



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE SISTEMAS
ELETRÔNICOS

FABIANA SEIDEL

**DETECÇÃO DE FALHAS MECÂNICAS EM MOTORES ELÉTRICOS BASEADA EM
MODELOS DE APRENDIZADO DE MÁQUINA EMBARCADOS EM
MICROCONTROLADOR**

DISSERTAÇÃO DE MESTRADO – DM – PPGESE – 003/2024

Joinville
2024

Fabiana Seidel

**DETECÇÃO DE FALHAS MECÂNICAS EM MOTORES ELÉTRICOS BASEADA EM
MODELOS DE APRENDIZADO DE MÁQUINA EMBARCADOS EM
MICROCONTROLADOR**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Sistemas Eletrônicos da Universidade Federal de Santa Catarina para a obtenção do título de Mestre em Engenharia de Sistemas Eletrônicos.

Orientador: Prof. Dr. Pablo Andretta Jaskowiak

Coorientadora: Profa. Dra. Aline Durrer Patelli Juliani

Joinville

2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Seidel, Fabiana

Detecção de falhas mecânicas em motores elétricos baseada em modelos de aprendizado de máquina embarcados em microcontroladores / Fabiana Seidel ; orientador, Pablo Andretta Jaskowiak, coorientadora, Aline Durrer Patelli Juliani, 2024.

82 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Campus Joinville, Programa de Pós-Graduação em Engenharia de Sistemas Eletrônicos, Joinville, 2024.

Inclui referências.

1. Engenharia de Sistemas Eletrônicos. 2. Aprendizado de máquina. 3. Modelos para detecção de falhas. 4. Detecção de falhas mecânicas em motores. 5. Motores elétricos trifásicos. I. Jaskowiak, Pablo Andretta. II. Juliani, Aline Durrer Patelli . III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Sistemas Eletrônicos. IV. Título.

Fabiana Seidel

Detecção de falhas mecânicas em motores elétricos baseada em modelos de aprendizado de máquina embarcados em microcontrolador

O presente trabalho em nível de Mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Dr. Ademir Nied
UDESC

Prof. Dr. Lucas Weihmann
UFSC

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Mestre em Engenharia de Sistemas Eletrônicos.

Prof. Dr. Lucas Weihmann
Coordenador do Programa

Prof. Dr. Pablo Andretta Jaskowiak
Orientador

Joinville, 02 de Agosto de 2024.

AGRADECIMENTOS

Agradeço ao Patrick, meu companheiro de vida, uma das pessoas que mais me deu suporte durante esse período. Suporte nas tarefas diárias enquanto eu estudava, no colo para me acalmar quando o cansaço batia e nas palavras de positivas e alegres de sempre. É um privilégio ter você ao meu lado em todas as fases da minha vida.

Agradeço muito ao meu orientador Pablo e coorientadora Aline, por toda a orientação e dedicação. Não poderia ter feito escolha melhor para me acompanhar durante todo esse processo.

Agradeço aos meus colegas de trabalho Thiago e Marcus pela ajuda durante a realização do trabalho e ao meu chefe Hugo, pela compreensão quando foi necessário mesclar os papéis de mestrandia e colaboradora.

Agradeço aos meus amigos mais próximos, que sempre estiveram presentes e precisaram ouvir diversas vezes que eu precisava estudar.

Por fim, durante a longa jornada de realização deste trabalho de mestrado, tive a oportunidade de confirmar quão rodeada de pessoas de bem estou. Reafirmo o agradecimento aos acima citados por todo o apoio, e estendo aos que de uma forma ou de outra participaram dessa jornada.

RESUMO

O movimento da indústria 4.0 tem atraído o interesse de diversas empresas que visam tornar seus processos mais eficientes e digitalizados. Uma área muito afetada por esse movimento é a manutenção industrial, onde novas tecnologias vêm sendo empregadas para monitoramento de condição de ativos. Motores elétricos de indução são parte muito importante das indústrias pois estão presentes em diversos processos industriais. Prevenir falhas nestes motores é o objetivo almejado por equipes de manutenção industrial. Este trabalho faz um estudo de caso para identificação de falhas mecânicas de desbalanceamento e desalinhamento em motores elétricos de indução trifásicos. O objetivo é identificar falhas executando modelos de aprendizado de máquina embarcados em um microcontrolador. Os sensores de internet das coisas (IoT) aplicados na digitalização das indústrias normalmente contam com microcontroladores internamente. Executar os algoritmos inteligentes dentro desses dispositivos com restrições de recursos de memória, processamento e energia, poder trazer vantagens como menor latência de resposta e independência de rede de internet. São treinados modelos de aprendizado de máquina utilizando dados coletados através de um sensor IoT comercial. Estes modelos são então embarcados no mesmo sensor para realizar a inferência da condição do motor no ensaio onde os dados foram coletados. Os modelos foram avaliados utilizando a técnica de validação cruzada. Os modelos supervisionados testados tiveram acurácia superior a 90% nos dados de treinamento e o modelo não supervisionado superior a 99%. Durante a inferência, os resultados para os modelos supervisionados também foram satisfatórios, alcançando acurácia média de 83.3% para o modelo de árvore de decisão e 71.5% para o SVM. O modelo não supervisionado se mostrou bastante dependente dos parâmetros selecionados, sendo que para os parâmetros definidos com o conjunto de dados de treinamento, obteve acurácia de 98.21% para os dados de teste e de apenas 9.5% para dados normais, no conjunto de dados de inferência.

Palavras-chave: Aprendizado de máquina; Sistemas embarcados; Detecção de falhas; Motores Elétricos

ABSTRACT

The Industry 4.0 movement has attracted the interest of various companies aiming to make their processes more efficient and digitalized. One area greatly affected by this movement is industrial maintenance, where new technologies are being employed in assets condition monitoring. Induction electric motors are a very important part of industries as they are present in various industrial processes. Preventing failures in these motors is the goal of industrial maintenance teams. This work presents a case study for identifying mechanical failures of imbalance and misalignment in three-phase induction motors. The objective is to identify faults by running machine learning models embedded in a microcontroller. The Internet of Things (IoT) sensors used in the digitalization of industries typically use microcontrollers. Running intelligent algorithms within these devices with memory, processing, and energy resource constraints can bring advantages such as lower response latency and internet network independence. Machine learning models are trained using data collected through a commercial IoT sensor. These models are then embedded in the same sensor to perform inference of the motor condition in the test where the data was collected. The models were evaluated using the cross-validation technique. The supervised models tested had an accuracy greater than 90% on the training data, and the unsupervised model greater than 99%. During inference, the results for the supervised models were also satisfactory, achieving an average accuracy of 83.3% for the decision tree model and 71.5% for the SVM. The unsupervised model was highly dependent on the selected parameters, obtaining an accuracy of 98.2% for the test data and only 9.5% for normal data in the inference dataset.

Keywords: Supervised Machine Learning; Embedded Systems; Condition Monitoring; Electrical Machines

LISTA DE FIGURAS

Figura 2.1 – Principais tipos de motores elétricos.	23
Figura 2.2 – Partes de um motor trifásico de indução.	24
Figura 2.3 – Arranjo das bobinas de um motor trifásico.	25
Figura 2.4 – Parte de um rotor bobinado.	26
Figura 2.5 – Parte de um rotor em gaiola.	27
Figura 2.6 – (a) Fluxo produzido por um enrolamento concêntrico em uma máquina de entreferro uniforme. (b) A FMM produzida no entreferro por uma corrente nesse enrolamento.	28
Figura 2.7 – A FMM de uma fase de um enrolamento trifásico distribuído de dois polos.	29
Figura 2.8 – Tipos de falhas em motores elétricos.	32
Figura 2.9 – Tipos de falhas em motores elétricos.	33
Figura 2.10 – Construção de um rolamento de esferas.	34
Figura 2.11 – Seção transversal do motor de indução: íntegro, com excentricidade estática e com excentricidade dinâmica.	35
Figura 2.12 – Representação do alinhamento e dos três tipos de desalinhamento em vista 2D.	36
Figura 2.13 – Motor com desbalanceamento estático (a) ou desbalanceamento acoplado (b).	37
Figura 2.14 – Monitoramento de falha online de um motor de indução.	39
Figura 3.1 – Principais técnicas de aprendizado e seus respectivos dados esperados.	42
Figura 3.2 – Fluxo para criação de modelo de aprendizado supervisionado. . .	43
Figura 3.3 – Representações da divisão da árvore de decisão	44
Figura 3.4 – Separação linear usando SVM para dados de duas dimensões. . .	46
Figura 3.5 – Separação não linear de dados de duas dimensões utilizando uma função Gaussiana como kernel	46
Figura 3.6 – Separação do espaço de dados na classificação de uma classe .	48
Figura 3.7 – Fluxo típico do tinyML.	52
Figura 4.1 – Fluxo do desenvolvimento do trabalho - geração da base de dados.	55
Figura 4.2 – Sistema Back-to-Back utilizado nos ensaios de qualificação. . . .	56
Figura 4.3 – Sistema Back-to-Back utilizado nos ensaios finais.	57
Figura 4.4 – Árvore de ensaios para geração da base de dados.	57
Figura 4.5 – Ajustes para gerar falhas na bancada.	58
Figura 4.6 – Espectro em frequência para diferentes condições do motor. . . .	59
Figura 4.7 – Sensor WEGscan100.	60
Figura 4.8 – Sensor fixado no motor de acionamento.	61

Figura 4.9 – Número de amostras de dados por condição e carga do motor. . .	62
Figura 5.1 – Fase de seleção e treinamento dos modelos.	63
Figura 5.2 – PCA em duas dimensões para o conjunto de dados coletados. . .	64
Figura 5.3 – Gráfico de boxplot dos dados do conjunto de dados coletados. . .	65
Figura 5.4 – Média das matrizes de confusão das 10 pastas para Árvore de Decisão.	67
Figura 5.5 – Média das matrizes de confusão das 10 pastas para o SVM. . . .	68
Figura 5.6 – Acurácia do modelo para diferentes Kernels.	70
Figura 6.1 – Fase de seleção e treinamento dos modelos.	71
Figura 6.2 – FFT da condição normal do motor para inferência.	73
Figura 6.3 – FFT da condição desalinhado do motor para inferência.	73
Figura 6.4 – FFT da condição desbalanceado do motor para inferência.	74
Figura 6.5 – Matriz de confusão do ensaio de inferência.	74
Figura 6.6 – Matriz de confusão do ensaio de inferência para 0% de carga. . .	75
Figura 6.7 – Matriz de confusão do ensaio de inferência para o modelo de SVM com 100% de carga.	76
Figura 6.8 – FFT da condição desalinhado com 0, 50 e 100% de carga.	77
Figura 6.9 – Matriz de confusão do modelo de One-Class SVM com kernel RBF.	78
Figura 6.10 – Matriz de confusão do modelo de One-Class SVM com kernel poli- nomial.	78

LISTA DE TABELAS

Tabela 3.1 – Comparação de bibliotecas de aprendizado de máquina para plataformas embarcadas.	53
Tabela 4.1 – Variáveis do conjunto de dados para treinamento de modelos de AM.	61
Tabela 5.1 – Desempenho do modelo de árvore de decisão.	66
Tabela 5.2 – Desempenho do modelo de SVM.	68
Tabela 5.3 – Acurácia para diferentes kernels.	69
Tabela 5.4 – Acurácia para diferentes valores de ν	70
Tabela 5.5 – Acurácia para diferentes valores de γ	70
Tabela 6.1 – Tamanho dos modelos portáteis.	71
Tabela 6.2 – Tabela de acertos dos modelos supervisionados.	73
Tabela 6.3 – Acurácia dos Modelos de Árvore de Decisão e SVM em Diferentes Cargas e Condições de Falha.	75
Tabela 6.4 – Desempenho do modelo de One-Class SVM com kernel RBF.	78
Tabela 6.5 – Desempenho do modelo de One-Class SVM com kernel polinomial.	79

SUMÁRIO

1	INTRODUÇÃO	19
1.1	OBJETIVOS	20
1.1.1	Objetivo Geral	20
1.1.2	Objetivos Específicos	20
1.2	ORGANIZAÇÃO DO TEXTO	21
2	CONCEITOS ENVOLVENDO MOTORES ELÉTRICOS DE INDUÇÃO	23
2.1	TIPOS DE MOTORES	23
2.2	FUNCIONAMENTO DE UM MOTOR ELÉTRICO DE INDUÇÃO	23
2.2.1	Principais componentes do motor de indução trifásico	24
2.2.2	Produção do campo magnético	27
2.2.3	Princípio de funcionamento	28
2.3	MANUTENÇÃO DE MOTORES ELÉTRICOS INDUSTRIAIS	30
2.3.1	Falhas mecânicas em motores elétricos de indução trifásicos	31
2.3.1.1	Rolamentos	33
2.3.1.2	Excentricidade	34
2.3.1.3	Defeitos relacionados à carga do motor	35
2.4	TÉCNICAS DE MONITORAMENTO E DETECÇÃO DE FALHAS	38
2.4.1	Diagnóstico de falhas pelos sinais de vibração mecânica	38
3	APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE FALHAS EM MÁQUINAS ELÉTRICAS	41
3.1	TIPOS DE APRENDIZADO DE MÁQUINA	41
3.2	ALGORITMOS DE APRENDIZADO DE MÁQUINA	43
3.2.1	Árvore de decisão	43
3.2.2	Support Vector Machine	45
3.2.3	One Class SVM	47
3.3	APRENDIZADO DE MÁQUINA NO ÂMBITO DE DETECÇÃO DE FALHAS	48
3.4	APRENDIZADO DE MÁQUINA EMBARCADO EM MICROCONTROLADORES	51
3.5	CONSIDERAÇÕES FINAIS	54
4	GERAÇÃO DA BASE DE DADOS	55
4.1	BANCADA DE ENSAIOS PRELIMINAR	55
4.2	BANCADA DE ENSAIOS DEFINITIVA	56
4.3	MANIPULAÇÃO DAS CONDIÇÕES DA BANCADA	57
4.4	AQUISIÇÃO DOS DADOS	60
5	TREINAMENTO E AVALIAÇÃO DOS MODELOS DE AM	63
5.1	TRATAMENTO DOS DADOS	63
5.2	SELEÇÃO E TREINAMENTO DOS MODELOS	65

5.2.1	Árvore de decisão	65
5.2.2	SVM	67
5.2.3	One-Class SVM	69
6	RESULTADOS DOS ALGORITMOS EMBARCADOS	71
6.1	CONVERSÃO E PORTABILIDADE DO MODELO	71
6.2	ENSAIOS DE INFERÊNCIA	72
6.2.1	Ajustes da bancada para inferência	72
6.2.2	Inferência para modelos supervisionados	72
6.2.3	Inferência para modelo não supervisionado	76
7	CONCLUSÕES	81
	Referências	83

1 INTRODUÇÃO

Com os avanços da indústria 4.0 e do IoT (do inglês, *Internet of Things*), as indústrias têm melhorado cada vez mais seus processos a fim de alcançar a excelência operacional. Um dos componentes principais dessas indústrias são os motores elétricos. A degradação de partes dessas máquinas é um cenário comum na indústria e para mantê-las operantes são necessárias equipes de manutenção industrial. Se o motor elétrico falha, todo o processo é interrompido, com perdas operacionais e de produção que podem gerar prejuízos consideráveis [You, 2013].

As equipes de manutenção têm se beneficiado dos recentes avanços tecnológicos, principalmente nas áreas de monitoramento inteligente e análise de dados, onde dados podem ser coletados automaticamente e analisados em tempo real [Zhang et al., 2016]. Da forma convencional, analistas de manutenção possuem analisadores de vibração de alto custo e se deslocam até as máquinas para fazer a avaliação. O tempo de análise de cada máquina é alto e as máquinas são avaliadas randomicamente, somente pela ordem de prioridade do processo. Além disso, existem outros pontos de atenção, como questões de segurança e insalubridade. Os sensores de monitoramento inteligente auxiliam neste processo através da informação de quais máquinas precisam de mais atenção, possibilitando um ranqueamento de prioridades.

Algoritmos de Aprendizado de Máquina (AM) têm sido aplicados para analisar esses dados de sensores inteligentes, a fim de classificar falhas ou prever seus comportamentos. Normalmente, são necessárias grandes quantidades de dados ou tempo de aprendizado, para que os modelos aprendam o comportamento das máquinas e identifiquem desvios operacionais. Após a fase de aprendizado, estes modelos possuem grande potencial de redução de custos por planejamento de manutenção, evitando paradas desnecessárias. Alguns estudos de aplicação de modelos de AM em manutenção preditiva são mostrados em Paolanti et al. [2018b], Prytz et al. [2015] e Huda and Taib [2013].

Normalmente, os dados são coletados por sensores fixados diretamente na máquina e enviados para a nuvem onde são executados os algoritmos de AM. A privacidade dos dados tem sido um assunto muito discutido recentemente, através de leis como a Lei Geral de Proteção de Dados (LGPD) [Brasil, 2018]. Uma alternativa ao processamento em nuvem é a execução dos modelos embarcados nos sensores IoT. Normalmente, estes sensores são dispositivos de baixo consumo de energia e alimentados por bateria, o que implica em dispositivos com menor poder de processamento e restrições com relação à memória e consumo energético.

Dentro deste contexto, a contribuição deste trabalho é a investigação sobre o uso de modelos de AM, executados dentro de um microcontrolador de um sensor IoT

comercial ¹, para identificação de falhas de desalinhamento e desbalanceamento em motores elétricos de indução trifásicos. Para tanto, foi construída uma bancada de ensaios que permite a coleta de dados das diferentes falhas avaliadas nesse trabalho. Foram realizados estudos para identificar algoritmos apropriados para identificação das falhas e estudadas as formas de embarcar esses algoritmos dentro do microcontrolador. Finalmente, esses modelos embarcados são testados para fazerem a identificação das falhas.

1.1 OBJETIVOS

Abaixo estão descritos o objetivo geral e os objetivos específicos deste trabalho.

1.1.1 Objetivo Geral

O objetivo geral é a identificação de falhas de desalinhamento e desbalanceamento em motores elétricos de indução trifásicos utilizando modelos de aprendizado de máquina embarcados em um microcontrolador. Para tal, são coletados dados das condições de um motor em uma bancada experimental construída para a realização do trabalho. É feito o treinamento de modelos de árvore de decisão, SVM e One-Class SVM com o conjunto de dados gerado. Os modelos treinados são embarcados em um microcontrolador e realizam a inferência com dados ainda não vistos.

1.1.2 Objetivos Específicos

Os objetivos específicos do trabalho são:

- Criação de uma base de dados das condições normal e de falhas de desbalanceamento e desalinhamento de motores elétricos;
- Identificação de algoritmos de aprendizado de máquina adequados para resolver o problema proposto;
- Criação de modelos de aprendizado supervisionado e não supervisionado, utilizando técnicas para otimização de parâmetros;
- Conversão de modelos de aprendizado de máquina para linguagem compilada, o que possibilita a execução embarcada em microcontroladores.
- Portabilidade dos modelos e avaliação da capacidade de um microcontrolador em executá-los;
- Avaliação da capacidade de modelos embarcados em classificar corretamente as condições do motor.

¹ Neste trabalho, o sensor IoT utilizado é o sensor WEGscan100 que contém um acelerômetro internamente para medição de vibração

1.2 ORGANIZAÇÃO DO TEXTO

O trabalho é estruturado da seguinte forma:

- O Capítulo 2 traz uma revisão sobre motores elétricos e seu funcionamento, abordando os principais componentes de um motor elétrico e suas funções. Também apresenta as principais falhas encontradas em motores elétricos.
- O Capítulo 3 apresenta uma breve revisão sobre os algoritmos de aprendizado de máquina e seus variações, tendo foco no aprendizado supervisionado.
- O Capítulo 4 aborda a geração da base de dados, mostrando a bancada construída para coleta dos dados e o sensor utilizado.
- No Capítulo 5 mostra a fase de criação dos estimadores e da avaliação dos mesmos com os dados do conjunto de dados utilizado no trabalho.
- No Capítulo 6 é mostrada a técnica utilizada para embarcar os modelos criados e os resultados obtidos durante a fase de inferência.
- Finalmente, no Capítulo 7 é feita a conclusão do trabalho comentando os resultados encontrados e possíveis trabalhos futuros.

2 CONCEITOS ENVOLVENDO MOTORES ELÉTRICOS DE INDUÇÃO

O objetivo deste capítulo é apresentar os conceitos básicos de funcionamento do motor elétrico de indução. O texto se embasa principalmente no livro Máquinas Elétricas de Umans [2014].

2.1 TIPOS DE MOTORES

Um motor elétrico é uma máquina capaz de converter energia elétrica em energia mecânica. Estas máquinas assumem diversas formas construtivas, podendo ser separadas em dois grandes grupos - motores com alimentação em corrente contínua (CC) ou de corrente alternada (CA). Os principais tipos de motores são mostrados no diagrama da Figura 2.1. No âmbito dos motores de alimentação CA, ainda pode haver a classificação em duas categorias, motores síncronos e motores assíncronos, também conhecidos como motores de indução. O foco deste trabalho está nos motores de indução. Nestas máquinas, as correntes são induzidas nos enrolamentos do rotor por meio da combinação da variação, no tempo, das correntes de estator e do movimento do rotor em relação ao estator. Os motores de indução são os mais difundidos em ambientes industriais, por possuírem vantagens como baixo custo, construção simples e versatilidade de adaptação às cargas dos mais diversos tipos [WEG, 2021], além disso possuem a possibilidade de acionamento a partir da rede elétrica, sem a dependência de um variador de frequência para acionamento.

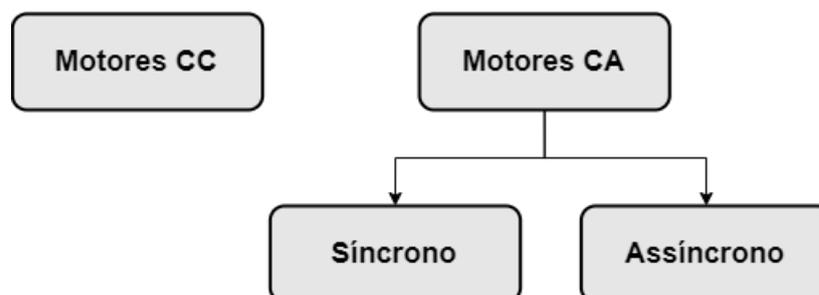


Figura 2.1 – Principais tipos de motores elétricos.

Fonte: Adaptado de Luiz Carlos de Freitas Júnior [2018]

2.2 FUNCIONAMENTO DE UM MOTOR ELÉTRICO DE INDUÇÃO

Para entender o funcionamento de um motor de indução é necessário conhecer todas as partes que compõe essa máquina. A seguir, são mostrados com mais detalhes seus principais componentes. A principal diferença entre um motor de indução trifásico para o monofásico está na quantidade e disposição das bobinas do estator. No âmbito deste trabalho é abordado o caso do motor de indução trifásico.

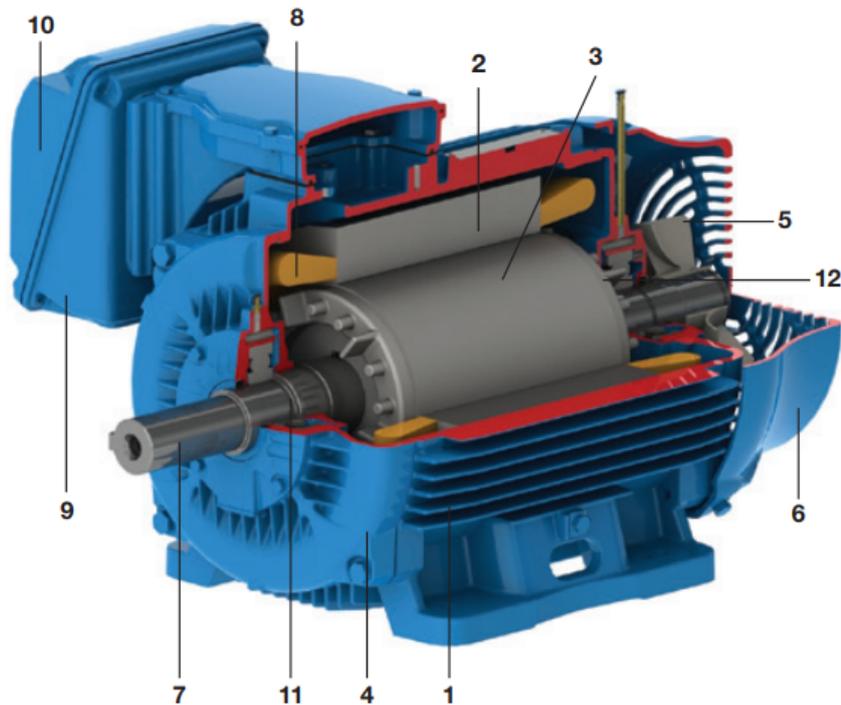


Figura 2.2 – Partes de um motor trifásico de indução.

Fonte: [WEG, 2021]

2.2.1 Principais componentes do motor de indução trifásico

A Figura 2.2 mostra um motor de indução trifásico aberto e as partes que compõe essa máquina são enumeradas. Durante o texto serão abordadas cada uma dessas partes.

Os principais componentes são o rotor, relacionado a parte rotativa, e o estator, relacionado à parte estática do motor. As partes integrantes do estator são:

- a carcaça (1), estrutura suporte do conjunto;
- o núcleo de chapas (2), construído em aço magnético;
- o enrolamento trifásico (8), que é o conjunto de bobinas que formam o enrolamento do estator.

O enrolamento trifásico do estator é composto por três bobinas monofásicas com defasagem de 120 graus entre si, podendo ser conectado em estrela ou triângulo. As bobinas são fabricadas, na maior parte das máquinas, com fios de cobre isolados entre si e alocados em ranhuras [Luiz Carlos de Freitas Júnior, 2018]. A Figura 2.3 representa como são arranjadas tais bobinas.

Já o enrolamento do rotor possui os terminais curto-circuitados para viabilizar a indução magnética de corrente. Um dos tipos de rotores dessas máquinas possuem a

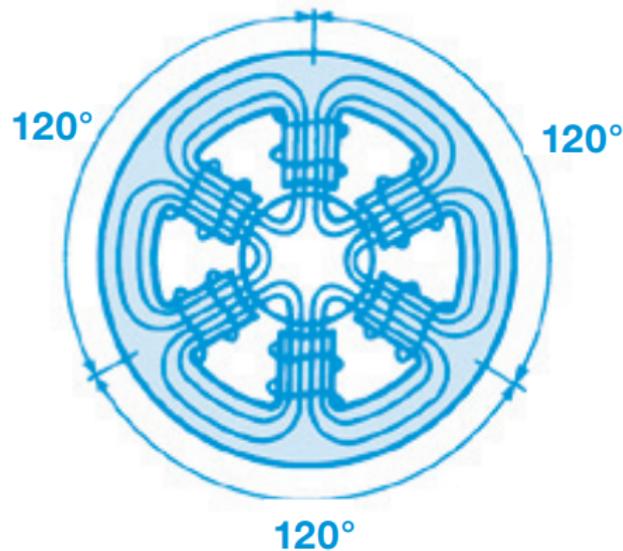


Figura 2.3 – Arranjo das bobinas de um motor trifásico.

Fonte: [WEG, 2021]

construção no formato de barras conectadas a anéis, sendo conhecidos como gaiola de esquilo [Luiz Carlos de Freitas Júnior, 2018]. O rotor é composto por:

- eixo (7), transmite a potência mecânica desenvolvida pelo motor;
- núcleo de chapas (3) similares às do estator;
- barras e anéis de curto-circuito (12), que formam o enrolamento do rotor do tipo gaiola.

Além das partes que formam o conjunto estator-rotor, são necessárias outras partes auxiliares em um motor comercial para garantir o seu correto funcionamento e eficiência. Tais partes são descritas em Luiz Carlos de Freitas Júnior [2018] e mostradas a seguir:

- Tampa (4) para fechamento frontal do motor;
- Ventilador (5), sistema de ventilação forçada para auxiliar na troca térmica do motor. É acionado pelo próprio giro do rotor (em ambos os sentidos). As aletas normalmente estão localizadas na carcaça do estator da máquina e são construídas de forma a permitir um fluxo de ar otimizado para a refrigeração da máquina;
- Tampa defletora (6), conectada após o ventilador, utilizada para deflexão;
- Caixa de ligação (9) permite acesso aos terminais do estator;
- Terminais (10) onde é feita a ligação da máquina a rede elétrica;

- Rolamento (11) peça importante que auxilia na rotação do eixo, diminuindo o atrito entre as peças. São normalmente do tipo esfera.

O rotor do motor de indução tem duas variantes construtivas possíveis, podendo ser o rotor bobinado ou rotor de gaiola, mostrados nas Figuras 2.4 e 2.5, respectivamente.

O rotor bobinado é composto por um enrolamento trifásico similar ao do estator. Os terminais das bobinas que formam o enrolamento são conectadas a três anéis coletores, isolados eletricamente entre si e montados sobre o eixo. Para ter acesso ao rotor, são utilizadas escovas de carvão apoiadas sobre os anéis, formando um sistema de contatos móveis e permitindo o acesso ao enrolamento mesmo com o rotor em movimento. Essa forma construtiva do rotor tem a vantagem de permitir a alteração de parâmetros elétricos do rotor por inserção de elementos externos de circuito, tipicamente resistores, promovendo alterações nas características do torque do motor para se adequar ao tipo de carga acionada.

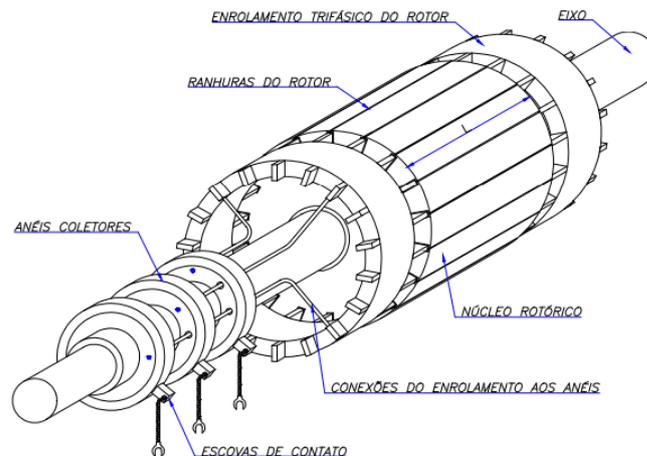


Figura 2.4 – Parte de um rotor bobinado.

Fonte: [Umans, 2014]

No rotor em gaiola, o enrolamento é formado por barras condutoras de cobre ou alumínio inseridas em ranhuras no aço elétrico do rotor e curto-circuitadas nas extremidades por anéis condutores dos mesmos materiais. A forma construtiva desse conjunto de barras forma um circuito elétrico fechado em curto, não permitindo nenhum acesso ao enrolamento. Nos motores de fabricação seriada, essa gaiola é obtida por um processo de fundição ou injeção de alumínio diretamente no núcleo do rotor, tornando possível uma grande taxa de automação e com isso contribuindo para a redução do custo de fabricação e aumento da confiabilidade.

Os motores de indução com rotor gaiola de esquilo, em diferentes tamanhos, são os mais utilizados em diversas aplicações. Isso se deve a forma construtiva do

rotor, que garante vantagens como maior robustez e simplicidade na fabricação. O motor apresentado na Figura 2.2 possui rotor de gaiola.

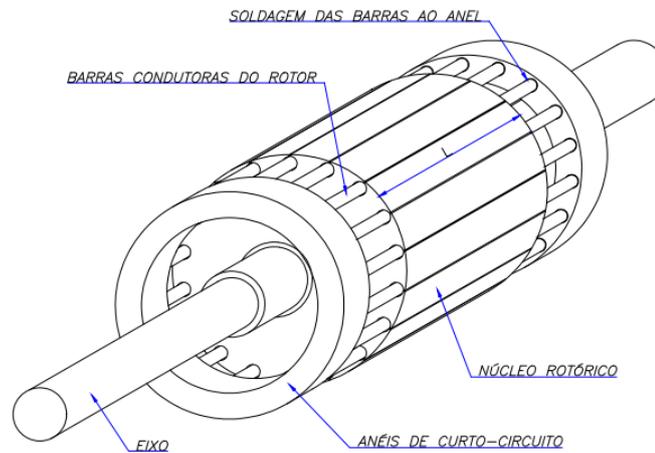


Figura 2.5 – Parte de um rotor em gaiola.

Fonte: [Umans, 2014]

O rotor da máquina de indução é normalmente montado dentro do estator de forma concêntrica, utilizando-se um sistema de mancais para sustentação da estrutura mecânica da máquina. Entre o rotor e o estator existe um espaço de ar que é chamado de entreferro, onde ocorre a conversão eletromecânica de energia [Luiz Carlos de Freitas Júnior, 2018] que será discutida na Seção 2.2.3.

2.2.2 Produção do campo magnético

A lei de Faraday da indução eletromagnética mostra que uma força eletromotriz (FEM) é gerada em uma bobina quando ela é submetida a uma variação de fluxo magnético em seu interior. A intensidade da força eletromotriz gerada é proporcional à taxa de variação do fluxo magnético que atravessa a área da bobina. Esta lei é fundamental para entender o princípio de funcionamento de uma máquina rotativa, pois a conversão eletromagnética de energia ocorre devido a variações de fluxo pela rotação das partes móveis da máquina.

Além da FEM, outro conceito importante para compreensão do funcionamento de um motor elétrico é a força magnetomotriz (FMM). A FMM é proporcional ao número de espiras da bobina e a intensidade da corrente que circula por elas.

Para melhor entender o conceito do campo magnético gerado, é mostrado na Figura 2.6(a) um enrolamento de uma única bobina com N espiras compreendendo 180 graus elétricos. Essa bobina é percorrida por uma corrente i , então conforme visto anteriormente, a FMM produzida terá intensidade $N.i$. Como cada linha de fluxo cruza o entreferro duas vezes, a FMM no entreferro deve ser igual à metade do total ou

$N * i/2$. Com isso, conforme pode ser visto na Figura 2.6(b), a força magnetomotriz produzida pode ser representada por uma onda quadrada de amplitude $N * i/2$.

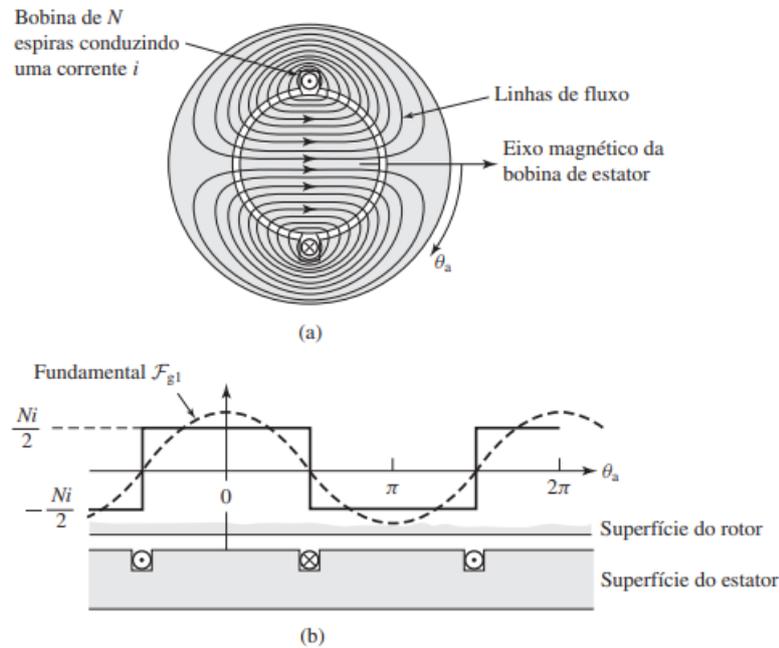


Figura 2.6 – (a) Fluxo produzido por um enrolamento concêntrico em uma máquina de entreferro uniforme. (b) A FMM produzida no entreferro por uma corrente nesse enrolamento.

Fonte: [Umans, 2014]

No projeto das máquinas CA, os enrolamentos são distribuídos para produzir uma distribuição espacial senoidal da FMM. Separar as bobinas de cada fase em um número maior de ranhuras otimiza o uso do aço do núcleo e do cobre dos enrolamentos, melhorando significativamente a forma da onda gerada e resultando em um desempenho mais eficiente da máquina. Tal comportamento pode ser visto na Figura 2.7, onde em (a) é mostrado o enrolamento de um motor trifásico de dois polos com bobinas separadas em 180 graus elétricos e em (b) a FMM resultante da distribuição do enrolamento em diversas ranhuras. É vista uma forma de onda que se assemelha a uma senóide, diferente do que é mostrado na Figura 2.6.

2.2.3 Princípio de funcionamento

As máquinas assíncronas são assim chamadas pois a velocidade do rotor será sempre em um valor próximo da velocidade síncrona, mas nunca igual a ela. Isso se deve ao fenômeno da indução magnética que ocorre na máquina, daí o nome máquina de indução [Luiz Carlos de Freitas Júnior, 2018]. No caso deste motor, somente o estator é ligado à rede de alimentação. O rotor não é alimentado e as correntes que circulam nele são induzidas eletromagneticamente pelas bobinas do estator.

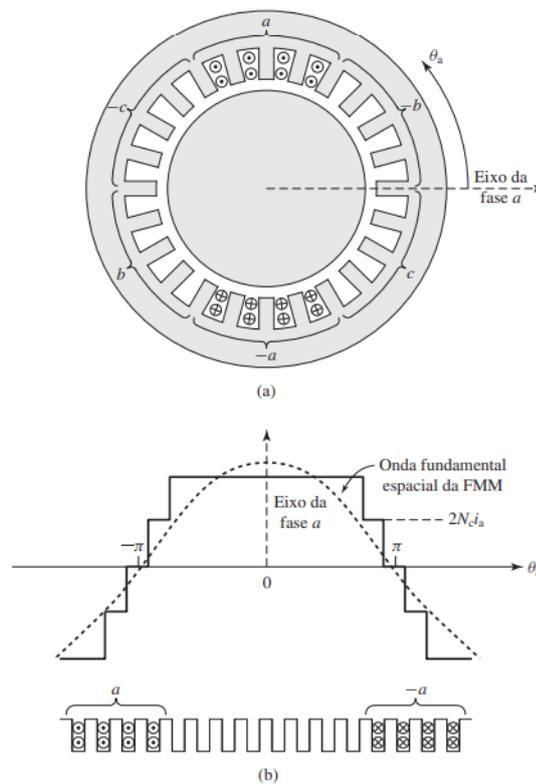


Figura 2.7 – A FMM de uma fase de um enrolamento trifásico distribuído de dois polos.

Fonte: Adaptado de [Umans, 2014]

Quando uma corrente elétrica percorre a bobina do estator, é gerado um campo magnético orientado na direção do eixo da bobina. O campo total resultante em cada instante da corrente alternada forma um "campo girante", onde a sua direção é alterada com a mudança de fase da corrente, mas com intensidade constante. O campo gerado no estator atravessa o entreferro e induz no rotor a força eletromotriz.

A força eletromotriz gerada induz circulação de corrente no enrolamento do rotor, criando um campo magnético no rotor, que tem polaridade oposta ao campo do estator. Os campos opostos então se atraem e o campo do rotor segue o campo girante do estator resultando na produção de torque, ou conjugado, de partida. A lei de Lenz diz que a força eletromotriz induzida por uma variação de fluxo magnético será sempre no sentido oposto a essa variação que o causou [HALLIDAY, 2004]. Isso determina que a direção do torque será oposta ao campo do rotor, ou seja, na mesma direção do campo do estator. Se esse conjugado for suficiente para superar a oposição à rotação criada pela carga no eixo, então o motor atingirá a sua velocidade de operação.

A velocidade com que o rotor irá girar será relativa à velocidade síncrona do motor n_s . Tal velocidade é definida através da Equação (2.1), onde f é a frequência de alimentação do motor e P o número de polos. Neste caso, se forem variados os parâmetros que definem a velocidade síncrona, o valor da velocidade de rotação mecânica também será alterado.

$$n_s = \frac{120 * f}{P} \quad (2.1)$$

O campo do rotor estará sempre tentando se alinhar com o campo do estator, porém sempre haverá uma defasagem devido à diferença dos parâmetros do estator e do rotor. Como resultado, a velocidade real do eixo do motor será muito próxima da velocidade síncrona, mas nunca igual [Luiz Carlos de Freitas Júnior, 2018]. Se ambas as velocidades se iguallassem, os condutores do rotor estariam estacionários em relação ao campo do estator, portanto não haveria indução de corrente neles e não seria produzido qualquer conjugado.

A diferença entre a velocidade síncrona e a velocidade do rotor é normalmente referida como velocidade de escorregamento do rotor e medido em rotações por minuto (rpm). O escorregamento é definido como uma fração da velocidade síncrona, pela Equação (2.2), onde n_s é a velocidade síncrona e n a velocidade do rotor. O controle da velocidade do motor pode ser feito pela variação da frequência de alimentação entregue ao motor.

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

2.3 MANUTENÇÃO DE MOTORES ELÉTRICOS INDUSTRIAIS

Os motores elétricos são componentes cruciais na maioria dos equipamentos e processos industriais. Uma falha nesses motores pode causar prejuízos significativos para a empresa, tanto em termos de custos de reparo quanto em perda de produtividade. Por isso, assegurar que esses motores estejam sempre em pleno funcionamento é uma prioridade constante para as equipes de manutenção industrial. A confiabilidade dos motores elétricos é fundamental para a eficiência operacional e o sucesso financeiro das indústrias. A manutenção feita combinando tecnologias avançadas e a formação adequada dos profissionais, constitui a melhor abordagem para garantir que esses motores funcionem de maneira ininterrupta e eficiente.

De acordo com a Associação Brasileira de Normas Técnicas (ABNT), por meio da NBR 5462/1994, a manutenção industrial “é a combinação de ações técnicas e administrativas, incluindo as de supervisão, destinadas a manter ou recolocar um item em um estado no qual possa desempenhar uma função requerida” [ABNT, 1994]. Ou seja, fazer com que os processos industriais estejam sempre operantes. Ainda de acordo com a NBR 5462, as três principais técnicas de manutenção aplicadas para garantir a confiabilidade fabril são a manutenção corretiva, preventiva e preditiva.

A manutenção corretiva é definida pela ABNT como a manutenção efetuada após a falha, para tornar o processo operante novamente. Essa forma de manutenção

costuma custar caro, devido à interrupção não programada do processo produtivo e também pela troca de componentes sem planejamento prévio.

Para prevenir falhas inesperadas e minimizar o tempo de inatividade, é essencial implementar um programa robusto de manutenção preventiva. Na manutenção preventiva, as ações de manutenção são realizadas antes mesmo de ocorrer uma falha, geralmente com base em critérios operacionais pré-estabelecidos, como o tempo [Costa and Cavalcante, 2022]. Este programa deve incluir inspeções regulares, testes de desempenho, e no caso das máquinas elétricas, deve compreender a análise de vibração e termografia para identificar sinais precoces de desgaste ou falha iminente. Além disso, a lubrificação adequada e a verificação periódica dos componentes elétricos e mecânicos são práticas indispensáveis.

Apesar de a manutenção preventiva ser uma boa forma de se evitar paradas inesperadas nos processos, com a evolução do conhecimento sobre as máquinas e técnicas de análise de falhas, foi desenvolvida a manutenção preditiva. Esta técnica conta com o uso de tecnologias avançadas, como sensores e a análise de dados em tempo real, e pode proporcionar uma visão mais aprofundada sobre a saúde dos motores elétricos. Isso permite que as equipes de manutenção intervenham antes que ocorra uma falha crítica, economizando tempo e recursos. Essa manutenção também pode ser chamada de “Manutenção Baseada em Condição” [Costa and Cavalcante, 2022]. O conceito de manutenção preditiva já existe há tempos, porém somente com os recentes avanços tecnológicos ela tem se tornado amplamente acessível [Nguyen et al., 2015]. Normalmente, envolve monitoramento de condição, diagnóstico de falhas, prognóstico de falhas e planos de manutenção.

Novas tecnologias têm sido desenvolvidas com potencial para identificar falhas preliminares em equipamentos e componentes de máquinas, ajudando a programação de manutenções baseadas na condição da máquina [Ran, 2019].

2.3.1 Falhas mecânicas em motores elétricos de indução trifásicos

Motores de indução são consideradas máquinas de alta confiabilidade, porém a condição de operação em que estão expostos pode levá-los a diferentes tipos de falhas. A ocorrência de qualquer falha acarretará em problemas operacionais que causarão a paralisação dos trabalhos e, conseqüentemente, prejuízos na produção, bem como possível risco à vida dos trabalhadores [Kiter et al., 2017]. Para obter um bom funcionamento e evitar estes cenários, o monitoramento da condição do motor durante a operação auxilia na detecção e diagnóstico de falha nos estágios iniciais.

O projeto de motores de indução é feito para que as forças eletromecânicas e mecânicas, intrínsecas ao seu funcionamento, tenham uma interação que leva a uma operação estável com o mínimo de ruído e vibrações. Quando ocorre uma falha, perde-se o equilíbrio entre essas forças, o que acarreta na degradação do sistema

[Singh and Ahmed Saleh Al Kazzaz, 2003].

Pradhan et al. [2013] divide os tipos de falhas em motores pela sua origem, podendo ser interna ou externa. As falhas de origem interna são causadas por erros de fabricação e deterioração dos materiais que constituem a máquina, enquanto as falhas de origem externa ocorrem como resultado de interações com o meio ambiente, a fonte de alimentação ou a carga. Em Jigyasu et al. [2018], as falhas em motores elétricos de indução são subdivididas entre falhas elétricas e mecânicas, conforme mostrado na Figura 2.8. Nas falhas elétricas estão as falhas nos enrolamentos do estator ou do rotor. As falhas mecânicas, que são o foco da detecção deste trabalho, podem ser subdivididas em falhas por excentricidade, nos rolamentos do motor ou relacionadas à carga acoplada no eixo.

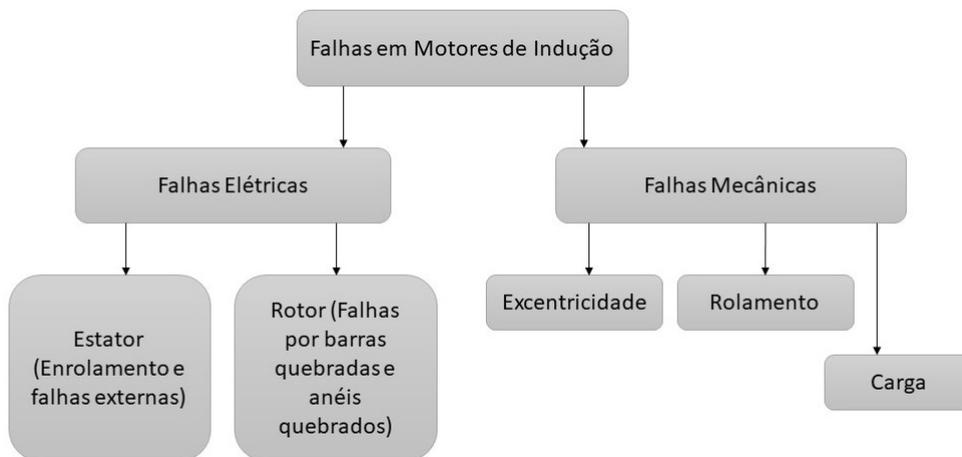


Figura 2.8 – Tipos de falhas em motores elétricos.

Fonte: Adaptado de [Jigyasu et al., 2018]

Na Figura 2.9, adaptada de Bonnett and Young [2004], são mostrados os 5 principais tipos de falhas em motores elétricos e a representatividade de cada uma no cenário industrial. Um estudo mostrado em Bazurto et al. [2016] estima que cerca de 70% das falhas em motores identificadas pelas equipes de manutenção são de origem mecânica. As falhas em rolamentos são as mais representativas, correspondendo a 51% do total de falhas, como observado na Figura 2.9. É seguida por falhas de origem externa relacionadas à carga acoplada, à tensão de alimentação ou ao ambiente, empatado com falhas no enrolamento do estator, que representam 16% cada.

A seguir, serão abordadas as principais causas de falhas mecânicas em motores elétricos de indução, que segundo Jigyasu et al. [2018], são as falhas relacionadas a defeitos de rolamento, excentricidade e defeito de carga.

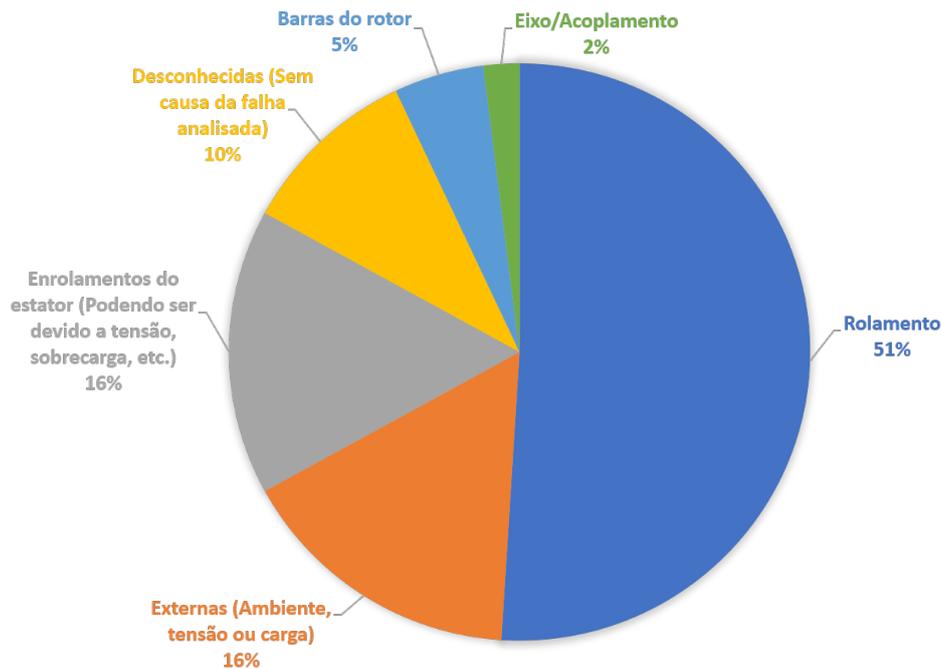


Figura 2.9 – Tipos de falhas em motores elétricos.

Fonte: Adaptado de [Bonnett and Young, 2004]

2.3.1.1 Rolamentos

Os rolamentos são componentes que acabam sendo afetados por outras falhas mecânicas do motor de indução, falhas como desalinhamento e desbalanceamento sobrecarregam os rolamentos. Portanto, há maior possibilidade de ocorrência de falha nestes elementos. Os motores empregados em aplicações industriais normalmente utilizam rolamentos de esferas ou de rolos, tendo anel de rolamento interno, corpos rolantes, anel de rolamento externo e gaiola para suportar os corpos rolantes [Jigyasu et al., 2018]. A Figura 2.10 mostra os detalhes de construção de um rolamento rígido de esferas.

Os rolamentos de esferas podem apresentar defeitos na pista externa, interna, na gaiola ou na própria esfera. Normalmente, esses defeitos iniciam com leves marcas e evoluem para falhas aparentes com remoção de massa dos componentes do rolamento [Zhongming and Bin, 2000]. Uma vez que a falha do rolamento começa, também tem início um efeito em cascata que acelera falhas graves no motor.

As falhas em rolamentos podem ser subdivididas em duas categorias, de acordo com [Kiter et al., 2017]. A primeira compreende os defeitos de fabricação, decorrentes de um processo de instalação mal feito, como pistas desalinhadas, pistas excêntricas, rolamentos fora do tamanho ou peças defeituosas. A segunda categoria abrange os defeitos locais, estes ocorrem no caso de uso prolongado do rolamento de esferas, incluindo uma série de danos como trincas, folgas, buracos e arranhões. Pode-se detectar esse tipo de falha com o monitoramento de frequências específicas do rolamento

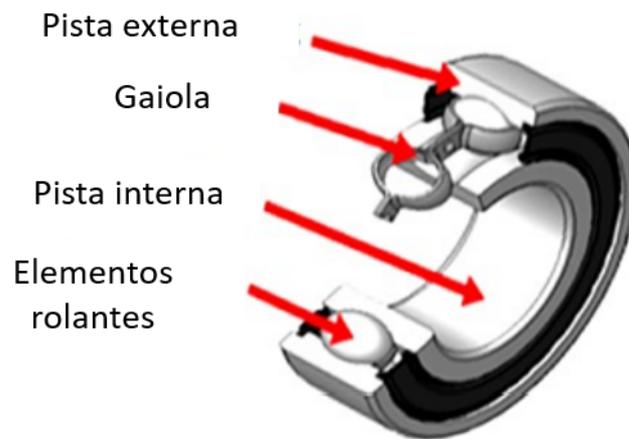


Figura 2.10 – Construção de um rolamento de esferas.

Fonte: Adaptado de [Jigyasu et al., 2018]

através do sinal de vibração.

Segundo Choudhary et al. [2018], as falhas no rolamento podem se dar por problemas de lubrificação no rolamento (falta ou excesso), desalinhamento, excesso de carga no eixo, vibração ou temperatura excessiva.

2.3.1.2 Excentricidade

Conforme explicado anteriormente, o entreferro é o espaço de ar entre o estator e o rotor onde o campo magnético se propaga. Em uma condição ideal, esse espaço é igual em todo o diâmetro do rotor. Quando o rotor não está concêntrico ao estator, ocorrem as falhas de excentricidade.

Nandi et al. [2005] separam a excentricidade em dois tipos: excentricidade estática do entreferro ou excentricidade dinâmica do entreferro. A excentricidade estática ocorre quando a posição do comprimento radial mínimo do entreferro é fixa no espaço porém o rotor não é concêntrico com o estator. Pode ser causada pela ovalização do núcleo do estator ou pelo posicionamento incorreto do rotor ou estator na instalação da máquina. No caso de excentricidade dinâmica, o centro do rotor não está no centro da rotação e a posição de entreferro gira com o rotor. Este desalinhamento pode ser causado devido a vários fatores, como eixo do rotor torto, desgaste ou desalinhamento do rolamento, ressonância mecânica em velocidade crítica, dentre outros. Na Figura 2.11 é mostrada a seção transversal de um motor íntegro, com excentricidade estática e dinâmica. Os principais sintomas de falha de excentricidade são vibrações e ruídos excessivos.

Faiz et al. [2008] realiza análise dinâmica baseada no método dos elementos finitos que identifica excentricidade mista (uma combinação de estática e dinâmica) por análise, sem acesso direto ao motor. O método supera a dificuldade de aplicar método

de elementos finitos ao comportamento transitório. É simulado o espectro da corrente de linha de um motor de linha de produção e comparado com o espectro de um motor íntegro conhecido para detectar excentricidade.

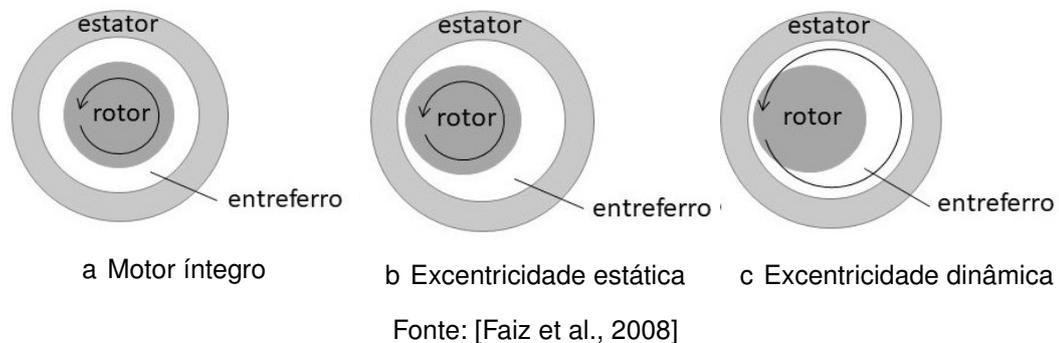


Figura 2.11 – Seção transversal do motor de indução: íntegro, com excentricidade estática e com excentricidade dinâmica.

2.3.1.3 Defeitos relacionados à carga do motor

Falhas relacionadas a carga são de origem mecânica, causadas por um torque não constante no tempo. Esse desequilíbrio no torque pode ser causado por problemas como desbalanceamento e desalinhamento no eixo [Jigyasu et al., 2018]. A seguir estas falhas são tratadas com mais detalhes.

a. Desalinhamento do eixo

Quando há falta de alinhamento entre o eixo do motor e a carga fixada ao eixo ocorre a condição chamada de desalinhamento do eixo. Esta falha pode ocorrer quando o motor ou carga mecânica é fixado em uma base desnivelada. A identificação do problema pode ser feita através da assinatura em frequência da corrente do estator e da vibração do motor nas harmônicas de rotação mecânica. No sinal da vibração, as harmônicas pronunciadas pela falha de desalinhamento são as de $2x$ a $4x$ a frequência de rotação mecânica.

O desalinhamento é dividido em três categorias principais:

- Desalinhamento do eixo paralelo: ocorre quando o eixo do motor está paralelo ao eixo de carga mecânica, e não em uma linha reta.
- Desalinhamento angular do eixo: ocorre porque existe um ângulo entre o eixo do motor e o eixo de carga mecânica.
- Desalinhamento composto: quando ambos o desalinhamento paralelo e angular estão presentes.

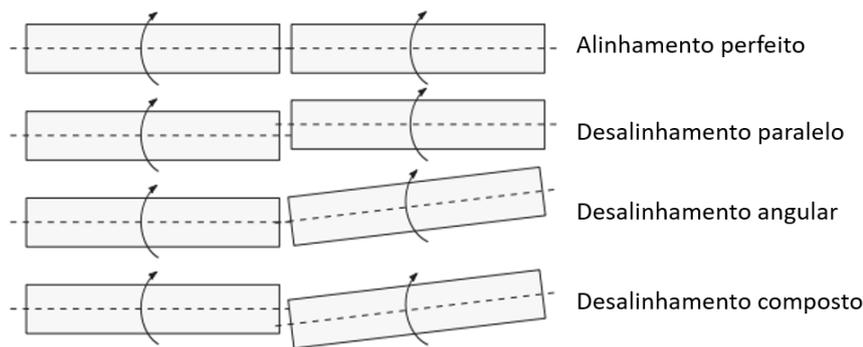


Figura 2.12 – Representação do alinhamento e dos três tipos de desalinhamento em vista 2D.

Fonte: Adaptado de Corne et al. [2017]

Na Figura 2.12 são representados os três tipos de desalinhamentos mencionados e o alinhamento ideal.

O desalinhamento não é o causador direto de uma falha que força a parada da máquina, porém esse acoplamento desalinhado faz com que os rolamentos do motor sejam submetidos a uma carga dinâmica e/ou estática que pode fazer esses componentes falharem prematuramente. [Corne et al., 2017].

Os principais sintomas de um problema de desalinhamento são ruídos excessivos, sobreaquecimento, consumo elevado de energia, excesso de vibração e baixo rendimento do motor. Algumas formas indicadas para evitar estes problemas são o uso de acoplamentos flexíveis sempre que possível, o alinhamento de eixos a laser e o monitoramento da temperatura e vibração de trabalho do motor [Fluke].

b. Desbalanceamento do eixo

Quando há uma distribuição desigual de massa no rotor, uma vibração anormal é produzida pela interação desta massa com a aceleração radial devido à rotação, gerando assim uma força centrífuga que transmite a vibração aos mancais do rotor. Neste caso, qualquer ponto do mancal experimentará esta força uma vez por revolução [MacCamhaoil, 2016].

No sinal de frequência, o desbalanceamento aparece em 1x a frequência da rotação mecânica do motor, fazendo com que essa frequência apresente amplitude maior que o normal [Jalan and Mohanty, 2009]. O desbalanceamento do rotor pode ser classificado como desbalanceamento estático ou acoplado, conforme ilustrado na Figura 2.13.

MacCamhaoil [2016] fornece as seguintes definições para cada tipo de desbalanceamento. Desbalanceamento estático é uma excentricidade do centro de

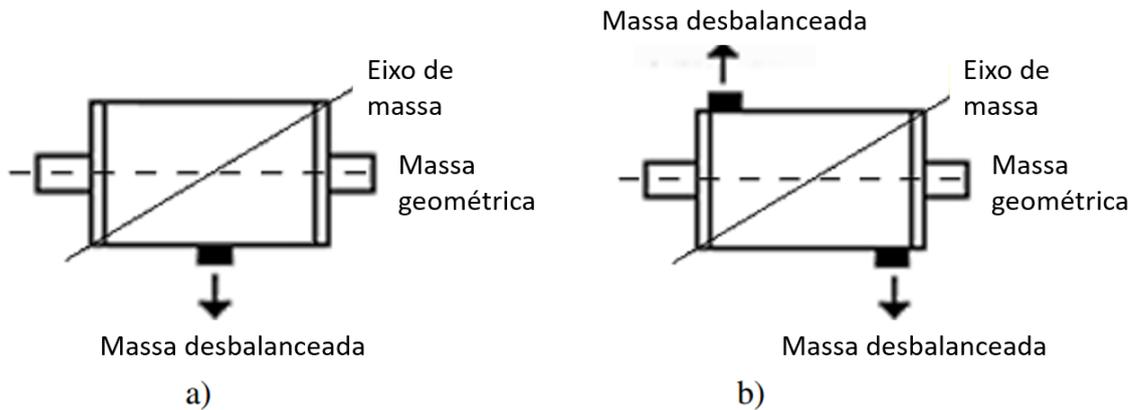


Figura 2.13 – Motor com desbalanceamento estático (a) ou desbalanceamento acoplado (b).

Fonte: Adaptado de Corne et al. [2017]

gravidade de um rotor, causada por uma massa pontual a um determinado raio do centro de rotação. Pode ser corrigido através de uma massa de mesmo tamanho no lado oposto. O desbalanceamento acoplado é encontrado quando existem duas massas iguais colocadas simetricamente em relação ao centro de gravidade, mas posicionadas a 180° uma da outra. O rotor está em equilíbrio estático mas quando o rotor gira, as duas massas provocam um deslocamento do eixo de inércia, fazendo com que ele não fique mais alinhado com o eixo de rotação, levando a fortes vibrações nos rolamentos. A correção do desbalanceamento é feita com a adição de massa de correção em dois planos através de medições de vibração com o rotor girando.

MacCamhaoil [2016] ainda introduz um terceiro desbalanceamento, chamado de desbalanceamento dinâmico. Este é uma combinação de desbalanceamento estático e acoplado. Para corrigir o desbalanceamento dinâmico, é necessário fazer medições de vibração com a máquina em funcionamento e adicionar massas de balanceamento em dois planos. O desbalanceamento do eixo por um acoplamento inadequado é um exemplo clássico de excentricidade dinâmica [Corne et al., 2017]. O desbalanceamento em uma máquina rotativa pode fazer com que peças se desgastem mais rapidamente, levando a falhas mais graves e causando paradas nas máquinas. Para diagnosticar falhas de desbalanceamento, é tradicionalmente utilizada a assinatura da vibração no espectro de frequência [Saleem et al., 2012].

2.4 TÉCNICAS DE MONITORAMENTO E DETECÇÃO DE FALHAS

O monitoramento contínuo de condição de máquinas industriais tem se tornado uma necessidade para garantir maior confiabilidade nos processos. Com o avanço dos métodos de medição e processamento de sinais, o monitoramento de condição tem se tornado cada vez mais simples e eficaz.

Akbar et al. [2023] aborda as técnicas mais convencionais empregadas em detecção de falhas em motores elétricos de indução. As principais delas são a análise da assinatura de corrente do motor, análise de assinaturas de tensão, análise de parâmetros como vibração, temperatura e fluxo magnético e diagnósticos envolvendo sobretensões e descargas parciais. Os sensores de monitoramento de condição empregados nestes casos são:

- a. Sensores de vibração, como acelerômetros e sensores de proximidade;
- b. Equipamento de detecção de temperatura, incluindo termopares e detectores de resistência e temperatura;
- c. Detectores de corrente, incluindo sensores de efeito Hall e transformadores de corrente;
- d. Microfones e sensores ultrassônicos, incluídos na categoria sensores óticos;
- e. Monitoramento baseado em níveis de emissão.

Com os dados medidos pelos sensores, podem ser realizadas: análises no domínio do tempo, usando valores característicos para determinar mudanças por configuração de tendência; análises de espectro para determinar tendências de frequências e relações de amplitude e fase são usadas como ferramentas de avaliação [El Hachemi Benbouzid, 2000].

O monitoramento da condição pode ser realizado quando o equipamento está em operação, o que é conhecido como *online*, ou quando está *offline*, ou seja, quando está parado e fora de operação. Este monitoramento deve ser feito continuamente ou periodicamente. A Figura 2.14 mostra um processo típico de monitoramento *online*, que consiste em um conjunto de sensores que fornece o sinal de falha para uma unidade de processamento de sinal, que posteriormente envia seu resultado para ser analisado por sistemas especialistas, onde a falha correspondente é finalmente detectada [Bhowmik, 2013].

2.4.1 Diagnóstico de falhas pelos sinais de vibração mecânica

Um motor em sua condição normal de operação tem níveis baixos de vibração. Quando uma falha interna ocorre, há aumento da vibração do motor e esse parâmetro

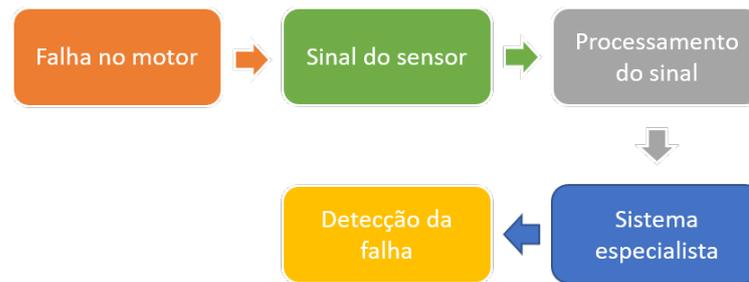


Figura 2.14 – Monitoramento de falha online de um motor de indução.

Fonte: Adaptado de [Bhowmik, 2013]

é útil para detecção do tipo de falha que está ocorrendo. Os sinais podem ser medidos em termos de velocidade, aceleração ou deslocamento, tanto em valores relativos como absolutos [Ahmed et al., 2011].

A análise da vibração assume um papel importante no campo industrial devido à sua facilidade de uso e tem capacidade para detectar a maioria das falhas mecânicas e algumas falhas elétricas [Kiter et al., 2017]. Falhas de rolamentos, excentricidades do rotor, falhas de engrenagens e rotores desbalanceados são as melhores candidatas para diagnósticos baseados em vibração. O monitoramento de vibração de máquinas elétricas é realizado através do uso de análise de banda larga, banda estreita ou análise espectral (assinatura) da vibração medida da máquina [MEHALA, 2010].

Segundo Jayaswal et al. [2008], o monitoramento de vibração online pode ser feito das três formas a seguir:

- Medições periódicas de campo com instrumentos portáteis: este método fornece informações sobre mudanças de longo prazo na condição da planta. Os instrumentos são operados por analistas com conhecimento nos equipamentos e costumam ter alto valor agregado;
- Monitoramento contínuo com instrumentos permanentemente instalados: é empregado quando falhas de máquinas ocorrem rapidamente e quando os equipamentos são mais críticos;
- Análise de assinaturas: coleta de informações, sinais ou assinaturas, diagnóstico e detecção de falhas por meio de uma análise desses dados, e o julgamento da gravidade das falhas para tomada de decisão é chamado de análise de assinatura. A técnica envolve o uso de instrumentação eletrônica especialmente projetada para este fim.

Para medir a vibração da máquina, o transdutor de aceleração (acelerômetro) é um dos tipos de sensores mais utilizados atualmente. Este é denominado como acelerômetro, devido ao sinal de saída proporcional à aceleração. Além disso, os

princípios de seu funcionamento dizem respeito à segunda lei de Newton $F = m * a$ [Silva and Cardoso, 2005].

A análise do sinal de vibração pode ser feita no domínio do tempo ou da frequência. Quando a análise do sinal é realizada no domínio do tempo, a forma mais simples de análise é obtida através do cálculo do valor RMS (do inglês, *root mean square*) de vibração [Kiter et al., 2017]. Além do RMS, os recursos de domínio do tempo usados com mais frequência incluem a análise da curtose, fator de crista, média e variância, etc., que são métodos estatísticos para avaliação de uma forma de onda de um sinal.

No domínio da frequência, a análise baseia-se na Transformada Rápida de Fourier (FFT, do inglês *Fast Fourier Transformer*), a técnica mais utilizada para monitoramento de condição *online*. Com isso, a partir do conhecimento dos parâmetros do motor, frequências de muitos sinais de falhas mecânicas e elétricas são determinadas [Jigyasu et al., 2018].

O monitoramento de condições baseado em vibração atraiu a atenção de muitos pesquisadores e ganhou aceitação industrial. R. Cardinali [1992] apresenta um modelo para diagnóstico de falhas baseado na simulação de vibração de máquinas aplicadas em hidroelétricas. Com base no comportamento dinâmico da máquina, a detecção de falhas é feita por comparação de medições indiretas de forças externas. A investigação experimental é realizada em uma bancada de testes e os resultados obtidos são comparados com os simulados para comprovar a validade do modelo proposto.

Renwick and Babson [1985] demonstram que a manutenção preditiva usando análise de vibração alcançou resultados significativos no diagnóstico de problemas de máquinas. São reduzidos o tempo de inatividade do maquinário e perdas de produção, assim como outros benefícios de custo mais sutis a longo prazo, que podem resultar de um cronograma de manutenção preciso.

Diversas falhas mecânicas podem ser identificadas utilizando o sinal de vibração no domínio da frequência, Singh and Ahmed Saleh Al Kazzaz [2003] apresentam um compilado de tipos de falhas e como elas são identificadas. A detecção de falha de desalinhamento deve considerar o monitoramento das 3 primeiras múltiplas da frequência da rotação mecânica enquanto o desbalanceamento se manifesta em 1x a frequência da rotação mecânica.

3 APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE FALHAS EM MÁQUINAS ELÉTRICAS

O intuito do presente trabalho é a identificação de falhas de desbalanceamento e desalinhamento em motores elétricos, de forma online, utilizando Aprendizado de Máquina (AM) embarcado em um microcontrolador. Antes disso, é preciso conhecer os conceitos relacionados à AM de uma maneira geral e as limitações encontradas ao embarcar modelos de AM em sistemas microcontrolados. O presente capítulo traz uma breve revisão dos conceitos básicos relacionados a estes temas.

3.1 TIPOS DE APRENDIZADO DE MÁQUINA

O Aprendizado de Máquina (AM) é uma área da Inteligência Artificial na qual modelos são criados a partir de dados. Usualmente, o objetivo é utilizar dados disponíveis para criar um modelo matemático capaz de inferir possíveis saídas para novos dados de entrada [Grus, 2019].

Os modelos de AM podem estar associados a diversos tipos de tarefas, sendo normalmente tarefas de classificação ou regressão. A diferença entre as duas é que na classificação o modelo deve classificar as amostras com rótulos de uma categoria ou classe para uma dada entrada, já na regressão o objetivo é prever um valor contínuo [Bishop, 2006]. Smola and Vishwanathan [2008] trazem alguns exemplos de aplicações como o ranqueamento de páginas web, tradução automática de documentos para diversas linguagens e reconhecimento facial baseado em fotos ou vídeos.

Bishop [2006] subdivide os tipos de aprendizado em três principais, mostrados na Figura 3.1. O que define o tipo de aprendizado a ser empregado são os dados disponíveis para treinamento. O aprendizado supervisionado é utilizado quando os dados de treinamento contém exemplos de valores de entrada e da saída esperada. Quando os dados de treinamento não possuem rótulos é aplicado o aprendizado não supervisionado. O aprendizado não supervisionado pode ser aplicado, por exemplo, para fazer a separação dos dados de entrada em grupos com características similares. Por fim, o aprendizado por reforço é uma abordagem em que o modelo interage com o ambiente, tomando alguma ação e recebendo uma recompensa quando acerta ou uma penalidade quando erra.

Durante a criação do modelo de AM são normalmente utilizados dados de treinamento, para ajustar os parâmetros do modelo. Esse procedimento é conhecido como fase de treinamento ou de aprendizado. Em problemas de classificação, com o modelo treinado, podem ser utilizados dados novos, chamados de dados de teste para avaliar o modelo final, sendo esta fase conhecida como fase de teste [Bishop, 2006]. A seleção do algoritmo de AM a ser utilizado é feita com base no problema que se deseja resolver, no tipo de dados de entrada, número de variáveis, e assim por diante

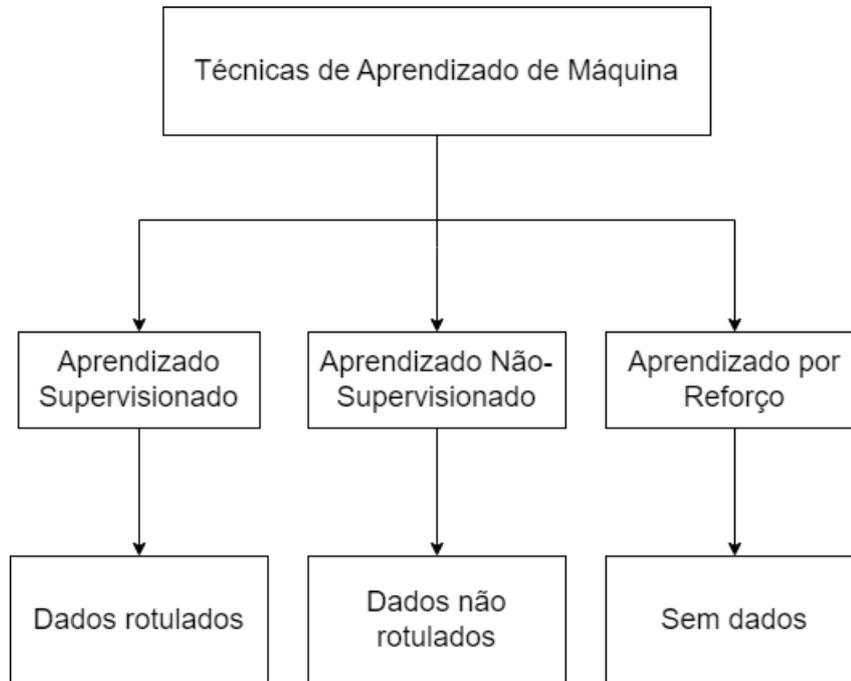


Figura 3.1 – Principais técnicas de aprendizado e seus respectivos dados esperados.

Fonte: Adaptado de [Mohammed et al., 2016]

[Mahesh, 2019].

Durante a fase de treinamento, é criado um modelo de aprendizado utilizando os dados de treinamento. É importante que o modelo final seja capaz de generalizar, pois espera-se que tenha um bom desempenho com novas amostras não vistas durante o processo de treinamento [Bonaccorso, 2017].

Na Figura 3.2 é mostrada uma representação do fluxo de criação de um modelo de aprendizado supervisionado, onde é treinado um modelo preditivo utilizando dados rotulados que, em seguida, é aplicado em novos dados. O objetivo é que o rótulo fornecido pelo modelo seja o esperado para a nova amostra.

Alguns exemplos de algoritmos de aprendizado supervisionado presentes na literatura são as Árvores de Decisão, Random Forests, K-Nearest Neighbors (KNN) e Support Vector Machines (SVM).

Já no aprendizado não supervisionado, os dados de entrada para criação dos modelos de AM são não rotulados. Nesse caso, existem as variáveis de entrada mas não existem as variáveis de saída esperadas para tais entradas. Han et al. [2011] define o aprendizado não supervisionado como um processo de agrupamento de dados por similaridade, também chamado de "clusterização". Como não há rótulo nos dados de entrada, apesar de separar em "K" grupos similares os dados, não são atribuídos rótulos à essas classes encontradas. Este tipo de aprendizado também é aplicado para redução de dimensionalidade. As técnicas de aprendizado não supervisionado são mais comumente encontradas em aplicações reais, pois não dependem de que os dados sejam manualmente rotulados [Murphy, 2013]

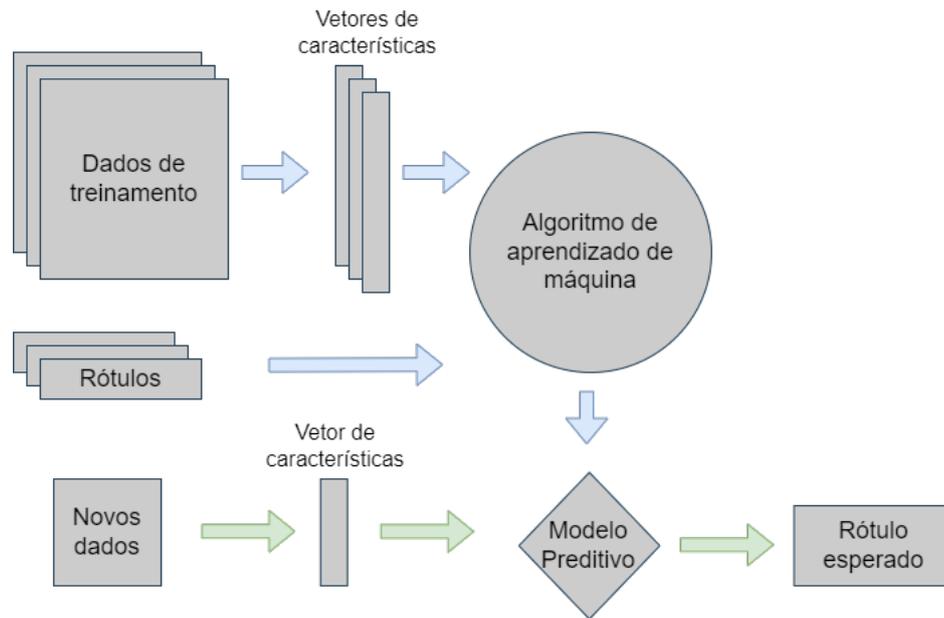


Figura 3.2 – Fluxo para criação de modelo de aprendizado supervisionado.

Fonte: Adaptado de [Nasteski, 2017]

Dentre os algoritmos de aprendizado não supervisionado estão o K-means, Gaussian Mixture Model e Principal Component Analysis.

3.2 ALGORITMOS DE APRENDIZADO DE MÁQUINA

Nesta seção, serão discutidos os métodos de aprendizado de máquina (AM) empregados neste estudo para identificação de falhas em motores elétricos.

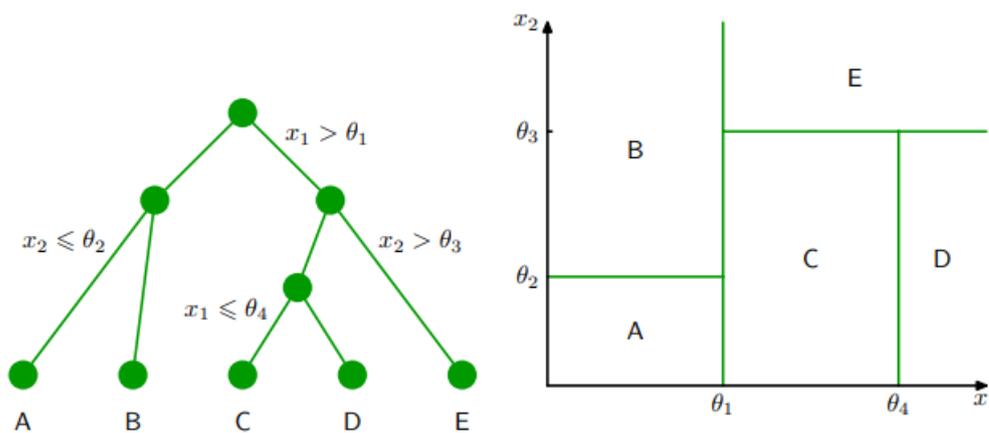
3.2.1 Árvore de decisão

A árvore de decisão é um algoritmo de aprendizado supervisionado que pode ser usado para classificação ou regressão [Dey, 2016]. Está entre os métodos de aprendizado mais populares e tem bons resultados para inferência de várias tarefas de aprendizado, como por exemplo, diagnósticos médicos ou análise de crédito para novos empréstimos [Mitchell, 1997].

As árvores de decisão são modelos que particionam os dados de entrada em regiões distintas do espaço de características, baseando-se no número de classes possíveis. O processo de seleção do modelo para uma nova amostra pode ser descrito como uma série de decisões sequenciais, semelhantes à estrutura de uma árvore binária que se divide em dois ramos em cada nó [Bishop, 2006]. Para todos os nós da árvore, um atributo específico dos dados é testado, e, dependendo do resultado desse teste, a árvore se ramifica em nós descendentes que representam os possíveis valores desse atributo [Mitchell, 1997].

Para melhor entendimento, é mostrado o exemplo apresentado por Bishop [2006], que descreve o algoritmo de árvore de decisão conhecido como CART [Breiman and Stone, 1984]. Além do CART, são encontrados na literatura outros algoritmos como o ID3 [Quinlan, 1986] e C4.5 [Quinlan, 1993].

A Figura 3.3 representa uma árvore de decisão e a sua divisão no espaço para um conjunto de dados de duas dimensões. Nas imagens, x_1 e x_2 representam as variáveis de entrada e θ_{1-4} os parâmetros do modelo criado. O espaço é dividido em 5 partições, de A a E, pois são 5 os possíveis valores de saída da árvore de decisão. Na ramificação da árvore, o primeiro passo da decisão, no nó chamado de "raiz", é testada se a variável x_1 da nova amostra de dados é maior, menor ou igual ao parâmetro θ_1 . A partir do resultado do primeiro passo, a amostra passa para o nó subsequente, onde a variável x_2 poderá ser comparada com os parâmetros θ_2 ou θ_3 . Caso a amostra for maior que θ_1 , por exemplo, o próximo passo será a última ramificação e o valor de saída será escolhido entre as classes A e B.



a Árvore binária de um conjunto de dados de duas dimensões

b Partição de dados no espaço de duas dimensões em cinco regiões distintas

Figura 3.3 – Representações da divisão da árvore de decisão

Fonte: [Bishop, 2006]

Todos os testes internos da árvore devem ter duas ou mais saídas. Se todos os testes tiverem duas saídas apenas, como o exemplo mostrado acima, a árvore é considerada uma árvore de decisão binária. Além disso, devem haver pelo menos duas classes nos dados de entrada [Nilsson, 1998].

Definir a estrutura da árvore de decisão é uma tarefa desafiadora, mesmo que seja definido um número fixo de nós para a árvore ainda há diversas possibilidades de variáveis e valores de comparação para cada nó [Bishop, 2006]. A definição de qual atributo pertence a cada nó pode ser feita utilizando o conceito de impureza. Idealmente, a impureza deve ser nula em um nó subsequente à divisão, o que significa que os próximos nós serão afetados somente pelos próximos atributos e não pelo atributo de decisão do nó atual [Bonaccorso, 2017].

Uma alternativa utilizada para medir a impureza em um conjunto de treinamento é a entropia. A entropia define a importância de um atributo na classificação dos dados através do ganho de informação gerado pela divisão dos dados de treinamento por cada atributo [Mitchell, 1997]. A função utilizada para cálculo do ganho é geralmente um parâmetro do modelo, para o CART, por exemplo, é utilizado o índice de Gini [Shalev-Shwartz and Ben-David, 2014]. O índice de impureza de Gini mede a probabilidade de um erro de classificação se um rótulo aleatório for escolhido, com base na distribuição das classes nos dados de treinamento. O índice varia de 0 a 1, sendo 0 o máximo de pureza, que é atingido quando todos os dados pertencem a mesma classe. [Bonaccorso, 2017].

Algumas das principais vantagens da utilização da árvore de decisão são a facilidade de interpretação quando comparado a outros métodos e possibilidade de manipulação de dados binários e não binários [Ray, 2019].

3.2.2 Support Vector Machine

Support Vector Machine (SVM) é uma técnica de aprendizado supervisionado que pode ser aplicada a tarefas de classificação [Dey, 2016]. Este método é muito aplicado pois pode trabalhar com dados que são ou não linearmente separáveis, tendo um bom desempenho na prática [Bonaccorso, 2017].

Para melhor entender os conceitos do SVM, considera-se um conjunto de dados que contém duas classes e é linearmente separável. A abordagem do SVM trata esse problema através do conceito de margem, que é a distância entre um hiperplano que divide as classes no espaço e os pontos mais próximos de cada classe. O intuito do SVM é ter a maior margem entre as amostras de cada classe e para isso são usados os vetores de suporte, para definir a melhor posição do hiperplano que gera a maior margem possível na separação [Bishop, 2006]. Um hiperplano é considerado de margem máxima ou ótimo se é capaz de separar um conjunto de vetores sem erro e com máxima distância entre vetores de classes opostas [Vapnik, 1999].

Uma exemplificação do caso de SVM com separação linear é mostrada na Figura 3.4, onde é criado um hiperplano linear de separação das classes de triângulos e quadrados. Os vetores de suporte são criados a partir dos pontos de cada classe destacados em preto, que são os pontos mais próximos do hiperplano.

O caso apresentado acima é linearmente separável, porém a maioria dos problemas reais não são. Para isso, foi introduzido por Boser et al. [1992] o conceito de kernel ao SVM. O kernel é uma função matemática não linear que transforma os vetores de suporte projetando-os para um espaço de maior dimensão onde esses vetores podem ser linearmente separáveis [Bonaccorso, 2017]. Dois kernels comumente utilizados são o kernel polinomial e o Gaussiano [Schölkopf et al., 2002]. Além desses, pode ser empregada uma função de base radial (RBF), por exemplo [IH., 2021]. Uma forma

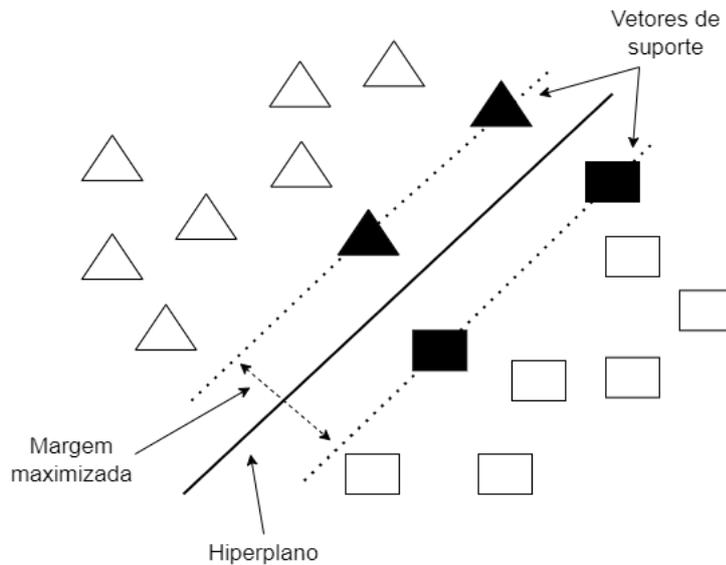


Figura 3.4 – Separação linear usando SVM para dados de duas dimensões.

Fonte: Adaptado de [Meyer, 2015]

utilizada para definir o melhor kernel para um conjunto de dados é definir alguns potenciais kernels e utilizar técnica de validação cruzada para definir o que melhor separa os dados. Isso implica em um maior custo de treinamento [Maglogiannis, 2007].

Bishop [2006] apresenta um exemplo de dados não separáveis linearmente em um espaço de duas dimensões. Na Figura 3.5 pode ser observado o resultado da separação das classes e a linha mais escura que representa o hiperplano criado por um kernel Gaussiano, no espaço transformado.

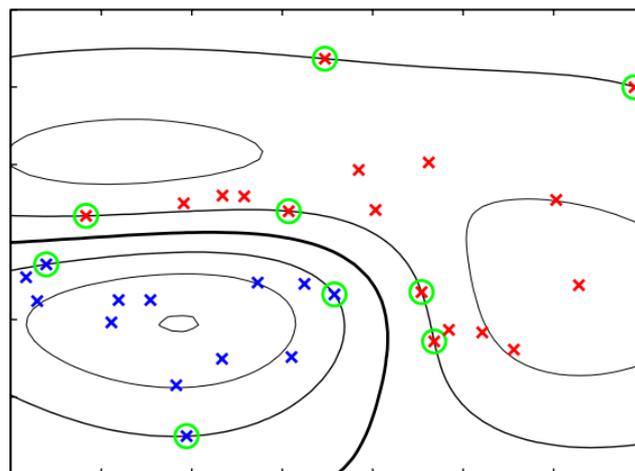


Figura 3.5 – Separação não linear de dados de duas dimensões utilizando uma função Gaussiana como kernel

Fonte: [Bishop, 2006]

Em sua formulação original, os SVMs foram concebidos para a solução de problemas de classificação binária. Para problemas de classificação multiclasse, duas técnicas principais são empregadas, sendo elas a One-vs-Rest e One-vs-One. Na

estratégia One-vs-Rest, o problema multiclasse é decomposto em vários problemas binários. Para cada classe k no conjunto de classes, um classificador SVM é treinado para fazer a distinção entre classes. Se houver K classes, serão treinados K classificadores. Para prever a classe de uma nova amostra, todos os classificadores são aplicados e a classe correspondente ao classificador que dá a resposta com a maior margem é escolhido. Esta técnica é abordada no estudo feito por Lorena and de Carvalho [2008]. Na estratégia One-vs-One, um classificador SVM é treinado para cada par de classes. Para um conjunto de K classes, isso resulta em $\frac{K(K-1)}{2}$ classificadores. Na classificação das novas amostras, cada classificador vota em uma das duas classes que ele distingue. Neste caso, a classe que recebe o maior número de votos é selecionada [Bishop, 2006].

3.2.3 One Class SVM

O método de One Class SVM consiste em uma adaptação do SVM para problemas de aprendizado não supervisionado, onde apenas uma classe de dados é utilizada para treinamento dos modelos. O conceito foi introduzido por Schölkopf et al. [2001] e propõe um algoritmo que calcula uma função binária que define áreas onde a maioria dos dados se encontra, de modo que a função é diferente de zero nessas regiões. Esse método é muito aplicado a problemas de detecção de anomalias, onde o conjunto de dados de treinamento possui apenas os dados normais e o algoritmo identifica novos dados que não estão dentro dessa definição de "normalidade".

A Figura 3.6 representa a função criada para representar o conjunto de dados de entrada numa classificação de uma classe. Considerando que o conjunto em verde corresponde aos dados de entrada do método de One Class SVM, a linha que circunda os dados é a função que os representa no espaço. Um novo dado dentro do conjunto em verde, pelo método proposto por Schölkopf, é classificado como "+1", ou seja, pertencente à classe. Já o ponto em vermelho na imagem é classificado como "-1", ou seja, não pertencente à classe.

Um exemplo de aplicação de algoritmos de classificação de uma classe é na detecção de falhas. A obtenção de dados da condição normal da máquina pode ser feita através de medições da condição atual de operação, já dados de falha normalmente não são conhecidos [Khan and Madden, 2010]. A geração destes dados de falha pode envolver altos custos, pois seria necessário a parada da máquina e a degradação proposital dos componentes que a compõe, sendo então esta técnica uma possibilidade melhor aceita para diagnóstico de falhas em aplicações reais. Os métodos de classificação de uma classe também são utilizados em aplicações de visão computacional, para detecção de anomalias em imagens e na área de biometria como Autenticação Ativa e anti-falsificação [Perera et al., 2021].

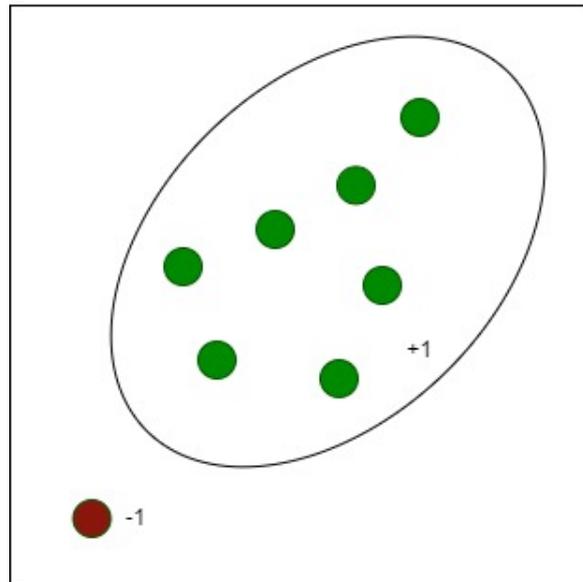


Figura 3.6 – Separação do espaço de dados na classificação de uma classe

Fonte: Adaptado de [Perera et al., 2021]

3.3 APRENDIZADO DE MÁQUINA NO ÂMBITO DE DETECÇÃO DE FALHAS

Conforme apresentado anteriormente, o contexto de detecção de falhas em máquinas elétricas tem forte relação com a manutenção preditiva de plantas industriais. O processo de diagnóstico de falhas envolve o monitoramento, diagnóstico e alerta precoce do status e falhas de equipamentos [Lin, 2021]. A detecção de defeitos em máquinas de forma precoce, permite que manutenções imediatas sejam feitas, economizando tempo e dinheiro em reparos de falhas [Yang et al., 2021]. Para isso, sensores e algoritmos inteligentes são essenciais.

Alguns conceitos sobre sistemas de manutenção preditiva inteligente utilizando dados são introduzidos em Yam et al. [2001]. Uma abordagem que se tornou importante nos últimos anos dentro da manutenção preditiva é a orientada a dados, onde através de dados históricos coletados das máquinas são criados modelos de AM que representam o comportamento do sistema [Paolanti et al., 2018a]. Em Shin and Jun [2015] são discutidas várias técnicas que podem ser aplicadas na manutenção preditiva, para processamento de dados, diagnósticos e prognósticos. Através dos modelos de AM é simulada a análise de um especialista em manutenção, trazendo informações importantes sobre o diagnóstico da máquina [Lin, 2021].

Abordagens de processamento de sinal combinadas com algoritmos de classificação (Support Vector Machines, árvores de decisão, KNN, entre outras) são frequentemente usadas na detecção de falhas para categorizar e identificar defeitos. O tratamento a ser empregado no sinal depende do tipo de falha que está sendo analisado, e geralmente envolve análises no domínio de tempo e frequência, análise

estatística, decomposição de Wavelet, entre outros métodos [Yang et al., 2021]. Na literatura são encontrados diversos trabalhos voltados a detecção inteligente de falhas em motores elétricos utilizando AM, uma breve revisão é apresentada a seguir.

Para detecção de falhas em rolamentos, Song et al. [2018] propõe um novo método de extração de características de sinal e diagnóstico de falhas para máquinas de baixa velocidade. São empregados filtros estatísticos e a transformada de Wavelet, combinados com um método de detecção de pico para extrair características de um sinal de falha no domínio da frequência. Já Soualhi et al. [2015] apresenta uma abordagem que combina a transformada de Hilbert-Huang com os algoritmos de Support Vector Machine (SVM) e Support Vector Regression (SVR) para o monitoramento de rolamentos de esferas. Delgado-Prieto et al. [2013] e Vaimann et al. [2018] utilizam Redes Neurais Artificiais (RNAs) para identificar falhas em rolamentos através dos sinais de vibração e som, respectivamente.

Com relação à falhas elétricas em motores de indução, Sadeghian et al. [2009] propuseram uma RNA junto com a decomposição de Wavelet para detecção de barras quebradas no rotor utilizando a corrente do estator. Os resultados experimentais mostram que o método proposto é capaz de detectar os casos de falha com alta precisão.

Conforme apresentado, o foco deste trabalho é detecção de falhas de desalinhamento e desbalanceamento. Existem algumas abordagens para esse fim na literatura, como em Mey et al. [2020], onde é apresentado um conjunto de dados de falha de desbalanceamento e são utilizadas RNAs e Random Forest, para testar o desempenho destes algoritmos no conjunto de dados coletado. A melhor acurácia para os dados de teste foi encontrada utilizando as redes neurais treinadas com dados do acelerômetro, no domínio da frequência.

Gangsar et al. [2021] fazem a detecção de falhas de desbalanceamento utilizando SVMs. Os autores utilizam um eixo e disco acoplados ao eixo de um motor como experimento para extrair os dados íntegros e com falha. O desbalanceamento é simulado adicionando massa ao disco. As variáveis utilizadas na base de dados são o desvio padrão, assimetria e curtose do sinal. Com a base de dados coletada, é utilizado um modelo de SVM para identificar no mesmo ensaio as condições do eixo. O trabalho apresenta a comparação entre o uso das variáveis calculadas a partir do sinal no domínio tempo e na frequência, mostrando que foi possível identificar as falhas com ambas. Os melhores resultados foram obtidos com as variáveis no domínio do tempo.

Em Moosavian et al. [2014] é desenvolvida uma abordagem para detecção de falha de desbalanceamento em máquinas rotativas. O ensaio é feito com as condições sem carga, com carga balanceada e com carga desbalanceada. Os dados são adquiridos e utilizados no domínio da frequência, através de 29 variáveis extraídas desses sinais. São utilizados os classificadores kNN e SVM e a conclusão do autor é que o kNN é mais rápido que o SVM e se mostrou com maior acurácia para o caso em

questão.

Para detecção de desalinhamento, Lee et al. [2021] utilizam o método de SVM para identificar os defeitos de desalinhamento no eixo de máquinas rotativas utilizando os dados de vibração em condições normais e anormais. Os dados são coletados, convertidos para o domínio da frequência e pré-processados utilizando PCA para extração de características e redução de dimensionalidade. Os dados pré-processados são então utilizados para treinar o algoritmo do SVM e classificar os dados com falha e sem falha de desalinhamento. Os autores fazem a comparação da utilização dos dados no domínio do tempo e da frequência e mostram que ao utilizar PCA obtiveram melhores resultados.

Para a análise de detecção de desalinhamento e desbalanceamento combinados, Martins et al. [2021] propõem um sistema supervisor para identificação de falhas de desbalanceamento e desalinhamento combinadas. Os classificadores são treinados com dados normais, com desbalanceamento, desbalanceamento associado com desalinhamento horizontal e desbalanceamento associado com desalinhamento vertical. Foram comparados os algoritmos de aprendizado supervisionado SVM, kNN e Random Forest. Os autores utilizaram validação cruzada para separar os dados de treino e teste e avaliar os modelos criados. A maior acurácia média, quando comparando todas as classes de falhas, foi encontrada utilizando o Random Forest, que chegou a 81.41% de acurácia na classificação.

Em Shin et al. [2005] é abordada a utilização do algoritmo One-Class SVM na detecção e classificação de falhas em máquinas. O estudo mostra que o One-Class SVM é eficaz na identificação de anomalias, especialmente em cenários onde os dados normais são abundantes, mas os dados de falhas são raros. A abordagem demonstrou uma alta taxa de precisão na detecção de falhas, destacando-se como uma ferramenta robusta para manutenção preditiva em ambientes industriais. No trabalho ainda é explorada a seleção dos parâmetros do modelo, como o kernel, γ e ν através de validação cruzada.

Martínez-Rego et al. [2016] também aplicam a classificação de uma classe para detecção de falhas, neste trabalho focando em problemas de rolamento, utilizando o sinal de vibração no domínio do tempo. O método utilizado demonstrou bons resultados e alta precisão na detecção das falhas.

A proposta do presente trabalho é similar ao proposto nos trabalhos citados acima porém, além da detecção de falhas de desbalanceamento e desalinhamento através do sinal de vibração, os algoritmos devem ser executados dentro de um dispositivo com restrição de recursos, algo não mencionado em nenhum dos trabalhos apresentados nesta seção. Uma revisão sobre os avanços tecnológicos para execução de algoritmos embarcados será visto na seção a seguir.

3.4 APRENDIZADO DE MÁQUINA EMBARCADO EM MICROCONTROLADORES

Com a evolução dos sistemas voltados à Internet das Coisas (IoT), utilizar algoritmos de AM nos dispositivos de borda tem se tornado cada vez mais relevante, gerando necessidades de melhora de desempenho e desenvolvimento de novas aplicações na área de processamento de dados [Lin et al., 2019].

Os principais motivadores dessa migração são a crescente preocupação com a privacidade dos dados, alto consumo de energia pelo envio de dados para a nuvem e execução dos algoritmos, latência de processamento e dependência da rede de internet [Ray, 2022]. Tendo em vista esses problemas, as principais vantagens da mudança de computação da nuvem para a borda, segundo Ren et al. [2021], são:

- Privacidade - Com o processamento na borda, os dados são coletados e tratados localmente, evitando problemas de violação em políticas de privacidade ou vulnerabilidade à interceptação.
- Menor latência - Os dispositivos que coletam os dados são capazes de executar os algoritmos e tomar decisões sem necessidade de comunicação externa, eliminando a dependência de conexão com a internet.
- Eficiência energética - Considerando toda a solução IoT, a transmissão de dados é uma das partes mais custosas em termos de energia, isso é eliminado através da execução local dos algoritmos.

Um dos grandes incentivadores dessa tecnologia é uma comunidade sem fins lucrativos conhecida como tinyML Foundation¹. Esta comunidade possui apoio de grandes empresas como ARM, Google e Qualcomm e promove eventos para discussões técnicas e pesquisas na área de aprendizado de máquina embarcado.

De acordo com essa comunidade, o tiny Machine Learning (tinyML) engloba hardwares, algoritmos e softwares capazes de realizar análises de dados em dispositivos com restrições de recursos de memória e energia. Quando aplicados à sistemas IoT por exemplo, possuem sensores capazes de medir variáveis do ambiente, informando sistemas que, dependendo da disponibilidade, podem agir com base nos resultados dos algoritmos de AM [Iodice and Naughton, 2022].

De acordo com Ray [2022], as principais restrições que atualmente dificultam o crescimento do tinyML são:

- Energia: dispositivos IoT normalmente possuem pouca energia disponível para manter o sistema operando. Fornecer a energia necessária para tarefas de aprendizado de máquina é um gasto a mais que pode causar bastante impacto;

¹ <https://www.tinyml.org/>

- Capacidade do processador: a maioria dos dispositivos de borda são baseados em microcontroladores, que são dispositivos de baixo poder de processamento;
- Memória: normalmente os microcontroladores possuem pouca memória de armazenamento, o que pode impedir que os modelos sejam guardados; e
- Custo: muitas vezes o custo para implantação em massa destas soluções é uma dificuldade encontrada.

Na Figura 3.7 é mostrado o fluxograma típico de um sistema tinyML. As partes já bem difundidas de um sistema de aprendizado de máquina são executadas em um dispositivo de maior capacidade de recursos, que compreendem as fases de seleção do algoritmo, treinamento do modelo com base nos dados previamente coletados e otimização do modelo gerado. A partir deste ponto, é iniciado o processo de envio do modelo para o microcontrolador, onde tem início a fase de aprendizado embarcado. Dentro do microcontrolador, que é um dispositivo com recursos limitados, ocorrem as fases de coleta de novos dados e inferência utilizando o modelo portado.

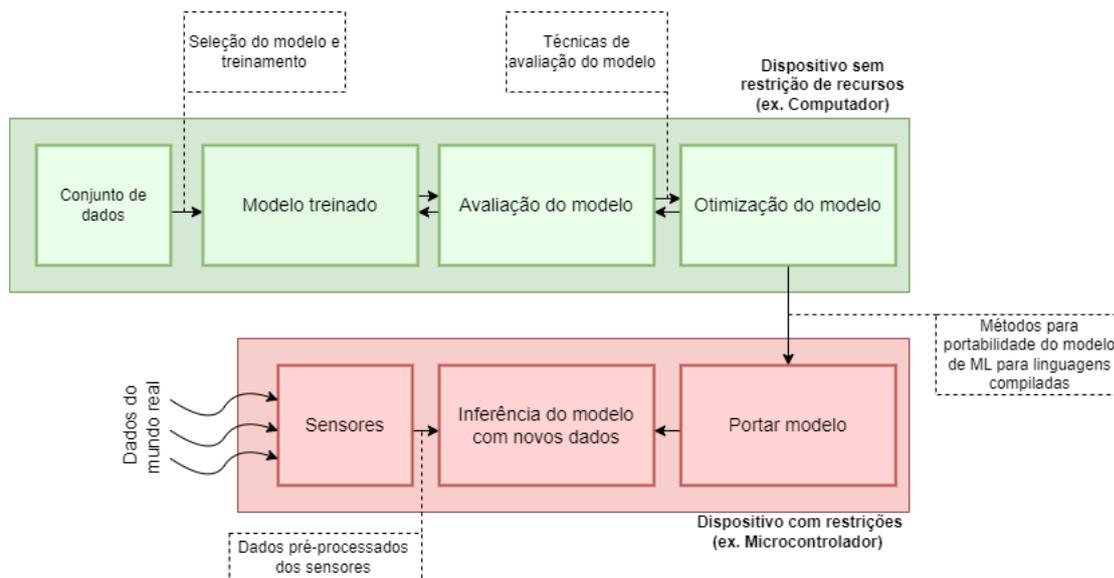


Figura 3.7 – Fluxo típico do tinyML.

Fonte: Adaptado de Rajapakse et al. [2023]

Para o treinamento do modelo podem ser utilizadas bibliotecas bem difundidas como TensorFlow, Pytorch e Scikit-Learn [Rajapakse et al., 2023]. O modelo treinado deve ser otimizado a fim de melhorar a precisão em dispositivos com restrição de recursos [Ray, 2022].

Grandes empresas têm demonstrado interesse na área e desenvolveram bibliotecas para execução de algoritmos em baixo nível, evitando sobrecarga computacional e mantendo consumo de energia baixo. Alguns exemplos dessas bibliotecas são mostrados na Tabela 3.1. Entre elas estão a TensorFlow Lite for Microcontrollers [David

et al., 2020], da Google e CMSIS-NN [Lai et al., 2018] da ARM. Ambas são bibliotecas para execução de redes neurais e consideram que o treinamento do modelo acontece na nuvem, sendo posteriormente carregado ao microcontrolador para fazer a inferência. Isso porque a inferência é muito menos intensiva computacionalmente do que a fase de treinamento [Sakr et al., 2020].

Biblioteca	Algoritmos	Plataformas compatíveis	Pública	Desenvolvedor
MicroMLGen	SVMs (SVC e One-ClassSVM) Árvore de Decisão Random Forest RVM	Arduino ESP32 ESP8266 ARM Cortex-M	Sim	Particular
m2cgen	Regressão Logística Regressão Linear SVM Árvore de Decisão Random Forest Redes Neurais	Múltiplas plataformas	Sim	Particular
AlfES	Redes Neurais	ARM Cortex-M4 Windows (DLL) STM32 Arduino Atmega8xx Raspberry Pi	Não	Fraunhofer IMS
CMSIS-NN	Redes Neurais	ARM Cortex-M	Sim	ARM
ELL	Redes Neurais	ARM Cortex-M ARM Cortex-A Arduino micro:bit	Sim	Microsoft
TensorFlow Lite	Redes Neurais	ARM Cortex-M	Sim	Google
ARM-NN	Redes Neurais	ARM Ethos ARM Mali ARM Cortex-A	Sim	ARM
STM 32Cube.AI	Redes Neurais	STM32	Sim	STMicroelectronics

Tabela 3.1 – Comparação de bibliotecas de aprendizado de máquina para plataformas embarcadas.

Como pode ser visto na Tabela 3.1, os algoritmos suportados por essas bibliotecas são majoritariamente redes neurais. Porém, outros modelos como árvores de decisão, SVM, Relevance vector machine (RVM) e Random Forest também podem ser embarcados. Entre os desenvolvedores estão grandes empresas e grupos de pesquisas renomados com o Fraunhofer², além de desenvolvedores particulares que disponibilizaram suas bibliotecas para uso público.

Abadade et al. [2023] fazem uma revisão sobre aplicações reais de algoritmos embarcados, voltados para áreas de saúde, meio ambiente, agronomia e industrial. No contexto industrial se trata de detecção de anomalias para monitoramento de condição de equipamentos e manutenção preditiva. Alguns trabalhos encontrados na literatura aplicam modelos embarcados para identificação de falhas, como em Mostafavi and Sadighi [2021], onde são utilizados dados de vibração de máquinas rotativas para treinar

² <https://www.fraunhofer.de/en.html>

modelos de aprendizado. Após a fase de treinamento, os modelos são armazenados na memória flash do microcontrolador e dados não vistos são classificados. Antonini et al. [2022] utiliza os dados de vibração e temperatura de bombas para identificação de anomalias. Um microcontrolador ESP32 é utilizado para coletar dados dos sensores, processá-los e treinar localmente um modelo de Random Forest.

3.5 CONSIDERAÇÕES FINAIS

Neste capítulo foram abordados os conceitos básicos sobre os métodos de aprendizado de máquina que serão utilizados durante o trabalho. Também foram apresentadas algumas referências onde estes métodos foram aplicados a problemas de identificação de falha, mostrando que houve bons resultados através do emprego de aprendizado de máquina na detecção de falhas em motores elétricos. Finalmente foram introduzidos os conceitos de aprendizado embarcado, ou tinyML foram introduzidos, apresentando as principais vantagens e desafios a serem vencidos para executar os algoritmos dentro de um microcontrolador.

Após conhecer os conceitos, os algoritmos serão criados a fim de provar a capacidade destas técnicas na identificação das falhas de desalinhamento e desbalanceamento. Também será testada a técnica de One-Class SVM para detecção de anomalia, onde apenas os dados de motores íntegros são utilizados. Nos capítulos a seguir, o processo de obtenção dos dados de treinamento e teste, treinamento dos modelos e portabilidade para o microcontrolador serão detalhados.

4 GERAÇÃO DA BASE DE DADOS

Para o desenvolvimento do trabalho, foi definido o fluxo mostrado na Figura 4.1. O primeiro passo compreende a geração de uma base de dados que contemple as condições do motor que serão classificadas, que são a condição normal de operação e as falhas de desbalanceamento e desalinhamento.

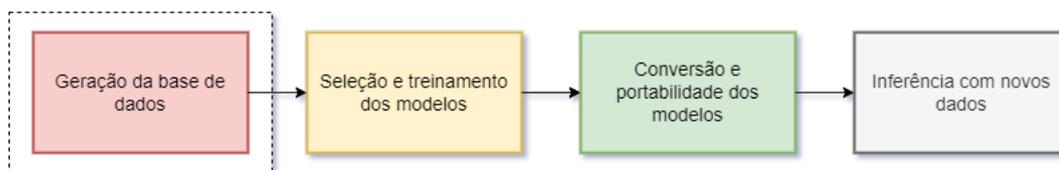


Figura 4.1 – Fluxo do desenvolvimento do trabalho - geração da base de dados.

A seguir serão abordados os experimentos de bancada e o sensor utilizado no trabalho para gerar os dados de treinamento para os modelos de AM.

4.1 BANCADA DE ENSAIOS PRELIMINAR

Para criar as condições de falhas no motor, foi utilizado um acoplamento de motores conectados como um sistema back-to-back. Nesse tipo de arranjo, dois motores são acoplados pelo eixo e um dos motores opera como carga e o outro como acionador. O motor acionador é alimentado pela rede elétrica e o motor de carga através de um variador de frequência, que permite controlar a rotação e carga aplicada ao motor acionador. O objetivo é simular a carga real de um motor de forma controlada em laboratório.

A bancada de testes utilizada durante os ensaios para a qualificação do presente trabalho é mostrada na Figura 4.2. Foram utilizados ambos motores trifásicos da WEG. O motor acionador é da carcaça NEMA 182/4T de 4 polos, frequência de alimentação de 60Hz e com potência nominal de 2,2kW. Este motor foi ligado diretamente à rede de alimentação trifásica. O motor de carga é um motor carcaça 132S, 4 polos, 60Hz e potência de 7,5kW, porém conectado a um inversor de frequência, que permite a variação da velocidade do eixo, causando uma variação na carga do motor acionador. A base utilizada para montagem dos motores não possuía a furação adequada para os pés dos motores utilizados, então a fixação foi feita através de presilhas metálicas. O acoplamento disponível para esta bancada é um acoplamento flexível. Este tipo de acoplamento é capaz de compensar desalinhamentos entre os eixos e absorver vibrações, não sendo recomendado em um sistema onde a falha de desalinhamento será induzida, pois esta pode ser compensada pelo próprio acoplamento.

O processo de identificação de falhas com os dados gerados através dessa bancada não obteve bons resultados e, a principal suposição levantada, foi que a bancada utilizada na geração dos dados de treinamento foi responsável pela baixa

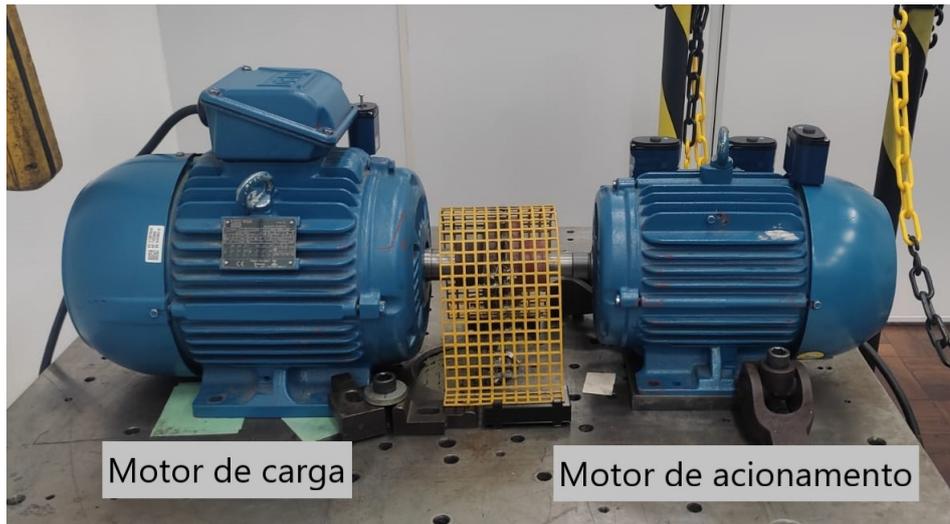


Figura 4.2 – Sistema Back-to-Back utilizado nos ensaios de qualificação.

qualidade dos modelos gerados. A continuação do trabalho contemplou então o projeto e construção de uma nova bancada de testes, que será abordada a seguir.

4.2 BANCADA DE ENSAIOS DEFINITIVA

Após os problemas encontrados na bancada de testes preliminar, foi proposta a construção de uma nova bancada tomando os devidos cuidados, a fim de evitar os problemas encontrados na primeira bancada.

Foram selecionados motores de indução da WEG, ambos da carcaça 63 de 2 polos, com frequência de alimentação de 60Hz e com potência nominal de 0.5kW. Um dos motores foi ligado diretamente à rede de alimentação trifásica e o outro conectado a um inversor de frequência para variação na carga do motor acionador. Foram escolhidos motores menores em vista de facilitar o manuseio da bancada durante os testes.

Os motores foram montados sob uma base projetada especialmente para a bancada, com as furações específicas para a distância entre pés da carcaça dos motores selecionados. Houve o cuidado da precisão ao fazer as furações para garantir o alinhamento dos eixos dos motores no sistema back-to-back. O acoplamento selecionado para esta nova bancada, foi um acoplamento rígido, que não absorve vibrações e desalinhamentos. A bancada de ensaios definitiva pode ser vista na Figura 4.3.

O planejamento do ensaio para geração da base de dados foi feito através de uma árvore de ensaios, mostrada na Figura 4.4. Foram coletados dados com o motor alinhado e balanceado, considerando essa como a condição normal do motor. Após, foram causadas as falhas de desbalanceamento e desalinhamento. Por definição, não foram geradas as falhas combinadas de desalinhamento e desbalanceamento. Também foram variadas as condições de carga de 0%, 50% e 100%, para cada uma



Figura 4.3 – Sistema Back-to-Back utilizado nos ensaios finais.

das condições do motor.

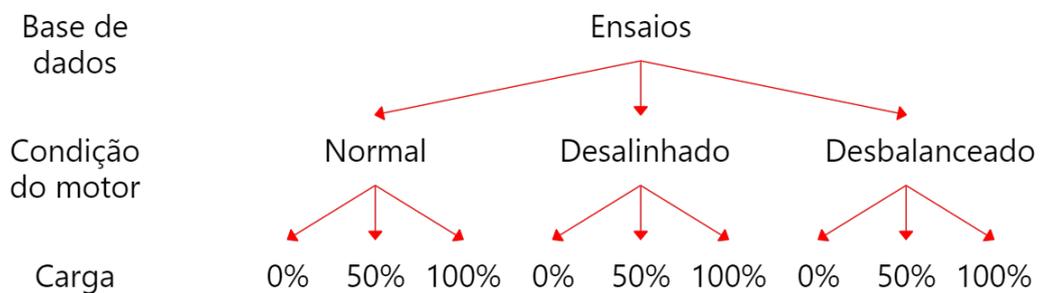


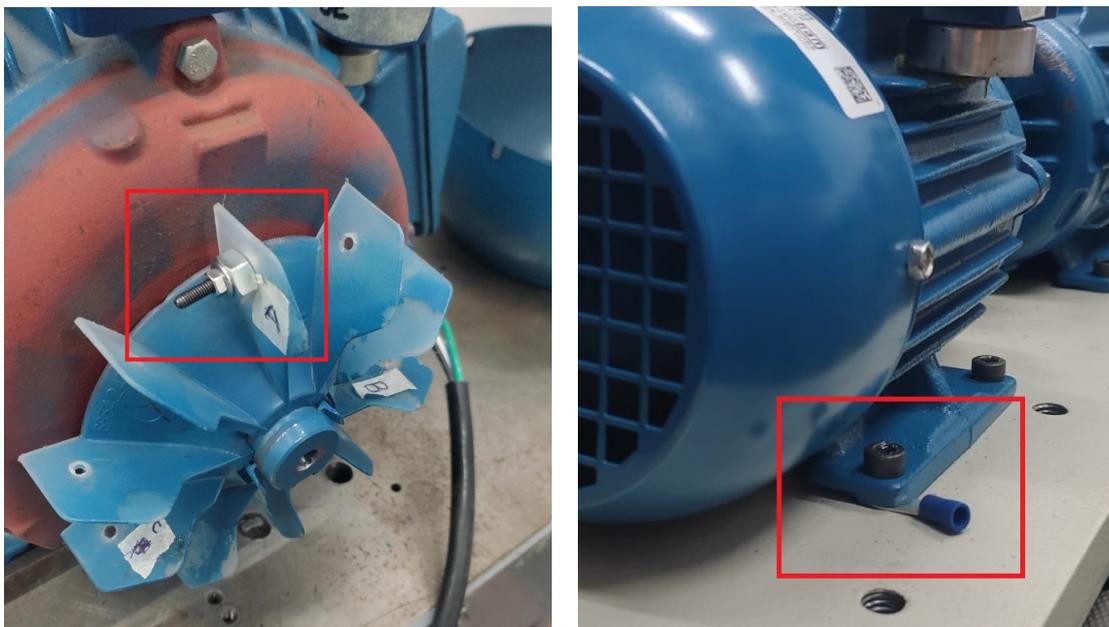
Figura 4.4 – Árvore de ensaios para geração da base de dados.

4.3 MANIPULAÇÃO DAS CONDIÇÕES DA BANCADA

Para gerar os dados que contemplam a base de dados desse trabalho, foram geradas as condições de motor “normal”, “desalinhado” e “desbalanceado” e medida a FFT do sinal de vibração do motor acionador. Através da FFT, podem ser diagnosticadas as falhas nas frequências específicas mostradas na Seção 2.4.1. As condições de falha foram geradas pelas interferências na bancada mostradas na Figura 4.5.

O ensaio de coleta de dados iniciou com a condição normal, sendo realizado o alinhamento e balanceamento do motor. O balanceamento foi medido através da técnica de balanceamento em um plano sem medição de fase. Nesta técnica, são feitos cálculos para definição da massa necessária para o balanceamento, com base na medição de vibração em 4 condições. As condições são: vibração inicial, massa

em uma posição arbitrária, massa na posição 120° da primeira posição e com massa a 240° da primeira posição. Todas as medições devem ser feitas com a mesma massa, somente com o reposicionamento dela [Wowk, 1991]. O desalinhamento por sua vez, foi avaliado apenas pelo sinal de vibração em frequência. A Figura 4.6a mostra o sinal do motor para a condição normal, no início do ensaio. Há alguns picos nas frequências de 1 e 2x a frequência da rotação (60 e 120Hz), porém com baixa amplitude, não chegando a representar falha no motor. A carga foi então variada entre as cargas planejadas para o ensaio e os dados coletados para essa condição.



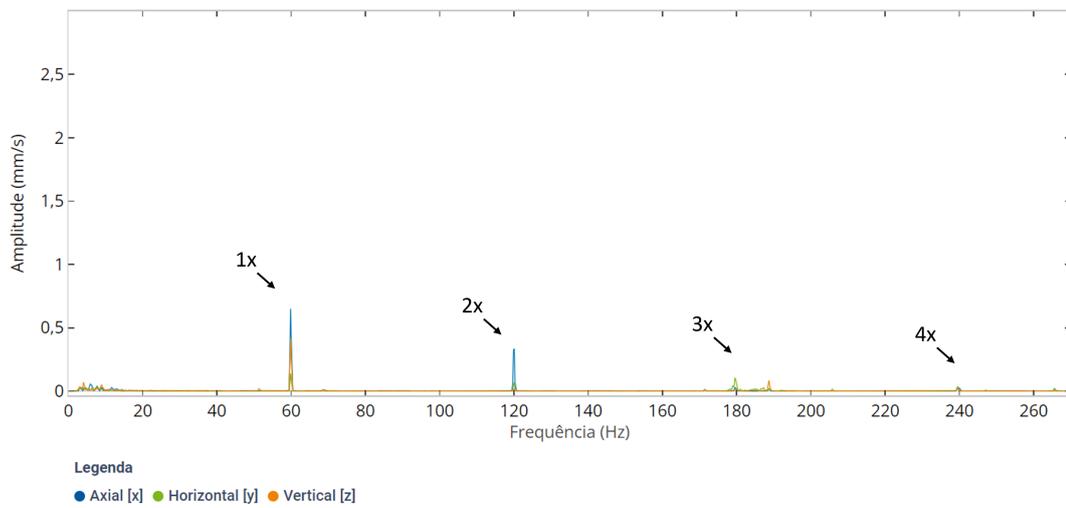
a Massa adicionada para desbalanceamento

b Calço para desalinhamento angular

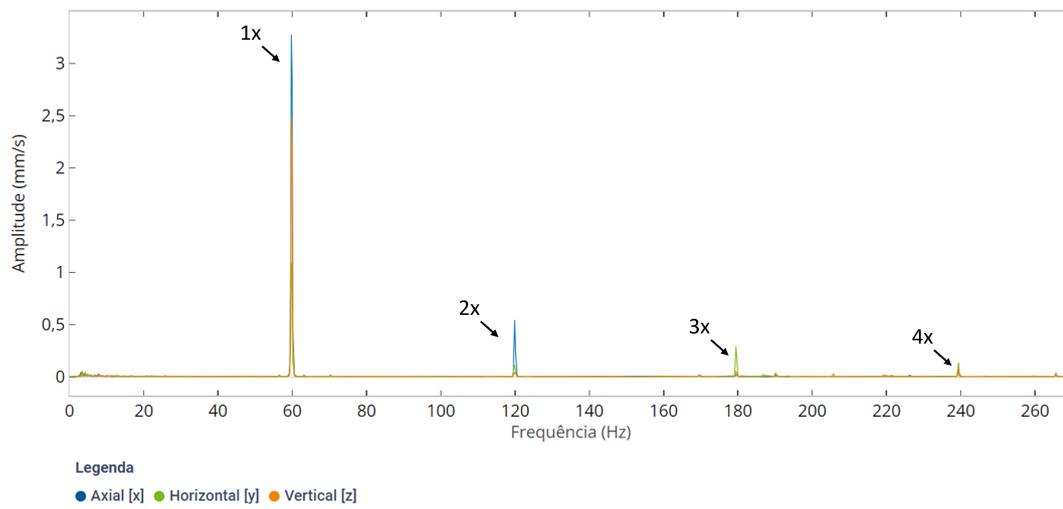
Figura 4.5 – Ajustes para gerar falhas na bancada.

Após isso, o motor foi desbalanceado através da adição de massa no ventilador do motor, conforme pode ser observado na Figura 4.5. A medição da FFT de vibração foi realizada e o sinal é mostrado na Figura 4.6b. Nesta condição, é evidenciado um pico em 60Hz, 1x a frequência da rotação mecânica do motor de 2 polos. Isso indica que o motor está desbalanceado, conforme discutido no Capítulo 2.4.1. Foram feitas as variações de carga e então o motor foi balanceado e alinhado novamente.

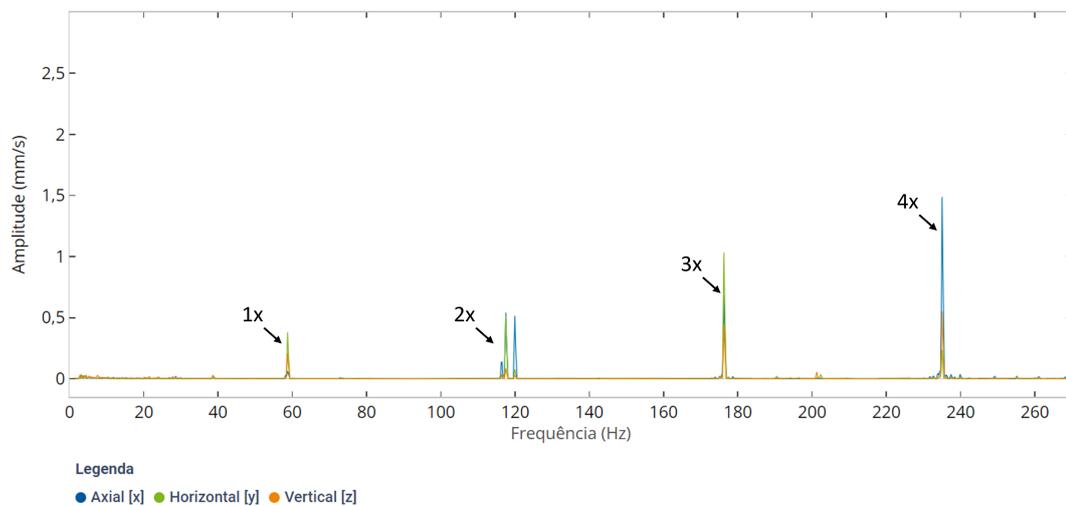
A última condição imposta no ensaio, foi o desalinhamento. Optou-se por realizar o desalinhamento angular, através da adição de calços na fixação traseira do motor, conforme mostrado na Figura 4.5a. O mesmo calço foi adicionado em ambos os pés traseiros do motor. A FFT da vibração para esta condição é mostrada na Figura 4.6c. Se comparada com a Figura 4.6b, pode-se observar que o pico em 60Hz não está mais presente, mas as múltiplas dessa frequência se evidenciam, principalmente de 2 a 4x. Assim como nas outras condições, foi variada a carga em 0, 50 e 100%.



a Condição normal



b Condição desbalanceada



c Condição desalinhada

Figura 4.6 – Espectro em frequência para diferentes condições do motor.

4.4 AQUISIÇÃO DOS DADOS

Para a medição de vibração, foi utilizado o sensor comercial WEGscan100. O WEGscan100 é um sensor para monitoramento de equipamentos e ativos através da coleta de dados como temperatura, vibração nos três eixos, campo magnético e outras informações relevantes do ativo. Para o presente trabalho, apenas os dados de vibração triaxial foram utilizados. O sensor pode ser visto na Figura 4.7. O microcontrolador presente no sensor possui memória flash de 1024MB e tecnologia Bluetooth® 5.1 para coleta dos dados. A alimentação do sensor WEGscan100 é feita por meio de baterias, que podem ser substituídas quando esgotadas. Conta ainda com um botão de acionamento e um led que indica se o sensor está ligado. Mais informações sobre as características do WEGscan100 podem ser obtidas em WEGscan [2023].



Figura 4.7 – Sensor WEGscan100.

Fonte: [WEGscan, 2023]

O sensor é parafusado a uma base intermediária de alumínio, e esta base é colada ao corpo do motor de acionamento. A Figura 4.8 mostra a fixação e o posicionamento do sensor no motor. São realizadas aproximadamente 4 medições por minuto, sendo cada medição com duração de 307ms, devido às configurações do equipamento. Esta medição no domínio do tempo é utilizada para calcular as variáveis que compõem o conjunto de dados de treinamento, mostradas na Tabela 4.1. Todas as variáveis apresentadas serão utilizadas posteriormente na criação dos modelos de aprendizado.

A cada troca de condição ou carga, os dados foram coletados do sensor e rotulados com a condição do ensaio em que foram adquiridos. Ao final do ensaio, a base de dados conta com 7 variáveis relacionadas ao sinal de vibração para cada eixo, resultando em 21 dados de entrada para o treinamento dos modelos. Ao final, a base possui 52371 objetos de dados, sendo 17457 linhas para cada condição do motor (normal, desbalanceado e desalinhado). Essa base de dados é chamada durante o presente trabalho, como base de dados de treinamento.

A Figura 4.9 mostra o histograma dos dados separando-os pela quantidade de dados de cada condição e carga. A base de dados resultante não é totalmente

Tabela 4.1 – Variáveis do conjunto de dados para treinamento de modelos de AM.

Variável	Descrição
accelRMS(X,Y,Z)	Valor RMS do sinal de vibração em aceleração nos três eixos
velRMS(X,Y,Z)	Valor RMS do sinal de vibração em velocidade nos três eixos
accelKurt(X,Y,Z)	Curtose nos três eixos
velocity(X,Y,Z)1F	1x a frequência de rotação mecânica nos três eixos
velocity(X,Y,Z)2F	2x a frequência de rotação mecânica nos três eixos
velocity(X,Y,Z)3F	3x a frequência de rotação mecânica nos três eixos
velocity(X,Y,Z)4F	4x a frequência de rotação mecânica nos três eixos



Figura 4.8 – Sensor fixado no motor de acionamento.

balanceada com relação às variáveis do ensaio, tendo a condição desalinhada mais amostras em carga 0% e a condição normal com mais amostras em 50%, por exemplo. O menor número de amostras está no caso desbalanceado com carga 50%.

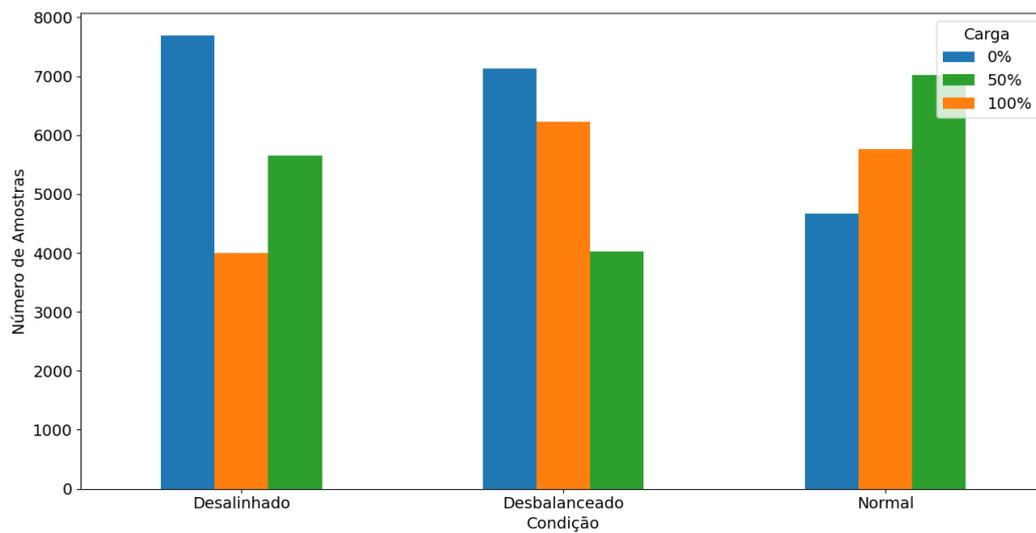


Figura 4.9 – Número de amostras de dados por condição e carga do motor.

5 TREINAMENTO E AVALIAÇÃO DOS MODELOS DE AM

A segunda etapa planejada para o desenvolvimento do trabalho, mostrada na Figura 5.1, é a seleção e treinamento dos modelos de aprendizado de máquina. Antes de utilizar a base de dados para treinamento dos modelos, é preciso conhecer os dados gerados e fazer um pré-tratamento dos mesmos para garantir a qualidade do conjunto de dados coletados. Com os dados tratados, pode-se partir então para a seleção dos algoritmos e treinamento dos modelos. A fase final é a avaliação destes modelos com dados não vistos para avaliar sua capacidade de identificação das falhas.

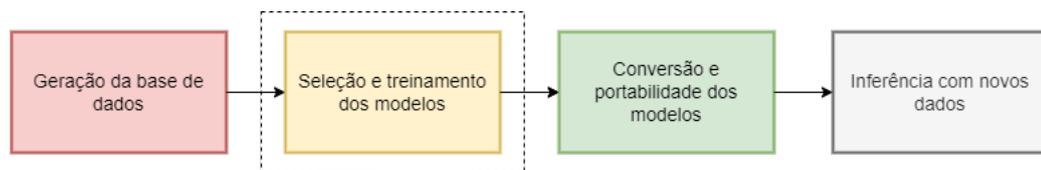


Figura 5.1 – Fase de seleção e treinamento dos modelos.

A seguir serão abordadas as etapas de tratamento dos dados, seleção, treinamento e avaliação dos modelos. Todas as análises foram feitas em Python utilizando a biblioteca Scikit-Learn [Pedregosa et al., 2011].

5.1 TRATAMENTO DOS DADOS

O primeiro tratamento aplicado à base de dados de treinamento foi a normalização através da padronização dos dados, técnica conhecida na literatura como *StandardScaler*. A padronização é feita calculando a média e o desvio padrão do conjunto de dados, então cada ponto é subtraído pela média e dividido pelo desvio padrão. Isso é feito para cada variável de forma independente. Todos esses coeficientes são exportados para serem utilizados na fase de inferência. Segundo a documentação oficial da biblioteca *scikit-learn*, essa técnica é utilizada para garantir que todas as características dos dados tenham a mesma escala, o que tende a melhorar o desempenho dos algoritmos, principalmente os mais sensíveis a variações na escala dos dados.

Após a normalização, foi aplicada uma técnica de redução de dimensionalidade nos dados, o *Principal Component Analysis* (PCA). O PCA pode ser aplicado para redução de dimensionalidade, extração de características e visualização de dados. Seu objetivo principal é reduzir a dimensionalidade de um conjunto de dados mantendo o máximo de variação possível dos dados originais. Isso é feito através da criação de um novo conjunto de dados, chamados de componentes principais (CP), onde os primeiros componentes capturam a maior variação das variáveis originais [T., 2002]. O número de componentes principais gerados pela técnica pode ser variado. Em termos de visualização de dados, normalmente são utilizados dois ou três componentes, uma vez que com mais que três dimensões, a visualização torna-se difícil de interpretar

[Wold et al., 1987]. Neste trabalho, o PCA foi utilizado apenas para visualização dos dados, sendo a experimentação feita no espaço de atributos original.

A visualização do PCA para três CPs é mostrada na Figura 5.2. A variância do conjunto de dados é representada 52.9% pelo componente principal 1, 23.3% pelo componente principal 2 e 7.3% pelo componente principal 3. Juntos, estes 3 componentes representam 83.5% da variância presente nos dados. Observando os gráficos, verifica-se que as três condições (Normal, Desalinhado, Desbalanceado) se sobrepõem em algumas dimensões dos componentes principais (PCs). Isso mostra que, em algumas combinações de PCs, as condições não são completamente separáveis. Essa sobreposição pode dificultar a tarefa de classificação correta das diferentes condições pelos modelos de AM. O PCA mostra uma maior facilidade de separação para os dados da condição desbalanceada, principalmente ao observar a relação entre os componentes PC3-PC1 e PC2-PC1. As condições normal e desalinhada se sobrepõem no espaço em alguns pontos, principalmente nas componentes PC3-PC1, mas não se sobrepõem totalmente, o que pode ser observado nas componentes PC2-PC1 e PC3-PC2.

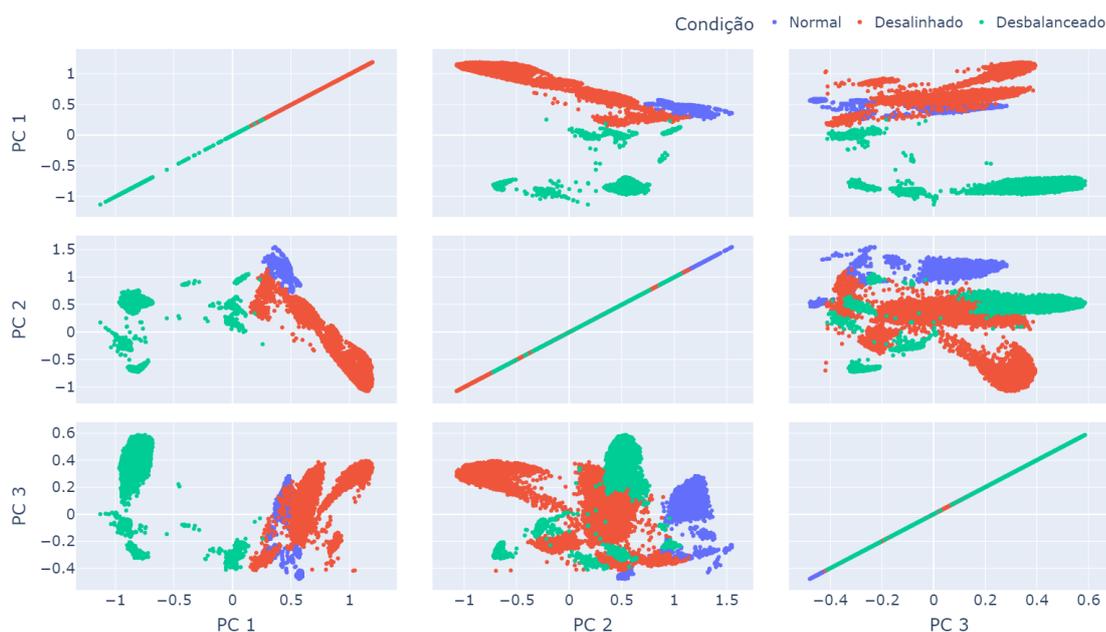


Figura 5.2 – PCA em duas dimensões para o conjunto de dados coletados.

Outro recurso utilizado para avaliação dos dados de cada condição do motor é o gráfico de boxplot. O boxplot ajuda a ter uma avaliação geral de um conjunto de dados, pois mostra graficamente os grupos de localidade, dispersão e assimetria de dados numéricos por meio de seus quartis. Além disso, relaciona a mediana, máxima e mínima de cada variável e identifica possíveis dados fora do padrão (outliers). A Figura 5.3 mostra o boxplot do conjunto de dados coletados no ensaio, separados através da condição do motor. Avaliando a variação mostrada pelo boxplot, pode se observar que há variações entre o padrão das medianas e quartis para cada condição. Para a

condição normal, os valores das medianas e quartis de todas as variáveis são menores, quando comparado às duas condições de falha. Para a condição desalinhada, é visto o aumento dos valores de aceleração, velocidade e das múltiplas de 2, 3 e 4 vezes a rotação do motor, nos três eixos. Já na condição de motor desbalanceado, a múltipla de 1x a rotação mecânica se evidencia e também há um aumento nos valores globais de vibração em aceleração e velocidade. Este comportamento é esperado para as variáveis em questão, como mostrado anteriormente.

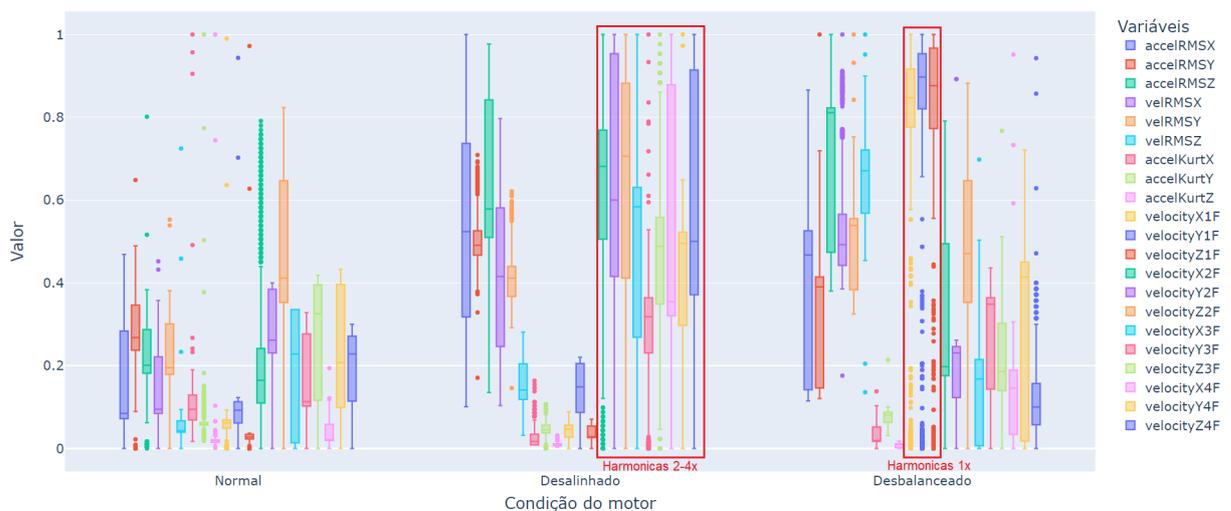


Figura 5.3 – Gráfico de boxplot dos dados do conjunto de dados coletados.

5.2 SELEÇÃO E TREINAMENTO DOS MODELOS

Após a fase de pré-processamento do dados, a base de dados é considerada apta para o treinamento dos modelos. A seleção dos algoritmos é feita avaliando a possibilidade de embarcar os modelos no microcontrolador disponível. Para isso, os modelos resultantes devem atender aos requisitos de memória, processamento e consumo do dispositivo a ser embarcado. Foi definida a utilização de modelos supervisionados de árvore de decisão e SVM. Também optou-se por avaliar a capacidade do modelo não supervisionado One-Class SVM em identificar as falhas, pois nesse caso apenas a condição do motor normal é utilizada para treinar o modelo. Todos os modelos escolhidos podem ser portados para o microcontrolador, conforme discutido na seção 3.4. Abaixo serão abordados com mais detalhes a fase de treinamento de cada um dos modelos.

5.2.1 Árvore de decisão

A etapa inicial do treinamento do modelo começa com a otimização dos parâmetros. No caso da árvore de decisão, foram avaliados diferentes valores para o

parâmetro que determina a profundidade máxima da árvore (max-depth) e o critério usado para avaliar o modelo (criterion).

Para escolher os melhores parâmetros para o modelo, foram empregadas as técnicas de validação cruzada e variação de parâmetros, utilizando as funções GridSearchCV e StratifiedKFold do Scikit-Learn.

O GridSearchCV, segundo a documentação oficial da biblioteca, recebe como entrada o tipo de modelo a ser treinado e um conjunto de parâmetros possíveis. Em seguida, ele explora diversas combinações desses parâmetros para determinar os mais eficazes na construção do melhor modelo de aprendizado de máquina para o conjunto de dados em análise. O StratifiedKFold separa o conjunto de dados em “k” pastas a serem definidas pelo usuário, mantendo a mesma porcentagem de amostras de cada classe alvo para cada pasta.

Na validação cruzada, foram utilizadas 10 pastas para o treinamento e os parâmetros variados foram os critérios de “Gini” ou “Entropia” e max-depth de 5, 10 e 15. A Tabela 5.1 são apresentados os resultados das métricas de acurácia, precisão e recall durante as fases de treino e teste para o modelo de árvore de decisão. A acurácia informa a quantidade de classificações corretas feitas pelo modelo e é comum de ser utilizada como métrica global em casos onde os dados são balanceados, como é o caso do conjunto de dados de treinamento deste trabalho. A precisão ajuda a identificar a relação de falsos positivos durante a classificação enquanto o recall, também chamado de sensibilidade, fornece uma visão sobre os falsos negativos. Percentuais altos dessas métricas indicam mais classificações corretas do que incorretas. Para fins de investigação, as mesmas também são mostradas na Tabela 5.1.

Os valores de acurácia, precisão e recall encontrados na fase de treino tiveram média entre 93 a 95% e durante a fase de testes acima de 99%, mostrando que o modelo teve bons resultados com os dados de treinamento. O resultado inferior durante o treino se dá pois nesta fase ainda estão sendo escolhidos os parâmetros que possuem melhor desempenho, enquanto a fase de teste o modelo utilizado é o estimador criado a partir dos melhores parâmetros definidos.

Tabela 5.1 – Desempenho do modelo de árvore de decisão.

	Acurácia(%)	Precisão(%)	Recall(%)
Treino	93.87 ± 0.10	95.37 ± 0.07	93.86 ± 0.10
Teste	99.51 ± 0.16	99.49 ± 0.14	99.48 ± 0.14

Na Figura 5.4 é mostrada a matriz de confusão média dos resultados das 10 pastas durante o treinamento do modelo de árvore de decisão, onde a cada iteração eram escolhidos os melhores hiperparâmetros. Pode ser observado que o menor erro de classificação ocorreu para a classe normal, onde apenas 0.06% da classe predita

foi diferente da verdadeira. Os maiores erros de classificação ocorreram na classe desbalanceada ser considerada desalinhada e vice versa, porém como pode ser visto, a porcentagem de erros é baixa.

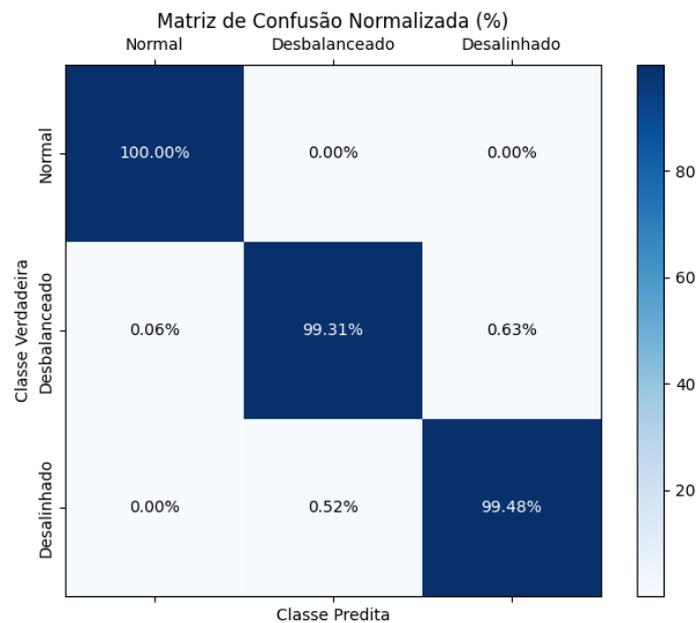


Figura 5.4 – Média das matrizes de confusão das 10 pastas para Árvore de Decisão.

A determinação dos melhores parâmetros foi realizada com base nos resultados obtidos em cada iteração da validação cruzada. Para cada parâmetro, foi selecionado aquele que apresentou o melhor desempenho com maior frequência ao longo das validações. Para o modelo de árvore de decisão, os melhores parâmetros foram identificados como o critério de Gini e número de ramificações igual a 15. Após a definição desses parâmetros, um estimador foi gerado e treinado utilizando todos os dados disponíveis na base de dados do experimento de coleta.

5.2.2 SVM

Para o modelo de SVM, os parâmetros variados foram o tipo de kernel e o C . Conforme apresentado na Seção 3.2.3, o tipo de kernel define a função a ser utilizada pelo SVM para resolver o problema de classificação. O parâmetro C adiciona uma penalidade para cada dado mal classificado. Se C for pequeno, a penalidade para pontos mal classificados é baixa, então um limite de decisão com uma grande margem é escolhido à custa de um maior número de erros de classificação. Se C for grande, o SVM tenta minimizar o número de exemplos mal classificados devido à alta penalidade que resulta em um limite de decisão com uma margem menor.

A técnica de validação cruzada com 10 pastas e variação de parâmetros foi aplicada, utilizando as funções do Scikit-Learn. Os parâmetros avaliados durante o

treinamento foram kernel “polinomial” ou “linear” e C igual a 10, 100 e 1000.

Os resultados encontrados durante a fase de treinamento são mostrados a seguir. Na Tabela 5.2, são apresentados os resultados de média e desvio padrão para acurácia, precisão e recall. Todas as métricas resultaram em valores acima de 90%, o que indica bons resultados encontrados. A acurácia média encontrada durante o treinamento foi de 91.23%, com desvio padrão de 0.09%. Os resultados de teste foram acurácia média de 99.55% com desvio padrão de 0.11%.

Tabela 5.2 – Desempenho do modelo de SVM.

	Acurácia(%)	Precisão(%)	Recall(%)
Treino	91.23 ± 0.09	94.14 ± 0.08	91.23 ± 0.09
Teste	99.55 ± 0.11	99.55 ± 0.11	99.55 ± 0.11

Na Figura 5.5 é apresentada a matriz de confusão média durante o processo de validação cruzada com 10 pastas. No caso do SVM, o caso com menos acertos é a condição desbalanceado, onde 0.57% das amostras foram classificadas como desalinhadas. Apesar disso, todas as classes tiveram acertos superiores a 99%, o que mostra que o modelo teve uma acurácia elevada para os dados da base de dados coletada.

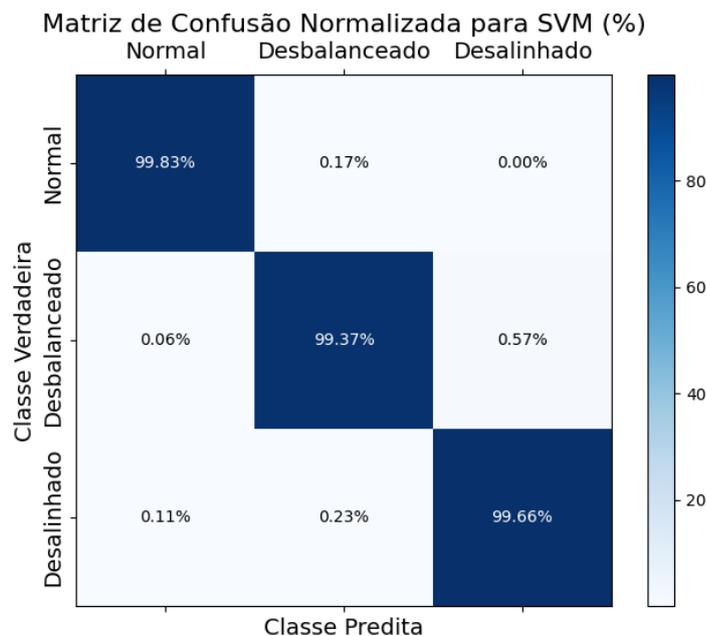


Figura 5.5 – Média das matrizes de confusão das 10 pastas para o SVM.

Assim como na árvore de decisão, os parâmetros do modelo foram determinados pela frequência em que foram escolhidos durante o treinamento. Os melhores

parâmetros encontrados foram kernel do tipo polinomial e $C=100$. O estimador gerado foi treinado com todas os dados disponíveis durante a fase de treinamento.

5.2.3 One-Class SVM

O treinamento do modelo de One-Class SVM, diferente dos outros modelos, utiliza somente os dados classificados como normais na base de dados. O classificador também foi avaliado utilizando validação cruzada com 10 pastas e variação de parâmetros, mas nesse caso, a variação de parâmetros foi feita manualmente, pois a função GridSearchCV é utilizada principalmente para modelos supervisionados. Como se trata de um modelo de SVM, o kernel é um parâmetro que deve ser escolhido adequadamente. Para isso, foram variados os kernels “linear”, “polinomial” e “rbf”. Além do kernel, foram variados também os parâmetros ν e γ .

A Figura 5.6 mostra a acurácia do modelo para cada um dos kernels testados, fixando $\nu=0.01$ e $\gamma=0.1$. Os dados normais foram separados em 10 pastas e o modelo treinado em cada uma das pastas. O teste do modelo foi feito com dados normais não vistos durante o treinamento e dados de falha.

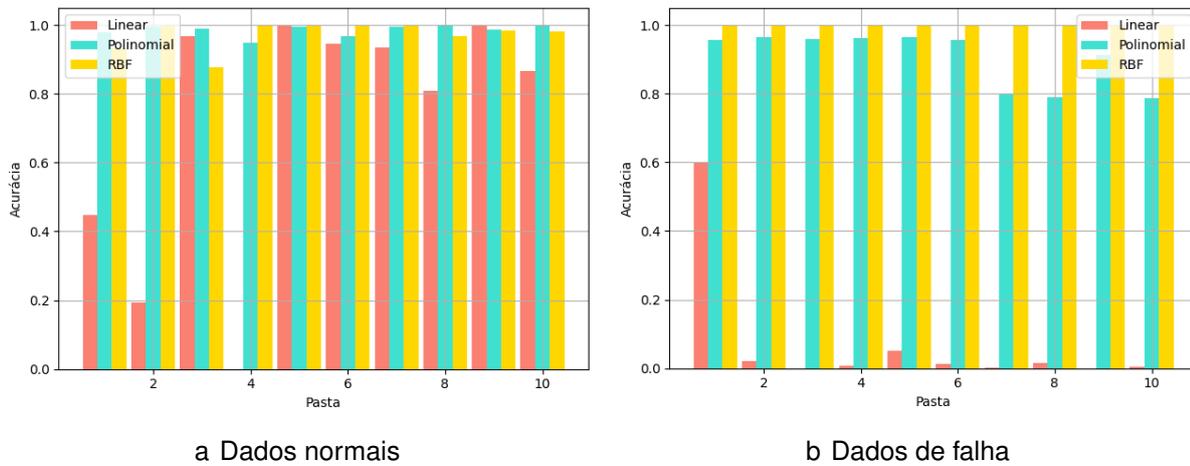
A Tabela 5.3 mostra a média e desvio padrão de cada um dos kernels nos dados normais, de falha e o geral, considerando a quantidade de dados de cada condição. Dentre os kernels testados, o linear teve o pior desempenho, tanto com dados normais como com falha. Para os dados normais, obtiveram bons desempenhos os kernels polinomial e RBF, tendo o primeiro uma média de acertos de 90.5% e desvio padrão de 1.2%. O kernel RBF obteve uma média ainda maior, de 94.4%, porém com desvio padrão alto entre as iterações. Para os dados de falha, ambos polinomial e RBF apresentaram acurácia alta, sendo o kernel RBF com a maior média de acertos e menor desvio padrão baixo. Observando os resultados gerais, é visto que o melhor resultado ocorre para o kernel RBF, tendo a maior média com menor desvio padrão geral entre os parâmetros avaliados.

Tabela 5.3 – Acurácia para diferentes kernels.

	Dados Normais	Dados de Falha	Geral
Linear	71.6 ± 34.8	7.1 ± 17.7	28.6 ± 14.3
Polinomial	90.5 ± 1.2	98.7 ± 7.6	93.2 ± 4.8
RBF	94.4 ± 9.2	99.8 ± 0.06	98.9 ± 1.2

A variação dos parâmetros ν e γ foram feitos fixando o kernel RBF. Na Tabela 5.4 são mostrados os resultados de acurácia obtidos para diferentes valores de ν . É visto que os melhores resultados são obtidos para os menores valores do parâmetro. Considerando a média e desvio padrão encontrados para ambos dados normais e de falha, o valor de $\nu=0.001$ foi selecionado para gerar estimador.

Figura 5.6 – Acurácia do modelo para diferentes Kernels.

Tabela 5.4 – Acurácia para diferentes valores de ν .

ν	Dados Normais	Dados de Falha	Geral
0.001	99.3 ± 1.6	99.4 ± 1.9	99.4 ± 0.4
0.01	98.5 ± 2.3	99.7 ± 0.1	99.3 ± 0.7
0.1	85.0 ± 13.5	100.0 ± 0.0	95.0 ± 4.5

Na Tabela 5.4 são vistos os resultados encontrados para diferentes valores de γ , mantendo kernel como RBF e $\nu=0.001$. Assim como o parâmetro ν , os melhores resultados de acurácia são encontrados para os menores valores de γ testados.

Os resultados obtidos com a variação dos parâmetros γ e ν mostram a pouca influência desses parâmetros nos resultados. Diferentemente do que foi visto para os kernels, onde a variação do parâmetro gerou grande impacto na acurácia final do modelo.

Tabela 5.5 – Acurácia para diferentes valores de γ .

γ	Dados Normais	Dados de Falha	Geral
0.001	98.5 ± 2.3	99.7 ± 0.1	99.3 ± 0.7
0.01	97.3 ± 3.8	99.7 ± 0.1	98.9 ± 1.2
0.1	94.4 ± 9.2	99.8 ± 0.1	98.0 ± 3.0

Ao final do treinamento, foi gerado o estimador do algoritmo de One-Class SVM com os parâmetros de kernel RBF, $\nu=0.001$ e $\gamma=0.001$.

6 RESULTADOS DOS ALGORITMOS EMBARCADOS

Com os modelos gerados, as próximas etapas do processo são a conversão dos modelos, portabilidade para o microcontrolador e a inferência com novos dados, conforme apresentado no fluxo da Figura 6.1. A inferência é feita na bancada de testes utilizada na geração dos dados de treinamento, para identificar o estado do motor. A seguir são abordadas cada uma das etapas com mais detalhes.

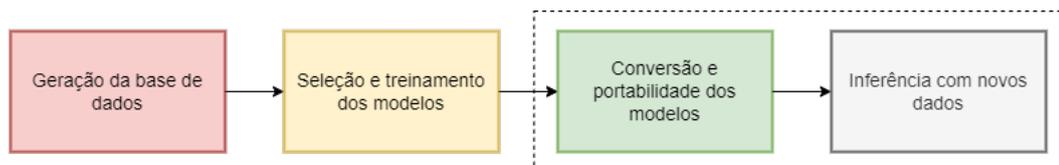


Figura 6.1 – Fase de seleção e treinamento dos modelos.

6.1 CONVERSÃO E PORTABILIDADE DO MODELO

Para realizar a conversão do modelo, foi realizada uma análise das bibliotecas publicamente disponíveis, e a biblioteca “micromlgen” foi escolhida. A biblioteca selecionada é uma ferramenta de código aberto que tem como objetivo facilitar a conversão de modelos de aprendizado de máquina para códigos otimizados o suficiente para serem executados em plataformas de microcontroladores e dispositivos IoT. A ferramenta é desenvolvida em Python e permite que os usuários convertam modelos treinados em diversas bibliotecas como Scikit-learn e TensorFlow Lite, para código C ou C++.

Os modelos de árvore de decisão, SVM e One Class SVM foram convertidos para arquivos C Header. O microcontrolador integrado ao sensor WEGscan100 possui 1024 kB de memória flash e 256 kB de RAM. Os modelos são armazenados na memória flash e declarados internamente como constantes, de modo que ao utilizá-los durante a execução do programa, possam ser acessados sem a necessidade de importá-los para a memória de execução (RAM). O tamanho após convertidos é apresentado na Tabela 6.1, onde pode ser visto que possuem tamanhos menores que 1024kB. É importante ressaltar que os modelos devem ser armazenados na memória flash para que não sejam perdidos a cada desenergização do sensor.

Tabela 6.1 – Tamanho dos modelos portáveis.

Modelo	Tamanho
Árvore de decisão	29kB
SVM	374kB
One Class SVM	168kB

Os modelos convertidos são utilizados para classificar ou prever novas amostras através da função “predict”. O retorno da função predict é 0 para condição normal,

1 para desbalanceado e 2 para desalinhado. No caso do modelo de One Class SVM, é retornado 0 para a condição normal e 1 para condição de falha. Por limitação de memória, os modelos de SVM e Árvore de Decisão foram portados para dentro de um sensor, e o modelo de One Class SVM para outro sensor separado.

6.2 ENSAIOS DE INFERÊNCIA

Após embarcados, os modelos treinados previamente foram testados na bancada do sistema back-to-back utilizado para a geração da base de dados. A seguir serão abordadas as condições do motor e os resultados obtidos nos ensaios de inferência.

6.2.1 Ajustes da bancada para inferência

As condições de motor foram avaliadas e as falhas geradas utilizando os mesmos métodos explicados no Capítulo 4. Inicialmente, foi realizado o balanceamento e alinhamento e utilizado o espectro da vibração para aferir a condição do motor como normal. O espectro medido é mostrado na Figura 6.2, onde são vistos alguns picos nas frequências múltiplas da rotação mecânica, porém com baixa amplitude. O mesmo foi feito para as condições de desalinhamento e desbalanceamento, mostradas nas Figuras 6.3 e 6.4, respectivamente. Na condição de desalinhamento, é visto o aumento da amplitude das harmônicas, principalmente as de 3 e 4x. Já na condição de desbalanceamento, há um aumento bem significativo da harmônica de 1x a frequência de rotação mecânica.

É importante ressaltar que por mais que os pesos de desbalanceamento e o calço de desalinhamento usados sejam os mesmos, não há como garantir que a condição resultante seja exatamente a mesma condição do primeiro ensaio. Por esse motivo, é importante repetir as medições de espectro e garantir que a condição resultante está próximo ao esperado.

6.2.2 Inferência para modelos supervisionados

Para cada condição ajustada na bancada, foram realizadas medições e utilizados os modelos embarcados para inferir sobre a condição do motor. Conforme mostrado anteriormente, o sensor faz a medição do sinal de vibração e calcula as variáveis de entrada dos modelos. Após o cálculo, são utilizados os coeficientes da técnica de normalização exportados na fase de treinamento, e normalizados os valores da nova amostra para realizar a inferência. Com o valor normalizado, o modelo é executado e o resultado de saída do modelo comparado com a condição atual do ensaio.

Durante o ensaio de inferência, foram avaliadas 63 amostras de motor normal, 82 de motor desbalanceado e 77 de motor desalinhado. A Tabela 6.2 mostra um com-

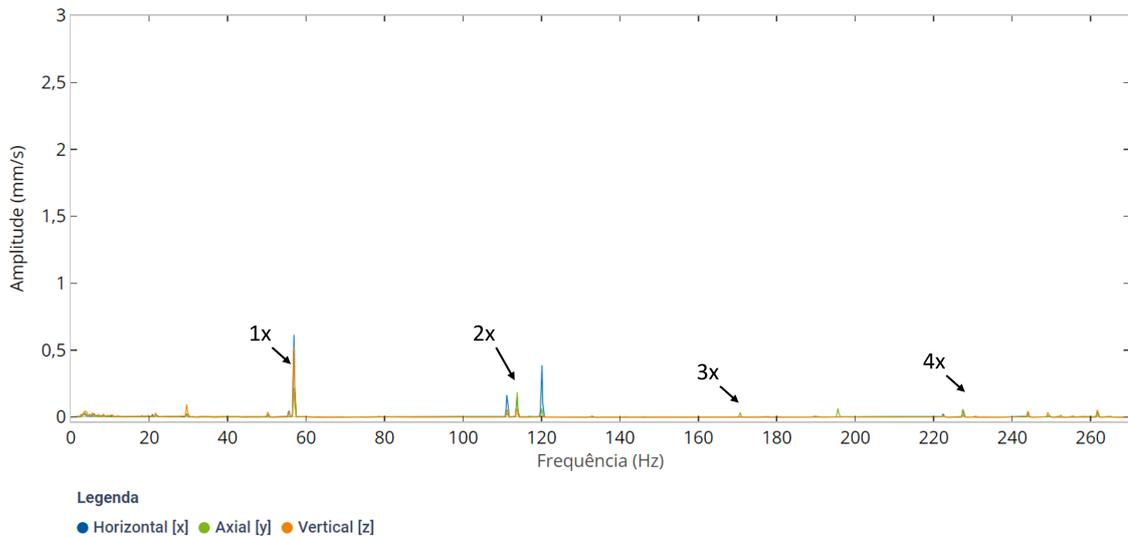


Figura 6.2 – FFT da condição normal do motor para inferência.

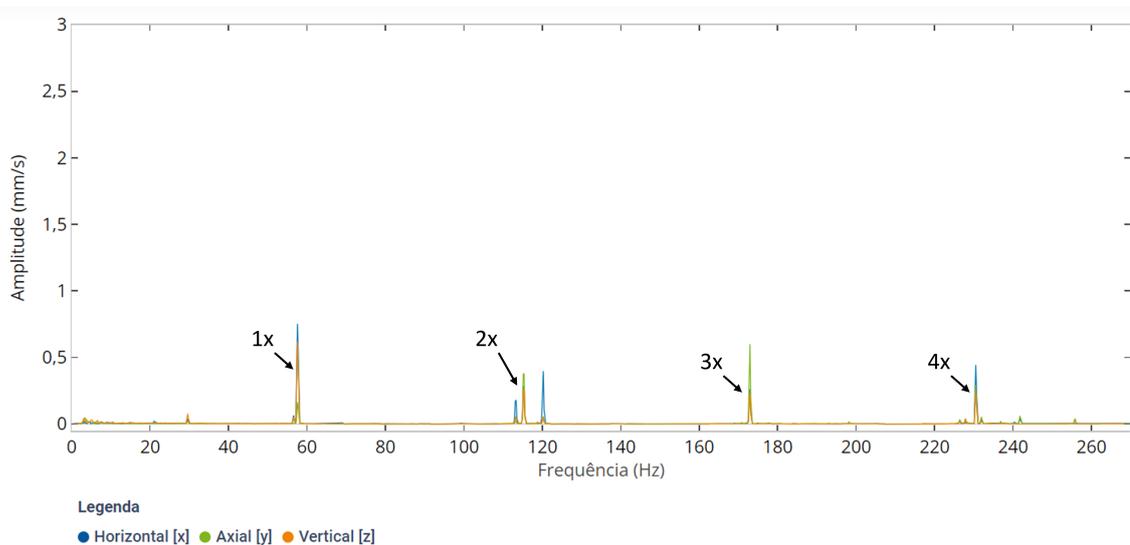


Figura 6.3 – FFT da condição desalinhado do motor para inferência.

pilado do resultado obtido para cada condição e para os dois modelos supervisionados testados. A acurácia média foi acima de 70% para os dois modelos, tendo melhor resultado geral para o modelo de árvore de decisão.

Tabela 6.2 – Tabela de acertos dos modelos supervisionados.

Modelo	Acurácia (%)			
	Normal	Desbalanceado	Desalinhado	Geral
Árvore de decisão	93.6	100.0	56.3	83.3
SVM	77.8	80.5	56.3	71.5

Na Figura 6.5 são mostradas as matrizes de confusão dos modelos de árvore de decisão e SVM. Pode ser observado que o maior erro em ambos os modelos ocorreu na condição de motor desalinhado, onde as amostras foram consideradas de motor

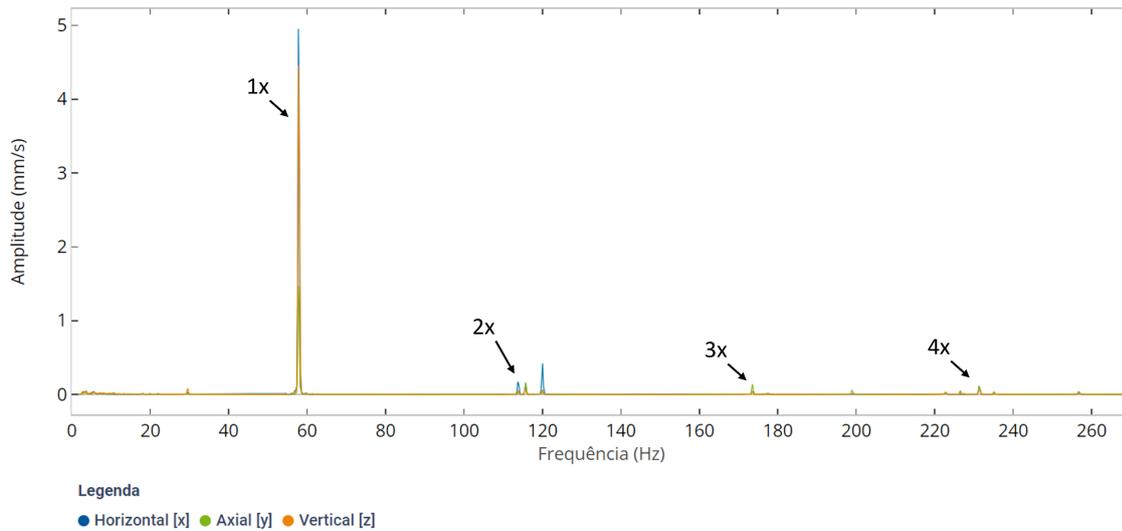
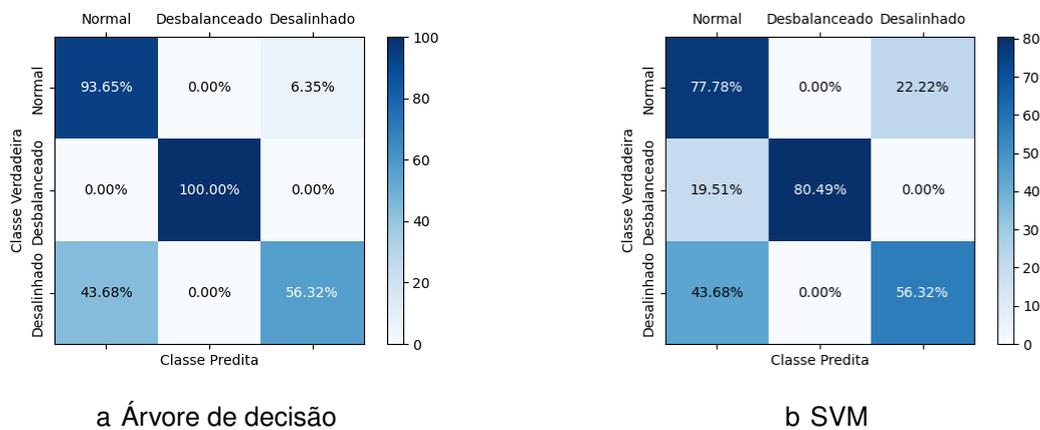


Figura 6.4 – FFT da condição desbalanceado do motor para inferência.

normal. Como a condição se repete para os modelos de árvore de decisão e SVM, é minimizada a possibilidade de o erro ser proveniente dos modelos. Já nas condições de motor normal e desbalanceado, o modelo de SVM teve mais classificações incorretas, quando comparado com o modelo de árvore de decisão.

Figura 6.5 – Matriz de confusão do ensaio de inferência.



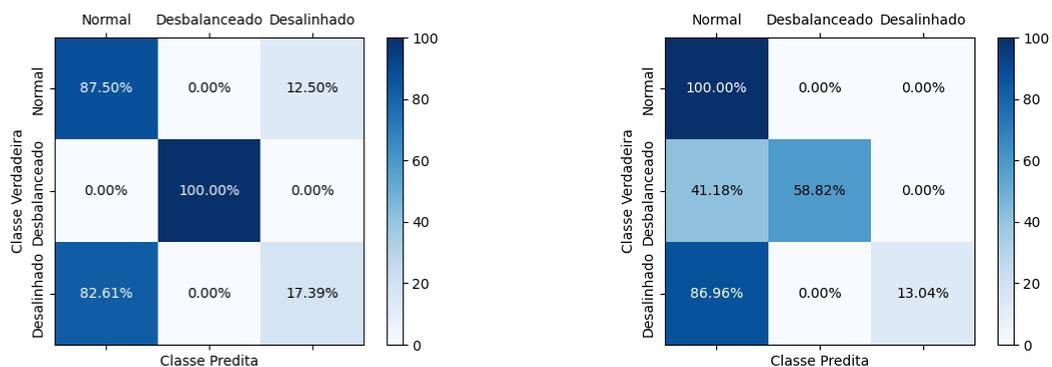
A fim de entender melhor o comportamento dos modelos nos testes de inferência, na Tabela 6.3 abaixo são mostrados os resultados encontrados para cada condição de carga. A condição de motor desalinhado com 0% de carga foi a maior causadora de erros de classificação. Através das matrizes de confusão mostradas na Figura 6.6, pode ser observado que a maior confusão ocorreu entre as classificações de motor normal e desalinhado.

Outra condição onde houve grande quantidade de erros de classificação foi para o modelo de SVM na condição de carga de 100%. Diferentemente do caso anterior, onde ambos os modelos tiveram o mesmo comportamento, neste caso apenas o

Tabela 6.3 – Acurácia dos Modelos de Árvore de Decisão e SVM em Diferentes Cargas e Condições de Falha.

Modelo	Carga	Condição		
		Normal	Desbalanceado	Desalinhado
Árvore de Decisão	0%	87.5	100.0	17.4
	50%	100.0	100.0	56.8
	100%	100.0	100.0	100.0
SVM	0%	100.0	58.8	13.0
	50%	100.0	89.8	59.1
	100%	6.7	75.0	100.0

Figura 6.6 – Matriz de confusão do ensaio de inferência para 0% de carga.



a Matriz de confusão da árvore de decisão

b Matriz de confusão do SVM

modelo de SVM teve acurácia baixa. Na matriz de confusão dessa condição mostrada na Figura 6.7 é visto que todas as amostras mal classificadas foram consideradas de motor desalinhado, sendo mais um caso de confusão entre as duas condições.

No ensaio de inferência, foi identificado que alguns modelos tiveram dificuldade de classificar entre normal e desalinhado para certas amostras. Obaid [2003] fazem uma avaliação do impacto da carga na detecção de desalinhamento angular através do sinal de vibração. No estudo, é mostrado que há um grande aumento no nível de vibração da harmônica de 3x a frequência de rotação mecânica quando a carga é variada, principalmente entre os níveis de 10 a 70% de carga. Foi feita a avaliação prática da bancada de testes, comparando os sinais de espectro de vibração da condição desalinhada para cada variação de carga para identificar a causa dos erros de classificação.

Na Figura 6.8 pode-se notar que o pico que representa as múltiplas da rotação mecânica é deslocado para esquerda com o aumento da carga, isso se dá devido ao

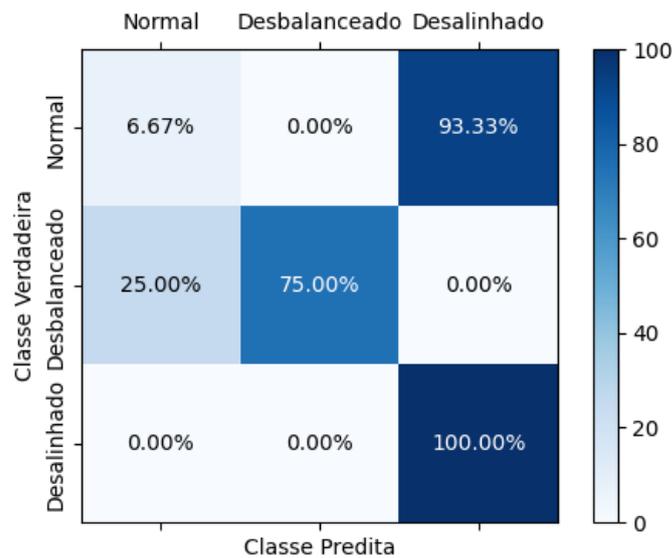


Figura 6.7 – Matriz de confusão do ensaio de inferência para o modelo de SVM com 100% de carga.

aumento do escorregamento nessas condições. Apesar disso, o sensor WEGscan100 é preparado para identificar o pico verdadeiro de rotação para cada condição. Pode ser vista uma diferença significativa entre as harmônicas de 3 e 4x a frequência de rotação mecânica conforme a carga é aumentada. A condição desalinhada com 0% de carga se assemelha ao que foi visto na Figura 6.2 para a motor normal, dando a entender que a confusão de classificação pode ter sido gerada gerada pela dificuldade de diferenciação entre as duas condições.

6.2.3 Inferência para modelo não supervisionado

Os dados adquiridos no teste de inferência foram testados com o modelo de One-Class SVM para diferenciar entre as condições normal e de falha. A matriz de confusão da Figura 6.9 mostra o resultado encontrado. O modelo classificou corretamente 98.2% dos dados de falha, porém errou ao classificar 90.5% dos dados normais. Na Tabela 6.4 são mostrados os resultados de acurácia, precisão e recall para as classificações do modelo. A proporção de dados de falha é maior do que de dados normais no conjunto testado, pois são considerados os dados de falha de desalinhamento e desbalanceamento. Devido a isso, e pela capacidade em classificar corretamente os dados de falha, a acurácia geral do modelo RBF encontrada foi de 78.7%. A acurácia geral é razoavelmente alta, mas a discrepância entre a acurácia para dados de falha e dados normais indica que o modelo pode estar enviesado para detectar falhas. A precisão para ambos os dados foi de 50%, indicando que há uma grande quantidade de falsos positivos entre os dados classificados como falha. O recall encontrado para a condição normal é muito baixo, sugerindo que o modelo não é confiável na detecção

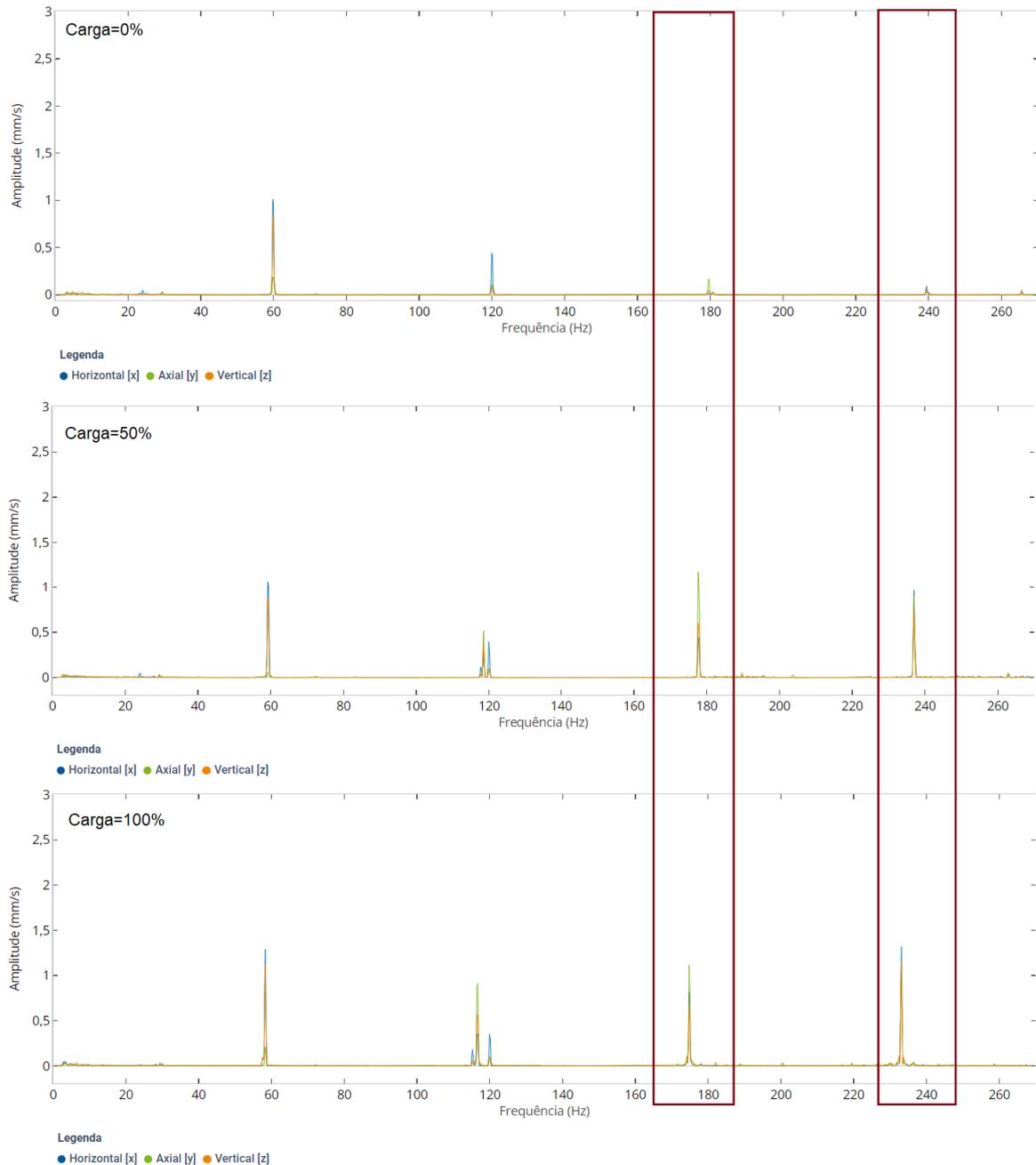


Figura 6.8 – FFT da condição desalinhado com 0, 50 e 100% de carga.

de estados normais.

Embora o modelo tenha um bom desempenho na detecção de falhas, ele não se mostra confiável para distinguir entre estados normais e falhos. Como a acurácia durante o treinamento foi muito superior ao encontrado na inferência, pode ser um indício de que o modelo gerado está sobreajustado para os dados de treinamento.

Como visto na seção 5.2.3, o resultado encontrado para o kernel polinomial também foi satisfatório, e devido ao resultado insuficiente encontrado com o kernel RBF o mesmo foi avaliado para os dados de inferência. A matriz de confusão da Figura 6.10 mostra o resultado obtido. O modelo gerado foi capaz de classificar corretamente 58.7%

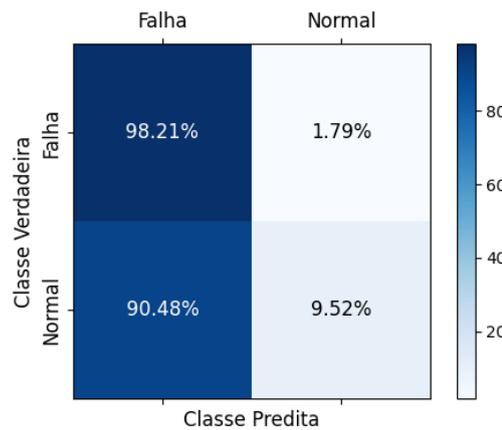


Figura 6.9 – Matriz de confusão do modelo de One-Class SVM com kernel RBF.

Tabela 6.4 – Desempenho do modelo de One-Class SVM com kernel RBF.

	Acurácia(%)	Precisão(%)	Recall(%)
Dados de Falha	98.2	50.0	49.1
Dados normais	9.5	50.0	4.7
Geral	78.7	69.7	53.9

dos dados normais e 85.6% dos dados de falha. A Tabela 6.5 apresenta o resultado das métricas para o kernel polinomial. Neste caso, a acurácia geral encontrada é de 79.0%, levemente superior ao kernel RBF. Isso é explicado pelo fato de que este kernel teve mais problemas na classificação dos dados de falha, que são a maior parte dos dados do conjunto testado. Houve menor diferença na acurácia dos dados normais e de falha, mas o modelo ainda classifica com mais exatidão os dados de falha. A precisão e recall se mantiveram baixos, mostrando a presença de falsos positivos e falsos negativos.

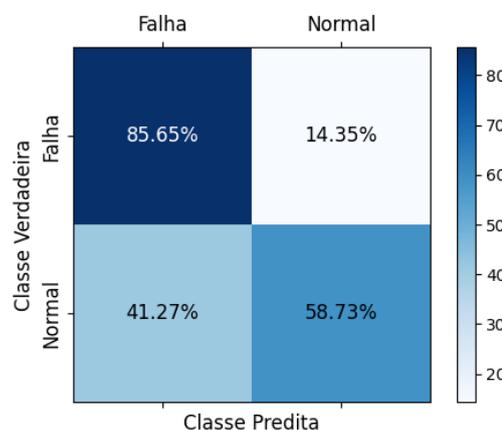


Figura 6.10 – Matriz de confusão do modelo de One-Class SVM com kernel polinomial.

Tabela 6.5 – Desempenho do modelo de One-Class SVM com kernel polinomial.

	Acurácia(%)	Precisão(%)	Recall(%)
Dados de Falha	85.6	50.0	42.8
Dados normais	58.7	50.0	29.4
Geral	79.0	70.0	71.7

7 CONCLUSÕES

O avanço do uso de sensoriamento e tecnologias em indústrias, motivado pelo movimento da indústria 4.0 traz novas perspectivas e oportunidades de aumento da eficiência das máquinas e plantas fabris. Uma das formas de aumento dessa eficiência é através da identificação de falhas e possíveis paradas em motores elétricos, evitando que estes equipamentos quebrem prematuramente e causem paradas imprevistas.

Este estudo avaliou a capacidade de modelos de aprendizado de máquina supervisionado e não supervisionado na identificação de falhas de desbalanceamento e desalinhamento em motores elétricos de indução. O trabalho contemplou as fases de geração da base de dados, treinamento e avaliação dos modelos, portabilidade para um microcontrolador e inferência com novas amostras.

A geração da base de dados foi feita utilizando uma bancada com dois motores no arranjo back-to-back. Foi utilizado o sensor comercial WEGscan100 para coleta dos dados. O conjunto de dados contemplou amostras de dados da condição normal do motor e das condições de falha sob avaliação.

Na fase de treinamento dos modelos supervisionados, todos os dados do conjunto foram utilizados para treinamento e avaliação com os estimadores. Já o modelo não supervisionado, utilizou apenas os dados da condição normal do motor em seu treinamento. Foram utilizadas técnicas de validação cruzada e variação de parâmetros para definição dos melhores modelos para o problema em questão.

Os modelos criados foram convertidos para linguagem C e portados para o microcontrolador. O sensor com os modelos embarcados foi fixado no motor e realizaram a inferência sobre novas amostras variando as condições de falha da bancada de testes. Ao final, foi constatada a capacidade dos modelos supervisionados em identificar as falhas, tendo acurácia geral superior a 70% para ambos os modelos de árvore de decisão e SVM. O modelo não supervisionado com o kernel RBF teve um bom resultado na identificação de falhas, porém teve dificuldade em classificar as amostras normais, gerando muitos falsos positivos, mas com acurácia geral de 78.7 devido ao desbalanceamento entre os dados normais e de falha. Já utilizando o kernel polinomial, foi encontrado um melhor resultado para os dados normais, porém com pouco impacto na acurácia geral (79%), novamente devido a maior quantidade de dados de falha do que normais. Para o modelo de One-Class SVM são necessários ajustes, como balanceamento dos dados de treinamento, alteração dos hiperparâmetros ou uso de técnicas adicionais de pré-processamento e seleção de características.

Considerando o uso prático dos modelos estudados, o mais adequado para uma aplicação industrial é o modelo não supervisionado, pois a necessidade de possuir dados das falhas dos motores já na fase de treinamento pode ser um limitador no uso da tecnologia. Para uma máquina em operação, coletar esses dados implicaria em

necessidade de intervenção no funcionamento do equipamento.

Como continuação deste trabalho, o uso de outros algoritmos não supervisionados como por exemplo, Redes Neurais Artificiais (RNA) do tipo Autoencoder podem ser uma alternativa para melhor identificação das falhas utilizando menos dados para treinamento.

REFERÊNCIAS

aaaa.

Youssef Abadade, Anas Temouden, Hatim Bamoumen, Nabil Benamar, Yousra Chtouki, and Abdelhakim Senhaji Hafid. A comprehensive survey on tinyml. *IEEE Access*, 11: 96892–96922, 2023. doi: <10.1109/ACCESS.2023.3294111>.

ABNT. *NBR 5462: Confiabilidade e Manutenibilidade*. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, Rio de Janeiro, 1994.

Intesar Ahmed, Manzar Ahmed, Kashif Imran, Muhammad Khan, and S. JunaidAkhtar. Detection of eccentricity faults in machine using frequency spectrum technique. *International Journal of Computer and Electrical Engineering*, pages 111–119, 01 2011. doi: <10.7763/IJCEE.2011.V3.300>.

Siddique Akbar, Toomas Vaimann, Bilal Asad, Ants Kallaste, Muhammad Sardar, and Karolina Kudelina. State-of-the-art techniques for fault diagnosis in electrical machines: Advancements and future directions. *Energies*, 16:6345, 09 2023. doi: <10.3390/en16176345>.

Mattia Antonini, Miguel Rodrigo Pincheira Caro, Massimo Vecchio, and Fabio Antonelli. A tinyml approach to non-repudiable anomaly detection in extreme industrial environments. pages 397–402, 06 2022. doi: <10.1109/MetroInd4.0IoT54413.2022.9831517>.

Alvaro Bazurto, Enrique Quispe, and Rosaura Castrillon. Causes and failures classification of industrial electric motor. 10 2016. doi: <10.1109/ANDESCON.2016.7836190>.

Pradhan S. Prakash M. Bhowmik, P. S. Fault diagnostics and monitoring methods of induction motor :a review. *International Journal of Applied Control, Electrical and Electronics Engineering(IJACEEE)*, 1(1), 2013.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media, LLC, Cambridge, UK, 2006.

Giuseppe Bonaccorso. *Machine Learning Algorithms*. Packt Publishing Ltd, Birmingham, UK, 2017.

A. Bonnett and C. Young. Explaining motor failure. 103:22–26, 10 2004.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, page 144–152, New York, NY, USA, 1992. Association for Computing Machinery. ISBN 089791497X. doi: <10.1145/130385.130401>. URL <https://doi.org/10.1145/130385.130401>.

Brasil. Lei nº 13.709, de 14 de agosto de 2018. dispõe sobre a proteção de dados pessoais e altera a lei nº 12.965, de 23 de abril de 2014 (marco civil da internet). *Diário Oficial [da] República Federativa do Brasil*, 2018. URL <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm.>

J. H. Friedman R. A. Olshen Breiman, L. and P. J. Stone. Classification and regression trees. *Wadsworth*, 1984.

- Anurag Choudhary, Deepam Goyal, Shimi Sudha Letha, and Aparna Akula. Condition monitoring and fault diagnosis of induction motors: A review. *Archives of Computational Methods in Engineering*, 26, 09 2018. doi: <10.1007/s11831-018-9286-z>.
- Bram Corne, Jos Knockaert, and Jan Desmet. Misalignment and unbalance fault severity estimation using stator current measurements. pages 247–253, 2017. doi: <10.1109/DEMPED.2017.8062363>.
- Lucas Q. M. da Costa and Cristiano A. V. Cavalcante. A review on the study of maintenance effectiveness. *Pesquisa Operacional*, 42, 2022. doi: <10.1590/0101-7438.2022.042nspe1.00263613>.
- Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Shlomi Regev, Rocky Rhodes, Tiezhen Wang, and Pete Warden. Tensorflow lite micro: Embedded machine learning on tinyml systems, 2020. URL <<https://arxiv.org/abs/2010.08678>>.
- Miguel Delgado-Prieto, Giansalvo Cirrincione, Antonio Garcia Espinosa, Juan Ortega, and Humberto Henao. Bearing fault detection by a novel condition-monitoring scheme based on statistical-time features and neural networks. *Industrial Electronics, IEEE Transactions on*, 60:3398–3407, 08 2013. doi: <10.1109/TIE.2012.2219838>.
- Ayon Dey. Machine learning algorithms : A review. 2016.
- M. El Hachemi Benbouzid. A review of induction motors signature analysis as a medium for faults detection. *IEEE Transactions on Industrial Electronics*, 47(5):984–993, 2000. doi: <10.1109/41.873206>.
- Jawad Faiz, Bashir Mahdi Ebrahimi, Bilal Akin, and Hamid A. Toliyat. Finite-element transient analysis of induction motors under mixed eccentricity fault. *IEEE Transactions on Magnetics*, 44(1):66–74, 2008. doi: <10.1109/TMAG.2007.908479>.
- Fluke. 13 common causes of motor failure. <<https://www.fluke.com/en-gb/learn/blog/motors-drives-pumps-compressors/13-common-causes-of-motor-failure>>. Acessado em 12-09-2022.
- Purushottam Gangsar, Rohit Kumar Pandey, and Manoj Chouksey. Unbalance detection in rotating machinery based on support vector machine using time and frequency domain vibration features. *Noise & Vibration Worldwide*, 52(4-5):75–85, 2021. doi: <10.1177/0957456521999836>.
- Joel Grus. *Data Science from Scratch: First Principles with Python*. O'Reilly Media, Inc., California, USA, 2019.
- RESNIK Robert KRANE Denneth S. HALLIDAY, David. *Física 3*. Ed. Rio de Janeiro., Rio de Janeiro, 2004.
- Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011. ISBN 0123814790.
- A.S.N Huda and Soib Taib. Application of infrared thermography for predictive/preventive maintenance of thermal defect in electrical equipment. *Applied Thermal Engineering*, 61, 08 2013. doi: <10.1016/j.applthermaleng.2013.07.028>.

- Sarker IH. Machine learning: Algorithms, real-world applications and research directions. 2021. doi: <doi:10.1007/s42979-021-00592-x>.
- Gian Marco Iodice and Ronan Naughton. *TinyML Cookbook*. Packt Publishing, 2022.
- Arun Kr. Jalan and Amiya Ranjan Mohanty. Model based fault diagnosis of a rotor-bearing system for misalignment and unbalance under steady-state condition. *Journal of Sound and Vibration*, 327:604–622, 2009.
- Pratesh Jayaswal, Arun Wadhvani, and Kavya Mulchandani. Machine fault signature analysis. *International Journal of Rotating Machinery*, 2008, 02 2008. doi: <10.1155/2008/583982>.
- Rajvardhan Jigyasu, Amandeep Sharma, Lini Mathew, and Shantanu Chatterji. A review of condition monitoring and fault diagnosis methods for induction motor. pages 1713–1721, 06 2018. doi: <10.1109/ICCONS.2018.8662833>.
- Shehroz S. Khan and Michael G. Madden. A survey of recent trends in one class classification. In Lorcan Coyle and Jill Freyne, editors, *Artificial Intelligence and Cognitive Science*, pages 188–197, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-17080-5.
- Riyah Kiter, Mohammed Moanes Ali, Yousif Al Mashhadany, and Fuad Salim. Detection and diagnosis of induction motor faults by intelligent techniques. *Journal of Engineering*, 23:29–47, 01 2017.
- Liangzhen Lai, Naveen Suda, and Vikas Chandra. Cmsis-nn: Efficient neural network kernels for arm cortex-m cpus, 2018. URL <https://arxiv.org/abs/1801.06601>.
- Yong Lee, Bok-Kyung Kim, Jun-Hee Bae, and Kyung Kim. Misalignment detection of a rotating machine shaft using a support vector machine learning algorithm. *International Journal of Precision Engineering and Manufacturing*, 22, 02 2021. doi: <10.1007/s12541-020-00462-1>.
- Li Lin, Xiaofei Liao, Hai Jin, and Peng Li. Computation offloading toward edge computing. *Proceedings of the IEEE*, 107:1584–1607, 07 2019. doi: <10.1109/JPROC.2019.2922285>.
- Shih-Lin Lin. Application of machine learning to a medium gaussian support vector machine in the diagnosis of motor bearing faults. *Electronics*, 10(18), 2021. ISSN 2079-9292. doi: <10.3390/electronics10182266>. URL <https://www.mdpi.com/2079-9292/10/18/2266>.
- Ana Carolina Lorena and André C.P.L.F. de Carvalho. Evolutionary tuning of svm parameter values in multiclass problems. *Neurocomputing*, 71(16):3326–3334, 2008. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2008.01.031>. URL <https://www.sciencedirect.com/science/article/pii/S092523120800307X>. Advances in Neural Information Processing (ICONIP 2006) / Brazilian Symposium on Neural Networks (SBRN 2006).
- Rafael Schincariol da Silva Luiz Carlos de Freitas Júnior. *Máquinas elétricas*. Editora e Distribuidora Educacional S.A., Londrina - PR - Brasil, 2018.

Macdara MacCamhaoil. Static and dynamic balancing of rigid rotors. *Bruel & Kjaer application notes, BO*, pages 0276–12, 2016.

I.G. Maglogiannis. *Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies*. Frontiers in artificial intelligence and applications. IOS Press, 2007. ISBN 9781586037802. URL <https://books.google.com.br/books?id=vLiTXDhr_sYC>.

Batta Mahesh. Machine learning algorithms -a review. 01 2019. doi: <10.21275/ART20203995>.

Dionisio Henrique Carvalho De Sá Martins, Denys Viana, Amaro Lima, Milena Faria Pinto, Luis Tarrataca, Fabricio Silva, Ricardo Gutierrez, Thiago Prego, Ulisses Monteiro, and Diego Haddad. Diagnostic and severity analysis of combined failures composed by imbalance and misalignment in rotating machines. *The International Journal of Advanced Manufacturing Technology*, 114, 04 2021. doi: <10.1007/s00170-021-06873-2>.

David Martínez-Rego, Oscar Fontenla-Romero, Amparo Alonso-Betanzos, and José C. Principe. Fault detection via recurrence time statistics and one-class classification. *Pattern Recognition Letters*, 84:8–14, 2016. ISSN 0167-8655. doi: <<https://doi.org/10.1016/j.patrec.2016.07.019>>. URL <<https://www.sciencedirect.com/science/article/pii/S0167865516301817>>.

NEELAM MEHALA. *CONDITION MONITORING AND FAULT DIAGNOSIS OF INDUCTION MOTOR USING MOTOR CURRENT SIGNATURE ANALYSIS*. ELECTRICAL ENGINEERING DEPARTMENT NATIONAL INSTITUTE OF TECHNOLOGY, KURUKSHETRA, INDIA, 2010.

Oliver Mey, Willi Neudeck, André Schneider, and Olaf Enge-Rosenblatt. Machine learning-based unbalance detection of a rotating shaft using vibration data. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 1610–1617, 2020. doi: <10.1109/ETFA46521.2020.9212000>.

David Meyer. Support vector machines. the interface to libsvm in package e1071. *R News*, 1, 01 2015.

Tom M. Mitchell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997.

Mohssen Mohammed, Muhammad Khan, and Eihab Bashier. *Machine Learning: Algorithms and Applications*. 07 2016. ISBN 9781498705387. doi: <10.1201/9781315371658>.

Ashkan Moosavian, Hojat Ahmadi, Babak Sakhaei, and Reza Labbafi. Support vector machine and k-nearest neighbour for unbalanced fault detection. *Journal of Quality in Maintenance Engineering*, 20, 03 2014. doi: <10.1108/JQME-04-2012-0016>.

Alireza Mostafavi and Ali Sadighi. A novel online machine learning approach for real-time condition monitoring of rotating machines. In *2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM)*, pages 267–273, 2021. doi: <10.1109/ICRoM54204.2021.9663495>.

- Kevin P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.], 2013. ISBN 9780262018029 0262018020.
- S. Nandi, H.A. Toliyat, and X. Li. Condition monitoring and fault diagnosis of electrical motors—a review. *IEEE Transactions on Energy Conversion*, 20(4):719–729, 2005. doi: <10.1109/TEC.2005.847955>.
- Vladimir Nasteski. An overview of the supervised machine learning methods. *HORIZONS.B*, 4:51–62, 12 2017. doi: <10.20544/HORIZONS.B.04.1.17.P05>.
- Kim-Anh Nguyen, Phuc Do, and Antoine Grall. Multi-level predictive maintenance for multi-component systems. *Reliability Engineering System Safety*, 144, 12 2015. doi: <10.1016/j.ress.2015.07.017>.
- Nils J. Nilsson. Introduction to machine learning. <<https://ai.stanford.edu/~nilsson/MLBOOK.pdf>>, 1998. Acessado em: 9-6-2023.
- T.G. Obaid, R.R. e Habetler. Effect of load on detecting mechanical faults in small induction motors. In *4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, 2003. SDEMPED 2003.*, pages 307–311, 2003. doi: <10.1109/DEMPED.2003.1234591>.
- Marina Paolanti, Luca Romeo, Andrea Felicetti, Adriano Mancini, Emanuele Frontoni, and Jelena Loncarski. Machine learning approach for predictive maintenance in industry 4.0. In *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–6, 2018a. doi: <10.1109/MESA.2018.8449150>.
- Marina Paolanti, Luca Romeo, Andrea Felicetti, Adriano Mancini, Emanuele Frontoni, and Jelena Loncarski. Machine learning approach for predictive maintenance in industry 4.0. In *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–6, 2018b. doi: <10.1109/MESA.2018.8449150>.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Pramuditha Perera, Poojan Oza, and Vishal Patel. One-class classification: A survey. 01 2021.
- Sourav Pradhan, Partha Sarathee Bhowmik, and Mangal Prakash. Fault diagnostic and monitoring methods of induction motor: a review. *International Journal of Applied Control, Electrical and Electronics Engineering*, 1:1–18, 05 2013.
- Rune Prytz, Slawomir Nowaczyk, Thorsteinn Rognvaldsson, and Stefan Byttner. Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. *Engineering Applications of Artificial Intelligence*, 41, 05 2015. doi: <10.1016/j.engappai.2015.02.009>.
- J. R. Quinlan. Induction of decision trees. *Machine Learning 1*, page 81–106, 1986.

- J. R. Quinlan. Programs for machine learning. *Morgan Kaufmann Publishers, Inc*, 1993.
- D. Weber R. Cardinali. Diagnosis on hydroelectric machinery, international conference on vibration in rotating machinery. pages 461–467, 1992.
- Visal Rajapakse, Ishan Karunanayake, and Nadeem Ahmed. Intelligence at the extreme edge: A survey on reformable tinyml. *ACM Comput. Surv.*, feb 2023. ISSN 0360-0300. doi: <10.1145/3583683>. URL <<https://doi.org/10.1145/3583683>>. Just Accepted.
- Zhou X. Lin P. Wen Y. Deng R. Ran, Y. A survey of predictive maintenance: Systems, purposes and approaches. *arXiv*, 2019. doi: <10.48550/arXiv.1912.07383>.
- Partha Pratim Ray. A review on tinyml: State-of-the-art and prospects. *Journal of King Saud University - Computer and Information Sciences*, 34(4):1595–1623, 2022. ISSN 1319-1578. doi: <<https://doi.org/10.1016/j.jksuci.2021.11.019>>. URL <<https://www.sciencedirect.com/science/article/pii/S1319157821003335>>.
- Susmita Ray. A quick review of machine learning algorithms. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 35–39, 2019. doi: <10.1109/COMITCon.2019.8862451>.
- Haoyu Ren, Darko Anicic, and Thomas Runkler. Tinyol: Tinyml with online-learning on microcontrollers, 2021. URL <<https://arxiv.org/abs/2103.08295>>.
- John T. Renwick and Paul E. Babson. Vibration analysis—a proven technique as a predictive maintenance tool. *IEEE Transactions on Industry Applications*, IA-21(2): 324–332, 1985. doi: <10.1109/TIA.1985.349652>.
- Alireza Sadeghian, Zhongming Ye, and Bin Wu. Online detection of broken rotor bars in induction motors by wavelet packet decomposition and artificial neural networks. *IEEE Transactions on Instrumentation and Measurement*, 58(7):2253–2263, 2009. doi: <10.1109/TIM.2009.2013743>.
- Fouad Sakr, Francesco Bellotti, Riccardo Berta, and Alessandro De Gloria. Machine learning on mainstream microcontrollers. *Sensors*, 20(9), 2020. ISSN 1424-8220. doi: <10.3390/s20092638>. URL <<https://www.mdpi.com/1424-8220/20/9/2638>>.
- Md Abdul Saleem, G Diwakar, and MRS Satyanarayana. Detection of unbalance in rotating machines using shaft deflection measurement during its operation. *IOSR J. Mech. Civ. Eng*, 3(3):08–20, 2012.
- Bernhard Schölkopf, John Platt, John Shawe-Taylor, Alexander Smola, and Robert Williamson. Estimating support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 07 2001. doi: <10.1162/089976601750264965>.
- Bernhard Schölkopf, Alex Smola, Alexander Smola, and A Smola. Support vector machines and kernel algorithms. *Encyclopedia of Biostatistics*, 5328-5335 (2005), 04 2002.
- Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, USA, 2014.

- Hyun Joon Shin, Dong-Hwan Eom, and Sung-Shick Kim. One-class support vector machines—an application in machine fault detection and classification. *Computers Industrial Engineering*, 48(2):395–408, 2005. ISSN 0360-8352. doi: <<https://doi.org/10.1016/j.cie.2005.01.009>>. URL <<https://www.sciencedirect.com/science/article/pii/S0360835205000100>>.
- Jong-Ho Shin and Hong-Bae Jun. On condition based maintenance policy. *Journal of Computational Design and Engineering*, 44, 04 2015. doi: <10.1016/j.jcde.2014.12.006>.
- J.L.H. Silva and António João Marques Cardoso. Bearing failures diagnosis in three-phase induction motors by extended park's vector approach. *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.*, pages 6 pp.–, 2005.
- G.K Singh and Sa'ad Ahmed Saleh Al Kazzaz. Induction machine drive condition monitoring and diagnostic research—a survey. *Electric Power Systems Research*, 64(2):145–158, 2003. ISSN 0378-7796. doi: <[https://doi.org/10.1016/S0378-7796\(02\)00172-4](https://doi.org/10.1016/S0378-7796(02)00172-4)>. URL <<https://www.sciencedirect.com/science/article/pii/S0378779602001724>>.
- Alex Smola and S.V.N. Vishwanathan. *Introduction to Machine Learning*. Cambridge University Press, Cambridge, UK, 2008.
- Liuyang Song, Huaqing Wang, and Peng Chen. Vibration-based intelligent fault diagnosis for roller bearings in low-speed rotating machinery. *IEEE Transactions on Instrumentation and Measurement*, 67(8):1887–1899, 2018. doi: <10.1109/TIM.2018.2806984>.
- Abdenour Soualhi, Kamal Medjaher, and Nouredine Zerhouni. Bearing health monitoring based on hilbert–huang transform, support vector machine, and regression. *IEEE Transactions on Instrumentation and Measurement*, 64(1):52–62, 2015. doi: <10.1109/TIM.2014.2330494>.
- Jolliffe I. T. *Principal Component Analysis*. Springer-Verlag New York, New York, USA, 2002.
- A. E. Umans, Stephen D. e Fitzgerald. *Máquinas Elétricas de Fitzgerald & Kingsley*. McGraw-Hill Companies, 2014.
- Toomas Vaimann, Jan Sobra, Anouar Belahcen, Anton Rassõlkin, Michal Rolak, and Ants Kallaste. Induction machine fault detection using smartphone recorded audible noise. *IET Science, Measurement Technology*, 12, 02 2018. doi: <10.1049/iet-smt.2017.0104>.
- Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, USA, 1999.
- WEG. *GUIA DE ESPECIFICAÇÃO DE MOTORES ELÉTRICOS*. Jaraguá do Sul - SC - Brasil, 2021.
- Manual WEGscan. Wegscan 100. <<https://www.weg.net/catalog/weg/BR/pt/Digital-Solutions/Dispositivos-para-Conectividade-e-Monitoramento/Sensores-de-Monitoramento-de-Condi%C3%A7%C3%A3o/WEGscan/>>

WEGscan-100/SENSOR-IOT-WEGSCAN-100-1-MFM/p/16437262#:~:text=Tamanho%3A%2010%20MB-,WEGscan%20100,-Data%3A%2023/06>, 2023.
Acessado em: 23.06.2024.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52, 1987. ISSN 0169-7439. doi: <[https://doi.org/10.1016/0169-7439\(87\)80084-9](https://doi.org/10.1016/0169-7439(87)80084-9)>. URL <<https://www.sciencedirect.com/science/article/pii/0169743987800849>>. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

V. Wowk. *Machinery vibration: measurement and analysis*. McGraw-Hill, New York, USA, 1991.

R. Yam, Peter Tse, L. Li, and Y.L. Tu. Intelligent predictive decision support system for condition-based maintenance. *International Journal of Advanced Manufacturing Technology*, 17:383–391, 02 2001. doi: <10.1007/s001700170173>.

Yuanyuan Yang, Md Muhie Menul Haque, Dongling Bai, and Wei Tang. Fault diagnosis of electric motors using deep learning algorithms and its application: A review. *Energies*, 14(21), 2021. ISSN 1996-1073. doi: <10.3390/en14217017>. URL <<https://www.mdpi.com/1996-1073/14/21/7017>>.

Ming-Yi You. Predictive component availability optimization with degradation correlated maintainability. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, 227(4):243–253, 2013. doi: <10.1177/0954408912456661>.

Yingfeng Zhang, Shan Ren, Yang Liu, and Shubin Si. A big data analytics architecture for cleaner manufacturing and maintenance processes of complex products. *Journal of Cleaner Production*, 142, 07 2016. doi: <10.1016/j.jclepro.2016.07.123>.

Ye Zhongming and Wu Bin. A review on induction motor online fault diagnosis. volume 3, pages 1353 – 1358 vol.3, 02 2000. ISBN 7-80003-464-X. doi: <10.1109/IPEMC.2000.883050>.