



CENTRO UNIVERSITÁRIO SENAI CIMATEC

PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM

COMPUTACIONAL E TECNOLOGIA INDUSTRIAL

Mestrado em Modelagem Computacional e Tecnologia Industrial

Dissertação de mestrado

**Análise, avaliação e validação do uso de técnicas de
inteligência artificial para o diagnóstico de falhas em
motores de indução**

Apresentada por: Álvaro Souza de Araújo
Orientador: Dr. Alex Álisson Bandeira Santos
Co-orientador: Dr. Oberdan Rocha Pinheiro

Julho de 2022

Álvaro Souza de Araújo

Análise, avaliação e validação do uso de técnicas de inteligência artificial para o diagnóstico de falhas em motores de indução

Dissertação de mestrado apresentada ao Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial, Curso de Mestrado em Modelagem Computacional e Tecnologia Industrial do Centro Universitário SENAI CIMATEC, como requisito parcial para a obtenção do título de **Mestrado em Modelagem Computacional e Tecnologia Industrial**.

Área de conhecimento: Interdisciplinar

Orientador: Dr. Alex Álisson Bandeira Santos

Co-orientador: Dr. Oberdan Rocha Pinheiro

Centro Universitário SENAI CIMATEC

Salvador
Centro Universitário SENAI CIMATEC
2022

CENTRO UNIVERSITÁRIO SENAI CIMATEC

Mestrado Acadêmico em Modelagem Computacional e Tecnologia Industrial

A Banca Examinadora, constituída pelos professores abaixo listados, aprova a Defesa de Mestrado, intitulada “**Análise, avaliação e validação do uso de técnicas de inteligência artificial para o diagnóstico de falhas em motores de indução**” apresentada no dia 26 de julho de 2022, como parte dos requisitos necessários para a obtenção do Título de Mestre em Modelagem Computacional e Tecnologia Industrial.

Assinado eletronicamente por:
Alex Álisson Bandeira Santos
CPF: ***.191.765-**
Data: 26/07/2022 16:14:41 -03:00

Sistema FIEB **SENAI**
PELO FUTURO DA INOVAÇÃO

Orientador:

Prof. Dr. Alex Álisson Bandeira Santos
SENAI CIMATEC

Assinado eletronicamente por:
Oberdan Rocha Pinheiro
CPF: ***.073.705-**
Data: 27/07/2022 08:03:36 -03:00

Sistema FIEB **SENAI**
PELO FUTURO DA INOVAÇÃO

Coorientador:

Prof. Dr. Oberdan Rocha Pinheiro
FIEB

Assinado eletronicamente por:
Roberto Luiz Souza Monteiro
CPF: ***.881.935-**
Data: 01/08/2022 10:19:03 -03:00

Sistema FIEB **SENAI**
PELO FUTURO DA INOVAÇÃO

Membro Interno:

Prof. Dr. Roberto Luiz Souza Monteiro
SENAI CIMATEC

Electronically signed by:
Marcelo Henrique Souza Bomfim
CPF: ***.646.836-**
Date: 7/26/2022 4:15:25 PM -03:00

Sistema FIEB **SENAI**
PELO FUTURO DA INOVAÇÃO

Membro Externo:

Prof. Dr. Marcelo Henrique Souza Bomfim
UFMG

Ficha catalográfica elaborada pela Biblioteca do Centro Universitário SENAI CIMATEC

A658a Araújo, Álvaro Souza de

Análise, avaliação e validação do uso de técnicas de inteligência artificial para o diagnóstico de falhas em motores de indução / Álvaro Souza de Araújo. – Salvador, 2022.

88 f. : il. color.

Orientador: Prof. Dr. Alex Álisson Bandeira Santos.
Coorientador: Prof. Dr. Oberdan Rocha Pinheiro.

Dissertação (Mestrado em Modelagem Computacional e Tecnologia Industrial) – Programa de Pós-Graduação, Centro Universitário SENAI CIMATEC, Salvador, 2022.
Inclui referências.

1. Detecção de falhas. 2. Análise de vibração. 3. Algoritmos de classificação. I. Centro Universitário SENAI CIMATEC. II. Santos, Alex Álisson Bandeira. III. Pinheiro, Oberdan Rocha. IV. Título.

CDD 006.3

Agradecimentos

Agradeço ao Prof. Dr. Alex Álisson Bandeira Santos pela orientação, incentivo, sugestões e auxílio na condução desta pesquisa.

Também agradeço ao Prof. Dr. Oberdan Rocha Pinheiro pela orientação, questionamentos e pontuações importantes para o andamento deste trabalho,

À minha família e amigos pelo auxílio e compreensão pela minha ausência devido à dedicação a este trabalho

À instituição Centro Universitário SENAI CIMATEC e os professores que fizeram parte do Programa de Pós-Graduação Stricto Sensu MODELAGEM COMPUTACIONAL E TECNOLOGIA INDUSTRIAL (MCTI) pela oportunidade de desenvolver este trabalho

Salvador, Brasil

Álvaro Souza de Araújo

30 de junho

Resumo

A manutenção preditiva utiliza de diversos métodos para acompanhamento das condições dos motores elétricos aplicados em plantas industriais. Dentre estes métodos, destaca-se a análise de vibração como método amplamente utilizado devido à possibilidade de identificação de uma vasta possibilidade de falhas. Este processo ainda depende muito da análise e processamento manual das informações coletadas, o que pode demandar um tempo maior para realizar a análise da condição de uma máquina, o que conseqüentemente gera mais custos para a empresa. A depender do número de equipamentos, esta tarefa pode levar muito tempo, necessitando de mais mão de obra. É possível, também, que erros na interpretação dos dados ocorram. Com o resultado desta pesquisa, e com o algoritmo testado e validado, erros de interpretação devem ser minimizados. Este trabalho apresenta o desenvolvimento de um modelo computacional dotado de um algoritmo classificador para processar sinais de vibração de motores elétricos e determinar se a máquina apresenta um comportamento de falha e, em caso positivo, qual falha. Foram comparados os desempenhos dos algoritmos *K-Nearest Neighbor*, *Support Vector Machines*, Rede Neural Artificial e *Random Forest*, de modo a determinar a metodologia mais adequada para a solução deste problema. Nesta pesquisa, foi utilizado o *dataset* MaFaulDa por conter leituras de vibração sob diversas condições de falha simuladas em laboratório, o que permite treinar o modelo para detectar uma variada gama de falhas. O modelo computacional realiza o pré processamento dos dados por meio da Transformada de *Fourier* (FFT), para que, então, os algoritmos sejam treinados com os sinais de vibração no espectro da frequência. Após a validação, foram comparados os resultados dos algoritmos para determinar o mais indicado para a solução do problema proposto neste trabalho. Após avaliar a precisão de todos os quatro modelos, o *Random Forest* apresentou a melhor precisão para o conjunto de dados, com 97.18% de acerto. O segundo algoritmo com a melhor taxa de acerto foi o *Support Vector Machines*, com 94.05% de acurácia com o *dataset* filtrado. O modelo *K-Nearest Neighbor* apresentou 91.39% de acerto, tornando-o o menos assertivo dentre os envolvidos no estudo, enquanto o Rede Neural Artificial apresentou uma vantagem relevante quando treinado com o *dataset* completo, sem os filtros de atributos.

Palavras-chaves: Detecção de falhas, análise de vibração, algoritmos de classificação.

Abstract

Predictive maintenance uses several methods to monitor the conditions of electric motors applied in industrial plants. Among these methods, vibration analysis stands out as a widely used method due to the possibility of identifying a wide possibility of failures. This process still depends a lot on the analysis and manual processing of the information collected, which can lead to a longer time to perform the analysis of the condition of a machine, which consequently generates more costs for the company. Depending on the number of equipment, this task can take a long time, requiring more manpower; It is also possible that errors in the interpretation of the data occur. With the result of this research, and with the algorithm tested and validated, interpretation errors must be minimized. This work presents the development of a computational model equipped with a classifier algorithm to receive vibration readings from electric motors and determine if the machine presents a failure behavior and, if so, which failure. Different classification algorithms will be compared in order to determine the most appropriate methodology to solve this problem. In this research, the MaFaulDa dataset was used since it contains vibration readings under different failure conditions simulated in the laboratory, which allows training the model to detect a wide range of defects. The computational model performs the pre-processing of the data through the Fourier Transform (FFT), then the algorithms are trained with the vibration signals in the frequency spectrum. After validation, the results of the algorithms were compared to determine the most suitable for the solution of the problem proposed in this work. After validation, the results of the algorithms were compared to determine the most suitable for the solution of the problem proposed in this work. After evaluating the accuracy of all four models, *Random Forest* showed the best accuracy for the dataset, with 97.18% accuracy. The second algorithm with the best hit rate was *Support Vector Machines*, with 94.05% accuracy with the filtered *dataset*. The *K-Nearest Neighbor* model presented 91.39% of accuracy, making it the least assertive among those involved in the study, while the Artificial Neural Network presented a relevant advantage when trained with the complete *dataset*, without attribute filters.

Keywords: fault detection, vibration analysis, classification algorithms

Sumário

1	Introdução	1
1.1	Definição do problema	2
1.2	Objetivo	4
1.3	Importância da pesquisa	4
1.4	Organização da Dissertação de mestrado	4
2	Revisão bibliográfica	6
2.1	Manutenção	6
2.1.1	Manutenção corretiva	7
2.1.2	Manutenção preventiva	8
2.1.3	Manutenção preditiva	9
2.2	Motores elétricos de indução	9
2.3	Análise de sinais no domínio da frequência	11
2.4	Análise de falhas	13
2.4.1	Tipos mais comuns de falhas em motores elétricos	14
2.4.1.1	Desbalanceamento	15
2.4.1.2	Desalinhamento	15
2.4.1.3	Danos no estator	17
2.4.1.4	Falhas em rolamentos	17
2.4.2	Análise termográfica	19
2.4.3	Análise de assinatura de corrente	21
2.4.4	Análise de sinais acústicos	22
2.4.5	Análise de vibração	23
2.5	Algoritmos classificadores	25
2.5.1	SVM	26
2.5.2	RNA	29
2.5.3	KNN	32
2.5.4	Random Forest	35
2.6	Trabalhos correlacionados	37
3	Projeto e desenvolvimento da pesquisa	43
3.1	<i>Dataset</i>	43
3.2	Modelo computacional	45
3.2.1	Pré processamento	46
3.2.2	Modelo de classificação	48
3.2.2.1	Filtragem de atributos relevantes	48
3.2.2.2	Redes Neurais Artificiais	49
3.2.2.3	K-Nearest Neighbors	51
3.2.2.4	Support Vector Machines	51
3.2.2.5	Random Forest	52
3.3	Método de validação dos modelos	52
3.4	Método de avaliação dos resultados	52

4	Avaliação e resultados experimentais	54
4.1	K-Nearest Neighbor	56
4.2	Support Vector Machines	58
4.3	Redes Neurais Artificiais	60
4.4	Random Forest	62
4.5	Comparação entre os algoritmos	64
5	Considerações finais	69
5.1	Conclusões	69
5.2	Contribuições	70
5.3	Atividades Futuras de Pesquisa	71
	Referências	72

Lista de Tabelas

2.1	Taxa de acerto	40
3.1	Variáveis estatísticas	47
3.2	Atributos selecionados após filtragem	49
4.1	Resultados dos modelos de classificação.	54
4.2	Taxa de classificações corretas	56
4.3	Taxa de classificações corretas para os <i>datasets</i> MaFaulda e CWRU	56
4.4	Resultados do algoritmo de classificação KNN.	57
4.5	Matriz de confusão do algoritmo KNN com o <i>dataset</i> filtrado	57
4.6	Resultados do algoritmo de classificação KNN.	58
4.7	Matriz de confusão do algoritmo KNN com o <i>dataset</i> completo.	58
4.8	Resultados do algoritmo de classificação SVM com o <i>dataset</i> filtrado.	59
4.9	Matriz de confusão do algoritmo SVM com o <i>dataset</i> filtrado.	59
4.10	Resultados do algoritmo de classificação SVM com o <i>dataset</i> original.	60
4.11	Matriz de confusão do algoritmo SVM com o <i>dataset</i> filtrado.	60
4.12	Resultados do algoritmo de classificação RNA com o <i>dataset</i> filtrado.	61
4.13	Matriz de confusão do algoritmo RNA com o <i>dataset</i> filtrado.	61
4.14	Resultados do algoritmo de classificação RNA com o <i>dataset</i> original.	62
4.15	Matriz de confusão do algoritmo RNA com o <i>dataset</i> original.	62
4.16	Resultados do algoritmo de classificação Random Forest com <i>dataset</i> filtrado.	63
4.17	Matriz de confusão do algoritmo <i>Random Forest</i> com <i>dataset</i> filtrado.	63
4.18	Resultados do algoritmo de classificação Random Forest com <i>dataset</i> original.	63
4.19	Matriz de confusão do algoritmo <i>Random Forest</i> com <i>dataset</i> original.	64

Lista de Figuras

2.1	Vista explodida de um motor de indução.	10
2.2	Estatísticas de causas de falhas em motores elétricos	11
2.3	Sinal no tempo e na frequência.	12
2.4	Diferença na amostragem de um sinal.	13
2.5	Dois tipos de excentricidade no entreferro: a) estática e b) dinâmica.	16
2.6	Componentes de um rolamento.	19
2.7	Desalinhamento angular e paralelo e respectivos comportamentos em frequência.	24
2.8	Relação entre falhas e assinaturas de frequência.	25
2.9	Planos de separação.	27
2.10	Transformação dos dados de um espaço bidimensional para tridimensional.	28
2.11	Exemplo de rede neural artificial.	29
2.12	Exemplo de função de ativação <i>threshold</i>	30
2.13	Exemplo de função de ativação sigmoide.	31
2.14	Exemplo de função de ativação tangente hiperbólica.	31
2.15	Exemplo de função de ativação ReLU	32
2.16	Algoritmo KNN classificando um dado novo com base na proximidade com outros dados.	33
2.17	Hiperparâmetro "k" influencia na classificação de um dado novo.	34
2.18	Estrutura de um algoritmo do tipo <i>Random Forest</i>	37
2.19	Matrizes de confusão do algoritmo RNA com e sem a implementação da técnica SMOTE.	42
3.1	Distribuição dos dados do <i>dataset</i> MaFaulDa	44
3.2	Arquitetura do modelo computacional.	46
3.3	Estrutura da rede neural utilizada.	50
4.2	Comparação entre os resultados dos algoritmos utilizando o <i>dataset</i> original.	55
4.1	Comparação entre os resultados dos algoritmos utilizando o <i>dataset</i> filtrado.	55
4.3	Comparação entre a precisão dos algoritmos utilizando o <i>dataset</i> filtrado.	65
4.4	Comparação entre a precisão dos algoritmos utilizando o <i>dataset</i> original.	65
4.5	Comparação entre a taxa de verdadeiros positivos dos algoritmos utilizando o <i>dataset</i> filtrado.	66
4.6	Comparação entre a taxa de verdadeiros positivos dos algoritmos utilizando o <i>dataset</i> original.	66
4.7	Comparação entre a taxa de falsos positivos dos algoritmos utilizando o <i>dataset</i> filtrado.	67
4.8	Comparação entre a taxa de falsos positivos dos algoritmos utilizando o <i>dataset</i> original.	68

Lista de Siglas

ACO	Auto-condicionamento do Operador
API	<i>Application Programming Interface</i>
ASCII	<i>American Standard Code for Information Interchange</i>
ARFF	<i>Attribute-Relation File Format</i>
CSV	<i>Commma-separated values</i>
DFT	<i>Discrete Fourier Transform</i>
FFT	<i>Fast Fourier Transform</i>
FP	<i>False Positives</i>
KNN	<i>K-Nearest Neighbor</i>
MAE	<i>Mean Absolute Error</i>
MCSA	<i>Motor current signature analysis</i>
PPGMCTI ..	Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial
PMSM	<i>Permanent magnet synchronous motor</i>
RF	<i>Random Forest</i>
RMSE	<i>Root Mean Square Error</i>
RNA	Redes Neurais Artificiais
SVM	<i>Support Vector Machines</i>
TP	<i>True Positive</i>

Introdução

Manutenção é um ponto crítico e de constante evolução nas indústrias. Mesmo com vários avanços tecnológicos, ainda é um desafio prever falhas antes que ocorram. A manutenção preditiva é indispensável, principalmente no ambiente industrial, onde a falha em um determinado equipamento que resulte na necessidade de manutenção corretiva, pode causar a parada parcial ou total da planta de produção, o que gera transtornos e prejuízos para a empresa.

Os motores de indução, também conhecidos como motores assíncronos, são geralmente usados como motores principais nas indústrias de manufatura, transporte, petroquímica e sistemas de energia devido ao seu design robusto, facilidade de controle, baixo custo, alta confiabilidade, alta capacidade de sobrecarga e eficiência. Devido aos longos tempos de operação, os motores estão sujeitos a diversos esforços elétricos e mecânicos, portanto a possibilidade de falhas não pode ser ignorada ([TOMA; PROSVIRIN; KIM, 2020](#)).

Estes equipamentos são formados por diversos componentes, entre outros: estator, enrolamentos do rotor, ímãs em máquinas síncronas de ímã permanente (PMSMs), eixo ou rolamentos. Todos esses componentes podem ser danificados devido à abrasão, sobrecarga, carga desequilibrada ou estresse elétrico devido ao controle com um inversor ([MBO'O; HAMEYER, 2016](#)).

Para reduzir danos significativos, prolongar a vida útil da máquina, aumentar a acessibilidade da máquina e reduzir a manutenção de avarias, o diagnóstico de falhas e o monitoramento de condição eficazes são importantes ([TOMA; PROSVIRIN; KIM, 2020](#)).

A detecção precoce de falhas é uma questão importante, pois uma pane inesperada ou dano fatal de todo o sistema podem ser evitados. Além disso, o custo de manutenção pode ser reduzido e a confiabilidade do sistema pode ser aumentada usando o monitoramento de condição ([MBO'O; HAMEYER, 2016](#)). Várias técnicas foram estudadas para a detecção e diagnóstico de falhas em motores de indução, como análise de vibração, monitoramento de campo eletromagnético, análise de sinal de corrente do motor (MCSA), análise química, análise termográfica, análise de sinal acústico ([TOMA; PROSVIRIN; KIM, 2020](#)).

O método de detecção de falhas por análise de vibração é uma das técnicas mais comuns de monitoramento de condição ([ABDELKRIM et al., 2019](#)), e isso se deve à sua capacidade de detectar, localizar e distinguir diferentes tipos de falhas desde o seu início antes que se tornem críticas e perigosas, essas falhas que podem ser distribuídas ou localizadas

([ABDULBARY; EMBABY; GOMAA, 2021](#)). Cada tipo de falha possui uma assinatura em frequência diferente, ou seja, um comportamento único no espectro de frequência do sinal de vibração. Dessa forma, a partir da análise do comportamento do perfil de vibração do equipamento é possível detectar o tipo de falha presente na máquina.

Dessa forma, com a utilização de um algoritmo dotado de um agente classificador e um *dataset* com diversas análises para permitir o treinamento deste sistema, o processo de avaliação das condições dos motores tornar-se-á mais rápido, eficiente, além de permitir reduções nos custos operacionais. O aprendizado de máquina é uma das áreas da ciência da computação que mais cresce, com aplicativos de longo alcance. Refere-se à detecção automatizada de padrões significativos nos dados ([OSISANWO et al., 2017](#)).

Serão avaliados, também, diferentes métodos de classificação: Redes Neurais Artificiais, *K-Nearest Neighbor*, *Support Vector Machines* e *Random Forest*, de modo a obter o que trará uma maior precisão e acurácia na detecção da falha, no menor tempo de execução.

1.1 Definição do problema

Qualquer máquina em operação gera vibrações. A maioria das falhas nas máquinas de indução se transforma em vibração. O sinal de vibração é considerado um indicador que auxilia no monitoramento das máquinas elétricas ([BESSOUS et al., 2017](#)). Portanto, a análise de vibração em motores é um método de manutenção preditiva cada vez mais utilizado na indústria, de modo a evitar que um equipamento entre em colapso devido a falhas.

Cada elemento da máquina induz seu próprio padrão de vibração, que identifica um distúrbio específico. Assim, uma medição do comportamento vibratório permite a identificação das possíveis causas da vibração, mostrando desta forma um diagnóstico mais preciso e indicando o possível problema (rolamento, fonte de alimentação e deposição de massa). O movimento vibratório de uma máquina é essencialmente um movimento harmônico, e o diagnóstico pode identificar os sinais harmônicos correspondentes ([PE-DOTTI; ZAGO; FRUETT, 2017](#)).

Este método consiste na instalação de sensores do tipo acelerômetros *Micro-Electro-Mechanical-Systems* ou do tipo piezoelétricos ([BESSOUS et al., 2017](#)) no equipamento em operação para então realizar a captura de dados de vibração ao longo do tempo. Estes sensores detectam a vibração por meio do deslocamento de uma massa no interior do sensor. Este deslocamento, em relação a outro elemento fixo causa a alteração na capacitância do sistema, o que permite a leitura da aceleração que gerou o deslocamento.

Posteriormente, estes dados são processados para que, então, uma análise dos resultados possa ser feita para determinar se o comportamento obtido está dentro da normalidade ou se apresenta característica de falha. A expansão da série *Fourier* é a técnica mais comum para obter mais detalhes do sinal. O sinal fornecerá informações relevantes, que também ajudarão no diagnóstico dos problemas. A quantidade de energia contida no sinal de vibração está relacionada à amplitude de cada componente harmônico obtido. Interpretar o papel de cada componente de frequência deste sinal é importante para que um diagnóstico da máquina possa ser feito e, conseqüentemente, decisões críticas possam ser tomadas (PEDOTTI; ZAGO; FRUETT, 2017).

De posse do espectro em frequência da vibração, o analista é capaz de determinar o tipo de falha e grau de severidade, ao tomar como referência as especificações técnicas do equipamento em questão, pois cada classe de máquina possui níveis de vibração considerados aceitáveis.

Este processo supracitado ainda depende muito da análise e processamento manual das informações, o que pode acarretar em um tempo maior para se analisar uma máquina, o que conseqüentemente gera mais custos para a empresa. A depender do número de equipamentos, esta tarefa pode levar muito tempo, necessitando de mais mão de obra. É possível, também, que erros na interpretação dos resultados ocorram. Com o resultado desta pesquisa, e com o algoritmo testado e validado, erros de interpretação devem ser minimizados.

Estudos atuais mostram algoritmos classificadores especializados em uma falha específica, em especial os defeitos em rolamentos. Estes são os elementos mecânicos mais essenciais das máquinas rotativas. São empregados para apoiar os eixos rotativos em máquinas rotativas (ABDULBARY; EMBABY; GOMAA, 2021). Esta especificidade dificulta a implementação destes algoritmos em um ambiente industrial para obtenção de indicações de falhas variadas, uma vez que um algoritmo especializado, como o proposto por AbdulBary, Embaby e Gomaa (2021), não cobrem a detecção de falhas oriundas de outras fontes. Com o intuito de cobrir esta lacuna, este projeto de pesquisa visa desenvolver um algoritmo classificador baseado em aprendizagem de máquina que, a partir de um *dataset* contendo leituras de diversos equipamentos, determinará a condição atual de um motor elétrico.

O modelo proposto por Abdelkrim et al. (2019) trata de um modelo de diagnóstico de defeitos em rolamentos de esferas em motores com base em indicadores estatísticos e no *Adaptive Neuro-Fuzzy Inference System* (ANFIS). Este trabalho consiste na realização de três etapas principais: A primeira é a extração de indicadores estatísticos da raiz quadrada média (RMS) dos sinais de vibração brutos medidos experimentalmente para diferentes estados do rolamento (íntegro e na presença de defeitos). A segunda etapa consiste na seleção dos indicadores mais relevantes e, por fim, na introdução desses indicadores na

rede ANFIS, a fim de classificar os vários defeitos no rolamento (falhas de pista interna e externa e falha combinada) (ABDELKRIM et al., 2019).

1.2 Objetivo

O objetivo geral desse trabalho de investigação é desenvolver um modelo computacional de modo a identificar e classificar uma eventual falha presente na máquina, para permitir uma intervenção antes que o equipamento esteja impossibilitado de operar nas suas devidas condições.

Os objetivos específicos são:

- Desenvolver um agente inteligente para realizar a classificação de novas leituras;
- Comparar diferentes métodos de classificação de modo a obter experimentalmente o algoritmo que melhor se adapta a classificar as falhas;
- Avaliar a viabilidade técnica do modelo por meio de testes.

1.3 Importância da pesquisa

Com uma demanda cada vez maior das indústrias por manutenção preditiva, é de grande valia utilizar a tecnologia para tornar esta metodologia mais acessível e eficiente. Para isso, esta pesquisa visa contribuir com a manutenção preditiva de modo a permitir que mais empresas adotem esta técnica e assim, reduzam o tempo de parada devido à falhas de equipamentos.

1.4 Organização da Dissertação de mestrado

O presente trabalho está dividido em quatro capítulos, conforme apresentado a seguir:

- Capítulo II: Revisão bibliográfica - são apresentados os conceitos sobre manutenção, motores de indução, análise de falhas, algoritmos classificadores;
- Capítulo III: Projeto e desenvolvimento da pesquisa - é apresentado o desenvolvimento do projeto do modelo de classificação de falhas em motores de indução por meio de análise de vibração;

- Capítulo IV: Avaliação e resultados experimentais - são apresentados os resultados experimentais do modelo de classificação proposto;
- Capítulo V: Considerações finais.

Revisão bibliográfica

2.1 *Manutenção*

Com o advento da Revolução Industrial, o uso de máquinas nos ambientes de produção tornou-se indispensável. Como consequência, surgiu a necessidade de realizar reparos nestas para reestabelecer o processo produtivo. Esta foi a primeira aplicação da manutenção corretiva não planejada, que visa apenas solucionar o problema para que o equipamento esteja apto a operar novamente.

Entretanto, com o avanço tecnológico, empresas buscaram estabelecer padrões mínimos de operação e isso fomentou a implementação de setores específicos para a realização de manutenção (OTANI; MACHADO, 2008). A manutenção é definida por Moraes¹ et al. (2020) como um conjunto de ações técnicas e administrativas cujo objetivo é preservar ou readmitir a função requerida de um equipamento.

A manutenção industrial resume-se a um conjunto de medidas e ações tomadas com o intuito de preservar a vida útil de um bem com o intuito de o manter no desempenho das funções que lhe foram conferidas (ARAÚJO, 2015).

Dessa forma evitam-se avarias e inesperadas quebras no ciclo de produção que poderão originar atrasos nas entregas, sendo conseqüentemente prejudiciais a nível econômico resultando em aumento de custos ou perda de mercado (WANG; GAO, 2006).

Após o início da Segunda Guerra Mundial, a demanda por produtividade aumentou de forma significativa, o que exigiu que as empresas adotassem novas estratégias de operação, o que envolvia buscar evitar a ocorrência de falhas, em vez de apenas corrigir quando estas ocorressem. Neste momento surgia a manutenção preventiva.

Na década de 60, o uso de computadores tornou-se mais presente, fato que permitiu a utilização destes a favor da manutenção. As técnicas de manutenção preventiva e preditiva passaram a ser associadas com um planejamento mais assertivo, com o auxílio destes computadores.

Atualmente, com a difusão de algoritmos computacionais para auxílio na detecção e predição de falhas, cada vez menos a manutenção corretiva é aplicada nas indústrias. Como consequência, o processo de manutenção torna-se cada vez mais eficiente e eficaz, de modo a minimizar as perdas financeiras durante o processo. Em um cenário onde

várias empresas concorrem entre si para oferecer produtos com o menor custo agregado, o foco na manutenção preditiva é indispensável para garantir a lucratividade.

No contexto atual, a necessidade de manutenção depende diretamente da correta especificação e da exatidão na função de determinado equipamento, já o planejamento depende do tamanho da empresa, da sua cadência produtiva e do custo relativo.

Objetivamente podem-se considerar ameaças à confiabilidade do processo os erros, falhas e avarias. Estes termos são definidos do seguinte modo (ARAÚJO, 2015):

- Falha é a operação incorreta ou incompleta caracterizada por um desvio anormal de pelo menos uma das propriedades ou parâmetros característicos do sistema, uma vez definido um comportamento aceitável. É a raiz do mau funcionamento do sistema;
- Erro é uma discrepância entre um valor assumido e o seu valor medido, estimado ou previsto. Com a presença de falhas surgem erros mensuráveis que são utilizados para detecção de falhas. Erros poderão ser aditivos ou multiplicativos. Desenvolvimento de um Sistema de Diagnóstico de Falhas em Motores Elétricos de Indução;
- Avaria é a perda permanente da capacidade de um sistema de operar normalmente, devido à presença de uma falha imposta. Na prática este termo é o mais utilizado num contexto de manutenção e pode ter vários significados. De um modo geral este termo pode significar que determinada peça ou equipamento ficou completamente inoperante; pode estar operante, mas ser incapaz de cumprir as suas funções ou estar seriamente afetada, não podendo ser confiável e insegura para utilização.

2.1.1 *Manutenção corretiva*

É a forma mais primária de manutenção, uma vez que só é realizada após a ocorrência do defeito e poderá conseqüentemente envolver avarias graves. Além do custo inesperado envolvido na correção, a falta de disponibilidade do equipamento implica a quebra da produção e possível perda do produto, bem como oportunidade de negócio. Uma quebra não esperada também pode representar um risco de segurança dependendo da situação (ARAÚJO, 2015).

Esta metodologia consiste em atuar em um equipamento após a falha do mesmo. Esta atuação pode ou não ser planejada, uma vez que, com um acompanhamento prévio do estado, é possível estimar o momento que a falha virá a ocorrer, sendo uma decisão gerencial manter o equipamento em operação até a ocorrência da falha (OTANI; MACHADO, 2008).

Entretanto, sem o devido acompanhamento das condições de operação, não há o planejamento para a correção da falha, quando esta vir a ocorrer. Ainda de acordo com [Otani e Machado \(2008\)](#), este tipo de manutenção gera custos mais elevados do que a corretiva planejada, pois causa perdas de produção devido à parada da planta produtiva e, por consequência da falta de monitoramento, os danos ao equipamento tendem a ser maiores.

Com a manutenção corretiva planejada, é possível manter o equipamento em operação até a ocorrência da falha, entretanto com um devido planejamento de como proceder no momento da quebra para reestabelecer as condições de operação em um menor tempo ([MORAES¹ et al., 2020](#)). As vantagens da adoção deste tipo de manutenção, de acordo com [Moraes¹ et al. \(2020\)](#), são: Redução do tempo de parada do equipamento e consequentemente do impacto gerado na produção, com base no alinhamento dos interesses de produção e manutenção; planejamento da execução do trabalho e garantia de equipe qualificada, materiais e ferramentas necessárias para tal, para que não haja avaria no equipamento.

2.1.2 *Manutenção preventiva*

Com esta técnica, existe o conceito do tempo de vida útil do equipamento. São realizadas inspeções e/ou reparações periódicas para substituição de componentes e lubrificação de partes, sendo que a periodicidade é determinada por parâmetros que levam em conta dados estatísticos e que são fundamentais para determinar o tempo de vida útil das peças, e desse modo reduzir a probabilidade de ocorrência de falha. Contudo, envolve igualmente a paragem do equipamento envolvendo os custos inerentes de paragem, bem como os de mão de obra e de estoque de peças ([ARAÚJO, 2015](#)).

Esta técnica de manutenção tem como base o cumprimento de um plano de atuação periódico, de modo a reduzir a ocorrência de falhas no equipamento e, consequentemente, reduzir a intervenção corretiva que, como citado anteriormente, acarreta em custos maiores. Entretanto, o intervalo entre intervenções deve ser cuidadosamente estabelecido para evitar paradas e substituição de componentes sem necessidade ([OTANI; MACHADO, 2008](#)).

A adoção deste tipo de manutenção deve se também pelo fator de segurança que o equipamento possui no processo e o risco a que expõe as pessoas. Além disso, outros fatores devem ser considerados para adoção da mesma como a possibilidade de realização de manutenção no equipamento, assim como os sistemas industriais que operam continuamente ([MORAES¹ et al., 2020](#)).

2.1.3 *Manutenção preditiva*

Esta técnica, também denominada verificação da condição, consiste no planeamento e na definição antecipada das intervenções corretivas, a partir do conhecimento da real condição de funcionamento das máquinas através da aplicação de uma monitorização contínua, capaz de detetar e diagnosticar com antecedência o surgimento de falha. Com este tipo de manutenção é possível estimar o tempo de vida de determinado componente. Tem a vantagem de reduzir as perdas originadas por reparos e paragens desnecessárias. As técnicas de monitoramento não invasivas utilizadas neste plano abrangem termografia, análise de vibração, análise de corrente elétrica, análise de ruído acústico e ferrografia (ARAÚJO, 2015).

Esta técnica corresponde ao acompanhamento das condições de funcionamento do equipamento através de dispositivos e/ou sensores com o objetivo de detetar previamente a necessidade de intervenção no mesmo, levando o ativo a operar dentro de condições controláveis até o fim de sua vida útil (MORAES¹ et al., 2020).

A manutenção preditiva visa garantir a qualidade da operação de um equipamento com base em técnicas de análise de condições com a intenção de reduzir e otimizar o número de intervenções por manutenção preventiva, assim como a corretiva.

Por meio de diversas técnicas de análise de condição, é possível determinar o momento ideal para intervenção em um equipamento, o que possibilita que o mesmo opere por mais tempo. Ao mesmo tempo, a manutenção é realizada de forma mais eficiente, uma vez que a intervenção ocorre com base em dados coletados, em vez de um cronograma periódico, conforme manutenção preventiva (OTANI; MACHADO, 2008).

Desta maneira, a Manutenção Preditiva promove a disponibilidade e confiabilidade à proporção que não estimula a intervenção desnecessária nos equipamentos (MORAES¹ et al., 2020).

2.2 *Motores elétricos de indução*

Dentre os vários tipos de motores elétricos, o motor de indução trifásico (MIT) evidencia-se pelo seu modo de funcionamento simplista extremamente versátil, construção resistente, rendimento aceitável e pouca necessidade de manutenção. O MIT é uma máquina elétrica de corrente alternada constituída por elementos comuns a todos os tipos e que são essencialmente o estator e o rotor, respetivamente a parte fixa e a rotativa (ARAÚJO, 2015).

O princípio de funcionamento do motor de indução trifásico baseia-se na conversão de energia elétrica em energia mecânica. É um componente amplamente utilizado pela indústria em variadas aplicações.

O motor de indução trifásico, ao ter uma tensão aplicada nas bobinas do estator, gera um campo magnético girante que, por sua vez, irá causar a movimentação do rotor, cuja velocidade depende do número de polos do equipamento e a frequência de operação da rede elétrica.

No estator estão instaladas três bobinas que, ao serem alimentadas, cada uma, por uma fonte trifásica, gera um campo magnético girante. De maneira similar a um transformador, o rotor produz uma força eletromotriz que tende a alinhar-se com o campo girante do estator, o que causa o movimento rotativo pelo rotor. É importante notar que, para o motor assíncrono, a velocidade de rotação do rotor não é a mesma do campo gerado pelo estator. Esta diferença é denominada fator de escorregamento. A Figura 2.1 descreve os componentes de um motor de indução (GANGSAR; TIWARI, 2020).

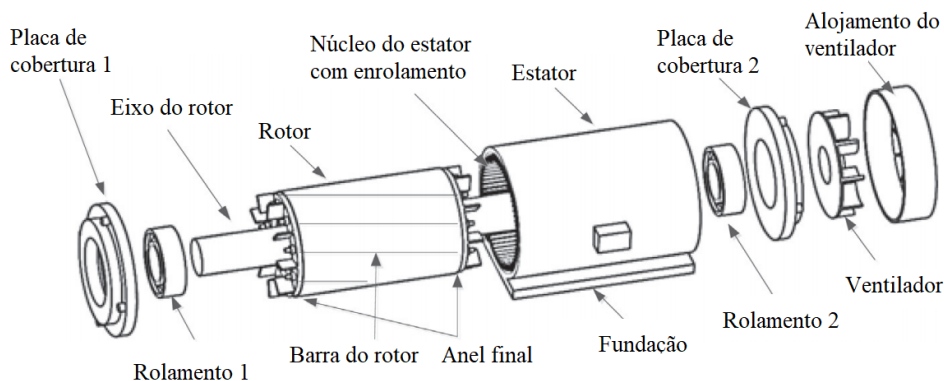


Figura 2.1: Vista explodida de um motor de indução.

O rotor é formado por uma bobina polifásica com os condutores curto-circuitados, energizando assim por submissão a fenômenos de indução magnética. A bobina é por sua vez atravessada por um veio mecânico assente em rolamentos. Pode apresentar duas configurações distintas, sendo a chamada “gaiola de esquilo” a mais utilizada devido ao fato de não apresentar contatos elétricos sujeitos a desgaste. Esta configuração é composta por barras de alumínio condutoras equidistantes e interligadas nos extremos (ARAÚJO, 2015).

A velocidade de rotação do campo girante criado no estator, denominada velocidade de sincronismo é ligeiramente superior à velocidade de rotação do campo criado no rotor, sendo por isso denominado um movimento assíncrono entre campos magnéticos (ARAÚJO, 2015). A velocidade de rotação de um motor de indução trifásico é fixa, entretanto depende da frequência da tensão de alimentação e do número de polos, de acordo com a Equação 2.1 a seguir:

$$N_s = 60 * \text{frequencia}/\text{polos} \quad (2.1)$$

O escorregamento é a relação entre as velocidades de rotação síncrona e velocidade do rotor, conforme Equação 2.2.

$$s = \frac{n_s - n}{n_s} * 100 \quad (2.2)$$

De acordo com análises estatísticas de falhas em motores elétricos (Figura 2.2), grande parte das causas destes defeitos estão relacionados com rolamentos. Dessa forma, diversos modelos de classificação e detecção de falhas são especializados em rolamentos. Entretanto, o modelo proposto neste trabalho busca cobrir uma gama maior de possibilidades de falha.

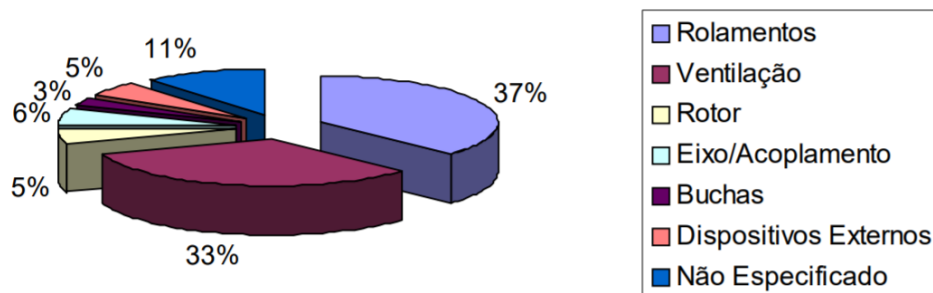


Figura 2.2: Estatísticas de causas de falhas em motores elétricos

2.3 Análise de sinais no domínio da frequência

Para poder analisar o sinal de vibração de um motor elétrico, é preciso realizar procedimentos matemáticos de modo a permitir a visualização do espectro em frequência deste, uma vez que a análise do sinal no tempo não permite a distinção das frequências contidas neste. Para tal, ferramentas como a Transformada de *Fourier* permite a transformação de um sinal para o campo da frequência.

A análise direta da vibração no tempo, normalmente, não apresenta muita informação útil. É necessária que ela seja processada adequadamente para que as suas características sejam identificadas e como tal procede-se à conversão da sua representação para o domínio da frequência. A resposta em frequência é conseguida através da transformada de *Fourier* (FFT), que mostra as frequências onde a energia vibratória se concentra, num espectro contínuo de um sinal não periódico no tempo. Neste caso como o problema apresentado

trata de sinais discretos é necessário implementar um algoritmo para transformar N amostras de dados no domínio do tempo para o domínio da frequência pelo que se recorre à Transformada Discreta de *Fourier* (DFT) (ARAÚJO, 2015).

- Transformada de *Fourier*

A Transformada de *Fourier* é uma ferramenta matemática que permite a decomposição de um sinal no domínio do tempo em somas de sinais senoidais. Dessa forma, é possível observar as componentes de frequência do sinal, o que não é possível se observar no sinal original (Figura 2.3). A partir desta transformada é possível, também, visualizar a magnitude de cada componente de frequência do sinal.

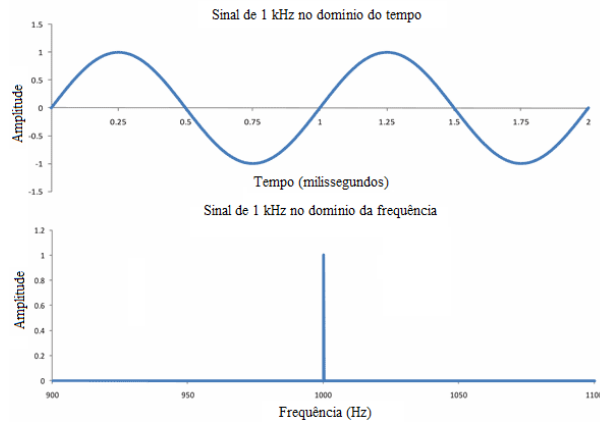


Figura 2.3: Sinal no tempo e na frequência.

- Transformada de *Fourier* de tempo discreto

Sinais amostrados digitalmente apresentam uma peculiaridade para o cálculo da Transformada de *Fourier*, uma vez que o sinal apresenta descontinuidades. Dessa forma, para processamento digital, é necessário aplicar a Transformada de *Fourier* de Tempo Discreto, própria para sinais amostrados (Equação 2.3).

$$X(w) = \sum_{n=-\infty}^{\infty} x[n].e^{-iwn} \quad (2.3)$$

Segundo Araújo (2015), a implementação direta da DFT sobre N amostras de dados requer um esforço computacional de aproximadamente N^2 multiplicações complexas e $N^2 - N$ adições complexas pelo que é necessário muito tempo de computação.

No entanto quando se tem uma sequência em que N é uma potência de 2, pode-se implementar um algoritmo para processar a DFT muito mais rapidamente com $\frac{1}{2}N * \log_2(N)$ multiplicações e $N * \log_2(N)$ adições. Esse algoritmo é denominado por Transformação Rápida de *Fourier* (FFT). Assim a FFT é um método eficiente para o cálculo da DFT, ganhando tempo e reduzindo esforço computacional, assim como uso de memória (ARAÚJO, 2015).

Ainda segundo Araújo (2015), de forma a ter-se uma representação discreta do sinal original e da frequência de vibração dos componentes na FFT, a frequência de amostragem dos dados deverá ser determinada com base no teorema de *Nyquist*. De acordo com o teorema, para que se possa representar o sinal adquirido com o mínimo de perda de informação, a frequência de amostragem deve ser igual ou maior a duas vezes a maior frequência máxima do espectro desse sinal (Figura 2.4).

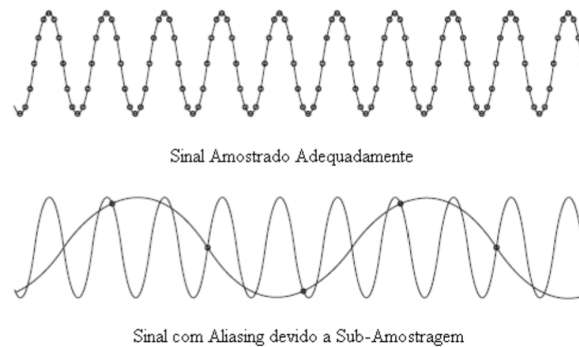


Figura 2.4: Diferença na amostragem de um sinal.

2.4 Análise de falhas

Motores elétricos são equipamentos amplamente utilizados na indústria para conversão de energia elétrica em mecânica. A extensa utilização destes motores elétricos pela indústria os submete a ambientes hostis, operação indevida, defeitos de fabricação, entre outras condições que podem acarretar em falha.

A confiabilidade e a disponibilidade dos motores de indução são cruciais para garantir uma operação contínua e sem problemas na indústria. No entanto, os motores de indução são expostos a muitas tensões inevitáveis, como as tensões mecânicas, elétricas, térmicas e ambientais, ao longo da operação devido à variação na carga externa, desvio na fonte de alimentação, calor excedente, lubrificação inadequada e vedação incompetente, ambiente empoeirado, defeito de fabricação e envelhecimento natural. Isso cria alguns modos de falhas inesperadas em diferentes componentes do motor (GANGSAR; TIWARI, 2020).

É importante a manutenção do bom estado de operação dessas máquinas, pois uma falha pode causar a parada parcial ou até total de uma planta de produção, o que causará prejuízos financeiros à empresa. De acordo com ZAREI, TAJEDDINI e KARIMI (2014), as maiores causas de falhas em motores elétricos estão relacionados com falha nos rolamentos ou nos estatores. Estas falhas, quando detectadas por meio da manutenção preditiva e mitigadas, prolongam a vida útil dos equipamentos. Entretanto, existem outras fontes de falha, capazes de causar danos e até a perda do equipamento, que devem ser detectadas

para garantir o bom funcionamento dos motores. A porcentagem esperada de falha de componentes IM é descrita como:

- Danos nos rolamentos: 41-42%
- Danos no enrolamento do estator: 28-36%
- Danos relacionados ao rotor: 8-9% e
- Outros danos: 14-28%.

As falhas do rolamento e as falhas do enrolamento do estator juntas respondem por pelo menos 69% do total de falhas do motor. Embora as falhas percentuais do rotor e eixo sejam muito baixas, os danos máximos do rolamento ocorrem devido ao rotor excêntrico, eixo desalinhado e desequilibrado e, ocasionalmente, falha do enrolamento que resulta devido à fricção do rotor defeituoso (desalinhado e quebrado) ([GANGSAR; TIWARI, 2020](#)).

O monitoramento remoto tem um papel importante em plantas com equipamentos críticos que necessitem de um monitoramento constante e intensivo. Neste tipo de monitoramento, evita-se a utilização de recursos humanos para a coleta de sinais, evitando-se a exposição de pessoal a áreas de risco desnecessariamente ([ÁLVARES, 2008](#)).

Existem diversas técnicas em aplicação em ambientes industriais para detecção de falhas, oriundas de variadas fontes, a exemplo da análise termográfica, análise de assinatura de corrente do motor (MCSA), análise de assinatura de potência instantânea (IPSA) e análise de vibração.

2.4.1 Tipos mais comuns de falhas em motores elétricos

Uma falha é uma irregularidade no funcionamento do equipamento que resulta em danos aos componentes, perdas de energia e eficiência reduzida da máquina. Tipos comuns de falhas de máquina são desbalanceamento, desalinhamento, falhas nos rolamentos, falhas em engrenagens e acoplamentos, rachaduras no rotor e eixos empenados. A amplitude aumentada das frequências predominantes indica a severidade da falha ([ABDULBARY; EMBABY; GOMAA, 2021](#)).

2.4.1.1 *Desbalanceamento*

O desbalanceamento em um rotor é definido como uma distribuição desigual de massa em torno do centro de rotação (GANGSAR; TIWARI, 2020). Este defeito, sempre aparece na forma de vibrações perceptíveis que prejudicam o funcionamento do equipamento quando excedem as tolerâncias. Isso encurta a vida útil de máquinas e reduz seu valor de utilidade. Na prática, não há rotores perfeitamente balanceados devido a erros de fabricação, tolerâncias e a mudanças geométricas no rotor durante a operação em campo (ABDUL-BARY; EMBABY; GOMAA, 2021). Isso requer que o rotor seja balanceado por meio da adição ou remoção de massas de correção em certas posições.

É uma falha com pouca expressão face às outras referidas, no entanto é uma das fontes mais comuns de vibração em máquinas rotativas e potencializa o surgimento de outras falhas dependendo da sua severidade. As suas principais causas são devidas a gradientes térmicos, desbalanceamento nos rolamentos por má colocação ou defeito, não homogeneidade do rotor por imprecisão na fabricação e excentricidades (ARAÚJO, 2015).

O desbalanceamento pode ser caracterizado como estático ou dinâmico. O desbalanceamento estático é devido ao desequilíbrio de forças em um único plano transversal, enquanto o desbalanceamento dinâmico é devido ao desequilíbrio de forças em diferentes planos transversais, que também geram o par de desbalanceamento. Este defeito gera forças centrífugas extremas e vibrações durante a operação que degradam a vida útil do rotor, rolamentos, acoplamento, vedações e engrenagens (GANGSAR; TIWARI, 2020).

Segundo Araújo (2015), na maioria dos casos é detetada uma alta amplitude no espectro na mesma frequência que a de rotação do rotor ou até três vezes essa mesma frequência, sendo que na direção radial a sensibilidade será maior e como tal aparecem maiores valores de amplitude. A amplitude de vibração é diretamente proporcional ao grau de desbalanceamento, esse fato é importante pois permite que o rotor possa ser balanceado minimizando o problema. Outro facto importante é que a amplitude é proporcional ao quadrado da velocidade do rotor. O balanceamento pode ser realizado de forma estática, onde a compensação de massas é feita no mesmo plano, ou de forma dinâmica em diferentes planos.

2.4.1.2 *Desalinhamento*

O desalinhamento existe devido ao acoplamento inadequado da máquina à carga, distorção térmica e assimetria na carga aplicada. Desalinhamento em máquinas rotativas resultam em forças de reação e momentos que atuam sobre o acoplamento que causam vibrações

no sistema rotativo. Embora teoricamente o alinhamento perfeito não cause vibração, na prática é impossível obter-se um rotor alinhado. No entanto, deve-se estar dentro de um nível aceitável de tolerância com base no tipo de motor em questão (ABDULBARY; EMBABY; GOMAA, 2021).

No desalinhamento paralelo, a distância mínima entre o estator e o rotor é fixa, e é determinada pelo posicionamento inadequado do rotor e do núcleo do estator durante o comissionamento. No caso de desalinhamento angular, o centro do rotor não coincide com o centro de rotação do rotor. (GANGSAR; TIWARI, 2020).

Se as excentricidades no entreferro, ou seja, a presença de irregularidades no campo girante entre o rotor e o estator forem negligenciadas, a máquina poderá sofrer sérios danos. Este tipo de falha frequente é independente do tipo de rotor utilizado, ocorrendo quando existe uma diferença substancial de tamanho entre duas regiões distintas no entreferro ou *airgap* do motor. Na Figura 2.5, as duas regiões estão representadas e fica evidente que numa situação de excentricidade os eixos geométricos do estator e do rotor deixam de ser coincidentes (pontos O_r e O_s) (ARAÚJO, 2015).

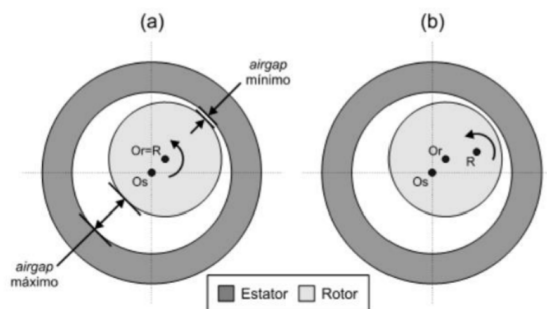


Figura 2.5: Dois tipos de excentricidade no entreferro: a) estática e b) dinâmica.

Na Figura 2.5 (a), este tipo de excentricidade caracteriza-se por um desalinhamento oval do estator e a posição do *airgap* mínimo é invariável. A manifestação da irregularidade resulta num desequilíbrio das forças radiais, passagem de corrente elétrica através dos rolamentos, ruído e numa vibração acrescida provocada por uma frequência igual ao dobro da frequência de rede (100 Hz). A amplitude de vibração aumenta com a carga e anula-se quando a alimentação do motor é desligada. A causa mais comum deste tipo de excentricidade são imperfeições no processo de manufatura, rotor e eixo não concêntricos, núcleo do estator oval e empeno térmico do rotor (ARAÚJO, 2015).

Na Figura 2.5 (b) a posição do *airgap* mínimo gira juntamente com o rotor. Se o rotor não for perfeitamente uniforme, no caso de uma velocidade baixa existirá uma frequência resultante de deslizamento. Esse fato ocorre devido a uma possibilidade de motivos tais

como veio do motor curvado, sistema trifásico desequilibrado, rolamentos danificados ou devido à existência de algum apoio do motor mal apertado ou empenado o que origina a deformação do estator e uma distribuição assimétrica do entreferro (NANDI et al., 2010).

2.4.1.3 Danos no estator

No estator poderão ocorrer vários tipos de falhas, sendo que 94% destas ocorrem ao nível dos seus enrolamentos e as restantes distribuídas pelo quadro e núcleo. A falha mais comum está relacionada com problemas de fase dos enrolamentos do estator devido a curto-circuito. Este fenômeno pode acontecer numa ou entre várias fases. Falhas no núcleo ferromagnético também poderão acontecer, contudo são menos frequentes.

A ocorrência das falhas mencionadas tem como origem a deterioração de um dos elementos intrínsecos mais vulneráveis numa máquina elétrica, o isolamento do estator. Um correto isolamento preserva o tempo de vida útil da máquina e do seu funcionamento, atuando como dissipador térmico reduzindo deste modo o risco de sobreaquecimento dos condutores. Outra função deste componente é garantir que a corrente circule no caminho desejável, prevenindo a ocorrência de curto circuitos na bobinagem (ARAÚJO, 2015).

2.4.1.4 Falhas em rolamentos

Os rolamentos são os componentes mais críticos de qualquer sistema rotativo. São utilizados para apoiar os eixos das máquinas rotativas. Assim, qualquer falha ou mau funcionamento nos mancais pode resultar em perdas no nível de produção e equipamentos, bem como ter um ambiente de trabalho inseguro para as pessoas (ABDULBARY; EMBABY; GOMAA, 2021). Dessa forma, o diagnóstico de falhas em rolamentos tem obtido grande atenção dos pesquisadores nos últimos anos, especialmente com a utilização de modelos classificadores de falhas dedicados a estes tipos de anomalias.

Rolamentos são, geralmente, compostos por um anel interno e outro externo, com um conjunto de esferas metálicas no seu interior, fixadas por meio de uma armadura metálica (ARAÚJO, 2015). Ainda de acordo com Araújo (2015), falhas nestes componentes podem ser diagnosticadas através das análises de vibração, detectando componentes com frequências características de defeitos e suas harmônicas. O prognóstico de falha não se baseia apenas na intensidade dessas componentes, mas também no padrão de distribuição de energia pelas diversas bandas no espectro de frequência, o que permite identificar o estado de degradação do rolamento.

Estes defeitos podem ser classificados como defeitos localizados e defeitos distribuídos. Os localizados são visíveis no próprio rolamento e aparecem nos principais componentes: na pista externa, pista interna, gaiola e nos elementos rolantes. Os distribuídos referem-se a danos que não estão concentrados num local específico, como pista externa desgastada ou elementos rolantes deformados.

O defeito localizado é considerado de cíclico pois produz um impacto entre a esfera e a pista gerando uma frequência e um ruído detetável e previsível que pode ser visualizado no gráfico da forma de onda no domínio do tempo. Essa frequência a que a vibração ocorre é previsível e depende da superfície do rolamento em que o defeito se encontra, bem como da dimensão geométrica dos elementos e da frequência de rotação do rotor (ARAÚJO, 2015). Apesar disso as frequências características de vibração no caso de falha não são síncronas, ou seja não são múltiplas inteiras da velocidade do rotor, e nesse caso pode permitir a identificação da zona de falha mesmo quando não se sabe o rolamento presente na máquina. É importante ainda ressaltar que estas frequências são verdadeiramente frequências de falha, isto é, elas só aparecem nos espectros de vibração quando os rolamentos estão defeituosos (SIDDIQUI; SAHAY; GIRI, 2014).

Segundo Siddiqui, Sahay e Giri (2014), as principais causas das falhas nos rolamentos são:

- Contaminação: o processo de contaminação ocorre através da deposição de pequenas partículas abrasivas de dureza variável que frequentemente contaminam a lubrificação do rolamento;
- Corrosão: o processo de corrosão ocorre com a presença de agentes oxidantes como água, ácido ou lubrificação contaminada. Com o avanço da corrosão partículas são expelidas resultando num processo idêntico à contaminação;
- Instalação inadequada de rolamento: no caso de o rolamento ser instalado de forma imprópria, ocorrem falhas causadas por forçar o rolamento contra o eixo ou contra a carcaça, desalinhamento, deflexão do eixo, empeno da pista deslizante interna ou externa, resultando em danos físicos. Os problemas referidos podem surgir como consequência de um carregamento mais pesado que o previsto, sobrecarga, ou de ajustes demasiado apertados. Estas falhas referidas podem ser facilmente confundidas com desequilíbrios nas fases do estator, devido à sua influência no comportamento do funcionamento do motor;
- Lubrificação imprópria: lubrificar inadequadamente, quer por utilizar lubrificante deteriorado devido a contaminação, quer por uma utilização carente ou excessiva do mesmo, a temperatura no interior do rolamento é influenciada. No caso de carência de lubrificante surge um aquecimento gerado por atrito, que deteriora a lubrificação existente potenciando o aceleração do processo de falha. Por outro lado, uma vedação ineficaz do isolante do rolamento pode permitir o derrame de lubrificante,

e conseqüente entrada de agentes erosivos. A utilização de lubrificante em excesso pode acarretar neste tipo de ocorrência.

Além destas, [Gangsar e Tiwari \(2020\)](#) mencionam outras possíveis causas para falhas em rolamentos, como alta vibração do rotor devido ao grande torque de carga de saída que, em última análise, leva a alta tensão de fadiga, deterioração de lubrificação devido à tensão do eixo direcionando para alta corrente no rolamento e, calor excessivo conduzido através do eixo.

Quando o rolamento apresenta falhas, a distribuição de carga é desequilibrada e resulta em mais vibração em determinadas frequências ([SANTOS et al., 2019](#)). Os elementos de um rolamento podem ser visualizados na Figura 2.6.

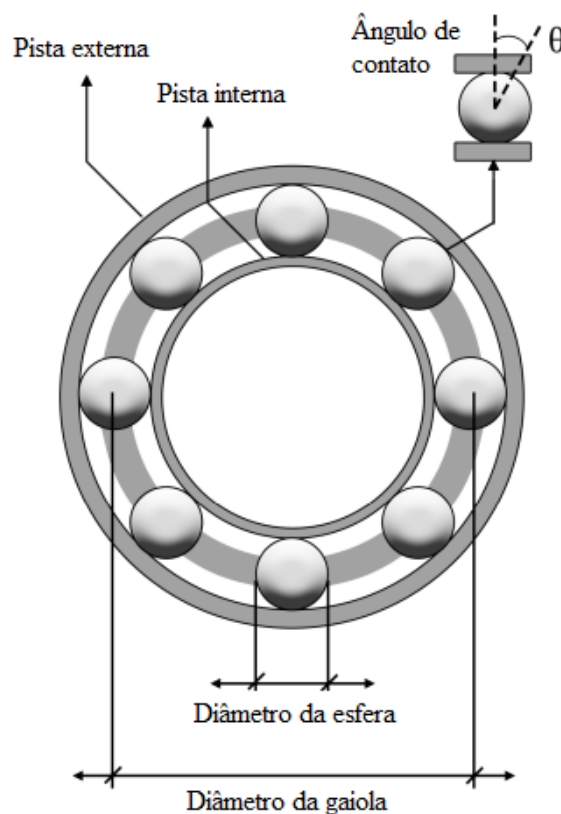


Figura 2.6: Componentes de um rolamento.

2.4.2 Análise termográfica

Durante o funcionamento, um motor elétrico realiza a conversão da energia elétrica da alimentação em energia mecânica e térmica. Quanto maior o trabalho mecânico realizado pelo equipamento, a dissipação térmica será proporcionalmente maior. Entretanto, existe um limite para a temperatura dos componentes internos e externos da máquina, uma vez

que uma temperatura demasiadamente elevada pode ser um indicativo de falha.

A análise termográfica é uma técnica não invasiva de avaliação de condições de um motor elétrico. Consiste em utilizar uma câmera capaz de captar radiação infravermelha para obter uma imagem que converte esta radiação em níveis de temperatura (ÁLVARES, 2008). Através da termografia é possível obter um mapeamento dos componentes do motor com o intuito de distinguir regiões de diferentes temperaturas superficiais, através da visualização artificial da luz dentro do espectro de infravermelhos (ARAÚJO, 2015).

Segundo Álvares (2008), o processo de radiação ocorre em transferência de energia de uma superfície com altas temperaturas para outra superfície de temperatura mais baixa, quando tais superfícies estão separadas no espaço, ainda que exista vácuo entre eles, através de ondas eletromagnéticas denominadas calor radiante, predominando os raios infravermelhos que viajam na velocidade da luz, a energia assim transferida é chamada de radiação térmica, e a capacidade de um corpo emitir energia infravermelha é chamada de emissividade.

A emissividade, demonstrada na equação 2.4, é uma grandeza adimensional que varia entre 0 e 1, sendo o menor valor correspondente a um corpo transparente, e o maior valor correspondente a um corpo negro ideal que absorve e emite toda a radiação incidente (PEREIRA, 2015). De acordo com a lei de *Stefan-Boltzmann*, a energia emitida é proporcional à emissividade do corpo e à quarta potência da temperatura, dessa forma o equipamento de medição de temperatura absorve esta energia e converte em sinais elétricos que são processados para fácil visualização.

$$E = \epsilon * \sigma * T^4 \tag{2.4}$$

Entretanto, o processo de determinação da emissividade de um corpo não é uma tarefa simples, e um valor incorreto pode causar uma variação na temperatura lida pelo sensor. Para facilitar este processo, fabricantes de sensores disponibilizam tabelas contendo valores para diversos materiais.

As câmeras são equipamentos portáteis, onde o operador aponta para o motor elétrico e observa as temperaturas por todo o motor, correlacionando as cores no visor com a escala determinada pelo aparelho. Embora estes leitores sejam bastante eficazes na determinação da temperatura no equipamento, estes apresentam um custo de aquisição elevado, e também podem não ser eficazes para a análise de equipamentos de grande porte, uma vez que a escala do aparelho pode não corresponder à temperatura do equipamento, causando distorções nas leituras.

O monitoramento da temperatura na carcaça do motor também é um método que pode ser utilizado pois permite detetar aquecimentos localizados em zonas específicas, consoante a localização do sensor térmico, contudo apenas dá indicação de sobreaquecimento indiscriminado uma vez que a temperatura medida é resultado da confluência das temperaturas geradas no estator, carga resistiva e da temperatura ambiente (ARAÚJO, 2015).

Este método de análise é capaz de prover medições de temperatura em diversos pontos do equipamento, em tempo real. Dessa forma, é possível identificar o ponto de origem da falha, assim como seu grau de severidade, uma vez que o equipamento, quando em perfeito estado, deve operar dentro de um intervalo de temperatura determinado pelo fabricante.

Outra forma possível de analisar termicamente um motor consiste na medição local de temperatura. Para isso são utilizados sensores de temperatura embebidos no isolamento dos condutores do estator ou na parte interior da carcaça que permitem medir a evolução da temperatura de forma precisa em pontos desejáveis (ARAÚJO, 2015).

Ainda de acordo com Araújo (2015), como alternativa mais barata têm sido desenvolvidas soluções de estimação de temperatura, com base na resistência dos enrolamentos do estator. Existem vários tipos de sensores sendo os mais utilizados o termopar, o resistivo e o de silício.

2.4.3 Análise de assinatura de corrente

A análise de assinatura de corrente tem seu princípio baseado na coleta de leituras de corrente por meio de sensores nas fases de alimentação, de modo a buscar por alterações no espectro de corrente. Esta medição pode ser realizada com sensores de corrente invasivos ou não invasivos.

Este método identifica anomalias nos motores, através da detecção de alterações nas correntes do motor pela sua análise espectral. Utiliza o motor como transdutor⁶ para detetar a presença e severidade de falhas elétricas e mecânicas em boa parte do conjunto eletromecânico. Através da monitorização de uma das três fases da corrente de alimentação, a técnica permite detetar quebras nas barras do rotor, enrolamentos do estator curto-circuitados, problemas de acoplamento e rolamentos danificados (ARAÚJO, 2015).

Ao apresentar um comportamento de falha, serão geradas frequências harmônicas a partir da modulação do campo magnético no entreferro (OBAID; HABETLER; GRITTER, 2000). Os sinais são obtidos pelos sensores no domínio do tempo, porém a análise é realizada no domínio da frequência.

Existe a possibilidade de realizar a aferição da corrente a partir de sensores instalados diretamente nas linhas de alimentação, ou por meio de sensores não invasivos, onde é necessário envolver o condutor a ser medido com o sensor, que consiste em um transformador cuja saída é proporcional à corrente que passa pelo condutor. Desta forma, esta técnica permite a manutenção preditiva em motores sem a necessidade de acesso à instalação dos mesmos, mas apenas ao painel de alimentação.

O objetivo desta técnica é coletar leituras da corrente na alimentação do motor para então permitir a obtenção do sinal denominado como assinatura de corrente. De acordo com [Bonaldi et al. \(2012\)](#), este sinal é necessário para identificar a magnitude e frequência de cada componente individual que constitui o sinal de corrente. Dessa forma, é possível identificar padrões de comportamento que indicam a presença de falhas em diversos componentes do motor.

Segundo [Araújo \(2015\)](#), a corrente do estator das máquinas de indução tem um espectro que varia harmonicamente devido ao facto da disposição dos enrolamentos das ranhuras não ser perfeitamente sinusoidal, devendo-se às imperfeições ou irregularidades ocorridas no fabrico do motor ou dos elementos constituintes da fonte de alimentação. Esse comportamento é previsível e mencionado pelo fabricante como sendo a assinatura do motor em que é indicada a sua frequência característica de alimentação proveniente da rede.

No caso de um motor ter uma falha no estator, a configuração do campo eletromagnético girante é influenciada e como consequência as componentes harmônicas da corrente do estator têm as suas amplitudes afetadas ([ARAÚJO, 2015](#)). Com a análise periódica do espectro de frequência das correntes do estator, identificam-se as componentes harmônicas diretamente relacionadas com as perturbações provocadas pelas tensões induzidas do campo magnético resultantes de falha ([MEHALA; DAHIYA, 2009](#)).

2.4.4 Análise de sinais acústicos

A emissão acústica é uma técnica de avaliação passiva e não destrutiva que monitora as ondas de tensão transitória dos materiais. Além disso, como resultado de mudanças microestruturais para os materiais ou estruturas, a emissão acústica mostra as ondas de estresse acústico que surgem a partir da liberação da energia ([ALSHORMAN et al., 2021](#)).

Esta técnica consiste em analisar o espectro em frequência de um sinal sonoro emitido pelo motor em operação e captado por um ou mais microfones. Utiliza-se esta metodologia quando não é possível fixar um sensor no equipamento devido a diversos fatores, como dificuldade de acesso ao mesmo, ou à impossibilidade de fixar um acelerômetro à carcaça do motor por causa de umidade e temperatura elevadas ([GERMEN; BAŞARAN; FIDAN,](#)

2014).

O funcionamento anormal da máquina resulta normalmente num barulho característico de elevadas frequências, que pode levar à identificação da falha derivada de problemas elétricos e mecânicos. A principal causa do ruído do motor está relacionada com as forças eletromagnéticas que atuam nas superfícies dos componentes e com a ventilação do motor devido à turbulência de ar causada por distúrbios de pressão impostos pelo ventilador (ARAÚJO, 2015).

Também é comum a combinação da análise de vibração com esta técnica, pois ao comparar os resultados obtidos pelas duas metodologias, é possível obter um diagnóstico mais preciso (GERMEN; BAŞARAN; FIDAN, 2014). Entretanto ruídos externos podem prejudicar ou impossibilitar a aplicação da análise de sinais acústicos. Estudos mostram que para a detecção de falhas em rolamentos, esta técnica mostra-se bem efetiva (GERMEN; BAŞARAN; FIDAN, 2014).

De acordo com (ALSHORMAN et al., 2021) esta técnica apresenta uma melhor performance para a detecção de falhas de origem elétrica do que para falhas de origem mecânicas, e consegue ser confiável para detectar falhas incipientes nos motores de indução. Entretanto, de acordo com Mehala e Dahiya (2009), a desvantagem desta técnica reside no fato de depender das condições de ruído sonoro de fundo, pelo que normalmente em ambiente industrial torna-se um método pouco viável.

2.4.5 Análise de vibração

Esta é uma técnica amplamente utilizada no ambiente industrial e que pode ser complementada com a Análise de Assinatura de Corrente, uma vez que distorções na alimentação do equipamento refletem em vibrações que podem ser detectadas por meio deste método. Dessa forma, ao comparar o espectro de corrente elétrica, é possível distinguir se a falha é causada por instabilidades na energia ou se realmente trata-se de um problema mecânico.

Para a detecção de falhas de origem mecânicas, é utilizada a análise de vibração, onde, por meio de um equipamento dotado de um sensor acelerômetro, é aferido um sinal de vibração em aceleração, velocidade ou deslocamento, para então observar estes dados no espectro do tempo ou frequência.

Ao considerar o sinal no domínio do tempo, é possível observar os níveis de vibração pico a pico, pico, ou RMS: os valores de pico a pico indicam uma falha severa, enquanto que o pico sinaliza uma falha prematura.

De acordo com [Li, Wang e Wang \(2019\)](#), os critérios de avaliação da vibração da máquina dependem de uma ampla gama de fatores e os critérios adotados variam significativamente para diferentes tipos de máquina. De acordo com a norma ISO 20816, existem três quantidades primárias de vibração: deslocamento, velocidade e aceleração. No entanto, na maioria das situações, é difícil fornecer tolerâncias absolutas de vibração para qualquer máquina. Existe, portanto, um risco óbvio de julgar os níveis de vibração medidos, perigosos quando não são, ou o contrário - não perigosos quando são perigosos. A experiência humana e a interpretação dos valores medidos ainda desempenham um papel importante no monitoramento da condição de vibração. Portanto, como aplicar abordagens de aprendizado de máquina para tomada de decisão com maior precisão em condições de vibração é sempre uma questão importante.

Para a captura de vibrações, são utilizados sensores transdutores de vibrações mecânicas. Existem diferentes tipos de sensores com esta finalidade, porém o escolhido e o mais usado é o acelerômetro por causa da sua versatilidade. Antes da realização de qualquer análise, a vibração precisa ser transformada para um sinal elétrico, sendo esta a finalidade do acelerômetro. É baseado na conversão de energia mecânica da superfície do motor em sinais elétricos, cuja forma de onda está relacionada com a vibração do mesmo.

O funcionamento deste sensor capacitivo consiste em uma massa acoplada a um sistema de molas e amortecedores com placas capacitivas em seu entorno. Em repouso, as distâncias entre massa móvel e as placas fixas são iguais. Entretanto, quando o sistema é submetido a um movimento, esta distância é alterada, o que causa variações na capacitância, pois este valor depende inversamente da distância entre os condutores.

Na Figura 2.7 é possível perceber o comportamento no espectro da frequência de duas falhas: o desalinhamento angular e paralelo, onde 1x representa a frequência dominante de operação do motor. Caso este esteja alimentado diretamente em uma rede elétrica de 60 Hz, esta será sua frequência dominante. Entretanto, ao inserir um inversor de frequência para controle de velocidade, este componente será alterado.

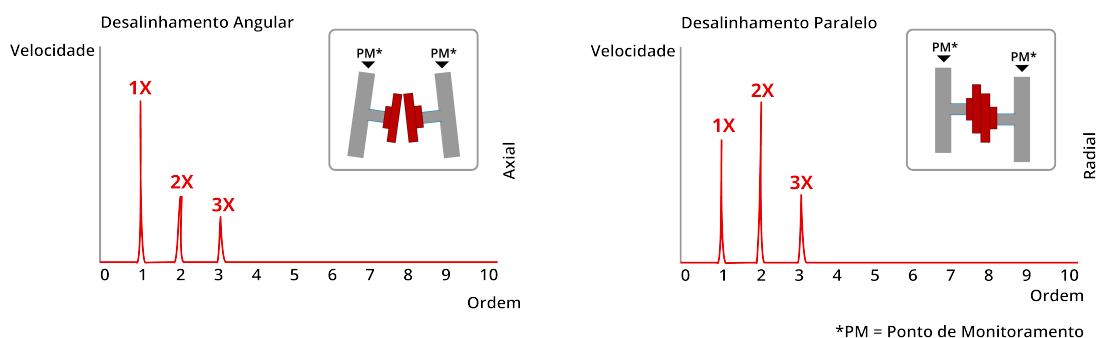


Figura 2.7: Desalinhamento angular e paralelo e respectivos comportamentos em frequência.

Dessa forma, outras falhas apresentam o que se chama de assinatura de frequência, ou seja, o comportamento da vibração gerado pela falha. Dentre as mais comuns, citadas na Figura 2.8, estão o desbalanceamento, desalinhamento, torção no eixo, lubrificação inadequada e rolamentos danificados.

CAUSA	FREQÜÊNCIA	VARIÁVEL	AMPLITUDE	OBSERVAÇÃO
Desbalanceamento	1 X rpm	d	Proporcional ao desbalanceamento. Maior no sentido radial	Geralmente aparece no primeiro harmônico
Desalinhamento ou eixo torto	1/2/3/4 X rpm	d	Grande em direção axial (maior que 50% da amplitude, radial)	Desalinhamento entre mancais ou entre ambas metades do "manchón"
Folga nos mancais	½ X rpm	d	Grande no sentido vertical	Como a frequência é menor que ½ rpm a fase pode ser variável
Falta de firmeza mecânica	2 X rpm	d	Instável	Geralmente afeta o alinhamento
Correias frouxas	1 X rpm	d	Instável	
Engrenagens defeituosas	alta, número de dentes X rpm	a/d	"blips"	Frequência entre 15000 e 40000 rpm
Almofadas antifricção deterioradas	Alta	a	"blips"	A frequência alcança valores maiores do que os registrados para engrenagens ou rolamentos
Rolamentos deteriorados	Alta	a	"blips"	Frequência entre 15000 e 40000 rpm
Lubrificação	Alta	a	-	A amplitude se modifica notadamente em função da lubrificação
Eixo torcido	1 X rpm	d	Grandes valores no sentido axial	A amplitude pode se apresentar de forma pulsátil

Figura 2.8: Relação entre falhas e assinaturas de frequência.

É possível combinar o uso de diversas técnicas de detecção de falha para uma maior assertividade e, também, para avaliar os efeitos causados pela falha. Ao combinar esta técnica com a análise termográfica é possível, por exemplo, avaliar a elevação de temperatura em componentes rotativos do equipamento na presença de desalinhamentos. De maneira análoga, um comportamento anômalo pode também causar distorções na corrente elétrica que pode ser observado com a análise de assinatura de corrente. Devido à possibilidade de detectar uma gama extensa de falhas por meio de análise de vibração, este método foi escolhido para o desenvolvimento deste trabalho. Em conjunto com um algoritmo de classificação, o modelo proposto será capaz de atuar na detecção de falhas comuns e também as não recorrentes, de modo a abranger o maior de possibilidades no ambiente industrial.

2.5 Algoritmos classificadores

Aprendizado de máquina, ou *Machine Learning*, é uma ramificação da inteligência artificial que trata da capacidade de algoritmos aprenderem com dados fornecidos, e assim poder realizar ações após este processo. Esses algoritmos podem ser classificados como

supervisionados ou não supervisionados. Segundo [Zhang \(2020\)](#), o primeiro consiste em trabalhar com dados de entrada devidamente rotulados para que, então, ao receber um conjunto de dados novos, sem anotação, seja possível realizar a identificação da classe destes. É necessário, portanto, um conjunto de dados para treinamento e outro conjunto para teste. A diferença entre estes consiste na ausência de anotações para os dados de teste, de modo que o algoritmo, a partir do treinamento, seja capaz de obter as classificações.

A aprendizagem não supervisionada, porém, obtém resultados a partir de dados não rotulados previamente, sendo assim destinados à organização de dados apenas tomando como base características e padrões de semelhança ([ZHANG, 2020](#)).

Algoritmos classificadores são aqueles que possuem como objetivo prever a classe de um novo dado, a partir do aprendizado sobre dados semelhantes. São uma subcategoria dos algoritmos de aprendizagem supervisionada, onde o objetivo é prever a categoria de um novo dado baseado em observações passadas ([RASCHKA; MIRJALILI, 2017](#)). Serão avaliados neste trabalho, a performance, acurácia e precisão de três algoritmos de classificação: *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN) e Redes Neurais Artificiais (RNA).

2.5.1 SVM

Support Vector Machine (SVM) é uma ferramenta de classificação e regressão que utiliza a teoria de aprendizado de máquinas para maximizar a acurácia na predição, enquanto automaticamente evita o *overfitting* da rede ([JAKKULA, 2006](#)). Este algoritmo realiza o aprendizado atribuindo anotações aos dados ([NOBLE, 2006](#)). Ainda de acordo com [NOBLE \(2006\)](#), o SVM pode ser usado em diversas aplicações, desde o reconhecimento de transações fraudulentas em cartões de crédito, detecção de padrões de escrita à mão, por meio de imagens, até aplicações na área de biologia, com o intuito de reconhecer anomalias em DNA.

Os algoritmos SVM são bem conhecidos por seu excelente desempenho na esfera da classificação estatística. Ainda assim, o alto custo computacional devido à complexidade do tempo de execução cúbico é problemático para os conjuntos de Big Data: o treinamento do classificador SVM requer a resolução de um problema de otimização quadrática ([CHAPELLE, 2002](#)).

O SVM utiliza conceitos de planos de separação para realizar a classificação. Dessa forma, dados que são facilmente separáveis, este algoritmo é capaz de construir uma reta ou plano que separa as classes. Na Figura 2.9, existem duas classes de dados linearmente separáveis. Entretanto, o algoritmo deve encontrar o melhor separador de modo a garantir

uma margem entre as duas classes para que um eventual dado novo seja corretamente classificado.

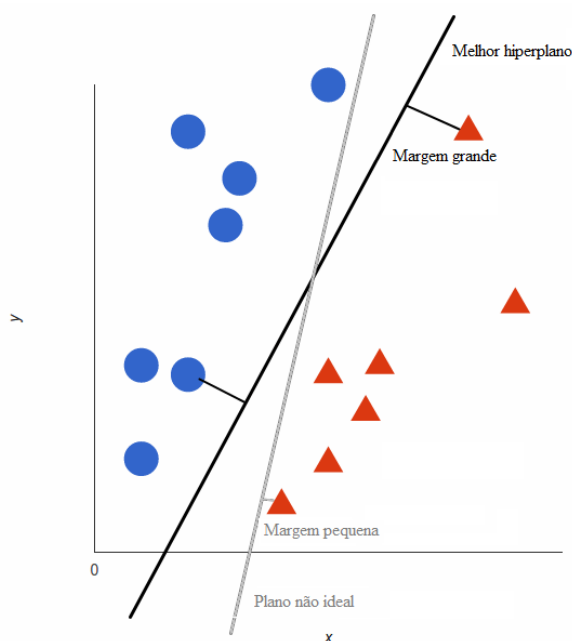


Figura 2.9: Planos de separação.

Por se tratar de um classificador binário, problemas de classificação que envolvam mais de duas classes requerem a aplicação de métodos multiclases. Os métodos multiclases mais utilizados são *one-vs-one* e *one-vs-all*. No primeiro, também conhecido como todos-contra-todos, dadas n classes são gerados $n(n - 1)/2$ classificadores binários, sendo cada um deles responsável por determinar apenas um par de classes. O resultado final é definido pelo método de votação, a classe mais votada é atribuída à amostra. No segundo método, conhecido também como um-contra-todos, para um problema com n classes são gerados n classificadores binários, cada um especializado na sua classe (LOBAO; PINTO, 2020).

No entanto, em situações reais, dados não podem ser separados facilmente com uma reta ou um plano, ou seja, não são linearmente separáveis. Para estas situações, o separador ideal tende a ser uma curva, onde o grau de complexidade deste separador aumenta conforme os dados estão sobrepostos.

De modo a reduzir a complexidade computacional para determinar o separador ideal, o SVM propõe a reorganização dos dados por meio de funções matemáticas denominadas *Kernels*. De acordo com Patle e Chouhan (2013), a função *kernel* projeta os dados de um espaço de dimensão inferior para um espaço com dimensão maior. Existem diversas funções para esta aplicação, dentre elas, o *kernel* linear, polinomial, radial e sigmoide. No exemplo abaixo, onde a curva de separação tem um grau elevado, ao utilizar um *kernel* linear para adicionar uma nova dimensão aos dados, é possível projetar um plano de

separação, conforme demonstrado na Figura 2.10.

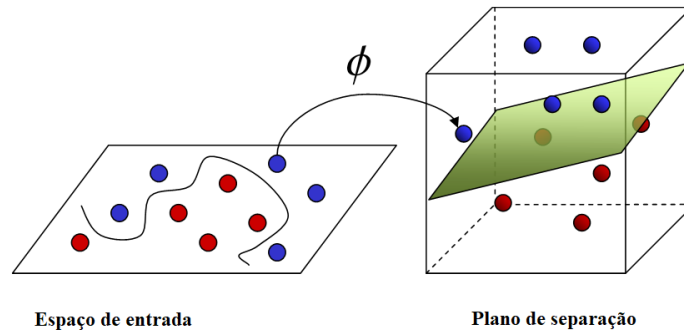


Figura 2.10: Transformação dos dados de um espaço bidimensional para tridimensional.

- *Kernel* linear

De acordo com [Cervantes et al. \(2020\)](#), esta função é a mais simples de implementar e, também, em uso de poder computacional. *datasets* com um grande volume de *features*, em geral, podem ser problemas linearmente separáveis, sendo esta função, portanto, ideal para tais casos. É utilizada para predição de cotação de ações em bolsas de valores ([HENRIQUE; SOBREIRO; KIMURA, 2018](#)) e detecção de *malwares* ([SANJAA; CHULUUN, 2013](#)) (Equação 2.5).

$$K(x_i, x_j) = \langle x_i, x_j \rangle + 1 + x_i x_j + 1 \quad (2.5)$$

- *Kernel* polinomial

Este *kernel* permite o mapeamento de um conjunto de dados para um espaço de dimensão maior que consiste em uma combinação de produtos de polinômios. Esta função tem ampla utilização em algoritmos desde diagnóstico de falhas em bombas centrífugas até processamento de linguagem natural ([CERVANTES et al., 2020](#)) (Equação 2.6).

$$K(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^r \quad (2.6)$$

- *Kernel* RBF

Esta função é a mais utilizada em algoritmos, sendo considerada como uma função genérica. Entretanto, o parâmetro deve ser definido com base em estudos ([CERVANTES et al., 2020](#)). Dentre as aplicações que se beneficiam da implementação desta função estão a classificação de imagens multiespectrais ([KUO et al., 2013](#)) (Equação 2.7).

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (2.7)$$

Contudo, o algoritmo apresenta um alto custo computacional quando aplicado em *datasets* grandes, uma vez que a matriz *kernel* aumenta de forma quadrática com o tamanho do *dataset*, o que torna o processo de aprendizagem lento nestes cenários (CERVANTES et al., 2020). Outra desvantagem está na dificuldade em obter os parâmetros otimizados para a aplicação, o que exige diversos testes e, conseqüentemente, demanda tempo para treinar a rede novamente, a cada alteração de parâmetro. Estas dificuldades se dão devido ao SVM ter sido projetado, inicialmente, para classificação em problemas binários (HSU; LIN, 2002).

2.5.2 RNA

As Redes Neurais Artificiais podem ser definidas como máquinas destinadas a modelar o princípio de funcionamento de um cérebro para executar uma tarefa (JUNIOR; ALMEIDA; GOMES, 2020). Os elementos básicos deste algoritmo são as conexões de entrada, onde cada uma possui um peso a ser determinado, camadas subsequentes de neurônios, um elemento acumulador para concentrar os sinais e, por fim, uma função de ativação, que pode assumir diversos formatos, para então, apresentar o valor da saída. Este princípio pode ser exemplificado na Figura 2.11:

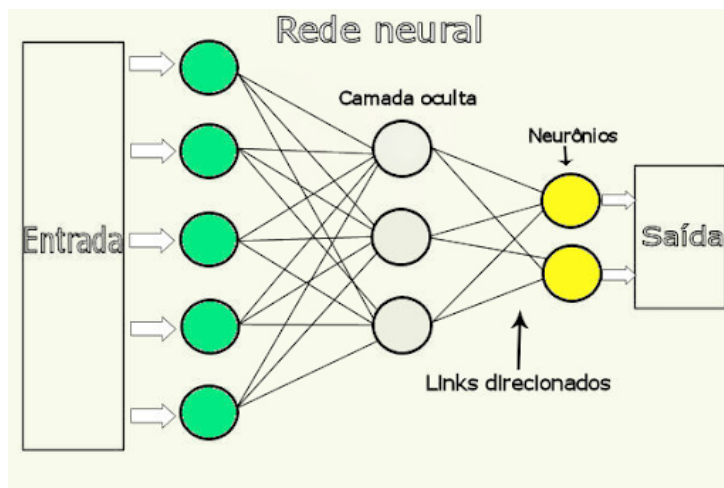


Figura 2.11: Exemplo de rede neural artificial.

Uma das maiores dificuldades encontradas no uso das redes neurais é a escolha da melhor arquitetura, uma vez que esse processo é experimental e demanda um grande tempo de execução. (FLECK et al., 2016) Dessa forma, a utilização desta técnica requer extensos testes com variadas configurações de modo a, experimentalmente, obter o modelo mais adaptado ao problema.

As redes neurais se assemelham ao cérebro humano em dois aspectos básicos: o conhe-

cimento é adquirido pela rede a partir de seu ambiente, por intermédio do processo de aprendizagem, e forças de conexão entre neurônios (pesos sinápticos) são utilizadas para armazenar o conhecimento adquirido (HAYKIN, 2001).

Na saída de cada neurônio, o sinal gerado passa por uma função de ativação, responsável por ponderar o efeito de cada saída na camada subsequente. Segundo HAYKIN (2001), existem diversos tipos de funções de ativação, dentre as mais conhecidas:

- Função *threshold*

Esta função apresenta um comportamento binário, resultando em uma saída em 1 ou 0 (Equação 2.8), dependendo do valor obtido pelo neurônio (Figura 2.12).

$$\begin{cases} 0 & \text{if } v < 0 \\ 1 & \text{if } v \geq 0 \end{cases} \quad (2.8)$$

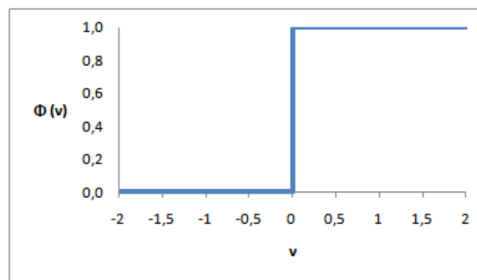


Figura 2.12: Exemplo de função de ativação *threshold*.

- Função sigmoide

O comportamento abrupto da função limiar (Equação 2.9) pode não ser indicado para alguns problemas. Dessa forma, a função sigmoide fornece níveis intermediários entre os valores 0 e 1, podendo ter o aclave ajustado conforme a necessidade (Figura 2.13).

$$q(v) = \frac{1}{1 + \exp(-\alpha v)} \quad (2.9)$$

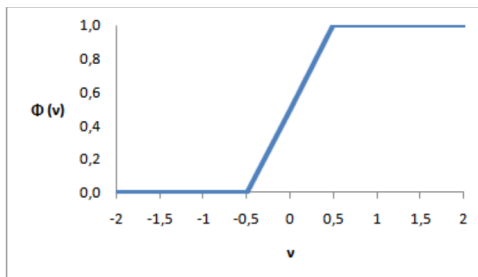


Figura 2.13: Exemplo de função de ativação sigmoide.

- Função tangente hiperbólica

Esta função apresenta um comportamento similar à sigmoide, entretanto a variação da saída está entre -1 e 1, em contraste com a anterior, cujo range compreende valores entre 0 e 1 (Figura 2.14) (Equação 2.10).

$$q(v) = \tanh(v) \quad (2.10)$$

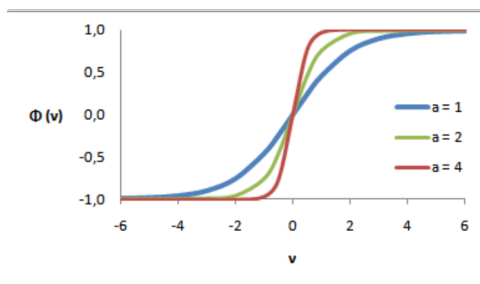


Figura 2.14: Exemplo de função de ativação tangente hiperbólica.

- Função ReLU

Esta função apresenta valor 0 para quaisquer saída negativa do neurônio (Equação 2.11). Entretanto, para valores positivos, esta função assume o próprio valor de saída. Em suma, esta função de ativação elimina saídas com valores negativos (Figura 2.15).

$$\begin{cases} 0 & \text{if } v < 0 \\ v & \text{if } v \geq 0 \end{cases} \quad (2.11)$$

A aprendizagem de uma rede neural é um processo onde os parâmetros livres são adaptados através de um processo de estimulação pelo ambiente em que a rede está inserida (FLECK et al., 2016). Ainda segundo FLECK et al. (2016), O aprendizado de uma RNA é realizado por meio de processos iterativos de ajustes aplicados aos pesos sinápticos, o chamado treinamento.

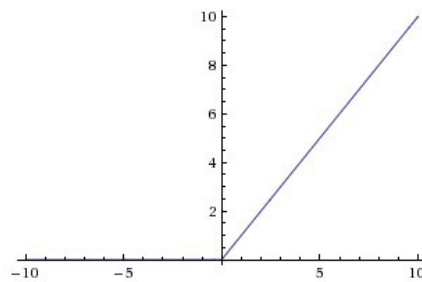


Figura 2.15: Exemplo de função de ativação ReLU

O treinamento pode ser ou não supervisionado. Para o primeiro caso, nenhuma saída desejada é informada ao algoritmo. Dessa forma, o algoritmo compara as saídas obtidas com os sinais de entrada, uma vez que não há valores de referência que devem ser buscados pelo treinamento. Nesta situação, critério de parada pode ser o número de épocas de treinamento, uma vez que não há a possibilidade de comparar o erro entre o resultado obtido com o desejado.

Para a aprendizagem supervisionada, o algoritmo recebe dados rotulados e sua tarefa é ajustar os parâmetros internos da rede de modo que, ao receber um dado novo, este algoritmo seja capaz de classificar corretamente.

2.5.3 KNN

O KNN é um classificador não paramétrico que atribui à amostra desconhecida a mesma classe das k amostras mais próximas, segundo um critério de decisão (LOBAO; PINTO, 2020). Para reconhecimento de padrões, o algoritmo KNN é um método para classificar objetos com base em exemplos de treinamento mais próximos no espaço (IMANDOUST; BOLANDRAFTAR, 2013). Este modelo foi proposto por Fukunaga e Narendra (1975) e é um classificador de implementação simples que pode obter resultados bem precisos, dependendo da aplicação.

O KNN usa os dados diretamente para classificação, sem construir um modelo primeiro (DREISEITL; OHNO-MACHADO, 2002). Assim, nenhum detalhe de construção do modelo devem ser considerados, e o único parâmetro ajustável no modelo é k , o número de vizinhos mais próximos para incluir na estimativa de classe adesão: o valor de $P(y|x)$ é calculado simplesmente como a proporção de membros da classe y entre os k mais próximos vizinhos de x . Variando k , o modelo pode ser feito mais ou menos flexível (pequenos ou grandes valores de k , respectivamente) (DREISEITL; OHNO-MACHADO, 2002).

Em outras palavras, o princípio sobre o qual o KNN se baseia é de que amostras similares

tendem a se concentrar em uma mesma região do espaço (LOBAO; PINTO, 2020). Dessa forma, é determinado como parâmetro o número de vizinhos que devem ser considerados para a classificação de um dado posterior. Este processo pode ser visualizado na figura 2.16.

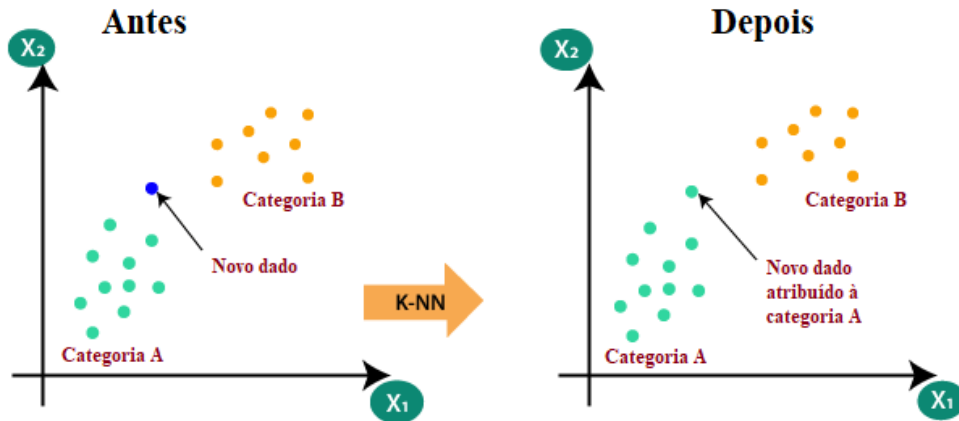


Figura 2.16: Algoritmo KNN classificando um dado novo com base na proximidade com outros dados.

Esta regra simplesmente retém todo o conjunto de treinamento durante o aprendizado e atribui a cada consulta uma classe representada pelo rótulo da maioria de seus vizinhos mais próximos no conjunto de treinamento (IMANDOUST; BOLANDRAFTAR, 2013).

Este método de classificação também apresenta uma boa acurácia e precisão, em comparação com os métodos citados anteriormente. O algoritmo consiste em estimar a distância entre os dados para determinar os limites de classificação. É importante ressaltar que é possível representar graficamente os limites de separação de dados representados por até três dimensões. Com dimensões superiores, o método é capaz de obter a classificação, entretanto não tem sentido físico a representação de mais dimensões.

Para a determinação do grau de semelhança de uma entrada em questão com os vizinhos mais próximos, existem diferentes métodos de cálculo da distância entre os dados, dentre eles a distância euclidiana, distância de *Hamming* e distância de *Manhattan*, sendo o primeiro o mais utilizado com variáveis de entrada do mesmo tipo:

- Distância euclidiana (Equação 2.12) - é calculada a partir da raiz quadrada da soma das diferenças quadráticas entre dois pontos, sendo um previamente conhecido, e o outro o dado novo que deseja-se obter a classificação.

$$(x, xi) = \sqrt{\sum((xj - xij)^2)} \quad (2.12)$$

- Distância de *Manhattan* (Equação 2.13) - também conhecido como "*City block*", é um método que determina a distância entre dois pontos a partir da soma das diferenças absolutas das coordenadas.

$$D_{\text{Manhattan}}(x, y) = \|\mathbf{x} - \mathbf{y}\|_1 = \sum_{i=1}^n |x_i - y_i| \quad (2.13)$$

Neste algoritmo, a escolha do número de vizinhos para avaliação deve ser escolhido com cautela, uma vez que a variação do valor deste hiperparâmetro pode alterar a classificação do dado, conforme pode ser visualizado na Figura 2.17, onde ao considerar três vizinhos, o dado novo é classificado como pertencente à classe B. Entretanto, ao elevar o número de vizinhos para sete, este dado torna-se pertencente à classe A. Essa característica é única a cada *dataset*, devendo ser analisada e testada, o que torna a escolha do parâmetro, uma etapa essencial no *tuning* do modelo.

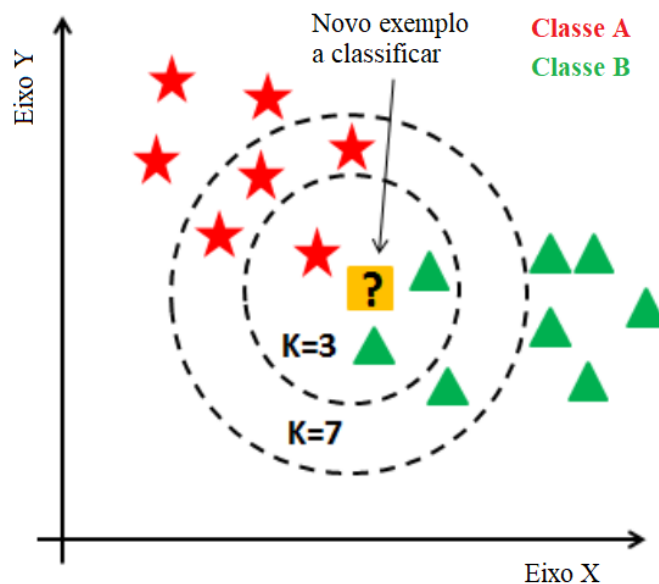


Figura 2.17: Hiperparâmetro "k" influencia na classificação de um dado novo.

Por fim, para a determinação da classe do dado, é necessário calcular a moda, ou seja, o dado mais frequente do conjunto obtido. Dessa forma, o algoritmo atribui ao dado novo a classificação destes dados que mais se repetem.

De acordo com Dreiseitl e Ohno-Machado (2002), a vantagem que o algoritmo *K-Nearest Neighbor* têm sobre outros algoritmos é o fato de que os vizinhos podem fornecer uma explicação para o resultado da classificação; essa explicação baseada em casos pode fornecer uma vantagem em áreas onde os modelos de caixa preta são inadequados.

Entretanto, a grande desvantagem dos k-vizinhos mais próximos está no cálculo da vi-

zinhança do caso: para isso, é preciso definir uma métrica que meça a distância entre os itens de dados. Na maioria das áreas de aplicação, não está claro como, a não ser por tentativa e erro, definir uma métrica de tal forma que a importância relativa, mas, desconhecida, dos componentes de dados seja refletida na métrica.

2.5.4 *Random Forest*

Este algoritmo divide repetidamente o conjunto de dados de acordo com um critério que maximiza a separação dos dados, resultando em uma estrutura em forma de árvore. O critério mais comum empregado é o ganho de informação; isso significa que a cada divisão, a diminuição na entropia devido a essa divisão é maximizada. A estimativa de $P(y|x)$ é a razão dos elementos da classe y sobre todos os elementos do nó folha que contém o item de dados x (DREISEITL; OHNO-MACHADO, 2002).

Uma grande desvantagem das árvores de decisão é dada pelo processo de construção ganancioso: em cada etapa, a combinação de única melhor variável e ponto de divisão ótimo é selecionada; no entanto, uma antecipação de várias etapas que considere combinações de variáveis pode obter resultados diferentes, e melhores.

Outra desvantagem reside no fato de que variáveis contínuas são implicitamente discretizadas pelo processo de divisão, perdendo informações ao longo do caminho. Ainda segundo Dreiseitl e Ohno-Machado (2002), em comparação com os outros métodos de aprendizado de máquina mencionados, as árvores de decisão têm a vantagem de não serem modelos de caixa preta, mas podem ser facilmente expressas como regras. Em muitos domínios de aplicação, essa vantagem pesa mais do que as desvantagens.

Melhorias significativas na precisão de algoritmos de classificação foram obtidos por meio da construção de conjuntos de árvores de decisão para que, então, estas votem pela classe mais popular para um determinado dado (BREIMAN, 2001). O *Random Forest* é um algoritmo de aprendizagem supervisionada baseado em árvores de decisão proposto por Breiman (2001) no qual um determinado número de árvores de decisão são criados em paralelo. A árvore de decisão é um classificador estruturado na forma de um grafo com nós de decisão e arcos, aos quais estão associadas às regras que serão aplicadas na classificação

De forma geral, a árvore de decisão apresenta bons desempenhos em tarefas de classificação. Contudo, para problemas que envolvam grandes conjuntos de dados e alta variância, a *Random Forest* pode ser mais adequada. Uma das formas de se implementar a *Random Forest* é a utilização do *bagging*. O *bagging* particiona o conjunto de treino em diversos subconjuntos.

Para cada subconjunto, *features* são selecionadas e um modelo é treinado formando diversas árvores cujos resultados são combinados. Ao estimar uma entrada, cada árvore irá classificar o subconjunto de forma independente, sendo que a decisão a respeito da classe que a amostra desconhecida pertence será resultado de um processo de votação, e a classe atribuída será aquela com maior número de votos (LOBAO; PINTO, 2020).

De acordo com Cutler, Cutler e Stevens (2012), de um ponto de vista computacional, Random Forests são interessantes, pois:

- São capazes de lidar naturalmente com a regressão e a classificação (multiclasse);
- São relativamente rápidos de treinar e prever;
- Dependem apenas de um ou dois parâmetros de ajuste;
- Possuem uma estimativa embutida de erro de generalização;
- Podem ser usados diretamente para problemas de alta dimensão;
- Podem ser facilmente implementados em paralelo.

De acordo com Cutler, Cutler e Stevens (2012), embora as *Random Forests* tenham a reputação de funcionar muito bem com os parâmetros padrão, existem três parâmetros que podem ser ajustados para fornecer maior precisão em situações específicas:

- m , o número de variáveis preditoras selecionadas aleatoriamente escolhidas em cada nó;
- J , o número de árvores na floresta;
- *tree_size*, medido pelo menor tamanho de nó para divisão ou o número máximo de nós terminais.

Ainda de acordo com Cutler, Cutler e Stevens (2012), O único desses parâmetros ao qual Random Forests é um pouco sensível é o m . Na classificação, o padrão padrão é $m = \sqrt{M}$, onde M é o número total de preditores. Na regressão, o padrão é $m = N/3$, onde N é o tamanho da amostra.

Um dos problemas observados em outros algoritmos de classificação é a possibilidade de *overfitting*, ou seja, o superdimensionamento do modelo, o que pode ocasionar o aprendizado de dados ruidosos, ao mesmo tempo que exige elevado poder computacional para resolução do modelo. Este fenômeno, entretanto, não ocorre no *Random Forest*, uma vez

que o próprio algoritmo faz com que as árvores de decisão individuais de uma floresta aleatória enfatizem diferentes características. A diversidade de árvores resultante pode capturar padrões de recursos mais complexos do que uma única árvore de decisão e reduz a chance de *overfitting* para dados de treinamento (DENISKO; HOFFMAN, 2018).

A estrutura deste algoritmo pode ser visualizada na Figura 2.18, onde um *dataset* de *input* (A (Figura 2.18)) contém três amostras de cada uma das cinco classes (x_1, x_2, x_3, x_4 e x_5). Em (B (Figura 2.18)), observa-se uma árvore de decisão, onde cada ponto de decisão possui uma regra que atribui uma amostra a um ramo ou outro, dependendo de um valor de característica. Os ramos terminam em folhas pertencentes à classe vermelha ou à classe amarela. Esta árvore de decisão classifica a amostra 1 para a classe vermelha (DENISKO; HOFFMAN, 2018).

Em (C (Figura 2.18)), outra árvore de decisão é definida, com regras diferentes em cada ponto de decisão. Esta árvore também classifica a amostra 1 para a classe vermelha. Em (D (Figura 2.18)), uma floresta aleatória combina votos de suas árvores de decisão constituintes, levando a uma previsão de classe final. Por fim, em (E (Figura 2.18)) A previsão de saída final é novamente a classe vermelha (DENISKO; HOFFMAN, 2018).

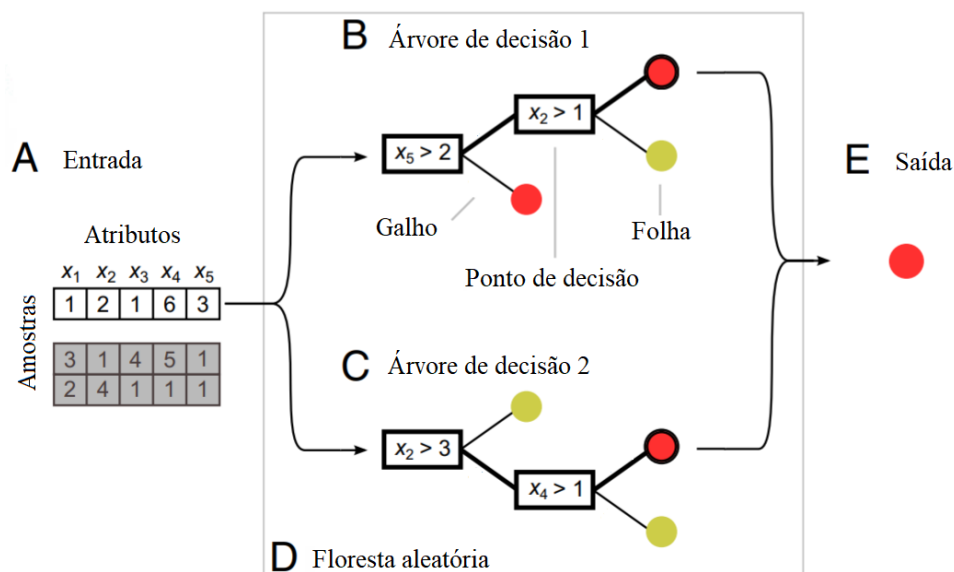


Figura 2.18: Estrutura de um algoritmo do tipo *Random Forest*.

2.6 Trabalhos correlacionados

Existem diversos trabalhos dedicados a desenvolver metodologias para classificação de falhas em motores elétricos por meio da análise de vibração. Trabalhos como o proposto por Marins et al. (2018) propõem a utilização de *Similarity-based models* para a classi-

ficação das falhas. O SBM é uma técnica pela qual a operação normal de um sistema é modelada para detectar falhas analisando sua semelhança com os estados normais do sistema. Proposto pela primeira vez há cerca de duas décadas, o SBM tem sido usado com sucesso para detecção de falhas em sistemas variados. Apesar desse sucesso, não há muitos estudos realizados na literatura sobre seu design, que englobe tanto métricas de similaridade quanto treinamento de modelos (MARINS et al., 2018).

Marins et al. (2018) utilizou dois *datasets* na sua pesquisa: o MaFaulDa, também utilizado nesta pesquisa, e o CWRU (Case Western Reserve University) (SMITH; RANDALL, 2015), composto por 161 cenários agrupados em quatro categorias. Cada cenário pode ser composto por sinais de aceleração em três direções: no rolamento final do eixo, que ocorre em todos os cenários; na carcaça do rolamento do eixo no lado do ventilador do motor, que ocorre na maioria dos cenários; e na placa base de suporte do motor, o que ocorre em alguns cenários. As taxas de amostragem utilizadas foram de 12 kHz para alguns cenários e 48 kHz para outros. Os sinais de vibração foram obtidos de diferentes estados dos rolamentos: condição normal, falha na pista interna, falha na esfera e falha na pista externa.

O banco de dados de rolamentos CWRU foi selecionado por dois motivos principais. A primeira é a sua disponibilidade pública. A segunda é a sua ampla utilização na literatura para reportar resultados de métodos automáticos de detecção de falhas em rolamentos, o que permite comparar o desempenho do método proposto com o de outros trabalhos. Neste artigo, apenas cenários contendo os sinais medidos no lado do ventilador do motor e na extremidade oposta foram utilizados, reduzindo o número total de cenários válidos para 153 (MARINS et al., 2018).

Assim como nesta pesquisa, Marins et al. (2018) precisou realizar etapas de pré processamento no sinal contido nos *datasets* de modo a reduzir os dados originais a um conjunto de valores mais informativos, relevantes e menos redundantes.

Dada a natureza distinta de cada um dos bancos de dados empregados neste trabalho, o bloco de pré-processamento deve ser diferente para cada banco de dados, embora sua finalidade seja a mesma para ambos. Os dois blocos de pré-processamento propostos por Marins et al. (2018) são descritos a seguir:

- MaFaulDa: três tipos de atributos foram extraídos da série temporal multivariada original: a frequência de rotação, 21 atributos espectrais adicionais e 24 outros atributos estatísticas. A frequência de rotação fr foi determinada diretamente a partir da transformada discreta de Fourier (DFT) do sinal do tacômetro. Os recursos estatísticos adicionais incluem, para cada um dos oito sinais medidos em cada cenário operacional, a média estatística, a entropia e a curtose. A variância não foi utilizada,

pois os sinais são normalizados para a variação da unidade para reduzir o efeito das variações de energia causadas por alterações na configuração de aquisição;

- CWRU : As características estatísticas apresentadas por [Marins et al. \(2018\)](#), juntamente com a média, variância e entropia foram extraídas de cada sinal, totalizando 36 características.

As características extraídas são então inseridas nos estágios subsequentes para realizar a detecção e classificação de falhas. As duas bases de dados são tratadas de forma independente para avaliação de desempenho dos métodos propostos.

Todo o banco de dados MaFaulDa foi separado aleatoriamente em dois conjuntos de treinamento e teste disjuntos, compreendendo respectivamente 90% e 10% dos cenários fornecidos. A escolha aleatória de cada conjunto foi restringida para que ambos apresentassem a mesma proporção de falhas de todo o banco de dados. O melhor conjunto de parâmetros foi escolhido usando um procedimento de *k-fold cross validation* nas amostras de treinamento, com $k = 10$. Em seguida, o desempenho dos melhores modelos é avaliado no conjunto de teste, produzindo os resultados finais.

Quanto ao banco de dados CWRU, foi aplicado um processo semelhante à validação cruzada de conjuntos de dados. As melhores configurações encontradas para o MaFaulDa são usadas diretamente no banco de dados CWRU. Assim, esta base de dados é utilizada para avaliar o poder de generalização dos classificadores obtidos com a metodologia proposta. Os resultados foram obtidos também usando *k-fold cross validation* com $k = 10$.

[Souza, Lepikson e Sperandio \(2019\)](#) apresentou uma revisão bibliográfica com o objetivo de identificar técnicas que estão sendo pesquisadas e apresentadas, através de trabalhos científicos, que abordem assuntos relacionados a aprendizagem de máquina, aprendizagem profunda e análise de vibração de equipamentos rotativos, apresentem o estado da arte e, através dos seus resultados, propor um modelo capaz de realizar a classificação de falhas com maior exatidão.

Um dos trabalhos encontrados por [Souza, Lepikson e Sperandio \(2019\)](#) foi a pesquisa proposta por [Li, Wang e Wang \(2019\)](#) que propõe a comparação de diversos algoritmos: *Deep Learning*, *Back Propagation Neural Network*, *Deep Belief Network*, *Support Vector Machines* e *K-Nearest Neighbor*. Entretanto, as condições de falha analisadas nesta pesquisa foram apenas desalinhamento, falha em rolamentos e atrito no eixo principal.

Em comparação com esta pesquisa, o trabalho supracitado não engloba a detecção de falhas como o desbalanceamento, ao passo que considera o desalinhamento nos eixos horizontal e vertical como a mesma condição. Além disto, não é detalhado o tipo de falha

no rolamento.

Os dados utilizados por [Li, Wang e Wang \(2019\)](#) foram coletados pelos autores em laboratório. Dentre as 10.216 amostras coletadas, 6.817 amostras foram coletadas para classificação de falhas e 3.399 amostras foram aplicadas para avaliação de degradação, respectivamente.

Este trabalho foi capaz de obter os seguintes valores de instâncias corretamente classificadas para os algoritmos, conforme Tabela 2.1:

Tabela 2.1: Taxa de acerto

Algoritmo	Acurácia
<i>Back Propagation Neural Network</i>	99.75%
<i>Support Vector Machines</i>	99.85%
<i>K-Nearest Neighbor</i>	99.85%
<i>Deep Belief Network</i>	99.80%
<i>Deep Neural Network</i>	99.87%

Por outro lado, o trabalho apresentado por [Souza et al. \(2021\)](#) propõe um modelo de manutenção preditiva (PdM) baseado em Redes Neurais Convolucionais (CNN) para a classificação das falhas. As CNN são amplamente utilizadas no campo da visão computacional, que são constituídas por pesos compartilhados espacialmente e subamostragem espacial (*pooling*) com sequência bidimensional (2D) ou tridimensional. Além disso, também tem sido usadas em aplicações de fala, áudio, texto e séries temporais com sequência unidimensional (1D). Em contraste com as imagens, que são distribuídas espacialmente em duas dimensões, os sinais de saída do sensor usados no domínio de aplicações de engenharia elétrica são mais frequentes e frequentemente encontrados como dados de séries temporais 1D.

No modelo, primeiramente, foram utilizados ReLU e normalização de peso. Em seguida, a *softmax* e a função de perda foram aplicadas para otimizar as falhas em máquinas rotativas. O modelo proposto foi validado aplicando e analisando seus resultados para dois conjuntos de dados disponíveis publicamente e do mundo real com diferentes classes de falhas e graus de desequilíbrio.

Para otimizar o desempenho do modelo, para cada dado do sensor de vibração, a Transformada Rápida de *Fourier* (FFT) correspondente foi extraída e utilizada como vetor de entrada para as tarefas de treinamento, validação e teste. A extração de recursos oferece uma grande vantagem ao reduzir drasticamente o tamanho do vetor de entrada em comparação com o número de amostras na série temporal. Além disso, esta etapa do processo permite a utilização de um vetor com maior capacidade de discriminação das classes de

falha do que o obtido com a série temporal original.

Em vez de utilizar todos os dados dos sensores de vibração, como foi realizado no trabalho proposto por [Marins et al. \(2018\)](#) e, também, como será feito nesta pesquisa, o modelo proposto foi projetado para utilizar apenas os dados dos sinais captados pelos três acelerômetros unidirecionais, o que equivale a um acelerômetro triaxial, instalado no drive fim ([SOUZA et al., 2021](#)).

[Ali et al. \(2019\)](#) apresentou um trabalho que consiste em um algoritmo do tipo *Multi-layer perceptron* para identificar falhas. Diferente dos outros apresentados, o autor utilizou apenas dados estatísticos e as frequências de rotação que foi obtida por meio da Transformada Discreta de Fourier (DFT). Entretanto foi implementado uma técnica denominada *Synthetic Minority Over-sampling Technique* (SMOTE) para mitigar o efeito dos desbalanceamento nos dados que ocorre quando os dados não são igualmente representados. Para resolver o problema, uma solução possível é reamostrar o *dataset* original. Isso pode ser feito por superamostragem da classe minoritária ou subamostragem da classe majoritária. Portanto, para reamostragem do conjunto de dados, [Ali et al. \(2019\)](#) utilizou a supracitada técnica SMOTE.

O objetivo buscado por [Ali et al. \(2019\)](#) foi propor a arquitetura mais simples da RNA para diferenciar, entre as rotações simuladas, falhas de máquinas abordadas no *dataset* e mostrando a eficácia da abordagem para lidar com classes de desequilíbrio com a ajuda da precisão do sistema proposto. Nesta pesquisa, também foi utilizado o *dataset* MaFaulDa (Figura 2.19).

Tabela I
Matriz de confusão sem SMOTE

Classe	Objetivo					
	Desal.Horiz.	Desbalanceamento	Normal	Anel externo	Anel interno	Desal.Vert.
Desal.Horiz.	31/59	0/100	0/15	5/154	0/168	23/90
Desbalanceamento	0/59	88/100	0/15	1/154	2/168	9/90
Normal	7/59	1/100	0/15	0/154	0/168	7/90
Anel externo	6/59	0/100	0/15	134/154	12/168	2/90
Anel interno	1/59	9/100	0/15	4/154	152/168	2/90
Desal.Vert.	8/59	7/100	0/15	4/154	0/168	71/90
Total	81.2%					

Tabela II
Matriz de confusão utilizando SMOTE

Classe	Objetivo					
	Desal.Horiz.	Desbalanceamento	Normal	Anel externo	Anel interno	Desal.Vert.
Desal.Horiz.	161/167	0/167	0/167	2/167	1/167	3/167
Desbalanceamento	0/167	164/167	0/167	0/167	3/167	0/167
Normal	0/167	0/167	168/167	0/167	0/167	0/167
Anel externo	4/167	0/167	2/167	154/167	3/167	5/167
Anel interno	1/167	3/167	0/167	2/167	162/167	0/167
Desal.Vert.	4/167	3/167	0/167	0/167	2/167	158/167
Total	96.2%					

Figura 2.19: Matrizes de confusão do algoritmo RNA com e sem a implementação da técnica SMOTE.

O algoritmo proposto, portanto, obteve uma acurácia de 96.2%, mostrando a efetividade do modelo e das técnicas utilizadas. Ao comparar com o MLP sem a implementação do SMOTE, o autor foi capaz de obter uma acurácia de 81.2%. O autor apresenta as matrizes de confusão do algoritmo em ambas situações na Figura 2.19. É importante notar que, além da melhoria nos valores gerais de acurácia, classes como a *horizontal misalignment* e *vertical misalignment* apresentaram classificações notoriamente mais assertivas, com grande redução nas classificações incorretas.

Por fim, o trabalho desenvolvido por Janssens et al. (2016) apresenta um modelo de classificação de falhas em rolamentos baseado em *Convolutional Neural Networks* (CNN) e *Random Forest*. Para tal, o *dataset* utilizado consiste em dados coletados em laboratório, com diferentes cenários de falhas induzidas, como defeitos nos anéis interno e externo e falta de lubrificação adequada. O modelo proposto apresentou acurácia de 93.61% para o classificador baseado em CNN, e 87.25% para o *Random Forest*.

Projeto e desenvolvimento da pesquisa

Este capítulo apresenta o desenvolvimento do modelo computacional para classificação de falhas em motores elétricos por meio de análise de vibração.

3.1 *Dataset*

Para implementação e treinamento deste modelo, foi selecionado *dataset* MaFaulDa (RIBEIRO et al., 2019) que contém 1951 séries multivariáveis contendo leituras de sensores de vibração. Este conjunto de dados abrange as seguintes condições do equipamento: funcionamento normal, desbalanceamento, desalinhamento axial e radial e falhas nos rolamentos internos e externos.

Cada leitura contida neste *dataset* é composta por três acelerômetros de um eixo, modelo IMI 601A01, posicionados, cada um, nos eixos radial, axial e tangencial, um acelerômetro triaxial, modelo IMI 604B31, um tacômetro analógico, modelo MT-190 e um microfone modelo Shure SM81. Para esta pesquisa, serão considerados apenas os dados referentes ao acelerômetros em todos os eixos supracitados.

O primeiro sensor, o acelerômetro uniaxial, possui uma sensibilidade de 100 mV/g, sendo capaz de aferir acelerações entre -50 g e +50 g, e com um espectro de frequência de 0.27 Hz até 10.000 Hz. O princípio de funcionamento deste equipamento é um sensor piezoelétrico, onde uma massa em contato direto com o elemento piezoelétrico e, ao ser submetido a uma aceleração, esta massa exerce uma força mecânica na carga que pode ser convertida em sinal elétrico, dessa forma, podendo ser lido por meio de um conversor analógico para digital conectado a um controlador. Este tipo de sensor é indicado para aplicações industriais que exigem resposta em frequência elevada enquanto devem manter respostas estáveis em ambientes com variação de temperatura.

O segundo sensor, o acelerômetro triaxial, também possui a mesma sensibilidade do anterior (100 mV/g) e o mesmo alcance de medição (-50 g até + 50g), entretanto sua resposta em frequência varia entre 0.5 e 5.000 Hz. Este equipamento possui elemento sensor de cerâmica, o que resulta em uma baixa interferência por ruídos externos, além de sofrer variações mínimas de leitura quando submetido a diferentes temperaturas.

No *dataset*, cada sequência de dados foi obtida com uma taxa de amostragem de 50 kHz

durante 5 segundos, totalizando 250.000 amostras (RIBEIRO et al., 2019), conforme distribuição apresentada na Figura 3.1. Esta taxa de amostragem é elevada, entretanto permite a reconstrução do sinal com maior fidelidade, além de permitir que sejam detectadas falhas que se apresentam em harmônicos elevados. O *dataset* completo é composto de diversos arquivos CSV (*comma-separated values*), organizados em pastas por tipo de falha. Cada arquivo está nomeado por valor de frequência de operação do motor em Hertz.

Sequência	Medidas
Normal	49
Desalinhamento horizontal	197
Desalinhamento vertical	301
Desbalanceamento	333
Rolamento inferior	
Falha na gaiola	188
Pista externa	184
Falha na esfera	186
Rolamento externo	
Falha na gaiola	188
Pista externa	188
Falha na esfera	137
Total	1951

Figura 3.1: Distribuição dos dados do *dataset* MaFaulDa

Marins et al. (2018) descreve com maior detalhamento o conteúdo e organização do *dataset*:

- Normal: esta classe representa o sistema operando em condições normais sem qualquer falha. Inclui um conjunto de 49 cenários distintos, cada um com uma velocidade de rotação fixa na faixa de 737 rpm a 3686 rpm com passos de aproximadamente 60 rpm;
- Desbalanceamento: para simular diferentes graus de operação desequilibrada, valores de carga distintos de 6 g, 10 g, 15 g, 20 g, 25 g, 30 g e 35 g foram acoplados ao

rotor. Para cada valor de carga abaixo de 30 g, a frequência de rotação assumiu os mesmos 49 valores empregados no caso de operação normal. Para cargas iguais ou superiores a 30 g, entretanto, a vibração resultante inviabiliza o sistema de atingir frequências de rotação acima de 3300 rpm, limitando o número de frequências de rotação distintas a apenas 44 nesses casos. Como tal, o banco de dados inclui um total de 333 cenários de operação de desequilíbrio diferentes;

- Desalinhamento horizontal paralelo: este tipo de falha foi induzida no motor deslocando o eixo do motor horizontalmente de 0,5 mm, 1,0 mm, 1,5 mm e 2,0 mm. Usando o mesmo intervalo para a frequência de rotação como na operação normal para cada deslocamento horizontal, um total de 197 cenários diferentes foram considerados para esta classe;
- Desalinhamento vertical paralelo: esta falha foi induzida no motor deslocando o eixo do motor verticalmente de 0,51 mm, 0,63 mm, 1,27 mm, 1,4 mm, 1,78 mm e 1,9 mm. Usando a mesma faixa de frequência de rotação que na operação normal para cada deslocamento vertical, um total de 301 cenários diferentes foram considerados para esta classe de falha;
- Falhas em rolamentos: por ser um dos elementos mais complexos da máquina, os rolamentos são os elementos mais suscetíveis à ocorrência de falhas. Foram utilizados três rolamentos defeituosos, cada um com um elemento defeituoso distinto (pista externa, corpos rolantes e pista interna), que foram colocados um de cada vez em duas posições diferentes na bancada experimental: entre o rotor e o motor (posição de balanço), ou na posição externa, tendo o rotor entre o mancal e o motor (posição de balanço). As falhas nos rolamentos são praticamente imperceptíveis quando não há desequilíbrio. Assim, as três massas de 6 g, 10 g e 20 g foram adicionadas para induzir um efeito detectável, com diferentes frequências de rotação como antes, levando a um total de 558 cenários de *underhang* e 513 cenários de *overhang*.

3.2 Modelo computacional

O modelo computacional consiste em duas camadas de software: a primeira para realizar o pré processamento dos dados, onde é realizada a transformação do domínio do tempo para a frequência e, posteriormente, a extração de dados estatísticos referentes aos conjuntos de dados. Por fim, a saída desta camada resulta em arquivo que contém os dados anotados com as respectivas classes. A segunda camada é responsável pela classificação, onde é realizado o treinamento por meio dos algoritmos selecionados nesta pesquisa: RNA, SVM, KNN e RF. Em seguida, a rede é treinada de modo a avaliar a eficácia de cada algoritmo. A Figura 3.2 descreve a arquitetura do modelo.

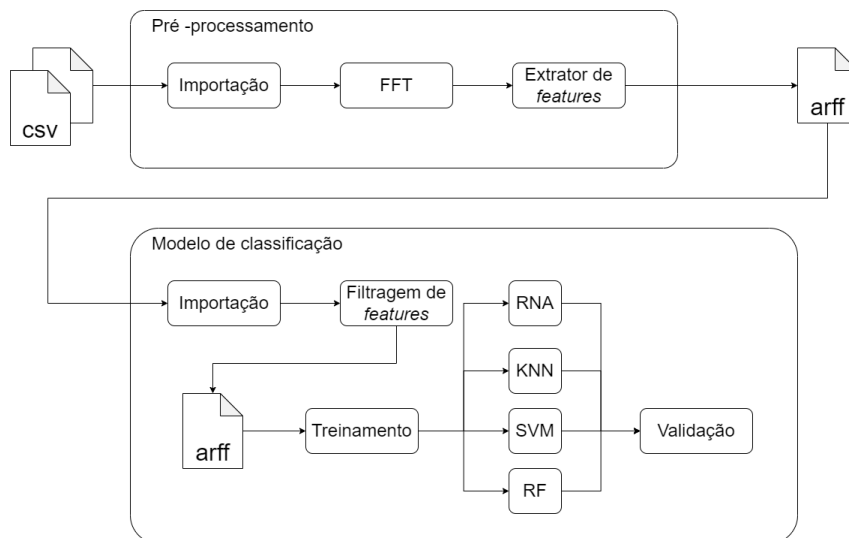


Figura 3.2: Arquitetura do modelo computacional.

3.2.1 Pré processamento

A etapa de pré processamento consiste em importar os dados do *dataset*, originalmente dispostos em arquivos CSV para variáveis de programa. Em seguida, deve-se realizar a Transformada Rápida de *Fourier* em todos os sinais para extrair o espectro de frequência. Em seguida, o algoritmo de pré-processamento recuperará os recursos estatísticos, como desvio padrão, curtose, média, assimetria e variância (TOMA; PROSVIRIN; KIM, 2020). Todas essas informações serão armazenadas em um arquivo ARFF que será utilizado para treinar os classificadores.

Esta camada foi desenvolvida na linguagem de programação Java, por meio da IDE *Visual Studio Code*. A escolha desta linguagem deve-se à possibilidade de executar os algoritmos em diversos sistemas operacionais, sem a necessidade de adaptar o código ao sistema. A IDE permite a instalação de extensões para executar o código em Java e realizar a depuração substituindo a IDE Eclipse.

A primeira etapa durante a execução do programa é buscar os arquivos do tipo CSV no diretório onde se encontra o *dataset* e carregar no algoritmo. Os dados estão distribuídos em pastas correspondentes à cada tipo de condição do equipamento. No algoritmo, é necessário manter este padrão de anotação.

Para cada um dos oito sinais contidos no *dataset* importado, é realizada a Transformada de *Fourier*. Esta operação converte os sinais de vibração para o espectro da frequência, onde é possível extrair o comportamento gerado por cada tipo de falha. Realizar a análise com o sinal no espectro do tempo torna-se impraticável, justamente devido ao sinal ser

composto pela soma de sinais em diferentes frequências e não possuírem componentes que facilmente permitam a distinção entre as condições.

A extração de atributos reduz a dimensão dos dados brutos iniciais, convertendo-os em dados menores, grupos mais gerenciáveis para processamento posterior. Na maioria das vezes, o conjunto de dados brutos contém um alto número de variáveis, exigindo assim um grande poder computacional para processamento. O principal objetivo deste método é selecionar e converter as variáveis em atributos, de modo a compactar efetivamente os dados que devem ser processados, e descrever precisamente o conjunto de dados brutos original (TOMA; PROSVIRIN; KIM, 2020).

Tabela 3.1: Variáveis estatísticas

Atributo	Equação	Atributo	Equação
Média	$\mu = \frac{1}{N} \sum_{i=1}^N x_i$	Distorção	$x_{skewness} = \frac{1}{N} \frac{\sum_{i=1}^N (x_i - \mu)^3}{\sigma^3}$
Mediana	$median = (\frac{(N+1)}{2})^{th} value$	Curtose	$x_{kurtosis} = \frac{1}{N} \frac{\sum_{i=1}^N (x_i - \mu)^4}{\sigma^4}$
Desvio padrão	$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N-1}}$	RMS	$RMS = \frac{\sqrt{\sum_{i=1}^N x_i^2}}{N}$
Variância	$\sigma^2 = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N-1}$	Soma	$Sum = \sum_{i=1}^N x_i$

Na Tabela 3.1 podem ser visualizados os atributos que foram extraídos dos sinais. Tais variáveis estatísticas foram obtidas para cada um dos oito conjuntos de dados contidos no *dataset* MaFaultDa, totalizando 80 atributos no total. Os atributos gerados pelo programa são divididos conforme a seguinte distribuição:

- mean;
- quadratic_mean;
- max;
- min;
- skewness;
- deviation;
- kurtosis;
- sum;
- sumSquares;
- variance.

A parte final do programa realiza a escrita dos valores obtidos em um arquivo do tipo ARFF (*Attribute-Relation File Format*). Este formato consiste em um arquivo de texto ASCII que descreve uma lista de instâncias que compartilham um conjunto de atributos. Os arquivos ARFF foram desenvolvidos pelo Departamento de Ciência da Computação da Universidade de Waikato para uso com o software de aprendizado de máquina Weka (*Waikato Environment for Knowledge Analysis*) (EIBE; HALL; WITTEN, 2016).

3.2.2 Modelo de classificação

Com o auxílio do software de aprendizado de máquina Weka (EIBE; HALL; WITTEN, 2016), o próximo passo é realizar uma análise para determinar quais recursos são relevantes para os modelos. Portanto, outro arquivo do tipo ARFF é gerado incluindo os recursos mais relevantes. Em seguida, o conjunto de dados filtrado é usado para treinar classificadores usando algoritmos KNN, SVM e RNA.

3.2.2.1 Filtragem de atributos relevantes

O algoritmo *CFS subset evaluator*, utilizado para filtragem do *dataset* foi desenvolvido por Hall (1998), e consiste em avaliar o valor de um subconjunto de atributos considerando a capacidade preditiva individual de cada recurso junto com o grau de redundância entre eles. São preferidos subconjuntos de recursos que são altamente correlacionados com a classe, embora tenham baixa intercorrelação (HALL, 1998).

Para realizar a busca entre os dados do *dataset*, o algoritmo de filtro usa o método *Best-first search*. Este método pesquisa o espaço de subconjuntos de atributos por escalada gananciosa, aumentando com uma facilidade de retrocesso. Definir o número permitido de nós consecutivos que não melhoram controla o nível de retrocesso feito. O *Best-first search* pode começar com o conjunto vazio de atributos e pesquisar para a frente, ou começar com o conjunto completo de atributos e pesquisar para trás, ou começar em qualquer ponto e pesquisar em ambas as direções, considerando todas as possíveis adições e exclusões de atributos únicos em um determinado ponto (EIBE; HALL; WITTEN, 2016).

Após submeter o *dataset* original, com 80 atributos, ao algoritmo de filtro, o *CFS subset evaluator* determinou que os seguintes atributos são os mais relevantes para o classificador. Dessa forma, utilizar o *dataset* completo, sem filtro, resultará em um maior tempo para construção e treinamento dos modelos, devido ao número elevado de dados, sem representar um aumento expressivo na acurácia e precisão.

Tabela 3.2: Atributos selecionados após filtragem

Posição	Atributo
3	sensibility601A01_mean
5	sensibility601A01_quadraticMean
12	frequency601A01_max
16	frequency601A01_skewness
17	frequency601A01_deviation
21	measurement601A01_kurtosis
26	measurement601A01_skewness
33	sensibility604B31_mean
34	sensibility604B31_min
37	sensibility604B31_deviation
38	sensibility604B31_sum
42	frequency604B31_max
43	frequency604B31_mean
44	frequency604B31_min
47	frequency604B31_deviation
48	frequency604B31_sum
56	measurement604B31_skewness
73	sound_mean
78	sound_sum

Dessa forma, o *dataset* utilizado para estudo dos algoritmos de classificação contém os 18 atributos mais relevantes conforme o filtro de atributos. Este *dataset* foi utilizado para o treinamento de modelos de classificação por meio dos algoritmos RNA, KNN e SVM.

3.2.2.2 Redes Neurais Artificiais

O algoritmo de RNA foi configurado com uma taxa de aprendizado de 0,3. Este parâmetro varia entre 0 e 1 e influencia na velocidade de aprendizagem da rede, uma vez que um número mais próximo de 0 reflete em uma rede mais lenta, porém menos suscetível a erros. Entretanto, um valor mais próximo de 1 resulta em uma rede mais rápida, porém provoca instabilidades no aprendizado, o que pode até resultar em um modelo que não converge.

O valor escolhido para tal parâmetro foi baseado em testes, onde 0,3 resultou em uma melhor assertividade na classificação. Ao utilizar um valor inferior, a rede tornou-se excessivamente lenta, porém sem apresentar melhorias na acurácia, o que significa que o valor definido é o equilíbrio entre a taxa de aprendizagem, tempo necessário para treinamento do modelo, e acurácia.

O modelo foi construído com uma camada oculta contendo 14 neurônios, conforme exemplificado na Figura 3.3. A camada de entrada contém um neurônio para cada atributo

presente no *dataset*, enquanto que a camada de saída contém 10 neurônios, correspondente ao número de classes possíveis. Não há uma fórmula exata para a determinação de tais hiperparâmetros.

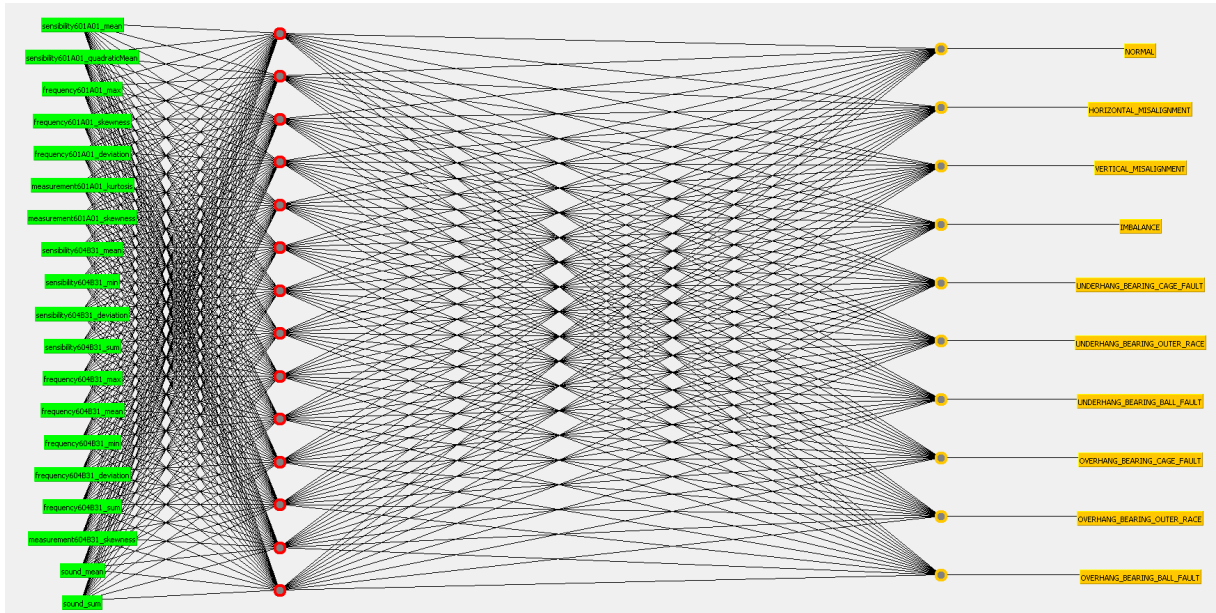


Figura 3.3: Estrutura da rede neural utilizada.

Entretanto, ao considerar o número de neurônios, um valor inferior ao utilizado no modelo faz com que o modelo não seja capaz de aprender os atributos do *dataset* com eficácia. Por sua vez, um número elevado demais resulta em *overfitting*, onde o classificador se torna mais suscetível a considerar ruídos no aprendizado, o que compromete a acurácia.

Neste caso, em vez do algoritmo aprender com base em uma tendência, um modelo superdimensionado pode memorizar atributos não previsíveis do conjunto de dados utilizado para treinamento (BILBAO; BILBAO, 2017).

Ainda segundo Bilbao e Bilbao (2017), o *overfitting* é percebido quando ocorre um erro de treinamento muito pequeno e, ao mesmo tempo, um erro de validação muito alto. E geralmente é devido a uma das seguintes causas:

- A primeira está relacionada ao tamanho ótimo da rede;
- A segunda, com a existência de *outliers* no conjunto de entrada (isso faz com que a variância dos parâmetros da rede seja alta);
- Uma terceira causa é quando algoritmos de resolução muito complexos são usados;
- E o último é quando o número de dados usados no treinamento é muito alto

Outro hiperparâmetro a ser considerado é o número de camadas ocultas. Da mesma forma que a quantidade de neurônios em uma camada pode prejudicar o algoritmo com valores demasiados ou insuficientes, o mesmo efeito pode ser observado no número de camadas

Por fim, o critério de parada pode ser escolhido como um valor fixo de épocas, ou um erro aproximado. O modelo em avaliação neste estudo foi limitado a 500 épocas e 4 casas decimais no erro. Estes valores foram obtidos por meio de testes, uma vez que incrementar tais limites resultam em um ganho mínimo na acurácia, enquanto demanda mais poder computacional.

3.2.2.3 *K-Nearest Neighbors*

Como o classificador KNN é um tipo de algoritmo mais simples, o hiperparâmetro a ser ajustado é o número de vizinhos. Para tal, também não há uma fórmula capaz de definir com exatidão o valor a ser escolhido, entretanto valores muito baixos podem tornar o classificador suscetível a interferência de ruídos no *dataset*, o que compromete a acurácia. Valores mais altos podem causar *overfitting* que acarreta em maior custo computacional e, também, redução na acurácia.

Após testes, o valor que apresentou o melhor resultado foi apenas 3 vizinhos, uma vez que o *dataset* possui um ruído muito baixo. Ao incrementar este valor, o modelo perde eficiência. Outra característica do modelo que pode ser alterada é a função de determinação da distância entre os vizinhos. Por meio de testes, o algoritmo apresentou um melhor resultado por meio da distância de *Manhattan*.

3.2.2.4 *Support Vector Machines*

O modelo SVM permite o ajuste de hiperparâmetros como o *gamma* e C, assim como a função *kernel*. Após testes comparando as funções disponíveis, o kernel polinomial apresentou o melhor resultado. O ajuste dos hiperparâmetros também foram feitos por meio de testes.

O parâmetro C (*cost*) determina o tamanho da margem da função de separação. Um valor maior resulta em uma margem menor, o que permite que o modelo realize a separação de maneira mais precisa. Entretanto, ao considerar um valor menor para C, uma margem maior pode resultar em dados incorretamente classificados. Para este modelo o parâmetro C definido para 10^4 , uma vez que o incremento deste valor não aumenta a acurácia de forma significativa, enquanto que os requisitos computacionais elevam-se.

3.2.2.5 *Random Forest*

Algoritmos do tipo *Random Forest* são uma combinação de preditores de árvore, de modo que cada árvore depende dos valores de um vetor aleatório amostrado de forma independente e com a mesma distribuição para todas as árvores na floresta (BREIMAN, 2001).

3.3 *Método de validação dos modelos*

Para realizar o treinamento dos três algoritmos, o *dataset* filtrado foi dividido em diversos blocos para validação cruzada. Este processo consiste em particionar o conjunto de dados em um determinado número n de seções, com aproximadamente a mesma quantidade de dados, de modo a realizar repetidos treinamentos utilizando $n - 1$ conjuntos, para então realizar a validação dos dados com o subconjunto restante (BERRAR, 2019).

Este processo repete-se n vezes, para que a cada treinamento uma nova combinação de subconjuntos seja utilizada para treinar a rede e, conseqüentemente, validar a rede com outros dados.

O resultado do modelo é dado pela média entre os valores obtidos por todas as validações, o que contribui para uma melhor acurácia do modelo, uma vez que múltiplas combinações entre os dados foram feitas tanto para treinamento como para validação.

É importante ressaltar que um número alto de divisões (*folds*) aumenta de forma notória o tempo de construção do modelo. Em contrapartida, considerar um número baixo pode contribuir negativamente para a precisão dos classificadores, uma vez que uma menor porcentagem dos dados será destinada para treinamento (BERRAR, 2019).

3.4 *Método de avaliação dos resultados*

Ao final da execução da classificação, o software Weka fornece outras métricas para a avaliação dos resultados das redes, além da acurácia geral de cada modelo, detalhadas a seguir:

- *Mean absolute error*: o MAE é um parâmetro básico de precisão que calcula a magnitude média dos erros dos resultados de previsão. Ele fornece as diferenças numéricas entre os valores reais e previstos. Em estatísticas, o erro médio absoluto (MAE) é uma quantidade usada para medir o quão próximas as previsões estão dos

resultados finais (USHA; BALAMURUGAN, 2016). É definido pela seguinte equação 3.1:

$$MAE = \frac{1}{m} \sum_m^{i=1} |x_i - y_i| \quad (3.1)$$

- *Root mean squared error*: O RMSE (Equação 3.2) é a diferença entre a previsão e os valores observados correspondentes são, cada um, ao quadrado e, em seguida, calculada a média da amostra. Por fim, a raiz quadrada da média é obtida (USHA; BALAMURUGAN, 2016).

$$RMSE = \sqrt{\frac{1}{m} \sum_m^{i=1} (x_i - y_i)^2} \quad (3.2)$$

Além destas, outra forma de avaliar o desempenho do modelo é por meio de um detalhamento da acurácia por classe. Para tal, as seguintes métricas estatísticas são utilizadas:

- *TP Rate e FP Rate*: o primeiro refere-se à taxa de verdadeiros positivos, ou seja, os valores corretamente classificados, enquanto que o segundo corresponde à taxa de falsos positivos, quando incorretamente classificados
- *Kappa statistic*: A característica Kappa fornece o nível de concordância entre as verdadeiras classes e as classificações. O valor 1 é o mais alto, mostrando concordância total (ABDULHAMID et al., 2018).
- *Precision*: refere-se à proporção de instâncias que são verdadeiramente de uma classe dividida pelo total de instâncias classificadas como aquela classe.
- *Recall*: é proporção de instâncias classificadas como uma determinada classe dividida pelo total real nessa classe (ABDULHAMID et al., 2018).
- *F-Measure*: é uma combinação matemática destes dois valores definida pela seguinte equação 3.3:

$$F_Measure = \frac{2 * recall * precision}{recall + precision} \quad (3.3)$$

Avaliação e resultados experimentais

Neste capítulo serão apresentados o desempenho do modelo de classificação para cada um dos algoritmos analisados: *Support Vector Machines* (SVM), *K-Nearest Neighbor* (KNN) e Redes Neurais Artificiais do tipo *Multi-layer Perceptron*.

O computador utilizado para a execução do modelo computacional possui um processador Intel Core i7 6700K com 4 núcleos e 8 *threads*, limitado a uma frequência de 4.20 GHz, 16 GB de RAM, e o sistema operacional Linux Zorin OS 15, uma distribuição Linux baseada no Ubuntu 20.04 LTS. Foi instalado o Weka na versão 3.8.

Após avaliar a precisão de todos os quatro modelos, o *Random Forest* apresentou a melhor precisão para o conjunto de dados, conforme Tabela 4.1. É importante notar que o tempo necessário para treinar o conjunto de dados usando este classificador é notavelmente menor do que SVM e RNA, devido à natureza do algoritmo, que exige menor poder computacional.

Tabela 4.1: Resultados dos modelos de classificação.

Algoritmo	Acurácia (<i>dataset</i> filtrado)	Acurácia (<i>dataset</i> original)
Random Forest	97.1809%	98.1548%
SVM	94.0543%	95.6433%
RNA	92.1066%	94.5115%
KNN	91.389%	91.3378%

O segundo algoritmo com a melhor taxa de acerto foi o SVM, com 94.0543% de acurácia com o *dataset* filtrado. O modelo KNN apresentou 91.389% de acerto, tornando-o o menos assertivo dentre os envolvidos no estudo, enquanto que o RNA apresentou uma vantagem relevante quando treinado com o *dataset* completo, sem os filtros de atributos. É importante notar que o algoritmo RNA realizou os cálculos em um tempo consideravelmente maior do que os outros algoritmos, o que torna estes outros modelos mais indicados para aplicações em sistemas embarcados ou com baixo poder computacional à disposição.

Na Figura 4.1 é possível visualizar a taxa de instâncias classificadas corretamente e incorretamente, para cada um dos algoritmos estudados, ao utilizar o *dataset* filtrado, assim como o RMSE e Kappa.

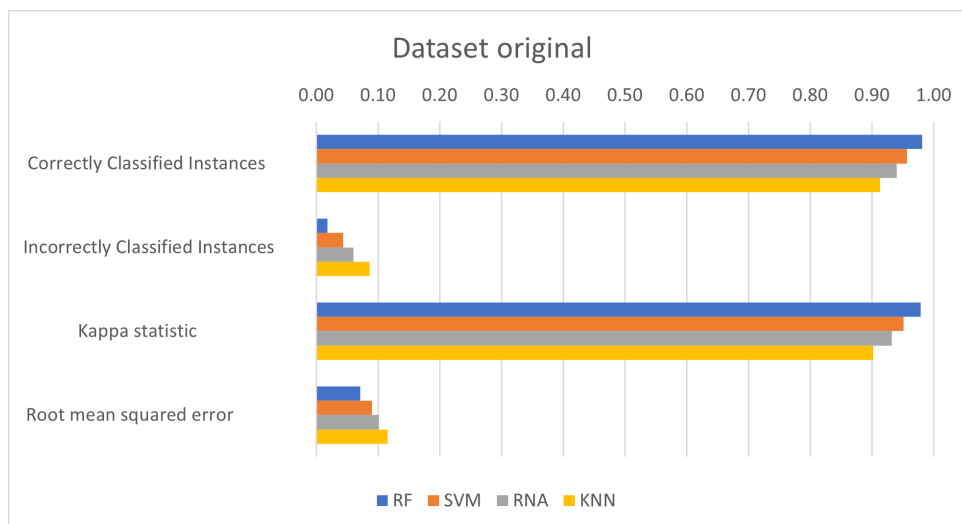


Figura 4.2: Comparação entre os resultados dos algoritmos utilizando o *dataset* original.

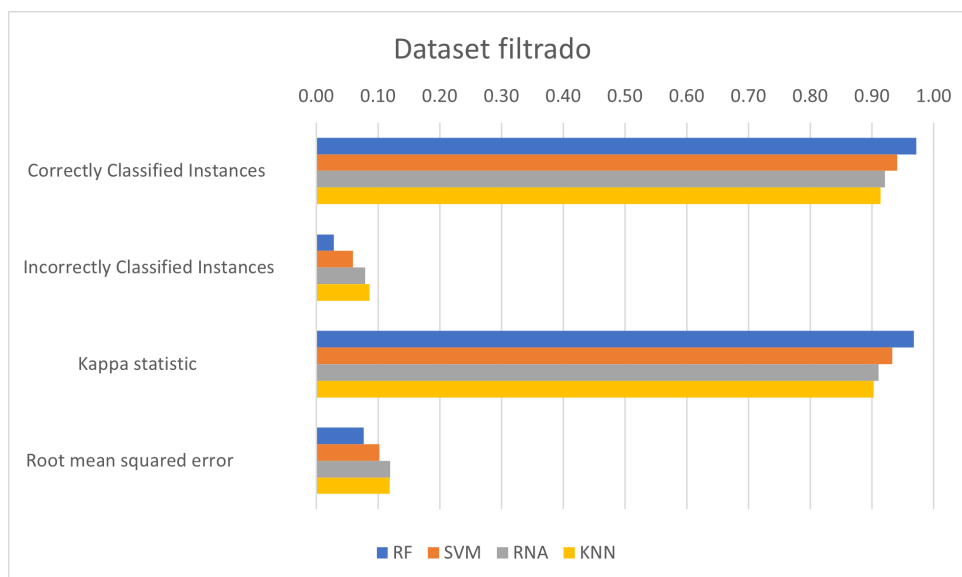


Figura 4.1: Comparação entre os resultados dos algoritmos utilizando o *dataset* filtrado.

É importante, também, realizar a comparação com o *dataset* inteiro, de modo a avaliar se a metodologia utilizada para o filtro do conjunto inteiro apresenta melhorias na classificação, uma vez que a intenção ao remover instâncias menos significativas é minimizar o aprendizado de amostras ruidosas que possam vir a prejudicar o modelo. Outro motivo para realizar a filtragem do *dataset* é a redução no tempo de treinamento e validação dos modelos. A Figura 4.2 mostra as estatísticas dos algoritmos ao utilizar o *dataset* original.

Ao analisar os resultados estatísticos dos modelos de classificação com o *dataset* filtrado, nota-se uma leve melhora nas estatísticas dos modelos. Contudo, o tempo de processamento aumenta significativamente, conforme será discutido em detalhes nas seções seguintes, dedicadas a cada algoritmo. O algoritmo KNN, embora tenha sido o com menos

acurácia dentre os quatro utilizando o *dataset* filtrado, sofreu uma influência mínima ao ser treinado com o *dataset* completo.

Ao comparar com os trabalhos apresentados no Capítulo 2, o modelo proposto por Li, Wang e Wang (2019) entregou valores de acerto acima de 99.74%, conforme pode ser visto na Tabela 4.2. Entretanto vale salientar que o *dataset* utilizado difere do que foi empregado nesta pesquisa. Além disto, a pesquisa de Li, Wang e Wang (2019) considera menos condições de falhas do que o modelo proposto neste trabalho.

Tabela 4.2: Taxa de classificações corretas

Algoritmo	Acertos
BPNN	99.74%
SVM	99.85%
KNNC	99.85%
DBN	99.80%
DNN	99.87%

Em contrapartida, os modelos desenvolvidos por Marins et al. (2018), voltado para a utilização de *Similarity-based Models*, foram capazes de obter mais de 98.91% de acerto, conforme Tabela 4.3 para o *dataset* MaFaulDa, o mesmo do presente trabalho. Por outro lado, para o *dataset* CWRU, os valores foram diferentes, porém muito próximos.

Tabela 4.3: Taxa de classificações corretas para os *datasets* MaFaulda e CWRU

Algoritmo	MaFaulDa	CWRU
Model A	98.95%	98.48%
Model B	98.91%	98.48%
Model C	98.91%	97.48%

Nas seções seguintes, os resultados de cada algoritmo serão detalhados de modo a avaliar com outras métricas o comportamento de cada modelo.

4.1 K-Nearest Neighbor

A Tabela 4.4 apresenta o resultado do teste do algoritmo, após o treinamento. É importante ressaltar o valor do *Kappa statistic*, próximo de 1, e o RMSE (*Root Mean Square Error*) com valor de 0.1186, próximo de 0. Estes indicadores, além da porcentagem de instâncias corretamente classificadas, em 91.389%, mostram que o modelo possui uma alta taxa de assertividade e, portanto, é capaz de realizar a correta classificação de um

dado novo.

Tabela 4.4: Resultados do algoritmo de classificação KNN.

Correctly Classified Instances	1783 - 91.389%
Incorrectly Classified Instances	168 - 8.611%
Total Number of Instances	1951
Kappa statistic	0.9027
Root mean squared error	0.1186

Ao observar a matriz de confusão, apresentada na Tabela 4.5, é possível notar que, para a classe NORMAL, o erro observado foi de 42.86%. Um dos fatores que contribuem para tal resultado é o número menor de dados disponíveis nesta categoria, apenas com 49 amostras, dentre 1951 do *dataset* inteiro, portanto o aprendizado para esta classe torna-se limitado e suscetível a falsos positivos.

Dessa forma, após o teste, foram observadas 8 das 49 amostras classificadas incorretamente como *UNDERHANG_BEARING_CAGE_FAULT*, 8 classificadas como *HORIZONTAL_MISALIGNMENT* e, por fim, 5 classificadas como *VERTICAL_MISALIGNMENT*.

Entretanto, para outras classes, a classificação resultou em erros consideravelmente menores, a exemplo das categorias *UNDERHANG_BEARING_OUTER_RACE*, *UNDERHANG_BEARING_BALL_FAULT* e *OVERHANG_BEARING_BALL_FAULT* com 100% de taxa de acerto. O modelo apresentou um erro na classificação de 19.29% dos sinais da classe *HORIZONTAL_MISALIGNMENT*, sendo este o mais expressivo, além da classe NORMAL citada anteriormente.

Tabela 4.5: Matriz de confusão do algoritmo KNN com o *dataset* filtrado

a	b	c	d	e	f	g	h	i	j	Classified as
28	8	5	0	8	0	0	0	0	0	a = normal
5	159	20	0	11	0	0	2	0	0	b = horizontal_misalignment
3	17	277	2	1	0	0	1	0	0	c = vertical_misalignment
0	2	7	319	2	0	0	3	0	0	d = imbalance
5	6	2	9	157	1	0	3	5	0	e = underhang_bearing_cage_fault
0	0	0	0	0	184	0	0	0	0	f = underhang_bearing_outer_race
0	0	0	0	0	0	186	0	0	0	g = underhang_bearing_ball_fault
1	4	0	3	3	0	0	164	13	0	h = overhang_bearing_cage_fault
2	0	0	1	2	0	0	11	172	0	i = overhang_bearing_outer_race
0	0	0	0	0	0	0	0	0	137	j = overhang_bearing_ball_fault

Por outro lado, ao considerar o *dataset* original, o modelo apresentou um resultado de

91.3378% de acurácia, um valor bem próximo do valor obtido pelo *dataset* filtrado, visualizado na Tabela 4.6. Dessa forma, a metodologia de filtro de atributos mostrou-se efetiva, uma vez que, além de reduzir o tempo necessário para computação do modelo de classificação, a acurácia obtida com o filtro apresentou uma leve melhora.

Tabela 4.6: Resultados do algoritmo de classificação KNN.

Correctly Classified Instances	1782 - 91.3378%
Incorrectly Classified Instances	169 - 8.6622%
Total Number of Instances	1951
Kappa statistic	0.9021
Root mean squared error	0.1157

Ao visualizar a matriz de confusão apresentada na Tabela 4.7, é possível observar que, para a classe *NORMAL*, ocorreram mais instâncias classificadas incorretamente, o que representou um erro de 57.14%. Para a classe *HORIZONTAL_MISALIGNMENT*, o erro aumentou para 21.32%, contra 19.29% de classificações incorretas com o *dataset* filtrado. Com estes resultados, a utilização dos algoritmos de filtro de *dataset* disponíveis no software Weka mostraram-se relevantes para este modelo.

Tabela 4.7: Matriz de confusão do algoritmo KNN com o *dataset* completo.

a	b	c	d	e	f	g	h	i	j	Classified as
21	11	6	0	11	0	0	0	0	0	a = normal
9	155	20	2	5	0	0	4	2	0	b = horizontal_misalignment
0	8	288	3	0	0	1	1	0	0	c = vertical_misalignment
0	2	14	313	2	0	0	1	1	0	d = imbalance
3	10	1	7	160	1	0	3	3	0	e = underhang_bearing_cage_fault
0	0	0	0	0	183	0	1	0	0	f = underhang_bearing_outer_race
0	0	0	0	0	0	186	0	0	0	g = underhang_bearing_ball_fault
0	7	0	0	2	1	0	169	9	0	h = overhang_bearing_cage_fault
1	2	0	1	1	0	0	13	170	0	i = overhang_bearing_outer_race
0	0	0	0	0	0	0	0	0	137	j = overhang_bearing_ball_fault

4.2 Support Vector Machines

Este algoritmo apresentou o segundo melhor resultado dentre os alvos do estudo. Na Tabela 4.8 é possível observar, além da porcentagem de instâncias corretamente classificadas, de 94.0543%, o valor do *Kappa statistic*, em 0.9329, um valor 3.34% superior ao obtido com o algoritmo KNN, com o *dataset* filtrado.

Tabela 4.8: Resultados do algoritmo de classificação SVM com o *dataset* filtrado.

Correctly Classified Instances	1835 - 94.0543%
Incorrectly Classified Instances	116 - 5.9457%
Total Number of Instances	1951
Kappa statistic	0.9329
Root mean squared error	0.1022

Na matriz de confusão (Tabela 4.9), as entradas, em geral, foram classificadas com erro mínimo, exceto para a classe NORMAL, que apresentou 73,47% de classificações corretas. A classe *HORIZONTAL_MISALIGNMENT* resultou em 85,28% de precisão, o que indica que o modelo apresenta uma certa dificuldade em distinguir sinais desta classe dos dados pertencentes à classe *VERTICAL_MISALIGNMENT*. Este valor de precisão está abaixo das outras classes, o que contribui negativamente com a acurácia do modelo como um todo, de 91.24%, assim como a classificação dos sinais da classe NORMAL.

Ao analisar a acurácia das demais classes, é possível observar uma taxa de acerto entre 92.69% para a classe *VERTICAL_MISALIGNMENT*, até 100% para as classes *OVERHANG_BEARING_OUTER_RACE* e *UNDERHANG_BEARING_OUTER_RACE*.

Tabela 4.9: Matriz de confusão do algoritmo SVM com o *dataset* filtrado.

a	b	c	d	e	f	g	h	i	j	Classified as
36	2	2	0	6	0	0	3	0	0	a = normal
3	168	16	0	0	0	0	10	0	0	b = horizontal_misalignment
2	13	279	1	4	0	0	2	0	0	c = vertical_misalignment
0	0	0	324	3	0	0	4	2	0	d = imbalance
4	3	1	0	177	1	0	1	1	0	e = underhang_bearing_cage_fault
0	0	0	0	0	184	0	0	0	0	f = underhang_bearing_outer_race
0	0	0	0	0	0	185	0	1	0	g = underhang_bearing_ball_fault
2	4	0	4	3	0	1	165	9	0	h = overhang_bearing_cage_fault
1	0	0	0	1	0	1	5	180	0	i = overhang_bearing_outer_race
0	0	0	0	0	0	0	0	0	137	j = overhang_bearing_ball_fault

Diferente do algoritmo KNN, o SVM obteve uma melhora de 1.68% na acurácia ao utilizar todos os atributos do *dataset* original, conforme observado na Tabela 4.10. De maneira análoga, o *Kappa statistic* obteve um acréscimo de 1.92%, enquanto que o RMSE foi reduzido de 0.1022 para 0.0901, o que resultou em um decréscimo de 11.84%.

Tabela 4.10: Resultados do algoritmo de classificação SVM com o dataset original.

Correctly Classified Instances	1866 - 95.6433%
Incorrectly Classified Instances	85 - 4.3567%
Total Number of Instances	1951
Kappa statistic	0.9509
Root mean squared error	0.0901

Na matriz de confusão, na Tabela 4.11, é possível notar que o modelo foi capaz de classificar corretamente 93.9% dos dados da classe *NORMAL*, enquanto que ao considerar o *dataset* filtrado, a assertividade para esta classe foi de 73,47%. A classe que apresentou o menor índice de verdadeiros positivos foi *IMBALANCE*, com 93.09% de acerto. Diante do exposto, o algoritmo SVM é beneficiado ao utilizar o *dataset* original, com todos os atributos, embora esta escolha acarrete em um tempo maior de processamento do modelo.

Tabela 4.11: Matriz de confusão do algoritmo SVM com o *dataset* filtrado.

a	b	c	d	e	f	g	h	i	j	Classified as
46	1	1	0	1	0	0	0	0	0	a = normal
5	187	1	0	2	0	0	2	0	0	b = horizontal_misalignment
0	10	285	0	6	0	0	0	0	0	c = vertical_misalignment
1	6	7	310	8	0	0	1	0	0	d = imbalance
4	4	0	1	176	1	0	1	1	0	e = underhang_bearing_cage_fault
2	0	0	0	0	182	0	0	0	0	f = underhang_bearing_outer_race
1	0	0	0	0	0	185	0	0	0	g = underhang_bearing_ball_fault
4	1	0	0	0	0	0	178	5	0	h = overhang_bearing_cage_fault
1	1	0	0	2	0	0	4	180	0	i = overhang_bearing_outer_race
0	0	0	0	0	0	0	0	0	137	j = overhang_bearing_ball_fault

4.3 Redes Neurais Artificiais

Os resultados do algoritmo RNA, vistos na Tabela 4.12 mostram que este modelo apresentou uma acurácia superior ao KNN, porém inferior ao SVM. Como consequência o *Kappa statistic* deste modelo mostrou-se maior do que o apresentado pelo algoritmo KNN em 0.9% e, em comparação com o SVM, 2.43% menor. O RMSE foi apenas 0.67% maior do que o KNN e 16.83% maior do que o SVM.

Tabela 4.12: Resultados do algoritmo de classificação RNA com o *dataset* filtrado.

Correctly Classified Instances	1797 - 92.1066%
Incorrectly Classified Instances	154 - 7.8934%
Total Number of Instances	1951
Kappa statistic	0.9107
Root mean squared error	0.1194

Ao visualizar a matriz de confusão na Tabela 4.13, nota-se uma baixa taxa de acerto para a classe *NORMAL*, em 42.85%, e 70.05% para a classe *HORIZONTAL_MISALIGNMENT*. Apesar destas taxas de erro, as demais classes apresentaram mais de 91% de acerto, pelo menos, a exemplo da *OVERHANG_BEARING_CAGE_FAULT* com 91.49% de acerto, até 99.46% de assertividade para a classe *UNDERHANG_BEARING_OUTER_RACE*.

Tabela 4.13: Matriz de confusão do algoritmo RNA com o *dataset* filtrado.

a	b	c	d	e	f	g	h	i	j	Classified as
21	7	12	0	8	0	0	1	0	0	a = normal
5	138	44	3	1	0	0	3	2	1	b = horizontal_misalignment
2	4	286	1	5	0	0	3	0	0	c = vertical_misalignment
0	2	3	320	1	0	0	4	2	1	d = imbalance
2	2	0	3	175	1	2	1	2	0	e = underhang_bearing_cage_fault
0	0	0	1	0	183	0	0	0	0	f = underhang_bearing_outer_race
0	0	0	0	0	0	184	1	1	0	g = underhang_bearing_ball_fault
0	5	0	1	2	0	0	172	8	0	h = overhang_bearing_cage_fault
0	0	0	0	1	0	0	5	182	0	i = overhang_bearing_outer_race
0	0	0	0	0	0	0	1	0	136	j = overhang_bearing_ball_fault

Ao realizar o treinamento com o *dataset* original, nota-se na Tabela 4.14 uma melhora de 1.89% na acurácia, em comparação com o *dataset* filtrado. Devido ao número superior de atributos a serem analisados, a camada de entrada precisou ser ajustada para 80 neurônios, ou seja, um para cada atributo. Entretanto, a camada oculta manteve-se inalterada para permitir uma comparação com o modelo anterior.

O RMSE foi reduzido em 15.24%, para 0.1012, enquanto que o *Kappa statistic* foi elevado de 0.9107 para 0.9322, representando um acréscimo de 2.36%. Estas métricas mostram que este algoritmo é levemente comprometido pela filtragem de atributos, uma vez que a quantidade reduzida de dados diminui a qualidade do aprendizado.

Tabela 4.14: Resultados do algoritmo de classificação RNA com o *dataset* original.

Correctly Classified Instances	1834 - 94.0031%
Incorrectly Classified Instances	117 - 5.9969%
Total Number of Instances	1951
Kappa statistic	0.9322
Root mean squared error	0.1012

A matriz de confusão apresentada na Tabela 4.15 mostra que o algoritmo RNA foi capaz de classificar corretamente 55.10% dos dados da classe NORMAL, o que representa uma melhora de 28.5% em relação ao resultado anterior, com o *dataset* filtrado. Da mesma forma, os valores para a classe *HORIZONTAL_MISALIGNMENT* apresentaram acurácia de 88.83%, em contraste com os 70.05% obtidos com o *dataset* filtrado, o que representa uma melhora de 26.81%.

Tabela 4.15: Matriz de confusão do algoritmo RNA com o *dataset* original.

a	b	c	d	e	f	g	h	i	j	Classified as
27	7	3	6	5	0	0	0	1	0	a = normal
4	175	11	0	2	1	0	2	1	1	b = horizontal _m isalignment
0	4	286	1	4	0	3	2	0	1	c = vertical _m isalignment
0	3	1	323	1	0	2	1	1	1	d = imbalance
1	3	5	6	167	1	0	4	1	0	e = underhang _b earing _c age _f fault
0	0	0	0	0	184	0	0	0	0	f = underhang _b earing _o uter _r ace
0	0	0	0	0	0	186	0	0	0	g = underhang _b earing _b all _f fault
0	1	7	2	3	2	0	172	1	0	h = overhang _b earing _c age _f fault
1	1	1	0	2	0	0	6	177	0	i = overhang _b earing _o uter _r ace
0	0	0	0	0	0	0	0	0	137	j = overhang _b earing _b all _f fault

4.4 Random Forest

Este algoritmo obteve o melhor resultado dentre os avaliados. A Tabela 4.16 mostra que o modelo foi capaz de classificar corretamente 97.18% das entradas durante o teste, também atingindo o menor RMSE e o *Kappa statistic* mais próximo de 1, o que indica uma maior assertividade.

Tabela 4.16: Resultados do algoritmo de classificação Random Forest com *dataset* filtrado.

Correctly Classified Instances	1896 - 97.1809%
Incorrectly Classified Instances	55 - 2.8191%
Total Number of Instances	1951
Kappa statistic	0.9682
Root mean squared error	0.077

A matriz de confusão apresentada na Tabela 4.17 mostra que, no geral, o *Random Forest* apresentou uma alta taxa de acerto, obtendo 100% de assertividade para classes como *UNDERHANG_BEARING_OUTER_RACE* e *OVERHANG_BEARING_BALL_FAULT*. Entretanto, para a classe *HORIZONTAL_MISALIGNMENT*, o algoritmo apresentou 90.86% de acerto, o valor mais baixo dentre as demais classes. É interessante notar, também, que com o algoritmo SVM, com o *dataset* original, a taxa de acerto para tal classe foi de 94.92%.

Tabela 4.17: Matriz de confusão do algoritmo *Random Forest* com *dataset* filtrado.

a	b	c	d	e	f	g	h	i	j	Classified as
45	2	0	0	1	0	0	1	0	0	a = normal
0	179	16	0	0	0	0	2	0	0	b = horizontal_misalignment
1	4	289	2	1	0	0	4	0	0	c = vertical_misalignment
0	2	2	325	1	0	0	1	2	0	d = imbalance
0	0	1	0	185	1	0	0	1	0	e = underhang_bearing_cage_fault
0	0	0	0	0	184	0	0	0	0	f = underhang_bearing_outer_race
0	0	0	0	0	0	185	0	1	0	g = underhang_bearing_ball_fault
0	1	0	0	0	1	0	185	1	0	h = overhang_bearing_cage_fault
0	0	0	0	1	0	0	5	182	0	i = overhang_bearing_outer_race
0	0	0	0	0	0	0	0	0	137	j = overhang_bearing_ball_fault

Os resultados deste algoritmo ao utilizar o *dataset* original podem ser observados na Tabela 4.18, que demonstrou a maior taxa de assertividade dentre todas as demais avaliações, com 98.15%. Conseqüentemente, o *Kappa statistic* obtido foi o maior dentre os demais, assim como este modelo apresentou o menor RMSE.

Tabela 4.18: Resultados do algoritmo de classificação Random Forest com *dataset* original.

Correctly Classified Instances	1915 - 98.1548%
Incorrectly Classified Instances	36 - 1.8452%
Total Number of Instances	1951
Kappa statistic	0.9792
Root mean squared error	0.0717

Por fim, a Tabela 4.19 traz a matriz de confusão do algoritmo *Random Forest*, porém considerando o *dataset* original, sem o processo de filtragem. Nesta tabela é possível notar que o algoritmo apresenta desvios mínimos, em comparação com os demais, em especial para as classes *NORMAL* e *HORIZONTAL_MISALIGNMENT*, nas quais os demais modelos apresentaram menores taxas de acerto na classificação.

Tabela 4.19: Matriz de confusão do algoritmo *Random Forest* com *dataset* original.

a	b	c	d	e	f	g	h	i	j	Classified as
46	2	0	0	0	0	0	1	0	0	a = normal
0	187	9	0	0	0	0	1	0	0	b = horizontal_misalignment
1	3	295	1	1	0	0	0	0	0	c = vertical_misalignment
0	1	0	327	3	0	0	1	1	0	d = imbalance
0	0	1	0	183	1	1	1	1	0	e = underhang_bearing_cage_fault
0	0	0	0	0	183	0	1	0	0	f = underhang_bearing_outer_race
0	0	0	0	0	0	186	0	0	0	g = underhang_bearing_ball_fault
0	0	0	0	0	1	0	186	1	0	h = overhang_bearing_cage_fault
0	0	0	0	1	0	0	2	185	0	i = overhang_bearing_outer_race
0	0	0	0	0	0	0	0	0	137	j = overhang_bearing_ball_fault

4.5 Comparação entre os algoritmos

Conforme apresentado nas seções anteriores, é notável que o algoritmo *Random Forest* é capaz de obter uma precisão muito boa de maneira contínua ao longo das classes, enquanto os outros algoritmos apresentam erros mais consideráveis para algumas das classes, o que pode ser visualizado pela Figura 4.3 que mostra os valores de precisão entre os algoritmos ao considerar o *dataset* filtrado, e Figura 4.4 para o *dataset* original. OS algoritmos KNN, SVM e RNA apresentaram maior dificuldade em realizar a classificação correta dos dados das classes *NORMAL*, *HORIZONTAL_MISALIGNMENT* e *IMBALANCE*, sendo a primeira classe a que apresentou uma taxa maior de erros.

O algoritmo RNA, por ter utilizado somente uma camada oculta, não obteve resultados acima de 94%. Este modelo pode ter uma precisão mais elevada com uma rede com maior número de camadas ocultas, entretanto o ganho na precisão acompanha uma elevação no custo computacional à medida que camadas adicionais são implementadas.

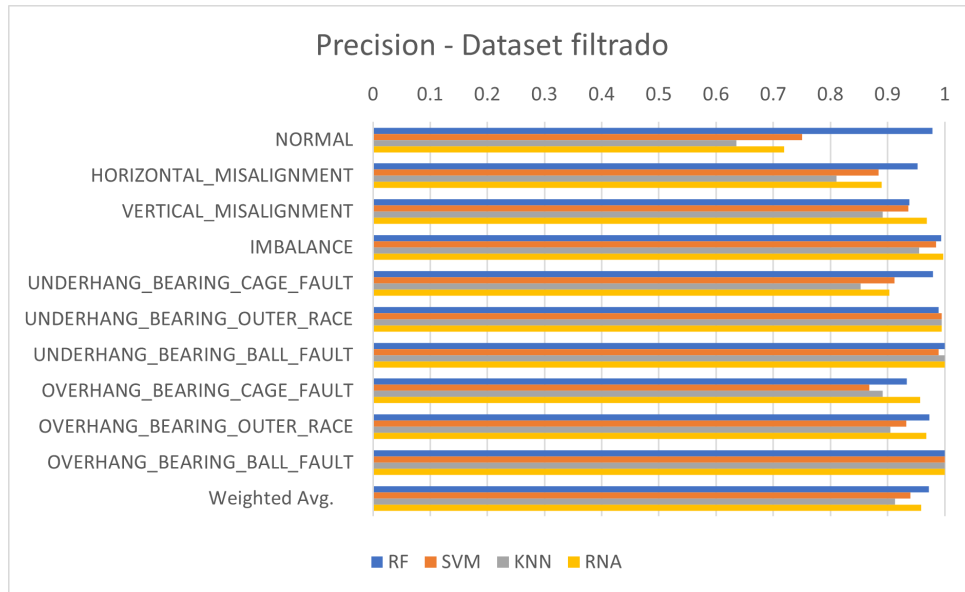


Figura 4.3: Comparação entre a precisão dos algoritmos utilizando o *dataset* filtrado.

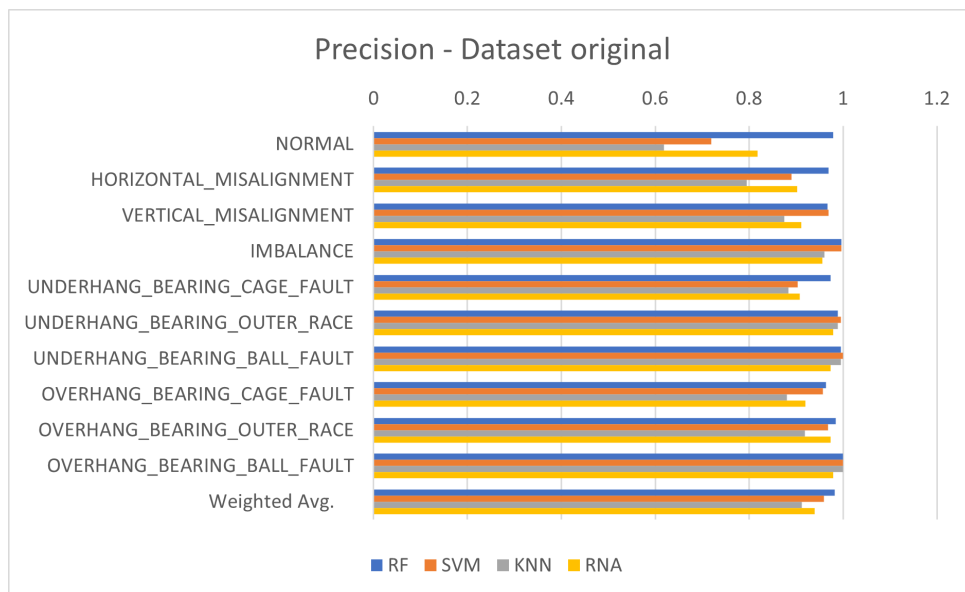


Figura 4.4: Comparação entre a precisão dos algoritmos utilizando o *dataset* original.

As taxas de verdadeiros positivos podem ser observadas nas Figuras 4.5 e 4.6 que contabilizam as instâncias corretamente classificadas. Novamente é possível notar que os algoritmos SVM e KNN apresentaram dificuldades ao corretamente classificar os dados correspondentes às duas primeiras classes.

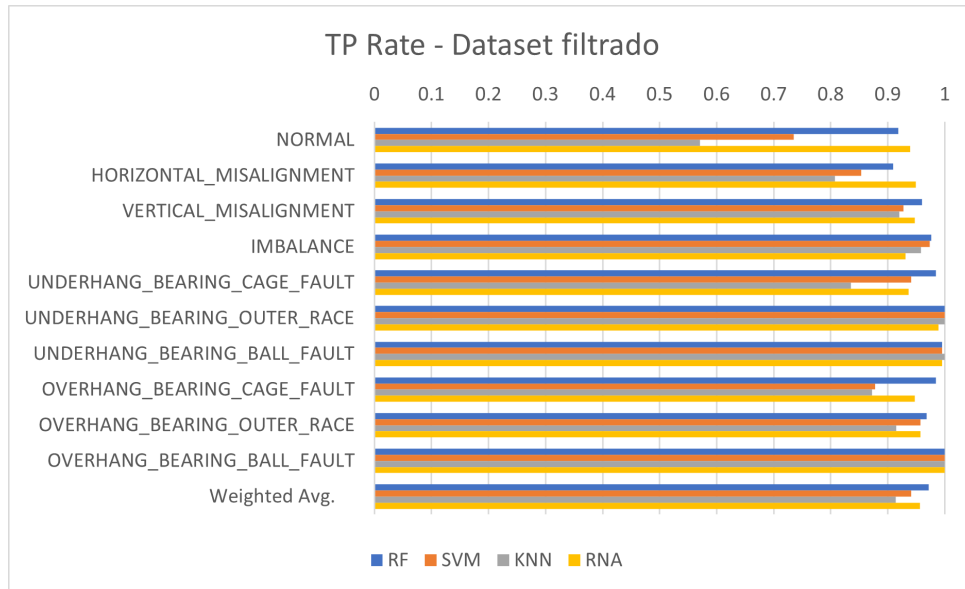


Figura 4.5: Comparação entre a taxa de verdadeiros positivos dos algoritmos utilizando o *dataset* filtrado.

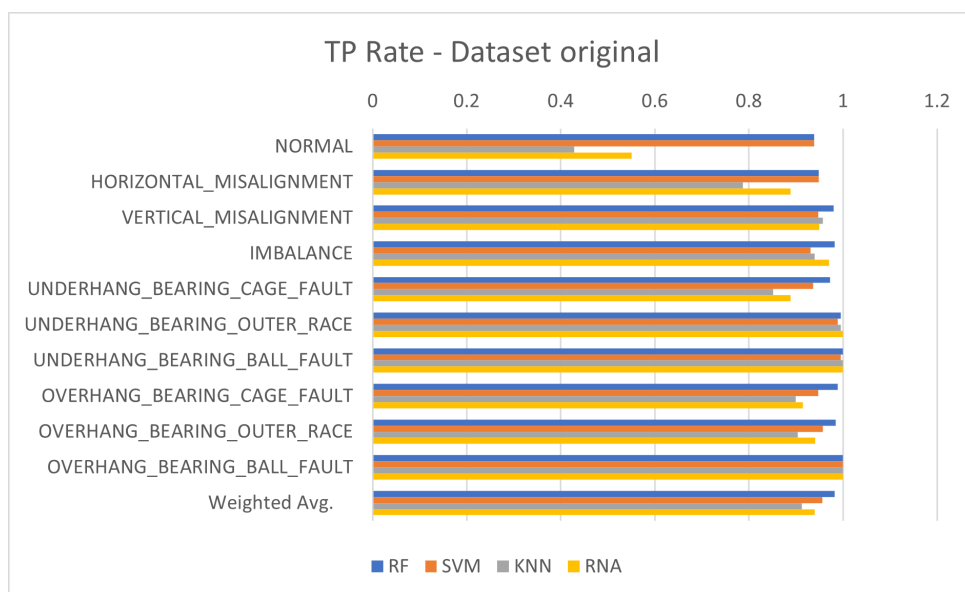


Figura 4.6: Comparação entre a taxa de verdadeiros positivos dos algoritmos utilizando o *dataset* original.

Além dos valores de verdadeiros positivos, é interessante, também, observar a taxa da falsos positivos, uma vez que esta métrica avalia o número de entradas que foram incorretamente classificadas como pertencentes a uma determinada classe, entretanto não correspondem à realidade. A Figura 4.7 mostra a taxa de falsos positivos para cada uma das classes de saída, para os quatro algoritmos analisados. É possível notar que o KNN apresentou uma taxa de FP elevada para classes como *HORIZONTAL_MISALIGNMENT*, *NORMAL* e *UNDERHANG_BEARING_CAGE_FAULT*.

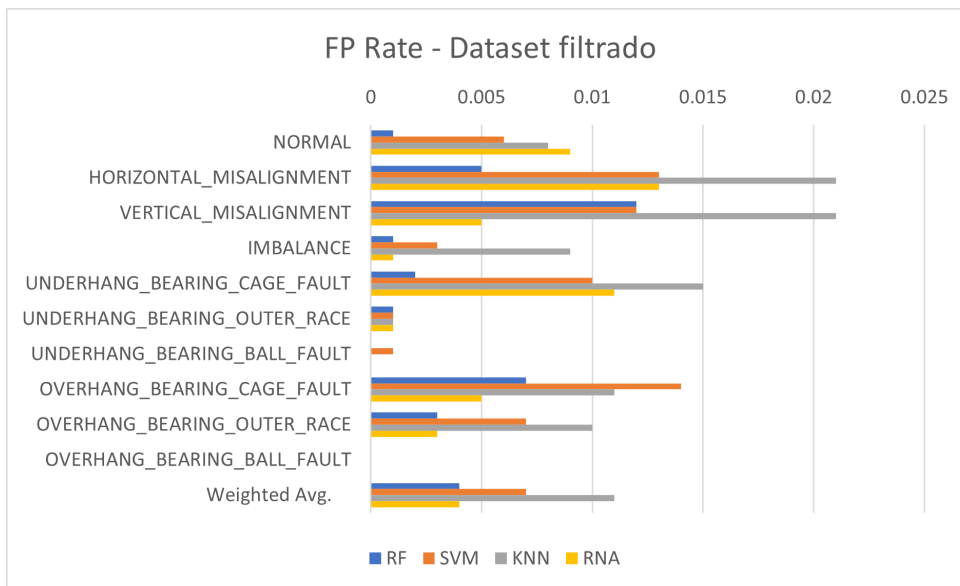


Figura 4.7: Comparação entre a taxa de falsos positivos dos algoritmos utilizando o *dataset* filtrado.

Por outro lado, na Figura 4.8, ao considerar o *dataset* original, é possível notar que o algoritmo KNN obteve uma taxa mais alta de falsos positivos para as classes *HORIZONTAL_MISALIGNMENT* e *VERTICAL_MISALIGNMENT*. Em contrapartida, o SVM foi capaz de classificar menos instâncias incorretamente como pertencentes a uma determinada classe.

O RNA, no geral, para esta métrica, apresentou resultados inferiores. É possível notar que em algumas classes houveram menos classificações incorretas, a exemplo da *NORMAL* e *HORIZONTAL_MISALIGNMENT*, enquanto que em outras apresentaram um aumento na taxa de FP, como pode ser observado nas classes *VERTICAL_MISALIGNMENT*, *IMBALANCE*. A classe *OVERHANG_BEARING_OUTER_RACE* não apresentou uma variação significativa.

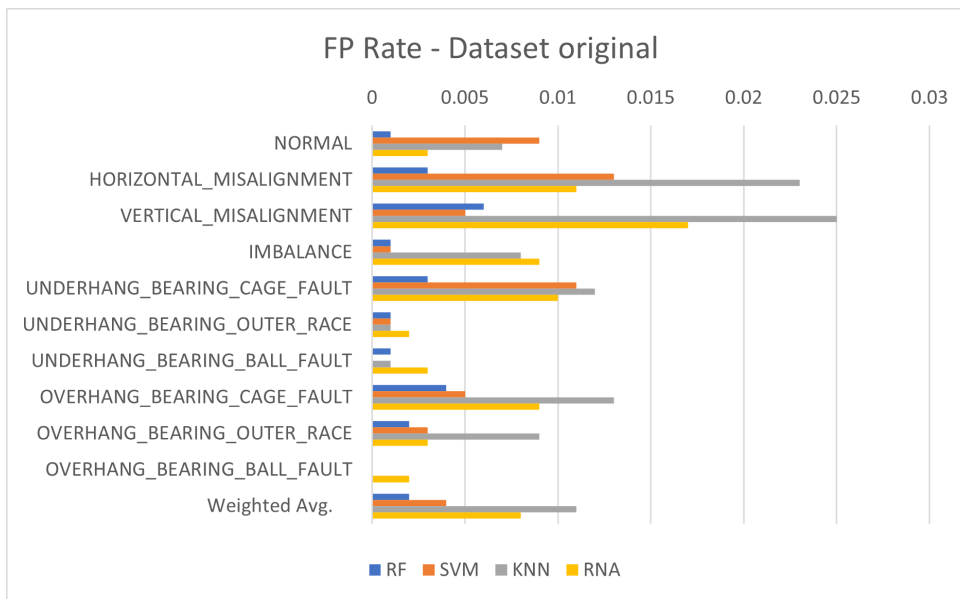


Figura 4.8: Comparação entre a taxa de falsos positivos dos algoritmos utilizando o *dataset* original.

Considerações finais

A manutenção preditiva é imprescindível, principalmente no ambiente industrial, onde a falha de determinados equipamentos que resulte na necessidade de manutenção corretiva pode ocasionar a paralisação parcial ou total da planta produtiva, o que acarreta transtornos e prejuízos para a empresa.

Em motores elétricos, para mitigar a possibilidade de falha, a técnica de análise de vibração é utilizada. A maioria dos problemas do motor reflete diretamente em sua vibração devido a vários fatores: desalinhamento, desequilíbrio ou falhas relacionadas aos rolamentos. Cada componente de um motor possui uma assinatura em uma frequência diferente, assim, a partir da análise do comportamento do perfil de vibração do equipamento, é possível detectar o tipo de falha presente na máquina.

Os estudos atuais mostram algoritmos de classificação especializados em uma falha específica, o que não permite sua implementação em um ambiente industrial para obter indícios de várias falhas, uma vez que a aplicação de um motor elétrico pode induzir defeitos em diferentes componentes deste equipamento.

Este estudo apresenta um modelo computacional baseado em algoritmos de classificação para determinar a presença e o tipo de falhas em motores elétricos através de análise de vibração. Foram estudados os algoritmos RNA (Redes Neurais Artificiais), KNN (K-Nearest Neighbor), SVM (Support Vector Machines) e RF (Random Forest). O modelo computacional consiste em uma camada de pré processamento, onde os sinais provenientes do dataset MaFaulDa são convertidos numericamente para o espectro da frequência, por meio do algoritmo FFT, e outra camada onde os algoritmos de classificação são treinados e avaliados.

5.1 Conclusões

O modelo proposto foi capaz de identificar corretamente diferentes tipos de falhas mecânicas em motores elétricos por meio de leituras de análise de vibração. Para obter este resultado, uma etapa inicial de processamento foi necessária para transformar os sinais para o espectro de frequência, extrair características estatísticas relevantes e filtrar a quantidade de dados fornecidos para treinar os classificadores.

O modelo foi treinado para detectar falhas em múltiplas velocidades de rotação do motor, para fornecer uma melhor precisão em um cenário do mundo real, uma vez que em muitas aplicações, a velocidade dos equipamentos é controlada por inversores de frequência.

Os algoritmos estudados neste trabalho apresentaram boa acurácia para a detecção de falhas a partir dos sinais de vibração. Contudo, o algoritmo *Random Forest* (RF) demonstrou ser o mais assertivo para tal finalidade, conforme detalhado no Capítulo 4, com 97.18% de acurácia.

Foi avaliada, também, a possibilidade de filtrar os dados fornecidos ao algoritmo de treinamento e validação de modo a otimizar o processo, para reduzir os custos computacionais e o tempo necessário para a resolução do problema. Ao comparar os resultados dos quatro algoritmos, com e sem o *dataset* filtrado, nota-se que, embora os algoritmos tenham sofrido uma perda na precisão de maneira geral, esta perda não é significativa para a classificação das falhas.

Para o algoritmo *Random Forest*, ao utilizar o dataset filtrado, foi gerada uma diminuição da acurácia de 0.97%, sendo o RNA mais afetado, com a redução de 2.41%. Entretanto, para o RNA, o dataset filtrado oferece uma redução significativa no tempo de processamento dos dados, tanto no treinamento, como na validação.

5.2 Contribuições

As principais contribuições desta pesquisa são:

- Desenvolvimento de algoritmos de pré processamento e classificação para sinais de vibração mecânica para a detecção de falhas em motores elétricos de indução.
- Avaliação de desempenho e acurácia de diferentes modelos de classificadores de modo a determinar a metodologia mais eficiente para a solução do problema.

A aplicação do modelo computacional apresentado neste trabalho de pesquisa em um ambiente industrial permite a fácil e rápida detecção de falhas em motores de modo a permitir uma intervenção mais assertiva e eficiente no equipamento. Esta metodologia evita que o motor entre em falha e cause a parada da produção, ao mesmo tempo que permite que os responsáveis pela manutenção das máquinas acompanhem a condição dos mesmos para programar o período ideal para intervenção.

5.3 Atividades Futuras de Pesquisa

A utilização de *Deep Learning* está definida como atividade futura desta pesquisa. Estes algoritmos são ideais para reconhecimento de padrões e podem ser aplicados para o problema descrito neste trabalho. Deve-se avaliar a acurácia desta proposta, assim como o custo computacional para o treinamento e validação do modelo.

Embora os algoritmos apresentados neste e em outros trabalhos correlatos demonstrem boa acurácia e precisão na classificação de falhas em equipamentos por análise de vibração, o uso do *Deep Learning* permite tornar o algoritmo ainda mais preciso e menos suscetível a erros.

Referências Bibliográficas

- ABDELKRIM, C.; MERIDJET, M. S.; BOUTASSETA, N.; BOULANOUAR, L. Detection and classification of bearing faults in industrial geared motors using temporal features and adaptive neuro-fuzzy inference system. *Heliyon*, Elsevier, v. 5, n. 8, p. e02046, 2019.
- ABDULBARY, M. B.; EMBABY, A. G.; GOMAA, F. R. Fault diagnosis in rotating system based on vibration analysis. *ERJ. Engineering Research Journal*, Menoufia University, Faculty of Engineering, v. 44, n. 3, p. 285–294, 2021.
- ABDULHAMID, S. M.; SHUAIB, M.; OSHO, O.; ISMAILA, I.; ALHASSAN, J. K. Comparative analysis of classification algorithms for email spam detection. *International Journal of Computer Network & Information Security*, v. 10, n. 1, 2018.
- ALI, M. A.; BINGAMIL, A. A.; JARNDAL, A.; ALSYOUF, I. The influence of handling imbalance classes on the classification of mechanical faults using neural networks. In: IEEE. *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*. [S.l.], 2019. p. 1–5.
- ALSHORMAN, O. et al. Sounds and acoustic emission-based early fault diagnosis of induction motor: A review study. *Advances in Mechanical Engineering*, SAGE Publications Sage UK: London, England, v. 13, n. 2, p. 1687814021996915, 2021.
- ÁLVARES, R. C. Diagnóstico de falhas em pára-raios utilizando termografia. Universidade Federal de Minas Gerais, 2008.
- ARAÚJO, R. G. C. *Desenvolvimento de um sistema de diagnóstico de falhas em motores elétricos de indução*. Tese (Doutorado), 2015.
- BERRAR, D. *Cross-Validation*. 2019.
- BESSOUS, N.; ZOZOU, S.; SBAA, S.; BENTRAH, W. A comparative study between the mcsa, dwt and the vibration analysis methods to diagnose the dynamic eccentricity fault in induction motors. In: IEEE. *2017 6th International Conference on Systems and Control (ICSC)*. [S.l.], 2017. p. 414–421.
- BILBAO, I.; BILBAO, J. Overfitting problem and the over-training in the era of data: Particularly for artificial neural networks. In: IEEE. *2017 eighth international conference on intelligent computing and information systems (ICICIS)*. [S.l.], 2017. p. 173–177.
- BONALDI, E. et al. A aplicação da análise da assinatura elétrica no setor de óleo e gás-procedimento de coleta e análise resultados e metodologia para redução do custo

com paradas não-programadas em refinarias brasileiras. *CD Do IV PCIC BR, Rio de Janeiro, RJ*, 2012.

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.

CERVANTES, J.; GARCIA-LAMONT, F.; RODRÍGUEZ-MAZAHUA, L.; LOPEZ, A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, Elsevier, v. 408, p. 189–215, 2020.

CHAPELLE, O. e. a. Choosing multiple parameters for support vector machines. *Machine learning*, v. 46, p. 131 – 159, 2002.

CUTLER, A.; CUTLER, D. R.; STEVENS, J. R. Random forests. In: *Ensemble machine learning*. [S.l.]: Springer, 2012. p. 157–175.

DENISKO, D.; HOFFMAN, M. M. Classification and interaction in random forests. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 115, n. 8, p. 1690–1692, 2018.

DREISEITL, S.; OHNO-MACHADO, L. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, Elsevier, v. 35, n. 5-6, p. 352–359, 2002.

EIBE, F.; HALL, M. A.; WITTEN, I. H. The weka workbench. online appendix for data mining: practical machine learning tools and techniques. In: *Morgan Kaufmann*. [S.l.: s.n.], 2016.

FLECK, L.; TAVARES, M. H. F.; EYNG, E.; HELMANN, A.; ANDRADE, M. Redes neurais artificiais: princípios básicos. *Revista Eletrônica Científica Inovação e Tecnologia*, v. 1, n. 13, p. 47–57, 2016.

FUKUNAGA, K.; NARENDRA, P. M. A branch and bound algorithm for computing k-nearest neighbors. *IEEE transactions on computers*, IEEE, v. 100, n. 7, p. 750–753, 1975.

GANGSAR, P.; TIWARI, R. Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review. *Mechanical systems and signal processing*, Elsevier, v. 144, p. 106908, 2020.

GERMEN, E.; BAŞARAN, M.; FIDAN, M. Sound based induction motor fault diagnosis using kohonen self-organizing map. *Mechanical Systems and Signal Processing*, Elsevier, v. 46, n. 1, p. 45–58, 2014.

HALL, M. A. Correlation-based feature subset selection for machine learning. *Thesis submitted in partial fulfillment of the requirements of the degree of Doctor of Philosophy at the University of Waikato*, 1998.

HAYKIN, S. *Sinais e Sistemas*. [S.l.]: Bookman, 2001.

- HENRIQUE, B. M.; SOBREIRO, V. A.; KIMURA, H. Stock price prediction using support vector regression on daily and up to the minute prices. *The Journal of finance and data science*, Elsevier, v. 4, n. 3, p. 183–201, 2018.
- HSU, C.-W.; LIN, C.-J. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, IEEE, v. 13, n. 2, p. 415–425, 2002.
- IMANDOUST, S. B.; BOLANDRAFTAR, M. Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International Journal of Engineering Research and Applications*, v. 3, n. 5, p. 605–610, 2013.
- JAKKULA, V. Tutorial on support vector machine (svm). *Washington State University*, v. 36, 2006.
- JANSSENS, O. et al. Convolutional neural network based fault detection for rotating machinery. *Journal of Sound and Vibration*, Elsevier, v. 377, p. 331–345, 2016.
- JUNIOR, R. F. R.; ALMEIDA, F. A. de; GOMES, G. F. Fault classification in three-phase motors based on vibration signal analysis and artificial neural networks. *Neural Computing and Applications*, Springer, v. 32, n. 18, p. 15171–15189, 2020.
- KUO, B.-C.; HO, H.-H.; LI, C.-H.; HUNG, C.-C.; TAUR, J.-S. A kernel-based feature selection method for svm with rbf kernel for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, IEEE, v. 7, n. 1, p. 317–326, 2013.
- LI, Z.; WANG, Y.; WANG, K. A deep learning driven method for fault classification and degradation assessment in mechanical equipment. *Computers in industry*, Elsevier, v. 104, p. 1–10, 2019.
- LOBAO, D. A.; PINTO, L. A. Diagnóstico inteligente de falhas em equipamentos dinâmicos através da transformada wavelet e dados estatísticos. In: *Congresso Brasileiro de Automática-CBA*. [S.l.: s.n.], 2020. v. 2, n. 1.
- MARINS, M. A.; RIBEIRO, F. M.; NETTO, S. L.; SILVA, E. A. da. Improved similarity-based modeling for the classification of rotating-machine failures. *Journal of the Franklin Institute*, Elsevier, v. 355, n. 4, p. 1913–1930, 2018.
- MBO’O, C. P.; HAMEYER, K. Fault diagnosis of bearing damage by means of the linear discriminant analysis of stator current features from the frequency selection. *IEEE Transactions on Industry Applications*, IEEE, v. 52, n. 5, p. 3861–3868, 2016.
- MEHALA, N.; DAHIYA, R. Condition monitoring methods, failure identification and analysis for induction machines. *International journal of circuits, systems and signal processing*, Citeseer, v. 3, n. 1, p. 10–17, 2009.
- MORAES¹, R. J. R. de; GOMES¹, G.; FRANCHINI, A.; FERNANDES, E.; SIQUEIRA, R. O desenvolvimento do conceito teórico referente à manutenção industrial. 2020.

- NANDI, S.; ILAMPARITHI, T. C.; LEE, S. B.; HYUN, D. Detection of eccentricity faults in induction machines based on nameplate parameters. *IEEE Transactions on Industrial Electronics*, IEEE, v. 58, n. 5, p. 1673–1683, 2010.
- NOBLE, W. S. What is a support vector machine? *Nature biotechnology*, v. 24, p. 1565 – 1567, 2006.
- OBAID, R. R.; HABETLER, T. G.; GRITTER, D. J. A simplified technique for detecting mechanical faults using stator current in small induction motors. *Industrial application of electrical energy world conference*, v. 35, p. 479 – 483, 2000.
- OSISANWO, F. et al. Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, v. 48, n. 3, p. 128–138, 2017.
- OTANI, M.; MACHADO, W. V. A proposta de desenvolvimento de gestão da manutenção industrial na busca da excelência ou classe mundial. *Revista Gestão Industrial*, v. 4, n. 2, p. 1–16, 2008.
- PATLE, A.; CHOUHAN, D. S. Svm kernel functions for classification. In: IEEE. *2013 International Conference on Advances in Technology and Engineering (ICATE)*. [S.l.], 2013. p. 1–9.
- PEDOTTI, L. A. D. S.; ZAGO, R. M.; FRUETT, F. Fault diagnostics in rotary machines through spectral vibration analysis using low-cost mems devices. *IEEE Instrumentation & Measurement Magazine*, IEEE, v. 20, n. 6, p. 39–44, 2017.
- PEREIRA, A. C. M. Emissividade dos materiais de construção-influência da medição em diferentes variantes. 2015.
- RASCHKA, S.; MIRJALILI, V. *Python machine learning*. [S.l.]: Packt Publishing Ltd, 2017.
- RIBEIRO, F. M.; MARINS, M. A.; NETTO, S. L.; SILVA, E. A. da. Rotating machinery fault diagnosis using similarity-based models. 2019.
- SANJAA, B.; CHULUUN, E. Malware detection using linear svm. In: IEEE. *Ifost*. [S.l.], 2013. v. 2, p. 136–138.
- SANTOS, H. L. d. et al. *Processamento de emissões acústicas aplicado a detecção de falhas em máquinas elétricas*. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2019.
- SIDDIQUI, K. M.; SAHAY, K.; GIRI, V. Health monitoring and fault diagnosis in induction motor-a review. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, v. 3, n. 1, p. 6549–6565, 2014.

- SMITH, W. A.; RANDALL, R. B. Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. *Mechanical systems and signal processing*, Elsevier, v. 64, p. 100–131, 2015.
- SOUZA, R. M.; NASCIMENTO, E. G.; MIRANDA, U. A.; SILVA, W. J.; LEPIKSON, H. A. Deep learning for diagnosis and classification of faults in industrial rotating machinery. *Computers & Industrial Engineering*, Elsevier, v. 153, p. 107060, 2021.
- SOUZA, R. M. de; LEPIKSON, H. A.; SPERANDIO, E. G. Aprendizagem de máquina e aprendizagem profunda para classificação de falhas em equipamentos rotativos: Uma revisão sistemática. 2019.
- TOMA, R. N.; PROSVIRIN, A. E.; KIM, J.-M. Bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 20, n. 7, p. 1884, 2020.
- USHA, T.; BALAMURUGAN, S. A. A. Seasonal based electricity demand forecasting using time series analysis. *Circuits and Systems*, Scientific Research Publishing, v. 7, n. 10, p. 3320–3328, 2016.
- WANG, L.; GAO, R. X. *Condition monitoring and control for intelligent manufacturing*. [S.l.]: Springer Science & Business Media, 2006.
- ZAREI, J.; TAJEDDINI, M. A.; KARIMI, H. R. Vibration analysis for bearing fault detection and classification using an intelligent filter. *Mechatronics*, v. 24, p. 151 – 157, 2014.
- ZHANG, X.-D. Machine learning. In: _____. *A Matrix Algebra Approach to Artificial Intelligence*. Singapore: Springer Singapore, 2020. p. 223–440. ISBN 978-981-15-2770-8. Disponível em: <https://doi.org/10.1007/978-981-15-2770-8_6>.

Trabalhos Publicados

Artigos publicados:

Álvaro Souza de Araújo, Oberdan Pinheiro Rocha, Alex Álisson Bandeira Santos. MODELO COMPUTACIONAL PARA ANÁLISE DE CONDIÇÕES E MONITORAMENTO DE EQUIPAMENTOS, International Symposium on Innovation and Technology VI (SI-INTEC) 2020.

Álvaro Souza de Araújo, Oberdan Pinheiro Rocha, Alex Álisson Bandeira Santos. MODELO COMPUTACIONAL PARA ANÁLISE DE CONDIÇÕES E MONITORAMENTO DE EQUIPAMENTOS, International Symposium on Innovation and Technology VII (SI-INTEC) 2021.

Análise, avaliação e validação do uso de técnicas de inteligência artificial para o diagnóstico de falhas em motores de indução

Álvaro Souza de Araújo

Salvador, Julho de 2022.