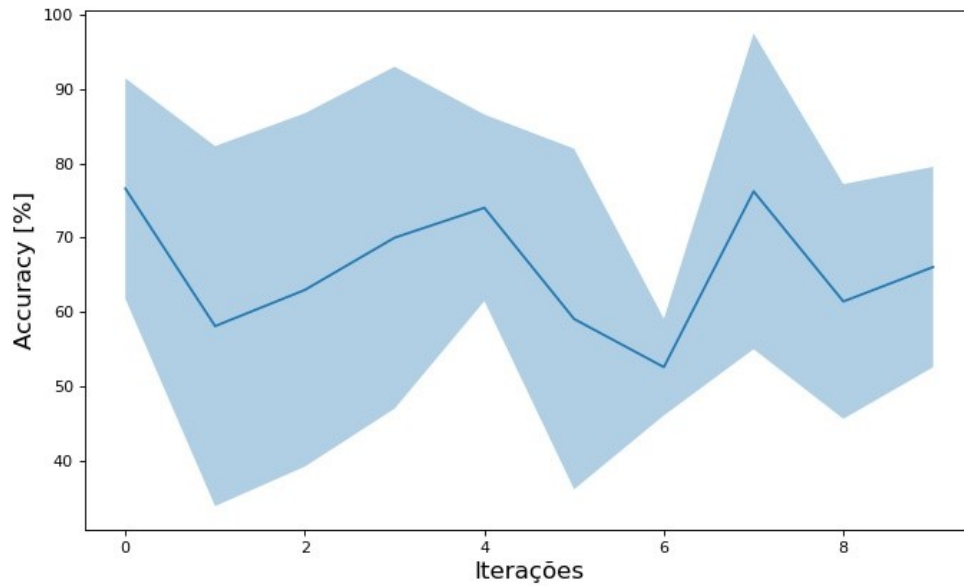
Fonte: O Autor.

Figura 34 – AdaBoost Clusterização de Características com Pesquisa Randômica.



Fonte: O Autor.

Figura 35 – AdaBoost PCA com Pesquisa Randômica.



Fonte: O Autor.

A partir dos processos de treinamento realizados, os melhores modelos foram avaliados com o conjunto de teste e tiveram sua métrica avaliativa referente à performance e custo computacional comparadas através das Tabelas 18 e 19 respectivamente.

Tabela 18 – Accuracy Médio Adaboost [%].

	PCA	Clusterização de Características
Algoritmo Genético	93,9	90,1
Pesquisa Randômica	93,8	80,6

Fonte: O Autor (2023)

Tabela 19 – Custo Computacional Médio Adaboost [s].

	PCA	Clusterização de Características
Algoritmo Genético	5987	4374
Pesquisa Randômica	2022	1987

Fonte: O Autor (2023)

Através os três cenários de aplicações possíveis, os modelos destaques ao longo dos cinco rounds de treinamento tem sua hiperparametrização exibida na Tabela 20, assim como sua respectivas métricas de desempenho e custo computacional. Nela pode-se perceber a mesma configuração para os três casos, definido a utilização do PCA como mais eficiente para determinação de modelos com melhores resultados de desempenho.

Tabela 20 – Configurações de hiperparâmetros de Adaboost destacadas em relação aos possíveis cenários.

	Performance	Custo Computacional	Performance-Custo Computacional
Estimador	Árvore de Decisão	Árvore de Decisão	Árvore de Decisão
Nº de Estimadores	150	150	150
Taxa de Aprendizado	10	10	10
Redução de Dimensionalidade	PCA	PCA	PCA
Método de hiperparametrização	Randômico	Randômico	Randômico
<i>Accuracy</i> [%]	98.27	93.87	93.87
Custo Computacional[s]	30057	780	780

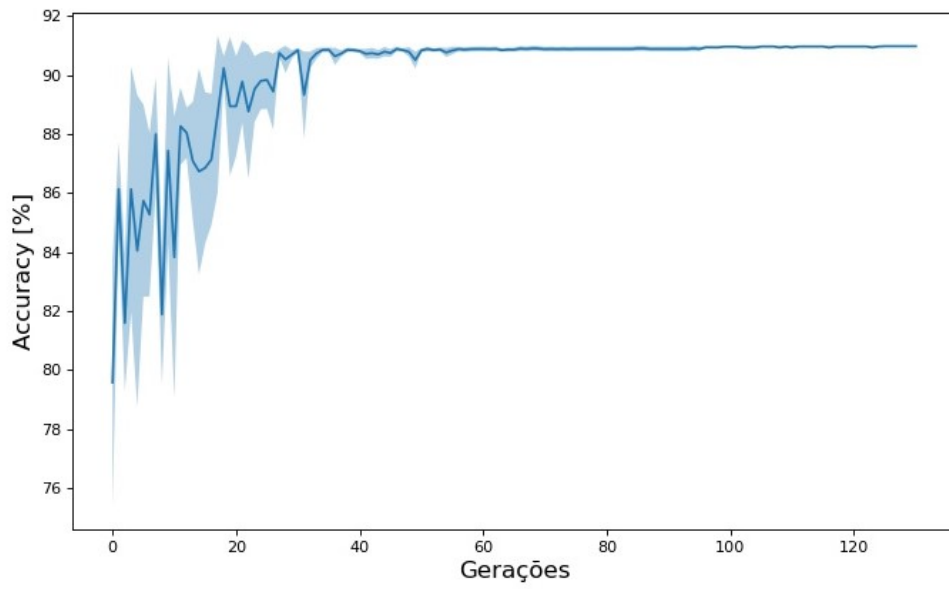
Fonte: O Autor (2023)

#### 4.1 FLORESTAS ALEATÓRIAS

Ao implementar os sistemas propostos no capítulo anterior para o modelo Florestas Aleatórias, foi possível verificar resultados adequados à implementação através de cinco rodadas de repetições entre as combinações dos dois métodos de redução de dimensionalidade e as duas metodologias de ajuste fino de hiperparâmetros.

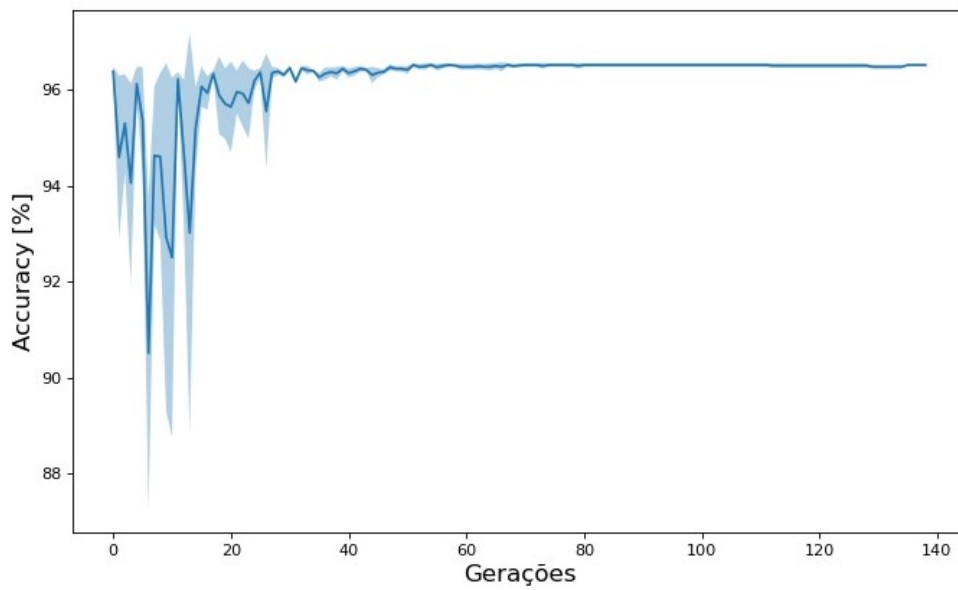
As curvas de *accuracy* médias entre as rodadas de repetições, ao longo do processo de configuração de hiperparâmetros, são exibidos pelas Figuras 36, 37, 38, 39, para o conjunto de validação dos dados, sintetizando seu valores médio pela curva destacada e o respectivo desvio padrão de cada época através do destaque da sua região.

Figura 36 – Floresta Aleatórias Clusterização de Características com Algoritmo Genético.



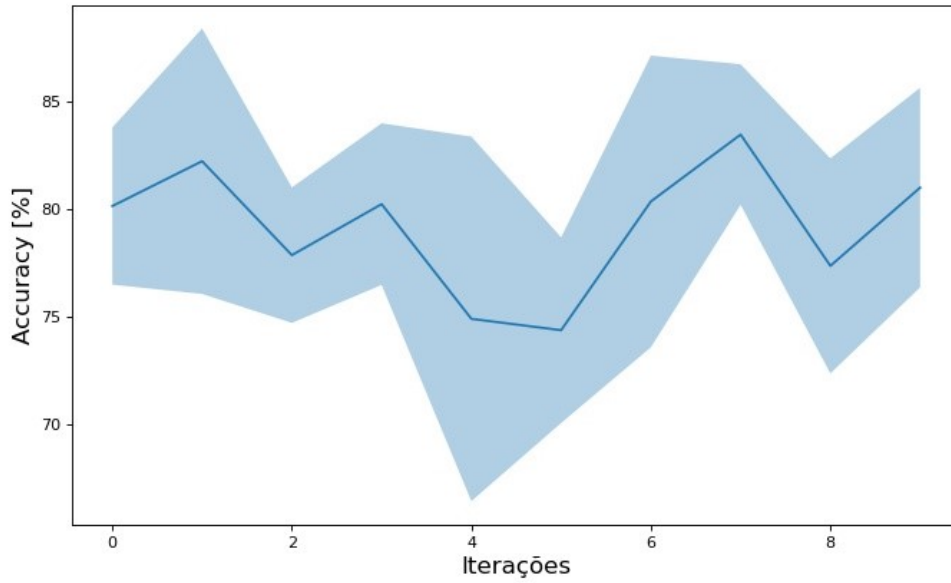
Fonte: O Autor.

Figura 37 – Floresta Aleatória PCA com Algoritmo Genético.



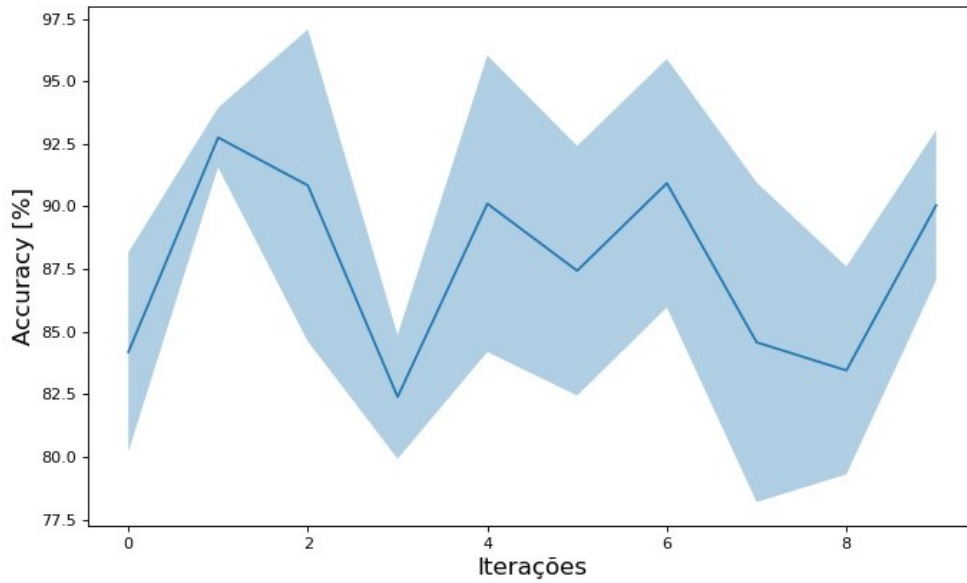
Fonte: O Autor.

Figura 38 – Floresta Aleatória Clusterização de Características com Pesquisa Randômica.



Fonte: O Autor.

Figura 39 – Floresta Aleatória PCA com Pesquisa Randômica.



Fonte: O Autor.

A partir dos processos de treinamento realizados, os melhores modelos foram avaliados com o conjunto de teste e tiveram sua métrica avaliativa referente à performance e custo computacional comparadas através das Tabelas 21 e 22 respectivamente.

Tabela 21 – Accuracy Médio Floresta Aleatória [%].

	PCA	Clusterização de Características
Algoritmo Genética	97,1	92,5
Pesquisa Randômica	96,1	89,9

Fonte: O Autor (2023)

Tabela 22 – Custo Computacional Médio Floresta Aleatória [s].

	PCA	Clusterização de Características
Algoritmo Genético	1357	610
Pesquisa Randômica	201	75

Fonte: O Autor (2023)

Através os três cenários de aplicações possíveis, os modelos destaques ao longo dos cinco rounds de treinamento tem seu ajuste fino de hiperparâmetros exibida na Tabela 23, assim como sua respectivas métricas de desempenho e custo computacional. Duas configuração foram destacadas, nas quais a utilização de PCA produziu um modelo com melhor desempenho, enquanto ao utilizar a clusterização de características gerou um modelo cujo custo computacional ao treinamento foi o menor.

Tabela 23 – Configurações de hiperparâmetros de Florestas Aleatórias destacadas em relação aos possíveis cenários.

	Performance	Custo Computacional	Performance-Custo Computacional
Nº de Estimadores	30	2	2
Profundidade Máxima	50	30	30
Nº Mín de Amostra por divisão	2	1	1
Bootstrap	Ausente	Ausente	Ausente
Critério de divisão	Entropia	Entropia	Entropia
Redução de Dimensionalidade	PCA	Cluster Característica	Cluster Característica
Método de Hiperparametrização	Randômico	Randômico	Randômico
<i>Accuracy</i> [%]	92.6	88.7	88.7
Custo Computacional[s]	676	47	47

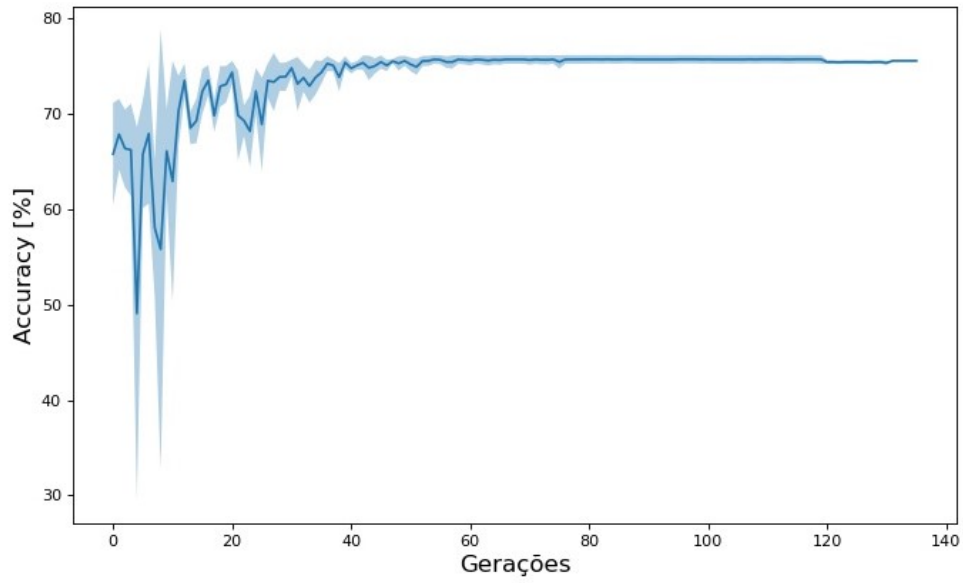
Fonte: O Autor (2023)

## 4.2 REDES NEURAIAS ARTIFICIAIS MLP

Ao implementar os sistemas propostos no capítulo anterior para o modelo de Redes Neurais MLP, foi possível verificar resultados adequados à implementação através de cinco rodadas de repetições entre as combinações dos dois métodos de redução de dimensionalidade e as duas metodologias de ajuste fino de hiperparâmetros.

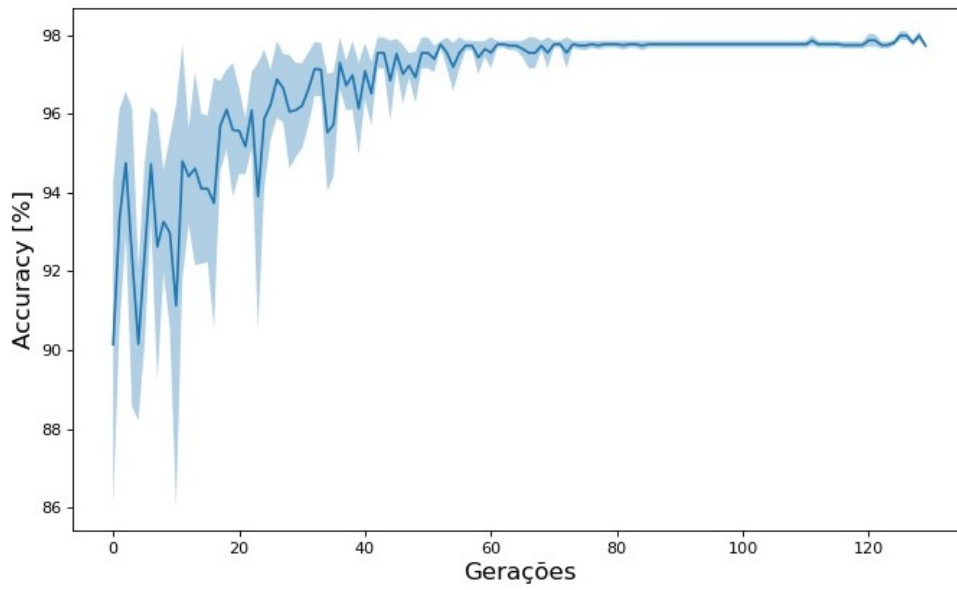
As curvas de *accuracy* médias entre as rodadas de repetições, ao longo do processo de configuração de hiperparâmetros, são exibidos pelas Figuras 40, 41, 42, 43, para o conjunto de validação dos dados, sintetizando seu valores médio pela curva destacada e o respectivo desvio padrão de cada época através do destaque da sua região.

Figura 40 – RNA MLP Clusterização de Características com Algoritmo Genérico.



Fonte: O Autor.

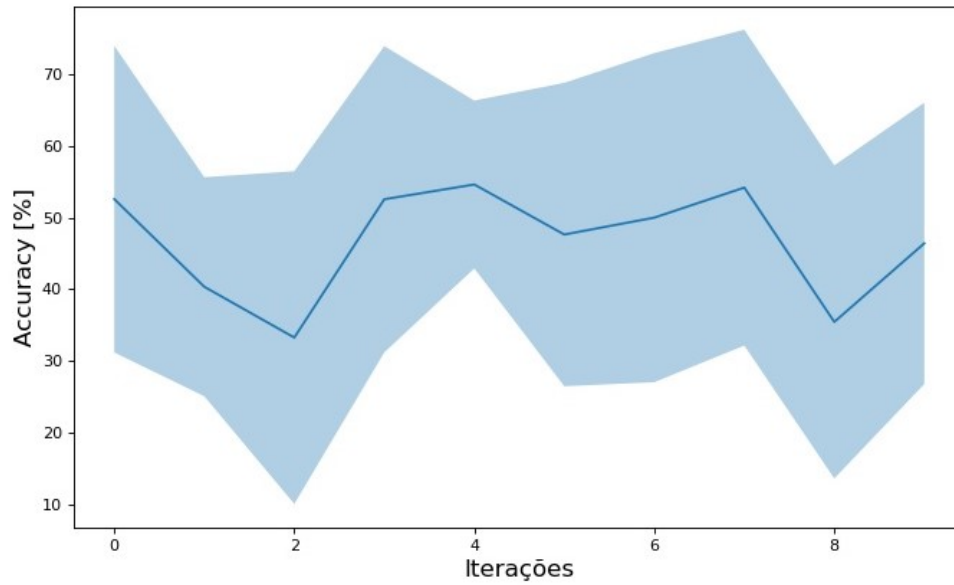
Figura 41 – RNA MLP PCA com Algoritmo Genérico.



Fonte: O Autor.

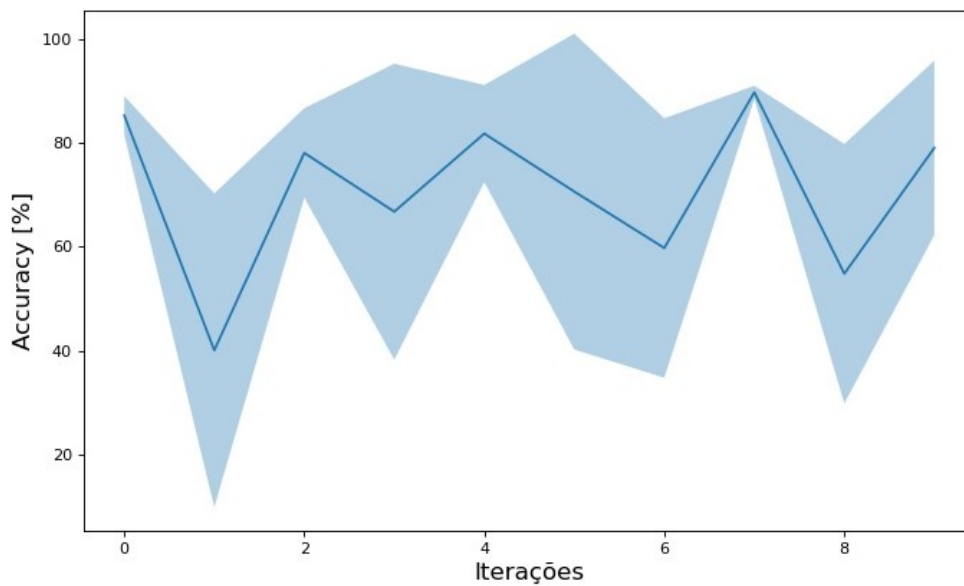


Figura 42 – RNA MLP Clusterização de Características com Pesquisa Randômica.



Fonte: O Autor.

Figura 43 – RNA MLP PCA com Pesquisa Randômica.



Fonte: O Autor.

A partir dos processos de treinamento realizados, os melhores modelos foram avaliados com o conjunto de teste e tiveram sua métrica avaliativa referente à performance e custo computacional comparadas através das Tabelas 24 e 25 respectivamente.

Tabela 24 – Accuracy Médio Redes Neurais MLP[%].

	PCA	Clusterização de Características
Algoritmo Genético	97,78	76,88
Pesquisa Randômica	94,58	74,79

Fonte: O Autor (2023)

Tabela 25 – Custo Computacional Médio Redes Neurais MLP [s].

	PCA	Clusterização de Características
Algoritmo Genético	8454	7545
Pesquisa Randômica	451	900

Fonte: O Autor (2023)

Através os três cenários de aplicações possíveis, os modelos destaques ao longo dos cinco rounds de treinamento tem seu ajuste fino de hiperparâmetros exibida na Tabela 26, assim como sua respectivas métricas de desempenho e custo computacional. Duas configuração foram destacadas, ambas utilizando o PCA como método de redutor de dimensionalidade das características.

Tabela 26 – Configurações de hiperparâmetros de Redes Neurais MLP destacadas em relação aos possíveis cenários.

	Performance	Custo Computacional	Performance-Custo Computacional
Nº de Camadas Ocultas	256	32	32
Nº de Neurônios por camada	3	4	4
Função de Ativação	Tanh	Tanh	Tanh
Otimizador	Adam	Adam	Adam
Regulador l2	0.1	0.1	0.1
Tipo de taxa de aprendizado	Adaptativo	Adaptativo	Adaptativo
Redução de Dimensionalidade	PCA	PCA	PCA
Método de Hiperparametrização	Genético	Randômico	Randômico
<i>Accuracy</i> [%]	98.2	95.3	95.3
Custo Computacional[s]	5650	183	183

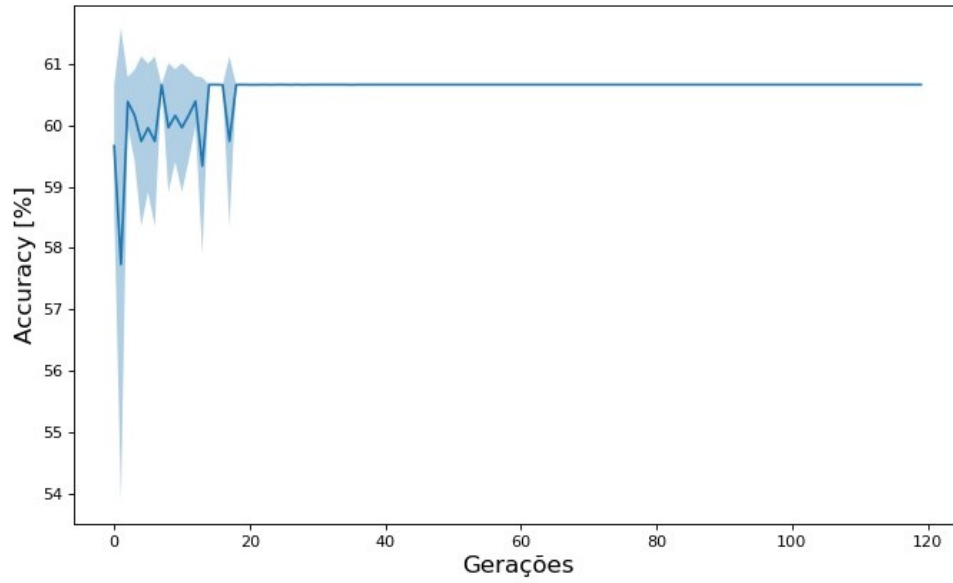
Fonte: O Autor (2023)

### 4.3 REGRESSÃO LOGÍSTICA

Ao implementar os sistemas propostos no capítulo anterior para o modelo de Regressão Logística, foi possível verificar resultados adequados à implementação através de cinco rodadas de repetições entre as combinações dos dois métodos de redução de dimensionalidade e as duas metodologias de ajuste fino de hiperparâmetros.

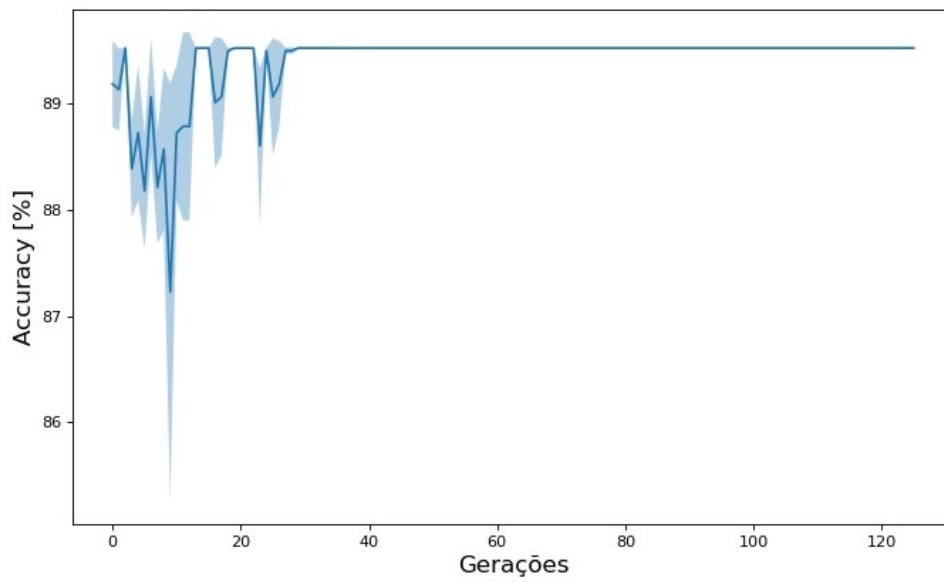
As curvas de *accuracy* médias entre as rodadas de repetições, ao longo do processo de configuração de hiperparâmetros, são exibidos pelas Figuras 44, 45, 46, 47, para o conjunto de validação dos dados, sintetizando seu valores médio pela curva destacada e o respectivo desvio padrão de cada época através do destaque da sua região.

Figura 44 – Regressão Logística Clusterização de Características com Algoritmo Genérico.



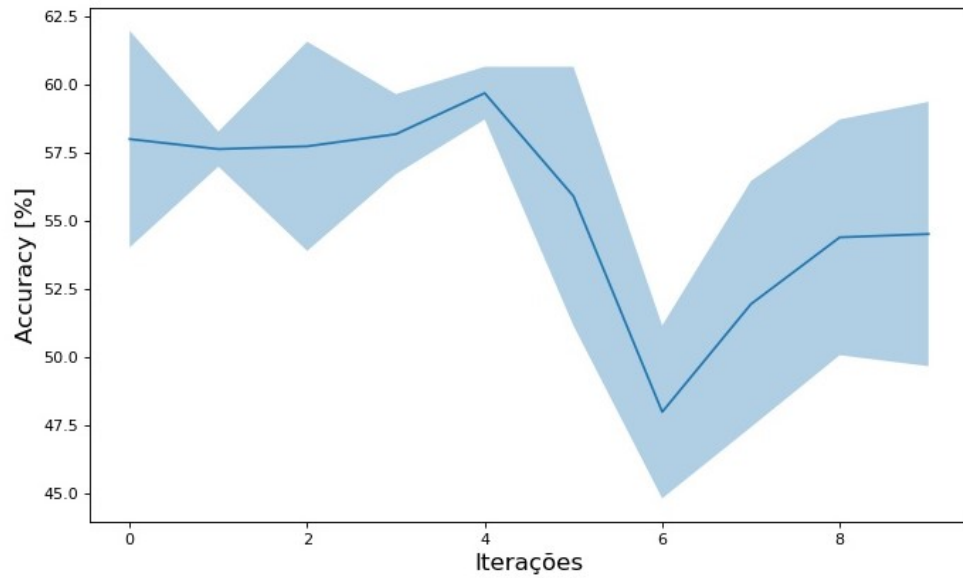
Fonte: O Autor.

Figura 45 – Regressão Logística PCA com Algoritmo Genérico.



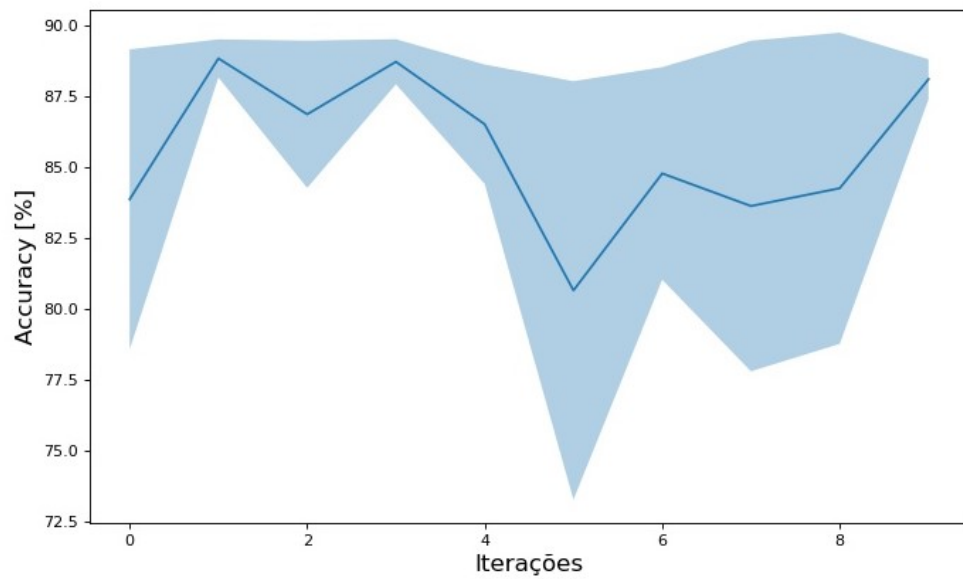
Fonte: O Autor.

Figura 46 – Regressão Logística Clusterização de Características com Pesquisa Randômica.



Fonte: O Autor.

Figura 47 – Regressão Logística PCA com Pesquisa Randômica.



Fonte: O Autor.

A partir dos processos de treinamento realizados, os melhores modelos foram avaliados com o conjunto de teste e tiveram sua métrica avaliativa referente à performance e custo computacional comparadas através das Tabelas 27 e 28 respectivamente.

Tabela 27 – Accuracy Médio Regressão Logística[%].

	PCA	Clusterização de Características
Algoritmo Genético	89,3	61,0
Pesquisa Randômica	89,3	61,0

Fonte: O Autor (2023)

Tabela 28 – Custo Computacional Médio Regressão Logística [s].

	PCA	Clusterização de Características
Algoritmo Genética	2132	1771
Pesquisa Randômica	653	470

Fonte: O Autor (2023)

Através os três cenários de aplicações possíveis, os modelos destaques ao longo dos cinco rounds de treinamento tem seu ajuste fino de hiperparâmetros exibida na Tabela 29, assim como sua respectivas métricas de desempenho e custo computacional. Duas configuração foram destacadas, a utilização de PCA como método de redutor de dimensionalidade das características voltados para modelos com melhores métricas de desempenhos *accuracy* enquanto a utilização de clusterização de características elaboram modelos com um menor custo computacional.

Tabela 29 – Configurações de hiperparâmetros de Regressão Logística destacadas em relação aos possíveis cenários.

	Performance	Custo Computacional	Performance-Custo Computacional
Penalidade l2	Ausente	Ausente	Ausente
Solver	SAG	SAG	SAG
C	0.1	2	2
Redução de Dimensionalidade	PCA	Clusters Características	Clusters Características
Método de Hiperparametrização	Randômico	Randômico	Randômico
<i>Accuracy</i> [%]	89.3	60.98	60.98
Custo Computacional[s]	525	48	48

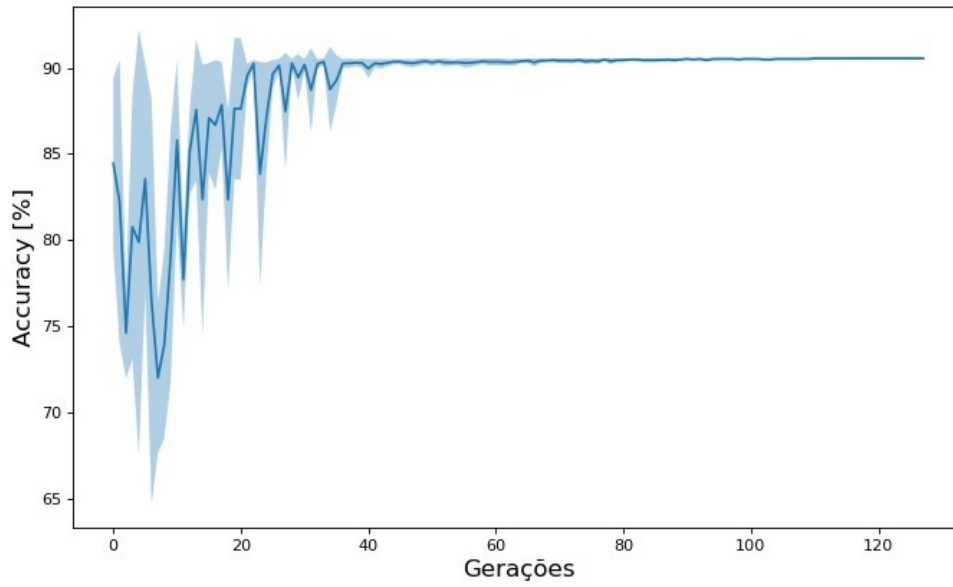
Fonte: O Autor (2023)

#### 4.4 XGBOOST

Ao implementar os sistemas propostos no capítulo anterior para o modelo de XGBoost, foi possível verificar resultados adequados à implementação através de cinco rodadas de repetições entre as combinações dos dois métodos de redução de dimensionalidade e as duas metodologias de ajuste fino de hiperparâmetros.

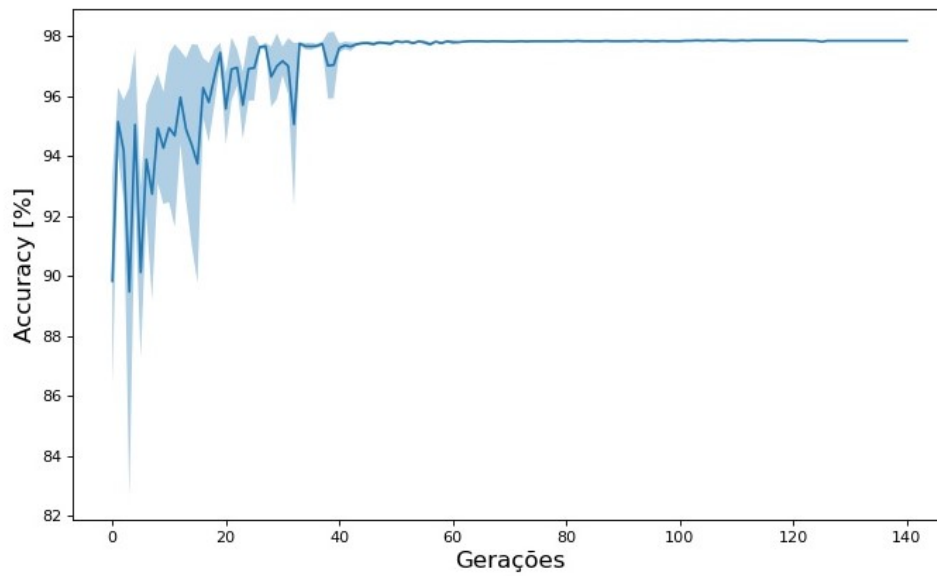
As curvas de *accuracy* médias entre as rodadas de repetições, ao longo do processo de configuração de hiperparâmetros, são exibidos pelas Figuras 48, 49, 50, 51, para o conjunto de validação dos dados, sintetizando seu valores médio pela curva destacada e o respectivo desvio padrão de cada época através do destaque da sua região.

Figura 48 – XGBOOST Clusterização de Características com Algoritmo Genérico.



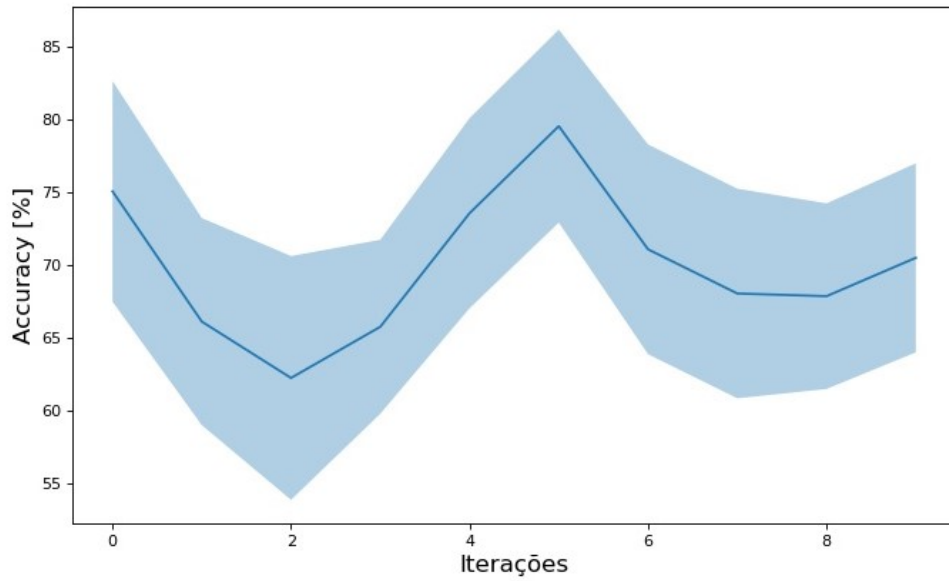
Fonte: O Autor.

Figura 49 – XGBOOST PCA com Algoritmo Genérico.



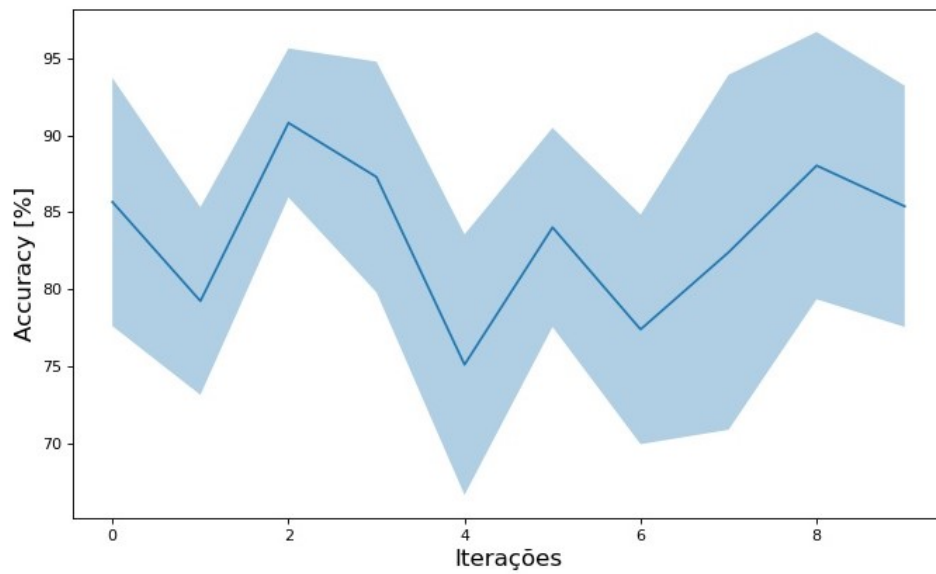
Fonte: O Autor.

Figura 50 – XGBOOST Clusterização de Características com Pesquisa Randômica.



Fonte: O Autor.

Figura 51 – XGBOOST PCA com Pesquisa Randômica.



Fonte: O Autor.

A partir dos processos de treinamento realizados, os melhores modelos foram avaliados com o conjunto de teste e tiveram sua métrica avaliativa referente à performance e custo computacional comparadas através das Tabelas 30 e 31 respectivamente.

Tabela 30 – Accuracy Médio XGBOOST[%].

	PCA	Clusterização de Características
Algoritmo Genético	98,28	92,11
Pesquisa Randômica	97,34	87,80

Fonte: O Autor (2023)

Tabela 31 – Custo Computacional Médio XGBOOST [s].

	PCA	Clusterização de Características
Algoritmo Genético	37017	18974
Pesquisa Randômica	3624	2629

Fonte: O Autor (2023)

Através os três cenários de aplicações possíveis, os modelos destaques ao longo dos cinco rounds de treinamento tem seu ajuste fino de hiperparâmetros exibida na Tabela 32, assim como sua respectivas métricas de desempenho e custo computacional. Três configuração foram destacadas, a utilização da clusterização de características auxiliou para a geração de modelos voltados para baixo custo computacional na etapa de treinamento, embora a utilização de PCA como método de redutor de dimensionalidade das características, produziu modelos com melhores métricas de desempenhos *accuracy* e melhor relação entre custo computacional e performance.

Tabela 32 – Configurações de hiperparâmetros de XGBOOST destacadas em relação aos possíveis cenários.

	Performance	Custo Computacional	Performance-Custo Computacional
Profundidade Máxima	6	5	4
Número de Estimadores	1024	256	256
Taxa de Aprendizado	0.5	0.5	0.5
Taxa de Subamostragem	1	0.7	0.3
Redução de Dimensionalidade	PCA	Clusters Características	PCA
Método de Hiperparametrização	Genético	Randômico	Randômico
<i>Accuracy</i> [%]	98.27	84.92	96.69
Custo Computacional [s]	30057	1065	1245

Fonte: O Autor (2023)

#### 4.5 COMPARAÇÕES ENTRE OS MODELOS E TESTES ESTATÍSTICOS

A partir dos três cenários abordados, é possível determinar os modelos mais indicados respectivamente. Dessa maneira, a tabela 33 os organiza, referenciando ao cenário em questão.



Tabela 33 – Comparação métricas.

Modelo	Cenário
XGBoost-PCA Evolutivo	Alta Performance
Floresta Aleatória-Cluster Pesquisa Randômica	Baixo Custo Computacional
Floresta Aleatória-Cluster Pesquisa Randômica	Máxima Performance em função do Custo Computacional

Fonte: O Autor (2023)

Através das Figuras do processo de configuração de hiperparâmetros é possível notar que a utilização de metaheurística para ajuste fino dos modelos, acataram em um maior tempo de processamento, uma vez que houveram maior número de iterações que a pesquisa randômica, no entanto a convergência entre as métricas de desempenho, ao longo dos testes realizados, ao final do ajuste detiveram um desvio padrão muito menor em todos os casos, garantido-se como um método mais eficaz, confiável e portanto aconselhável para ajuste de hiperparâmetros dos modelos investigados.

Para relacionar os resultados obtidos nas métricas de performance e custo computacional dos modelos, elaborou-se um teste DoE fatorial completo utilizando como fatores controláveis os modelos, técnicas de redução de dimensionalidade e metodologia de ajuste fino dos hiperparâmetros utilizados.

Os dados utilizados para elaboração do DoE foram adquiridos aleatoriamente e os respectivos modelos produzidos pelo teste de hipótese tiveram sua normalidade da distribuição de resíduos validados através do teste de Ryan-Joiner, possuindo constante RJ maior que 0.05 indicando a não-violação da premissa de normalidade da distribuição de ruído para os modelos gerados, dentro do intervalo de confiança de 95%, assim como para o teste de Levene que validou a igualdade de variâncias do modelo produzido dentro do mesmo intervalo de confiança. Dessa forma pode-se considerar os resultados da ANOVA, exibidos na Tabela 34 e 35 como exatos, uma vez que suas premissas foram respeitadas.

Tabela 34 – Resultados da Análise de Variância do DoE proposto à desempenho dos modelos.

Fator Controlável	Valor-p
Modelo	0.00
Método Redução dimensionalidade	0.00
Método de Ajuste de Hiperparâmetro	0.01
Modelo*Método Redução dimensionalidade	0.00
Modelo*Método de Ajuste de Hiperparâmetro	0.65
Método Redução dimensionalidade*Método de Ajuste de Hiperparâmetro	0.18
Modelo*Método Redução dimensionalidade*Método de Ajuste de Hiperparâmetro	0.46

Fonte: O Autor (2023)

Tabela 35 – Resultados da Análise de Variância do DoE proposto ao custo dos modelos

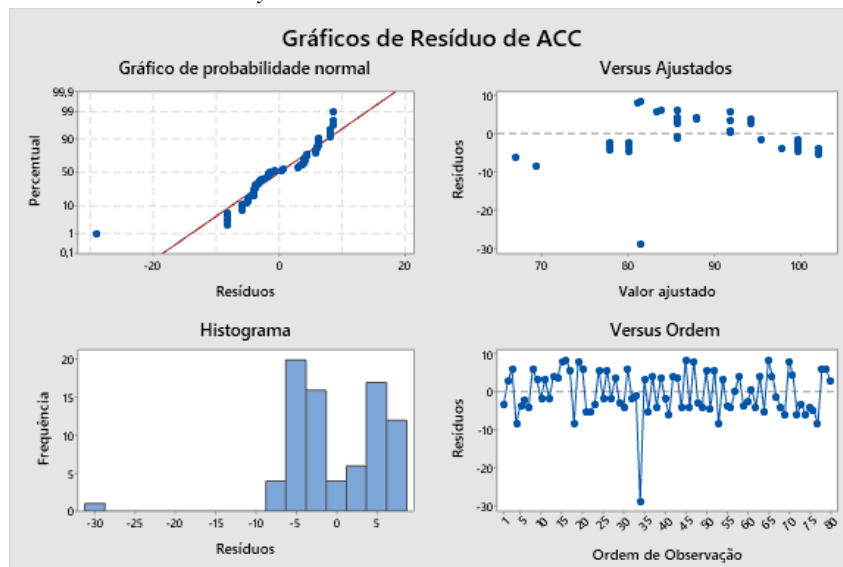
Fator Controlável	Valor-p
Modelo	0.00
Método Redução dimensionalidade	0.00
Método de Ajuste de Hiperparâmetro	0.01
Modelo*Método Redução dimensionalidade	0.00
Modelo*Método de Ajuste de Hiperparâmetro	0.00
Método Redução dimensionalidade*Método de Ajuste de Hiperparâmetro	0.00
Modelo*Método Redução dimensionalidade*Método de Ajuste de Hiperparâmetro	0.00

Fonte: O Autor (2023)

A partir dos valores gerados pelo DoE é possível validar a hipótese que o modelo utilizado, assim como seu método de redução de dimensionalidade e metodologia de ajuste fino de hiperparâmetros influenciam tanto na performance quanto custo computacional do treinamento dos modelos. No entanto as combinações inter-fatores não refletem esse comportamento em todos os casos, quando observa-se a performance do modelo, uma vez que os valores-p maiores 0.05 em quatro das cinco possíveis combinações de fatores.

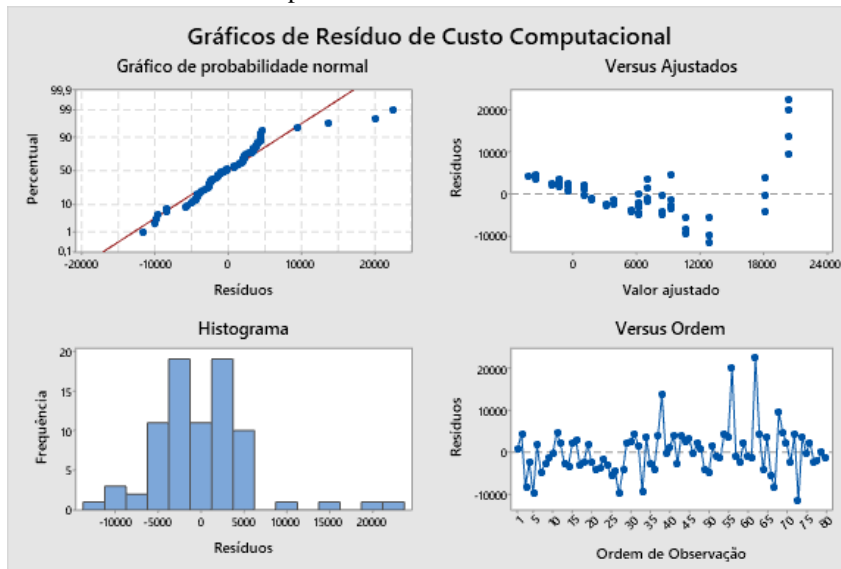
Para ilustrar o comportamento dos modelos produzidos pelo DoE, as Figuras e exibem suas de distribuição de resíduos.

Figura 52 – Resíduos - Modelo *Accuracy*.



Fonte: O Autor.

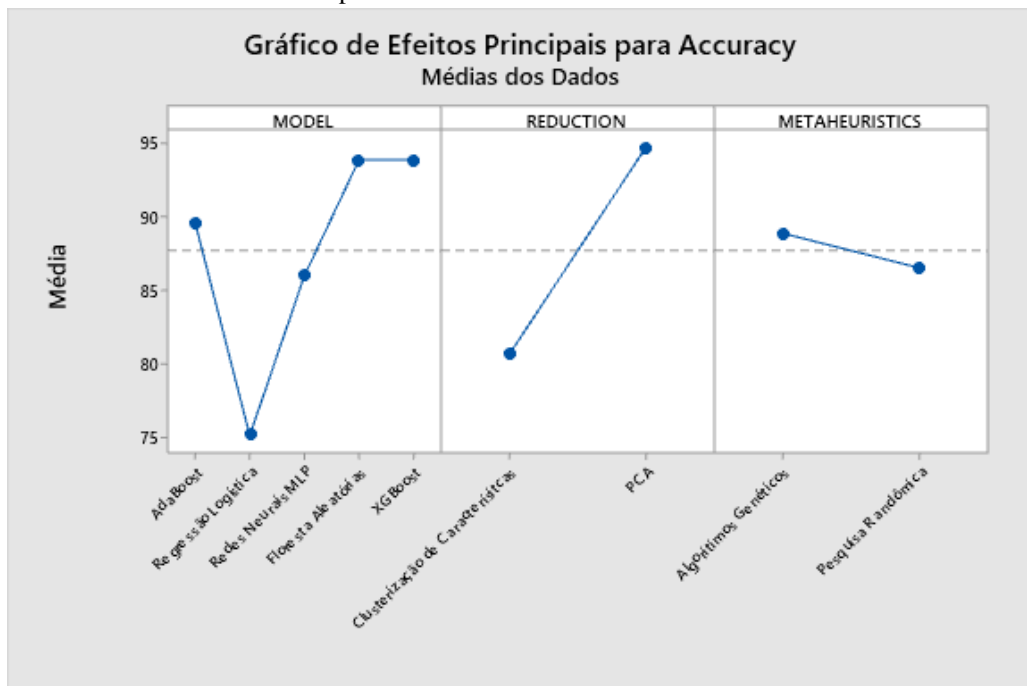
Figura 53 – Resíduos - Modelo Custo Computacional.



Fonte: O Autor.

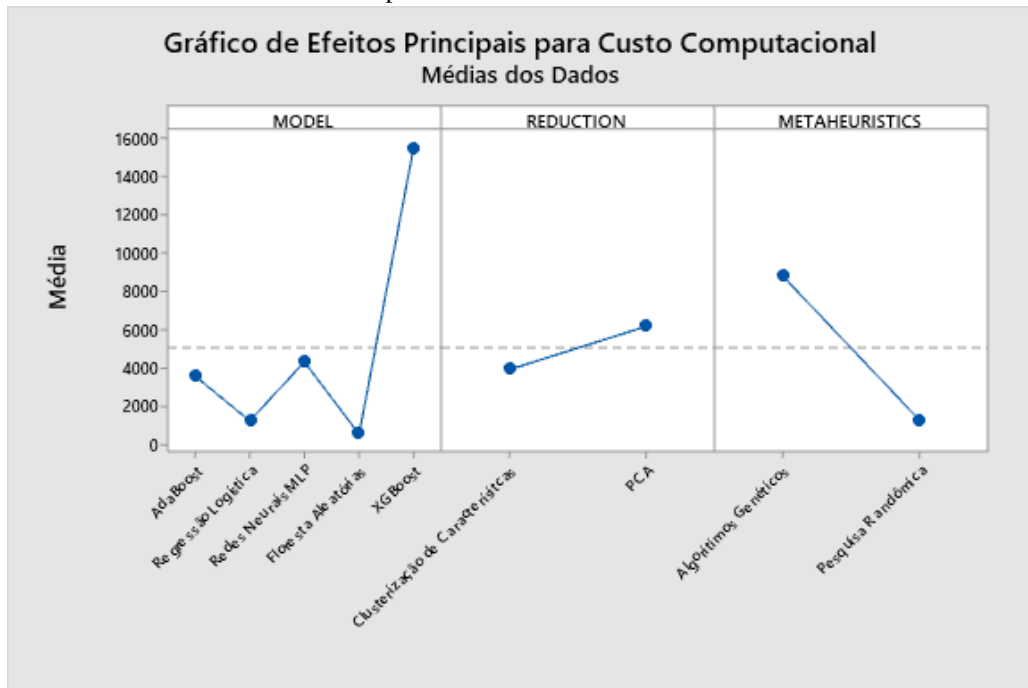
As Figuras 54 e 55, por sua vez ilustram os gráficos de efeitos entre os fatores controlados e as médias de *accuracy* e custo computacional, respectivamente, onde se exhibe a maior performance entre os modelos XGBoost e Floresta Aleatória com o uso de PCA e uso de Algoritmo genético e o maior custo computacional entre XGBoost com o mesmo método PCA e uso de Algoritmo genético.

Figura 54 – Gráfico de Efeitos - Desempenho.



Fonte: O Autor.

Figura 55 – Gráfico de Efeitos - Custo Computacional.



Fonte: O Autor.

Com base na bibliografia previamente analisada e discutida sucintamente no capítulo 2.4.1, verificou-se uma discrepância nas etapas de pré-processamento nas séries temporais trabalhadas. Nos trabalhos analisados, optou-se por abordagens que utilizaram redes recorrentes ou convolucionais, evitando a utilização de técnicas de extração de características. Como resultado, os modelos obtidos foram mais custosos computacionalmente. As faixas de performance obtidas situam-se entre 92.83% e 99.82% de *accuracy*, conforme ilustrado na Tabela 36, que destaca os principais artigos que utilizaram métodos inteligentes profundos com a mesma métrica avaliativa da monografia. É possível inferir que os resultados obtidos por estes estudos foram próximos aos resultados do presente trabalho, uma vez que diversas configurações avaliadas obtiveram *accuracy* médios acima de 92.83%, destacando os classificadores XGBoost Evolutivos que superaram o percentual de 98%.

Tabela 36 – Resultados Modelos referenciados.

Artigos	Accuracy
Wang et al. (2018)	99.68%
Wen, Gao e Li (2017)	99.82%
Glowacz et al. (2018)	92.83%
Jia et al. (2018)	99.22%
Shao et al. (2018)	99.76%

Fonte: Shao et al. (2018)

## 5 CONCLUSÃO

Este trabalho de conclusão de curso descreveu o processo de elaboração comparativa de um sistema de classificação inteligente utilizando como métodos inteligentes: RNA MLP, AdaBoost, Florestas Aleatórias, XGBOOST e Regressão Logística para predição de falhas em motores de indução trifásicos do tipo desbalanço de carga, desalinhamento vertical e horizontal no eixo, falhas nos rolamentos posteriores e inferiores da transmissão, podendo ser tanto no anel externo e interno dos rolamentos quanto nas esferas.

As falhas em máquinas rotativas podem ser analisadas pela ótica econômica, a qual indica uma contração na sustentabilidade da operação fabril, devido a parada da produção e danos a cargas acopladas e substituição do próprio maquinário. Através da ótica da saúde estrutural do ambiente instalado e do maquinário em si, justamente por motores em falhas serem consideradas fontes de vibrações excessivas que relativo a maneira e local de instalação pode apresentar sérios perigos a toda construção e operadores nela inserido, devido a possibilidade de ressonância entre a vibração do maquinário e a frequência natural de oscilação da estrutura. O impacto das falhas de motores no ambiente fabril é perceptível, tornando necessário a utilização de análises que prevejam-as e assim possam impedi-las, antes mesmo que as falhas ocorram Spamer (2009).

Análise Exploratória dos dados provenientes do *dataset* MaFaulDa, foram responsáveis por determinar as variáveis de entrada do sistema - curtose, amplitude média RMS, entropia de shannon, magnitude espectral média, máxima, centróide e curtose espectral. A segmentação de 10 ms com sobreposição de 10% para as séries temporais apresentaram bons resultados na implementação final, assim como o impacto realizado pela análise de componentes principais (PCA) conseguindo reduzir 85% dos dados mantendo uma covariância de 99% do conjunto de dados originais, atenuando consideravelmente a carga computacional do sistema, assim como a utilização de Aglomeração de Característica que conseguiu reduzir a mesma quantidade características, embora os testes elucidaram a utilização do PCA como método de redução de característica com maior taxa de *accuracy* para o modelo observado.

Ao revisar a literatura pertinente, foi constatado uma discrepância nas etapas de pré-processamento das séries temporais empregadas em outros estudos. Em geral, os modelos utilizaram redes recorrentes ou convolucionais, o que aumentou significativamente os custos computacionais. As faixas de desempenho obtidas oscilaram entre 92.83% e 99.82% de acurácia. Contudo, os modelos propostos neste trabalho adotaram uma abordagem mais eficiente, ao

incorporar a extração de características na etapa de processamento da série temporal, em contraposição à extração interna ao aprendizado dos modelos de referência. Como resultado, foi possível empregar técnicas com o mesmo nível de generalização para as dez classes de operação tratadas no projeto, com maior eficiência computacional e indicado para sistemas embarcados.

Um dos pontos levantados no estudo é o impacto de diferentes modelos inteligentes, destacando sistemas de metaclassificadores como *Random Forest* e XGBOOST, ambos com ajuste fino de hiperparâmetros otimizados por algoritmo genético ou pesquisa randômica, alcançando métricas de *accuracy* equivalentes a até 98%, com processo de treinamento de apenas 47 segundos. Isso valida estatisticamente os principais fatores influentes para as métricas performáticas dos modelos, produzindo aplicações adequadas para o ambiente industrial.

Portanto, reforçado pela pesquisa bibliométrica elaborada e o design de experimentos realizado como forma de validar a metodologia desta monografia, é possível afirmar que os resultados obtidos de desempenho dos modelos estão dentro da faixa de valores dos trabalhos publicados referidos, no entanto, com a proposta da utilização de métodos de aprendizado de máquinas clássicos, gerando modelos com menor carga computacional e, conseqüentemente, mais indicados para implementação de sistemas embarcados do que os modelos profundos, como as redes CNN, destacados nas publicações referenciadas

## 6 TRABALHOS FUTUROS

Algumas recomendações para o incremento dessa monografia para trabalhos futuros seria a incrementação do DoE viabilizando análise de potência estatísticas reforçando assim as conclusões obtidas pelos teste ANOVA. Assim como a realização de um estudo que compare além dos fator observados no presente trabalho como modelo inteligente, o algoritmo de redução de dimensionalidade e a metodologia de ajuste de hiperparâmetros, mas que avalie as diferenças no treinamento utilizando bases balanceadas e não balanceadas.

Uma terceira sugestão para implementação em trabalhos futuros leva em conta os resultados obtidos para elaborar a implementação dos modelos em um sistema embarcado

## REFERÊNCIAS

- ABB (EUA). **Motors don't just fail...do they?: A guide to preventing failure**. [S.l.]: ABB Release, 2015.
- AMBRÓSIO, Paulo Eduardo. **Redes neurais artificiais no apoio ao diagnóstico diferencial de lesões intersticiais pulmonares**. Tese (Doutorado) — Universidade de São Paulo, 2002.
- BAZURTO, Alvaro Jose; QUISPE, Enrique Ciro; MENDOZA, Rosaura Castrillon. Causes and failures classification of industrial electric motor. In: **2016 IEEE ANDESCON**. [S.l.: s.n.], 2016. p. 1–4.
- BEARDS, C.F. **Structural Vibration: Analysis and Damping**. Arnold, 1996. (Structural Vibration: Analysis and Damping). ISBN 9780340645802. Disponível em: <<https://books.google.com.br/books?id=tjaRX6fbfOYC>>.
- BEN-DAYA, M.B.D.; KUMAR, U.; MURTHY, D.N.P. **Introduction to Maintenance Engineering: Modeling, Optimization, and Management**. Wiley, 2016. ISBN 9781118926581. Disponível em: <<https://books.google.com.br/books?id=AVvDjwEACAAJ>>.
- BERGSTRA, James; BENGIO, Yoshua. Random search for hyper-parameter optimization. **Journal of machine learning research**, v. 13, n. 2, 2012.
- BHOWMIK, Partha Sarathee; PRADHAN, Sourav; PRAKASH, Mangal. Fault diagnostic and monitoring methods of induction motor: a review. **International Journal of Applied Control, Electrical and Electronics Engineering (IJACEEE)**, v. 1, n. 1, p. 1–18, 2013.
- BINDU, S; THOMAS, Vinod V. Diagnoses of internal faults of three phase squirrel cage induction motor — a review. In: **2014 International Conference on Advances in Energy Conversion Technologies (ICAECT)**. [S.l.: s.n.], 2014. p. 48–54.
- BREIMAN, Leo. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.
- BREIMAN, Leo et al. **Classification and regression trees**. [S.l.]: Routledge, 2017.
- CARDOSO, Marcela de Oliveira. Indústria 4.0: a quarta revolução industrial. **Universidade Tecnológica Federal do Paraná**, 2016.
- CHAPMAN, S.J. **Fundamentos de Máquinas Elétricas - 5ed**. AMGH Editora, 2013. ISBN 9788580552072. Disponível em: <<https://books.google.com.br/books?id=Fyg6AgAAQBAJ>>.
- CHEN, Tianqi; GUESTRIN, Carlos. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. [S.l.: s.n.], 2016. p. 785–794.
- COSTA, Cesar da; KASHIWAGI, Masamori; MATHIAS, Mauro Hugo. Rotor failure detection of induction motors by wavelet transform and fourier transform in non-stationary condition. **Case Studies in Mechanical Systems and Signal Processing**, Elsevier, v. 1, p. 15–26, 2015.
- CUTLER, D Richard et al. Random forests for classification in ecology. **Ecology**, Wiley Online Library, v. 88, n. 11, p. 2783–2792, 2007.



FARRAR, Charles R; WORDEN, Keith. An introduction to structural health monitoring. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, The Royal Society London, v. 365, n. 1851, p. 303–315, 2007.

FLECK, Leandro et al. Redes neurais artificiais: Princípios básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, v. 1, n. 13, p. 47–57, 2016.

FREUND, Yoav; SCHAPIRE, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. **Journal of computer and system sciences**, Elsevier, v. 55, n. 1, p. 119–139, 1997.

GHANBARI, Teymoor. Autocorrelation function-based technique for stator turn-fault detection of induction motor. **IET Science, Measurement & Technology**, IET, v. 10, n. 2, p. 100–110, 2016.

GLOWACZ, Adam et al. Early fault diagnosis of bearing and stator faults of the single-phase induction motor using acoustic signals. **Measurement**, Elsevier, v. 113, p. 1–9, 2018.

GRUS, Joel. **Data Science from Scratch**. [S.l.]: O’Reilly, 2019.

GUEDES, Manuel Vaz. O motor de indução trifásico: Seleção e aplicação. **Faculdade de Engenharia da Universidade do Porto**, 1994.

HUNG, Wen-Liang; LEE, E Stanley; CHUANG, Shun-Chin. Balanced bootstrap resampling method for neural model selection. **Computers & Mathematics with Applications**, Elsevier, v. 62, n. 12, p. 4576–4581, 2011.

IEEE Motor Reliability Working Group. Report of large motor reliability survey of industrial and commercial installations, part i. **IEEE Transactions on Industry Applications**, IA-21, n. 4, p. 853–864, 1985.

JIA, Feng et al. Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. **Mechanical Systems and Signal Processing**, Elsevier, v. 110, p. 349–367, 2018.

JIANG, Guoqian et al. Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox. **IEEE Transactions on Industrial Electronics**, IEEE, v. 66, n. 4, p. 3196–3207, 2018.

KHAN, Asif; HWANG, Hyunho; KIM, Heung Soo. Synthetic data augmentation and deep learning for the fault diagnosis of rotating machines. **Mathematics**, v. 9, n. 18, 2021. ISSN 2227-7390. Disponível em: <<https://www.mdpi.com/2227-7390/9/18/2336>>.

LI, Xiang; DING, Qian; SUN, Jian-Qiao. Remaining useful life estimation in prognostics using deep convolution neural networks. **Reliability Engineering & System Safety**, Elsevier, v. 172, p. 1–11, 2018.

LIASHCHYNSKYI, Petro; LIASHCHYNSKYI, Pavlo. Grid search, random search, genetic algorithm: a big comparison for nas. **arXiv preprint arXiv:1912.06059**, 2019.

LIBONI, Luisa Helena Bartocci. **Diagnóstico de falhas em motores de indução trifásicos baseado em decomposição em componentes ortogonais e aprendizagem de máquinas**. Tese (Doutorado) — Universidade de São Paulo, 2017.

MARR, Bernard. What is industry 4.0? here's a super easy explanation for anyone. **Forbes**, p. 1-2, 2018.

MATIĆ, Dragan et al. Support vector machine classifier for diagnosis in electrical machines: Application to broken bar. **Expert Systems with Applications**, Elsevier, v. 39, n. 10, p. 8681–8689, 2012.

MATTEDE, Henrique. **Motor trifásico, características e aplicações!** 2020. Disponível em: <<https://www.mundodaeletrica.com.br/motor-trifasico-caracteristicas-aplicacoes/>>.

MCCULLOCH, Warren S; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.

NANDI, Subhasis; TOLIYAT, Hamid A; LI, Xiaodong. Condition monitoring and fault diagnosis of electrical motors—a review. **IEEE transactions on energy conversion**, IEEE, v. 20, n. 4, p. 719–729, 2005.

NIED, Ademir. Treinamento de redes neurais artificiais baseado em sistemas de estrutura variável com taxa de aprendizado adaptativa. Universidade Federal de Minas Gerais, 2007.

PARK, Cheong Hee. A feature selection method using hierarchical clustering. In: **Mining intelligence and knowledge exploration**. [S.l.]: Springer, 2013. p. 1–6.

PEARSON, Karl. “DAS FEHLERGESETZ UND SEINE VERALLGEMEINER-UNGEN DURCH FECHNER UND PEARSON\*.” A REJOINER. **Biometrika**, v. 4, n. 1-2, p. 169–212, 06 1905. ISSN 0006-3444. Disponível em: <<https://doi.org/10.1093/biomet/4.1-2.169>>.

PEREIRA; GOMES, Thiago Silva; COUTINHO, Luciano. Análise da geometria e do fluxo magnético do motor síncrono á relutância auxiliado por imã permanente. In: **Conferência de Estudos de Engenharia Elétrica**. Uberlândia, MG: [s.n.], 2019.

PINHEIRO, Hélio Henrique Cunha; MAITELLI, André Laurindo; SALAZAR, Andrés Ortiz. Sistema para detecção e diagnóstico de falhas em motores elétricos de indução utilizando fasores e lógica fuzzy. In: **Congresso Brasileiro de Inteligência Computacional (CBIC 2011)**. [S.l.: s.n.], 2011.

RIBEIRO, Jéssica Mariana. Análise de falhas em motores de indução trifásica, através do método de detecção de falhas análise de vibração, utilizando controle por lógica fuzzy. Fundação de Ensino e Pesquisa do Sul de Minas, 2016.

ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.

SHAO, Siyu et al. Highly accurate machine fault diagnosis using deep transfer learning. **IEEE Transactions on Industrial Informatics**, IEEE, v. 15, n. 4, p. 2446–2455, 2018.

SIDDIQUE, Arfat; YADAVA, GS; SINGH, Bhim. A review of stator fault monitoring techniques of induction motors. **IEEE transactions on energy conversion**, IEEE, v. 20, n. 1, p. 106–114, 2005.

SPAMER, Fernanda Rosa. Técnicas preditivas de manutenção de máquinas rotativas. **Universidade Federal do Rio de Janeiro. Rio de Janeiro**, 2009.

SUN, Yanmin; WONG, Andrew KC; KAMEL, Mohamed S. Classification of imbalanced data: A review. **International journal of pattern recognition and artificial intelligence**, World Scientific, v. 23, n. 04, p. 687–719, 2009.

WANG, Jinjiang et al. Deep learning for smart manufacturing: Methods and applications. **Journal of manufacturing systems**, Elsevier, v. 48, p. 144–156, 2018.

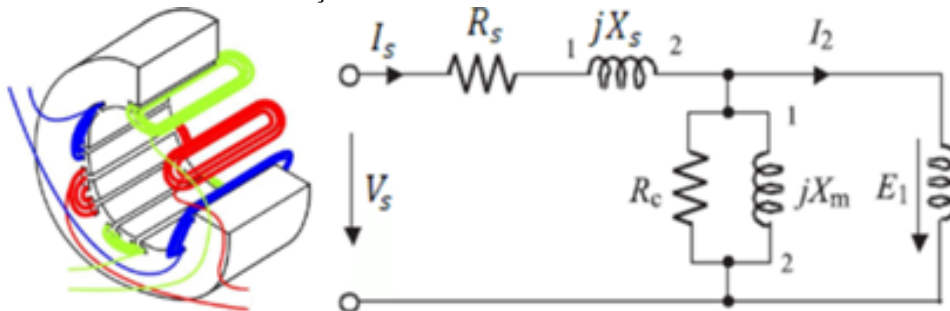
WEN, Long; GAO, Liang; LI, Xinyu. A new deep transfer learning based on sparse auto-encoder for fault diagnosis. **IEEE Transactions on systems, man, and cybernetics: systems**, IEEE, v. 49, n. 1, p. 136–144, 2017.

## MOTOR ELÉTRICOS DE INDUÇÃO TRIFÁSICA

### .1 MOTOR DE INDUÇÃO TRIFÁSICO

Motor de Indução trifásico é considerado a máquina elétrica de corrente alternada mais utilizada nos sistemas eletromecânicos devido sua robustez e simplicidade de manutenção (GUEDES, 1994), com aspecto construtivo simples, baixo custo e alta versatilidade (RIBEIRO, 2016). Para analisar seu funcionamento inicialmente divide-se em dois principais segmentos mecânicos: o estator e o rotor. O estator é formado de um núcleo ferro laminado com ranhuras que garantem fixação dos enrolamentos indutivos para cada uma das três fases, como a Figura 56 sugere contrastando o modelo real e equivalente em circuito para uma fase.

Figura 56 – Estator de um motor de indução trifásico.



Fonte: Adaptado de Pereira, Gomes e Coutinho (2019).

O estator é energizado por uma fonte de tensão elétrica e conseqüentemente é percorrido por uma intensidade de corrente elétrica  $I$ . A Equação (1) descreve a matriz correspondente de cada uma das fases (A, B e C) do estator.  $I_p$  define o valor de pico da corrente elétrica, a qual é descrita como uma senoide de frequência angular  $\omega$  sob o domínio do tempo  $t$ , descrevendo sua unidade de medida em Ampére [A].

$$\begin{bmatrix} I_{\phi A} \\ I_{\phi B} \\ I_{\phi C} \end{bmatrix} = I_p \begin{bmatrix} \sin(\omega t) \\ \sin(\omega t - 120^\circ) \\ \sin(\omega t - 240^\circ) \end{bmatrix} \quad [A] \quad (1)$$

O fluxo de corrente elétrica vigente sob as espiras do estator produz um campo magnético  $H$ , descrito pela Lei de Ampère, exibida na Equação (2), como um vetor de intensidade resultante da quantidade líquida de corrente passante em uma área fechada, denominada Amperiana ( $\partial R$ ) e ortogonal ao sentido do fluxo de corrente.

$$\oint_{\partial R} \vec{H} \cdot d\vec{l} = I \quad [A] \quad (2)$$

Portanto, pode-se determinar a produção de campo magnético estático, descrito pela Equação (3). Definindo campo H em cada uma das fases, com comportamento senoidal e valor de pico definido por  $H_p$ , em unidades de Ampere-Espira por metro.

$$\begin{bmatrix} \vec{H}_{\phi A} \\ \vec{H}_{\phi B} \\ \vec{H}_{\phi C} \end{bmatrix} = I_p \begin{bmatrix} \sin(\omega t) \angle 0^\circ \\ \sin(\omega t - 120^\circ) \angle 120^\circ \\ \sin(\omega t - 240^\circ) \angle 240^\circ \end{bmatrix} \left[ A \cdot \frac{e}{m} \right] \quad (3)$$

A conversão das grandezas de intensidade de campo magnético H e o vetor de indução magnética B, com unidade de medida em Tesla [T] é descrita na Equação (4) através da transformada linear dada pela constante de permeabilidade magnética  $\mu$  e o vetor de intensidade de campo magnético H. Permitindo estabelecer a equação de matricial-vetorial da indução magnética em uma espira, definida pela Equação (5).

$$\vec{B} = \mu \vec{H} \quad (4)$$

$$\begin{bmatrix} \vec{B}_{\phi A} \\ \vec{B}_{\phi B} \\ \vec{B}_{\phi C} \end{bmatrix} = \mu H_p \begin{bmatrix} \sin(\omega t) \angle 0^\circ \\ \sin(\omega t - 120^\circ) \angle 120^\circ \\ \sin(\omega t - 240^\circ) \angle 240^\circ \end{bmatrix} \left[ A \cdot \frac{e}{m} \right] \quad (5)$$

Para determinar a quantidade resultante de fluxo magnético estático  $B_s$ , deve-se somar os três campos produzidos em cada uma das componentes do plano x-y, como ilustram as Equações (6) a (8):

$$\vec{B}_s = \sum_n^N \vec{B}_n = B_p [\sin(\omega t) \angle 0^\circ + \sin(\omega t - 120^\circ) \angle 120^\circ + \sin(\omega t - 240^\circ) \angle 240^\circ] \quad (6)$$

$$\alpha_\Theta = \sin(\omega t - 120^\circ) + \sin(\omega t - 240^\circ) \quad (7)$$

$$\vec{B}_s = B_p \left[ \left( \sin(\omega t) - \frac{1}{2} \alpha_\Theta \right) \hat{x} + \left( \frac{\sqrt{3}}{2} \alpha_\Theta \right) \hat{y} \right] [T] \quad (8)$$

Utilizando identidades trigonométricas é possível estabelecer o vetor resultantes como mostra a Equação (9).

$$\vec{B}_s = \frac{3}{2} B_p [\sin(\omega t) \hat{x} - \cos(\omega t) \hat{y}] [T] \quad (9)$$

Conclui-se então que a o campo resultante produzido pela energização trifásica do estator, apresentará o mesmo valor 1.5 Bp de intensidade e continuará girando com a velocidade angular ao longo da distribuição das espiras. Com a produção de um campo magnético variante dentro do maquinário, através da Lei de Faraday, ilustrada na Equação (10), assumisse que qualquer condutor ao efeito de uma variação temporal de fluxo magnético, definido pela integral de superfície, terá induzido uma tensão elétrica  $\vec{E}$  proporcional à sua intensidade.

$$\oint_{\partial S} \vec{E} \cdot d\vec{l} = \frac{\partial}{\partial t} \iint_{\Sigma} \vec{B} \cdot d\vec{S} \quad [V] \quad (10)$$

A produção dessa tensão induzida em um laço de condutor fechado  $\partial s$  é percebida como uma corrente elétrica variante que por sua vez produz um segundo campo magnético que tenderá a perseguir seu campo opositor (SPAMER, 2009). A produção do torque resultante entre a interação dos campos de indução magnéticas do rotor e estator são definidos pela Lei de Lorentz através das Equações (11) e (12), indicado a produção de torque ao longo de um braço de alavanca de distância  $r$  e uma força  $F$ , gerada através da interação de um campo de indução magnética  $B$  e uma corrente elétrica  $i$  que flui no sentido do caminho  $l$  de um fio. Resultando no torque eletromagnético final  $\tau$  descrito pela Equação (13).

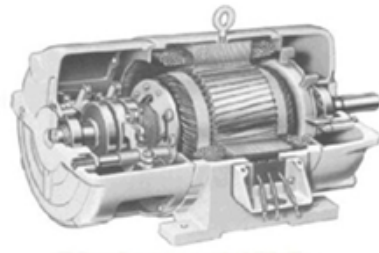
$$\vec{F} = i\vec{l} \times \vec{B} \quad (11)$$

$$\vec{\tau} = r \times \vec{F} \quad (12)$$

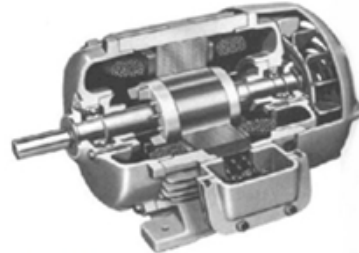
$$\vec{\tau} = k \left( \vec{B}_{estator} \times \vec{B}_{rotor} \right) \quad (13)$$

Na constituição do motor, o segundo agente construtivo relevante para a descrição de seu funcionamento atua como o laço condutor em curto, responsável pela produção do segundo campo magnético induzido pelo rotor e agente da produção de torque, é chamado de rotor e comumente elaborado como um conjunto de espiras igualmente deslocadas ou um conjunto de gaiola formadas por finas barras condutoras responsáveis por curto circuitar os dois anéis que compõem sua geometria (CHAPMAN, 2013). A Figura 57 ilustra os tipos de rotores citados.

Figura 57 – Comparativos entre tipos de rotores.



**Rotor bobinado**



**Rotor gaiola**

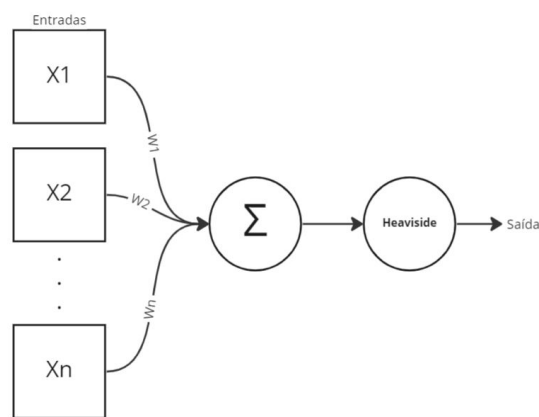
Fonte: Adaptado de Mattede (2020).

## MODELOS DE INTELIGENCIA COMPUTACIONAL

### .2 REDES NEURAIIS ARTIFICIAIS

O primeiro modelo matemático do neurônio biológico para aplicações em redes computacionais foi proposto por McCulloch e Pitts (1943), denominado neurônio McCulloch-Pitts, conceituado como um sistema de  $n$  entradas, multiplicadas por pesos  $w$  somadas a um comparador de limiar como a Figura 58 abaixo sugere

Figura 58 – Topologia Neurônio MCP.



Fonte: O Autor.

Inicialmente o modelo MCP foi concebido com a função de saída limiar respeitando o funcionamento de um neurônio biológico, porém após a proposta de topologia de Rosenblatt (1958) de um rede de interconexão de neurônios MCP, iniciou-se um processo de separação entre os estudos de elaboração de redes neurais artificiais e a mimesis dos sistemas biológicos (ROSENBLATT, 1958), destas diferenças destacam a função de ativação neuronal que não necessariamente deve ser um comparador limiar, podendo assumir outras configurações como a função logística, tangentes hiperbólicas, identidade entre outras comumente usadas em inúmeras aplicações. Assim o modelo de neurônio baseado nos MCP, denominados Perceptrons tem sua equação característica de saída y definida pela Equação 14

$$y = f \left( \sum_{i=0}^N w_i x_i \right) \quad (14)$$

A topologia apresentada por Rosenblatt (1958), é uma forma simples de RNA cuja principal aplicação se dá nos problemas de classificação de padrões (FLECK et al., 2016). Redes uma camada simples de perceptron são capazes de resolver problemas apenas linearmente sepa-

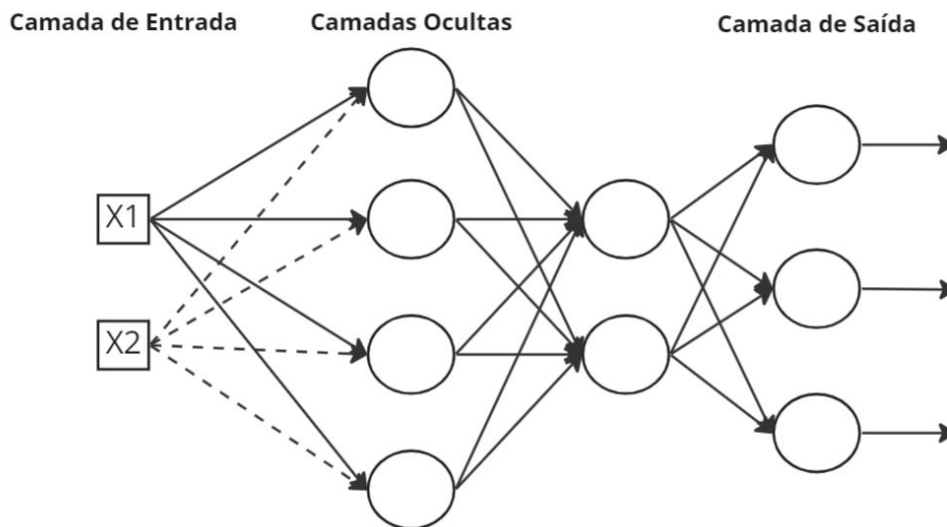


ráveis, tornando necessário a utilização de múltiplas camadas de processamento para conseguir abordar situações não-linearmente separáveis (AMBRÓSIO, 2002).

A topologia das redes MLP são elaboradas respeitando as segmentações de processamento, divididas em três camadas: entrada, intermediária(s) ou escondida(s) e camada de saída (NIED, 2007).

A composição de rede neural MLP é elaborada através do alinhamento das camadas neuronais e suas interligações, porém o processamento de informações não é uniformemente distribuído entre as camadas. A responsabilidade da camada de entrada é fornecer os dados a serem computados pelas camadas intermediárias, principal fonte de processamento sistêmico, para fornecer informações para a camada de saída, responsável por processar sua entrada de forma a estabelecer uma saída dentro do padrão fornecido pelos dados de supervisão. A Figura 59 ilustra a arquitetura de uma Rede Neural Artificial de Múltiplas Camadas de Perceptron (RNA MLP) composta de duas camadas ocultas

Figura 59 – Exemplo de Rede Neural MLP.



Fonte: O Autor.

A aprendizagem de uma rede neural MLP por algoritmo de *backpropagation* ou retropropagação é composta das etapas de propagação *feedforward* e retropropagação para ajustes dos pesos sinápticos.

A etapa de propagação *feedforward* é composta do disseminamento dos dados de entrada através do processamento neuronal, entre camadas até a camada de saída, produzindo a resposta da rede. Enquanto o estágio de retropropagação, há a alteração dos pesos das sinapses conforme a regra de aprendizagem estabelecida. Esta regra utiliza o sinal de erro propagado da camada

de saída em direção a camada de entrada de forma a atualizar os valores dos pesos visando a aproximação da resposta desejada (NIED, 2007).

As regras de aprendizagem ou ajuste de peso são inúmeras, neste trabalho utiliza-se o método do Gradiente Estocástico Descendente, do inglês *Stochastic Gradient Descent* (SGD) que consiste em encontrar o mínimo da função custo do sistema através de um laço iterativo que utiliza-se da variação do erro de saída para a topologia de pesos  $w(t)$  como forma de ajustar os mesmos para a próxima rodada de treinamento  $w(t + 1)$  para cada um dos neurônios indexados, como a Equação 15 sugere

$$w_i(t + 1) = w_i(t) + \eta \nabla_{w_i} L(w) \quad (15)$$

### .3 ÁRVORE DE DECISÃO E FLORESTAS ALEATÓRIAS

Árvore de decisão é um modelo estatístico utilizado em aprendizado supervisionado para a classificação e regressão de dados através da elaboração de regras de decisão simples inferidas pelo conjunto de características do sistema. A criação das regras baseia-se no particionamento binário recursivo em regiões com aumento de homogeneidade em relação à variável da classe de saída.

O processo de formação da árvore de decisão realiza uma otimização ao selecionar um novo nó (CUTLER et al., 2007) - região homogênea em relação a uma classe - resultando em subgrupos mais homogêneos para os dados ainda não bem descritos, quantificados através de métricas como a entropia ou índice de Gini (BREIMAN et al., 2017), complexificando a árvore até que novas subdivisões não reduzam mais a métrica visada.

Árvores de decisão são consideradas simples de entender e interpretar, requerendo pouca preparação nos conjuntos de dados, não necessitando de normalização e promovendo uma curva logarítmica de custo em relação ao número de dados de treinamento. Considerado um modelo caixa branca, no qual há facilidade na interpretação do processo de determinação da saída do modelo, podendo ser validado por testes estatísticos possibilitando estimar o nível de confiabilidade do modelo.

Os sistemas que utilizam árvores de decisão podem apresentar baixo poder de generalização, decorando o conjunto de treinamento ao invés de aprendê-lo, causando *overfitting*. Métodos como a poda, definindo o número mínimo de amostras necessárias em um nó ou a definição de profundidade máxima da árvore podem evitar este problema.

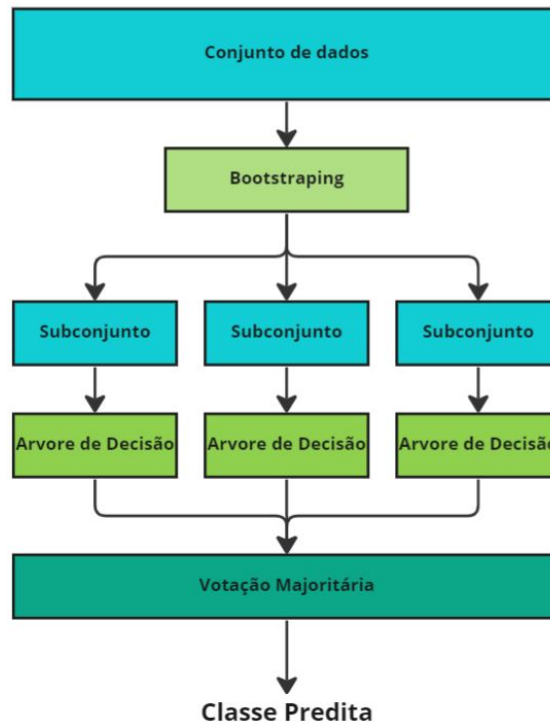
Treinamento de árvores de decisão podem apresentar instabilidades em razão a pequenas variações nos dados de treinamento que são potenciais agentes em gerações de árvores completamente diferentes. Esse problema é atenuado pelo uso de árvores de decisão em um conjunto.

Como forma de atenuar problemas de baixa generalização e da má distinção de mínimos locais e globais, é sugerido a utilização de modelos de metaclassificadores, nos quais diversas árvores são utilizadas como componentes de um sistema maior, como as florestas aleatórias.

Floresta aleatórias é um meta-algoritmo de aprendizagem de máquina, desenvolvido por Cutler et al. (2007) e Breiman (2001), responsável por combinar a saída de uma população e árvores de decisão através da moda das classes preditas, atuando de maneira análoga ao sistema de voto de peso único.

O algoritmo de treinamento para florestas aleatórias aplica a técnica de agregação por *bootstrap*, na qual o conjunto de treinamento é amostrado aleatoriamente e uma árvore de decisão, definida aleatoriamente e ajustada para sua respectiva partição para posteriormente fornecer sua predição para a etapa de votação, fornecendo a classe predita final do processo como a Figura 60 ilustra

Figura 60 – Topologia de Floresta Aleatória.



Fonte: O Autor.

Os principais hiperparâmetros das Florestas Aleatórias (Random Forest) são a presença

do *Bootstrap* na segmentação dos dados, a densidade máxima das árvores envolvidas no sistema, número máximo de nós, o número estimadores e métrica de avaliação, podendo

A presença do *Bootstrap* define a segmentação ou não do conjunto de treinamento aos estimadores. Densidade máxima da árvore, estipula a expansão máxima de nós ao longo das árvores elaboradas. O número máximo de nós define o número mínimo de amostras necessárias para configurar uma folha. O agente segmentador que relaciona as características de entrada com a formação de nós é feita através da métrica de avaliação, sendo o índice de gini ou valor de entropia do conjunto de dados.

#### .4 REGRESSÃO LOGÍSTICA

A Regressão Logística é um método que semelhante à Regressão Linear, estabelece uma relação entre as variáveis explicativas, no entanto, no lugar de retornar uma reta que melhor descreve os dados observados, a regressão logística estabelece o ponto no universo de discurso, que melhor segmenta as classes dentro do grupo de dados observado, explicitando-se a probabilidade do dado observado pertencer ou não a um fenômeno estudado

Uma das premissas da regressão logística binária é a respeito da distribuição binomial dos dados observados, garantindo a modelagem a partir do conjunto de variáveis independentes e suas respectivas com as probabilidade finais. Devido a não linearidade observada nas distribuições binomiais, sua linearização é feita através da aplicação de logaritmo, formando a Equação 16, como o principal modelo de regressões logísticas.

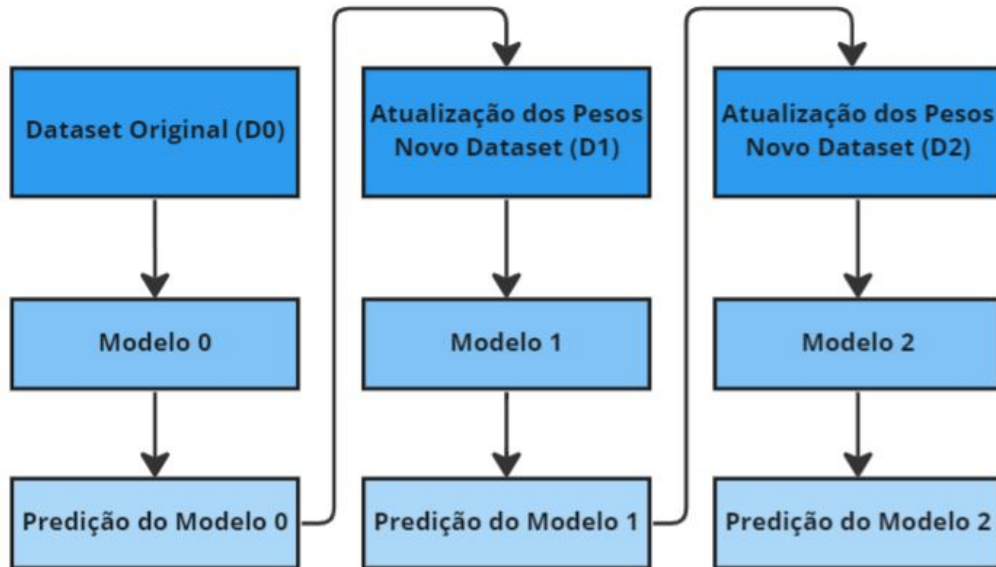
$$y = \frac{1}{1 + e^{-f(x)}} \quad (16)$$

#### .5 ADABOOST

Descrito pela primeira vez por Freund e Schapire (1997), o AdaBoost é um método de aprendizado de máquina, utilizado como um meta-classificador sequencial para maximizar a performance de um conjunto de algoritmos de aprendizagem bases (classificadores fracos/*weak learners*) através do aprendizado por reforço em cascada, no qual após todo treinamento de um *weak learners* há um ajuste no peso do conjunto de treinamento indicando quais amostras devem ter seu aprendizado reforçado na classificação subsequente com um novo *dataset* elaborado através através de uma seleção por roleta, onde o peso de cada uma das amostras indica sua probabilidade de escolha, produzindo assim um sistema encadeado de classificadores menores

que são adaptativamente elaborado conforme a resposta de seu antecessor, como a Figura 61 sugere

Figura 61 – Topologia do Adaboost.



Fonte: O Autor.

Seu funcionamento é baseado em classificações feitas em um processo de cascada, no qual as classificações subsequentes são ajustadas a favor das instâncias classificadas erroneamente pelos classificadores anteriores.

O Algoritmo itera repetidas vezes classificadores fracos, com atualização das distribuição de pesos amostrais, responsável por indicar a relevância da mesma no processo de aprendizagem por reforço, ao aumentando o peso para cada má classificação, aumenta-se a tendência de treinamento das amostras mal classificadas para o novo classificador trabalhe, conseguindo uma maior taxa de acerto global.

## .6 XGBOOST

XGBoost do inglês, *Extreme Gradient Boost* é um modelo de aprendizado de máquina proposto por Chen e Guestrin (2016), que utiliza a metodologia de *gradient boosting* na integração dos algoritmos de aprendizados bases (alunos fracos) para a formação do meta-classificador.

A formação do conjunto de treinamento dos *weak learners* não é feito por reamostras dos dados que necessitam de maior reforço de aprendizado como o *boosting* adaptativo, mas através *boosting* por gradiente, o qual define o aprendizado do aluno fraco no pseudo-resíduo obtido pela integração dos classificadores anteriores, dotando cada um dos alunos de capacida-

des fatores relevantes na contenção de más classificações através da variação residual gerada a partir da ação dos classificadores antecessores (CHEN; GUESTRIN, 2016).

No caso dos XGBoost, a principal diferença para um meta-classificador de árvores de decisão por *gradient boosting* seria a utilização de regularizadores L2 (CHEN; GUESTRIN, 2016), responsáveis por reduzir os índices de *overfitting* e tornar o modelo mais leve através da modulação da função custo, como a equação 17 sugere.

$$\begin{aligned} Loss &= \sum_i^N L(y, \hat{y}) + \sum_k^N \Omega(f_k) \\ \Omega(f_k) &= \gamma T + \frac{1}{2} \lambda \|w\|^2 \end{aligned} \quad (17)$$

## MÉTODOS DE OTIMIZAÇÃO DE HIPERPARÂMETROS

### .7 PESQUISA EXTENSIVA E PESQUISA RANDÔMICA

A pesquisa em grade é considerada como o método tradicional de otimização de hiperparâmetros (LIASHCHYNSKYI; LIASHCHYNSKYI, 2019). Fazendo a busca completa entre todas as combinações de hiperparâmetros pertencentes ao universo de discurso previamente delimitado e definindo a combinação mais adequada para o modelo em questão.

A pesquisa extensiva rapidamente pode ter seu custo computacional ampliado devido sua relação exponencial com a dimensão de hiperparâmetros estabelecida, não sendo considerada o método de otimização mais eficaz.

Pesquisa Randômica é proposta para substituir a seleção completa de todas as combinações de hiperparâmetros, como na pesquisa por grade, por uma seleção aleatória de subconjuntos dentro do universo de discurso delimitado previamente. Permitindo através da parametrização da busca uma maior controle do número de combinações a serem testadas, melhorando assim o custo computacional do processo.

Embora a pesquisa em grade consiga determinar o melhor resultado combinatório entre os hiperparâmetros através da testagem de cada uma das alternativas, a busca randomizada, apesar de não garantir o melhor dos resultados, tem um custo computacional significativamente reduzido e devido a distribuição probabilística da alocação de combinações ao longo do hiperplano ser uniforme é comum sua combinação resultante apresentar métricas satisfatórias na implementação do modelo (BERGSTRÄ; BENGIO, 2012)

### .8 ALGORITMOS GENÉTICOS

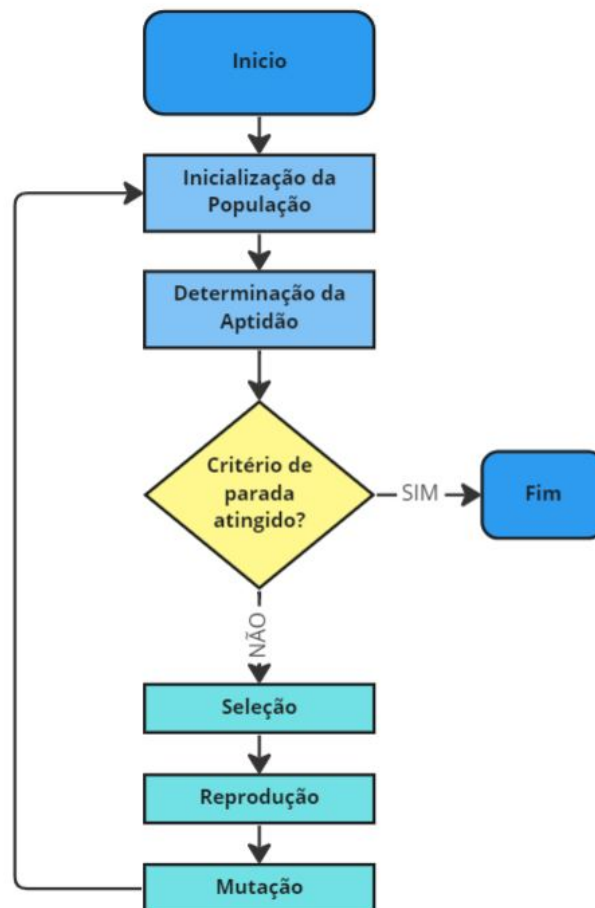
Metodologia de busca e otimização baseada nos processos descritos pela biologia evolutiva, com o objetivo de elaborar uma população de diferentes representações de modelos e através da evolução da mesma ao longo de gerações, tenda a estipular um modelo mais adequado com a função de aptidão estabelecida inicialmente como premissa do sistema.

O Funcionamento dos algoritmos genéticos é dado inicialmente pela população é formada com conjuntos de possíveis soluções formadas aleatoriamente, que após serem ranqueadas por seu valor de aptidão à métricas do sistema, são selecionadas para cruzamento, seguindo alguma metodologia de seleção - como por roleta, torneio, entre outras - para após a seleção dos potenciais cruzadores, estabelecer a metodologia de criação de novos indivíduo, podendo tam-

bém definir uma probabilidade de mutações e sua respectiva forma, para elaborar uma cadeia de novos indivíduos (filhos) que por sua vez irão eventualmente substituir os pais na formação do conjunto de possíveis soluções - a depender do escore de aptidão - e posteriormente serão potenciais candidatos de cruzamento para uma nova iteração, fornecendo novas cadeias de indivíduos que tendendo ao longo das gerações, otimizarão o processo, devido o aumento gradativo do escores de aptidão.

A Figura 62, ilustra por meio de uma fluxograma o processo de elaboração de soluções otimizadas por algoritmos genéticos

Figura 62 – Fluxograma do Algoritmo Genético.



Fonte: O Autor.