Universidade Federal do Rio Grande do Norte - UFRN

Centro de Tecnologia - CT

Curso de Engenharia Mecatrônica

Monitoramento de Falhas em Motores de Indução Utilizando Inteligência Artificial Embarcada

Sthefania Fernandes Silva

Natal-RN, Brasil

2025

Sthefania Fernandes Silva

Monitoramento de Falhas em Motores de Indução Utilizando Inteligência Artificial Embarcada

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Mecatrônica, da Universidade Federal do Rio Grande do Norte, como requisito parcial à obtenção do título de Bacharel em Engenharia Mecatrônica.

Universidade Federal do Rio Grande do Norte - UFRN

Centro de Tecnologia - CT

Curso de Engenharia Mecatrônica

Orientador: Samaherni Morais Dias

Coorientador: Valbério Gonzaga de Araújo

Natal-RN, Brasil

2025

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Silva, Sthefania Fernandes.
Monitoramento de falhas em motores de indução utilizando inteligência artificial embarcada / Sthefania Fernandes Silva. - 2025.
48f.: il.

Monografia (Graduação) - Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Engenharia Mecatrônica, Natal, 2025.
Orientação: Dr. Samaherni Morais Dias.
Co-orientador: Dr. Valbério Gonzaga de Araújo.

1. Motor de Indução Trifásico - Monografia. 2. Análise de Vibração - Monografia. 3. Redes Neurais - Monografia. 4. ESP32-S3 - Monografia. 5. Sistemas Embarcados - Monografia. I. Dias, Samaherni Morais. II. Araújo, Valbério Gonzaga de. III. Título.

RN/UF/BCZM CDU 621.865.8

Elaborado por Raimundo Muniz de Oliveira - CRB-15/429

Sthefania Fernandes Silva

Monitoramento de Falhas em Motores de Indução Utilizando Inteligência Artificial Embarcada

Monografia apresentada ao curso de graduação em Engenharia Mecatrônica, da Universidade Federal do Rio Grande do Norte, como requisito parcial à obtenção do título de Bacharel em Engenharia Mecatrônica.

Aprovada em: 10 de janeiro de 2025

Prof. Dr. Samaherni Morais Dias
Orientador
Universidade Federal do Rio Grande do Norte

Prof. Dr. Valbério Gonzaga de Araújo
Coorientador
Instituto Federal do Rio Grande do Norte

Prof. Dr. Kurios Iuri Pinheiro de Melo Queiroz
Membro interno
Universidade Federal do Rio Grande do Norte

Prof. Dr. Heitor Medeiros Florencio
Membro interno
Universidade Federal do Rio Grande do Norte

Agradecimentos

Agradeço à minha mãe, que com muito esforço me deu acesso à educação de qualidade e à possibilidade de mudança de realidade. Ao meu irmão, que sempre me desejou boa sorte.

Ao meu namorado, que me motivou nos momentos em que duvidei de mim mesma. Aos meus amigos da universidade, que tornaram os dias mais leves, mesmo em meio às inúmeras provas, trabalhos e projetos. Aos meus amigos e familiares de fora do curso, que, mesmo sem compreender exatamente o que faço, sempre acreditaram em mim.

Ao meu orientador Samaherni pelo auxílio, sugestões e paciência. Ao meu co-orientador Valbério, que, mesmo com nosso contato recente, me auxiliou na coleta de dados e compartilhou generosamente seus conhecimentos.

Agradeço à equipe Embarcados pela generosidade ao fornecer as placas, pelos cursos e pela troca de conhecimentos, que foram fundamentais ao longo dessa jornada.

Resumo

Estima-se que existam cerca de 20 milhões de motores de indução trifásicos em operação, sendo este o tipo mais utilizado na indústria. O motivo é a robustez, versatilidade, baixo custo e alta eficiência. Para garantir um ótimo desempenho e evitar paradas não programadas do motor, é necessário realizar o monitoramento regular das suas condições. Neste contexto, este trabalho busca desenvolver um sistema de monitoramento e detecção preditiva de defeitos em motores de indução trifásicos utilizando uma rede neural perceptron de multicamadas implementada em um microcontrolador ESP32-S3. Para isso, foi realizada a coleta de dados para o treinamento e teste da rede neural em uma bancada de testes, utilizando o acelerômetro ADXL335 e um ESP32. As condições monitoradas incluem: operação normal, desbalanceamento de corrente e defeito no rolamento. Após a coleta, foi adicionado ruído Gaussiano e foi realizada a expansão dos dados. A rede neural foi definida com duas camadas intermediárias "totalmente conectadas", cada uma contendo 8 neurônios e utilizando a função de ativação ReLU, já a camada de saída possui 3 neurônios, correspondentes às classes normal, desbalanceamento e rolamento, e utiliza a função de ativação Softmax. A rede foi treinada e quantizada, sendo implementada no ESP32-S3. O modelo obteve uma acurácia de 98%, atendendo aos objetivos propostos.

Palavras-chave: Motor de Indução Trifásico, Análise de Vibração, Detecção de Falhas, Aprendizado de Máquina, Redes Neurais, ESP32-S3, Sistemas Embarcados.

Abstract

The literature estimates that around 20 million three-phase induction motors are in operation, making them the most commonly used motor in the industry due to their robustness, versatility, low cost, and high efficiency. Periodical condition monitoring is necessary to ensure optimal performance and unplanned motor shutdowns. From this perspective, this work aims to develop a monitoring and predictive fault detection system for three-phase induction motors using a multilayer perceptron neural network implemented on an ESP32-S3 microcontroller. For this purpose, data collection for training and testing the neural network was conducted on a test bench using the ADXL335 accelerometer and an ESP32. The monitored conditions include regular operation, current imbalance, and bearing defect. After data collection, we add Gaussian noise and data augmentation.The neural network was defined with two fully connected intermediate layers, each containing 8 neurons and using the ReLU activation function. The output layer has 3 neurons, corresponding to the classes normal, imbalance, and bearing fault, and uses the Softmax activation function. The neural network was trained and quantized, and implemented on the ESP32-S3. The model achieved an accuracy of 98%, achieving the proposed objectives.

Keywords: Three-Phase Induction Motor, Vibration Analysis, Fault Detection, Machine Learning, Neural Networks, ESP32-S3, Embedded Systems.

Lista de ilustrações

Figura 1 – Motor elétrico. 16
Figura 2 – Distribuição percentual dos componentes com maior ocorrência de falhas. 16
Figura 3 – Fluxograma básico de detecção de falhas em motores. 18
Figura 4 – Sinais no domínio do tempo e da frequência. 19
Figura 5 – Modelo não-linear de um neurônio. 22
Figura 6 – Exemplo de quantização de FP32 para INT8. 23
Figura 7 – Bancada de testes. 25
Figura 8 – Sensor ADXL335. 26
Figura 9 – ESP32. 27
Figura 10 – ESP32-S3-BOX. 27
Figura 11 – Motor com sensor ADXL335. 28
Figura 12 – Esquema de aquisição de dados. 28
Figura 13 – Rolamento usado na condição de operação falha de rolamento. 30
Figura 14 – Circuito para desbalanceamento elétrico. 30
Figura 15 – Fluxograma de aquisição de dados. 30
Figura 16 – Arquitetura da rede neural quantizada. 33
Figura 17 – Fluxograma de implementação da rede com TensorFlowLite. 34
Figura 18 – Vibração do motor em diferentes rotações por minuto na condição normal. 36
Figura 19 – Vibração do motor em diferentes rotações por minuto com desbalanceamento de corrente. 36
Figura 20 – Vibração do motor em diferentes rotações por minuto com rolamento defeituoso. 37
Figura 21 – Evolução da Acurácia de Treinamento por Época. 38
Figura 22 – Resultados da avaliação do modelo com grupo 1 de teste. 39
Figura 23 – Resultados da avaliação do modelo com grupo 2 de teste. 40
Figura 24 – Comparação de acurácia entre o modelo base e o modelo quantizado. . 40
Figura 25 – Comparação do tamanho do modelo base e o modelo quantizado. . . . 41
Figura 26 – Resultados da inferência no ESP32-S3. 42
Figura 27 – Resultado da inferência fazendo uma previsão incorreta. 42

Lista de tabelas

Tabela 1 – Causas das falhas em rolamentos, estatores e rotores de MITs. 17
Tabela 2 – Frequência de rotação e de falhas do rolamento 6204-2Z da SKF a 1200 RPM. 20
Tabela 3 – Frequência de rotação e de falhas do rolamento 6204-2Z da SKF a 1800 RPM. 20
Tabela 4 – Frequência de rotação e de falhas do rolamento 6203-2Z da SKF a 1200 RPM. 20
Tabela 5 – Frequência de rotação e de falhas do rolamento 6203-2Z da SKF a 1800 RPM. 21
Tabela 6 – Características do Motor Trifásico de Indução WEG Gaiola. 25
Tabela 7 – Características do acelerômetro ADXL335. 26
Tabela 8 – Características do Módulo ESP32-DEVKit-V1. 26
Tabela 9 – Características do Hardware - ESP32S3-BOX. 27
Tabela 10 – Condições do motor coletadas. 29

Lista de abreviaturas e siglas

FFT	Transformada Rápida de Fourrier
WT	Transformada de Wavelet
FEM	Métodos de Elementos Finitos
IA	Inteligência Artificial
MLP	*Multi Layer Perceptron*
PMC	Perceptron de Múltiplas Camadas
ML	*Machine Learning*
RBF	*Radial Basis Function*
RNA	Redes Neurais Artificiais
DC	*Direct Current*
MIT	Motor de Indução Trifásico
IEEE	*Institute of Eletrical and Electronics Engineers*
EPRI	*Electric Power Research Institute*

Lista de símbolos

Ω	unidade de medida da resistência elétrica (ohm).
V	unidade de medida da tensão elétrica (volts).
A	unidade de medida da corrente elétrica (amperes).
W	unidade de medida de potência (watts).
cv	unidade de medida de potência (cavalo-vapor).
Hz	unidade de medida de frequência (hertz).
RPM	unidade de medida de velocidade angular (rotações por minuto).
s	unidade de medida de tempo (segundos).
m	unidade de medida de comprimento (metros).
g	unidade de aceleração definida como 9,80665 m/s^2
°C	unidade de temperatura na escala Celsius (graus celsius)
k	prefixo do SI que indica que a unidade de medida padrão foi multiplicada por mil (quilo).
M	prefixo do SI que indica que a unidade de medida padrão foi multiplicada por um milhão (mega).
m	prefixo do SI que indica que a unidade de medida padrão foi dividida por mil (mili).
μ	prefixo do SI que indica que a unidade de medida padrão foi divida por um milhão (micro).

Sumário

1 INTRODUÇÃO 12
1.1 Objetivos 13
1.2 Estrutura do Trabalho 13

2 FUNDAMENTAÇÃO 15
2.1 Motor de Indução Trifásico (MIT) 15
2.1.1 Principais Falhas dos MITs 16
2.2 Procedimento de Diagnóstico de Falhas em MITs 17
2.3 Técnicas para Detecção de Falhas em MITs 18
2.3.1 Análise de Vibração Utilizando FFT 18
2.3.1.1 Falhas no Rolamento 19
2.3.1.2 Desbalanceamento Elétrico 21
2.4 Classificação de Falhas Utilizando Aprendizado de Máquina 21
2.4.1 Redes Neurais Artificiais 22
2.5 Sistemas Embarcados 22
2.6 Quantização 23

3 DESENVOLVIMENTO 24
3.1 Definição dos Hardwares Utilizados no Desenvolvimento do Projeto 24
3.2 Coleta de Dados 28
3.3 Condições Monitoradas 29
3.4 Tratamento dos Dados 31
3.5 Arquitetura da Rede Neural 31

4 RESULTADOS 35
4.1 Construção da Base de Dados 35
4.1.1 Treinamento da Rede Neural 37
4.1.2 Desempenho da Rede Neural 38
4.2 Implementação da Rede Neural no ESP32-S3 39
4.3 Discussão dos Resultados 42

5 CONCLUSÕES 44

REFERÊNCIAS 45

1 Introdução

Em 2017, de acordo com dados levantados pela Empresa de Pesquisa Energética 35,7% do consumo de energia elétrica no Brasil foi utilizada pelo setor industrial e 60% desse consumo foi destinado aos sistemas motrizes. Isso significa que os motores representam mais de 26% do consumo total de eletricidade no país (GGCE, 2020).

Neste contexto, o motor de indução trifásico (MIT) é o mais utilizado na indústria nas últimas décadas (KOLEHMAINEN, 2017; SANTOS, 2021), estimando-se aproximadamente 20 milhões de unidades em operação. O motivo é a robustez, versatilidade, baixo custo e alta eficiência. Para garantir um desempenho operacional ideal, maximizar a produtividade e evitar paradas não programadas, é essencial realizar o monitoramento regular das condições dos motores. Nesse sentido, a manutenção desempenha um papel crucial.

A manutenção é definida como um conjunto de medidas para garantir a confiabilidade e o pleno funcionamento dos equipamentos (KARDEC; NASCIF, 2009). Dentre os diferentes tipos de manutenção — preventiva, corretiva e preditiva — que se distinguem pelo tipo de intervenção realizada, a manutenção preditiva tem se destacado nos últimos anos, especialmente em sua aplicação em motores elétricos.

O motivo do destaque é que nesse tipo de manutenção é possível predizer as condições operacionais de um equipamento, através do acompanhamento de parâmetros diversos, identificando problemas ainda em seu estágio inicial. Isso permite que a intervenção para correção seja feita de maneira programada e controlada (KARDEC; NASCIF, 2009).

Considerando isso, observa-se um crescente interesse por sistemas automatizados de manutenção preditiva capazes de realizar esse tipo de diagnóstico de falhas. Em particular, tem havido uma ampla discussão sobre sistemas de manutenção preditiva que utilizam técnicas de *Machine Learning*.

No cenário de detecção de falhas em motores utilizando redes neurais, é possível citar alguns trabalhos. VITOR et al. (2014) propõem o uso de uma rede RBF para a detecção de falhas em motores de indução. Especificamente, a rede foi utilizada para identificar falhas de curto-circuito entre as espiras do estator, quebras das barras do rotor, defeitos nos rolamentos e também para distinguir condições normais de operação. Os autores consideraram que os resultados encontrados foram satisfatórios.

Tavares (2021) desenvolveu uma rede neural com a capacidade de classificar as condições operacionais de máquinas rotativas, visando identificar falhas. Para isso, ele utilizou dados de vibração, considerando diferentes condições de operação, velocidades

de rotação e níveis de defeito e treinou a rede neural com diversas arquiteturas. O autor concluiu que a rede apresentou um bom desempenho.

Nesse contexto, a proposta deste trabalho é desenvolver um sistema de monitoramento das condições de funcionamento de um motor trifásico de indução com rotor em gaiola, capaz de detectar e identificar falhas.

1.1 Objetivos

O objetivo geral deste trabalho é desenvolver um sistema de monitoramento e detecção de defeitos em motores de indução trifásicos utilizando uma rede neural quantizada para ser implementada em um microcontrolador ESP32-S3. Para alcançar este objetivo geral, foram definidos os seguintes objetivos específicos:

- **Treinamento *offline* de RNA**: Testar diferentes tipos de redes neurais, identificando suas vantagens e limitações com relação ao problema de detecção de defeitos. Com base nessa análise, treinar um modelo de Rede Neural Artificial (RNA) que atenda a demanda;
- **Quantização de RNA para ser embarcada**: Realizar a quantização da RNA treinada para implementação em microcontroladores.
- **Implementação da RNA no microcontrolador ESP32-S3**: Adaptar a RNA quantizada utilizando TensorFlow Lite e integrá-lo ao microcontrolador ESP32-S3.

1.2 Estrutura do Trabalho

A monografia é dividida em 5 capítulos, descritos a seguir:

- Na **introdução** temos uma contextualização do tema, uma visão ampla dos trabalhos encontrados na literatura e o objetivo da solução proposta;
- Na **fundamentação** concentra-se todo o embasamento técnico e teórico para o entendimento do trabalho. Assim, o texto inclui a definição e funcionamento de um motor de indução trifásico, as principais falhas que acometem esse tipo de motor, assim como os procedimentos e técnicas para identificar essas falhas. Além disso, é explorada a definição de aprendizado de máquina e redes neurais artificiais que será a técnica utilizada para identificar falhas no motor. Por fim, é definido o conceito de sistemas embarcados;
- No **desenvolvimento** é discutida a metodologia utilizada no trabalho, abrangendo os dispositivos utilizados, como foi realizada a coleta de dados, os testes e a definição

da rede neural e a adaptação do modelo treinado para implementação em um microcontrolador;

- Nos **resultados** é mostrado os detalhes da construção da base de dados, o processo de treinamento da rede, em seguida é feita a análise do desempenho da rede neural. Por último, é feita a discussão dos resultados;
- Na **conclusão** é retomada a motivação do trabalho, o que foi feito e as sugestões para trabalhos futuros.

2 Fundamentação

Neste capítulo, são reunidos todos os aspectos técnicos necessários para a compreensão do trabalho. Começando com o entendimento do motor de indução trifásico, explicando seu funcionamento e suas principais características.

Em seguida, são abordadas as falhas mais comuns que afetam esses motores, como também os procedimentos e técnicas utilizadas para o seu diagnóstico. Direcionando o foco para a análise de vibração utilizando a Transformada Rápida de Fourier (FFT), com ênfase nas falhas que serão avaliadas no trabalho, levando em consideração as alterações na vibração causadas por essas falhas no motor.

Além disso, o capítulo explora o conceito de aprendizado de máquina e redes neurais artificiais, com o objetivo de realizar a classificação das falhas do motor. Por fim, é explicado o conceito de sistemas embarcados e uma técnica de otimização de ponto flutuante conhecida como quantização.

2.1 Motor de Indução Trifásico (MIT)

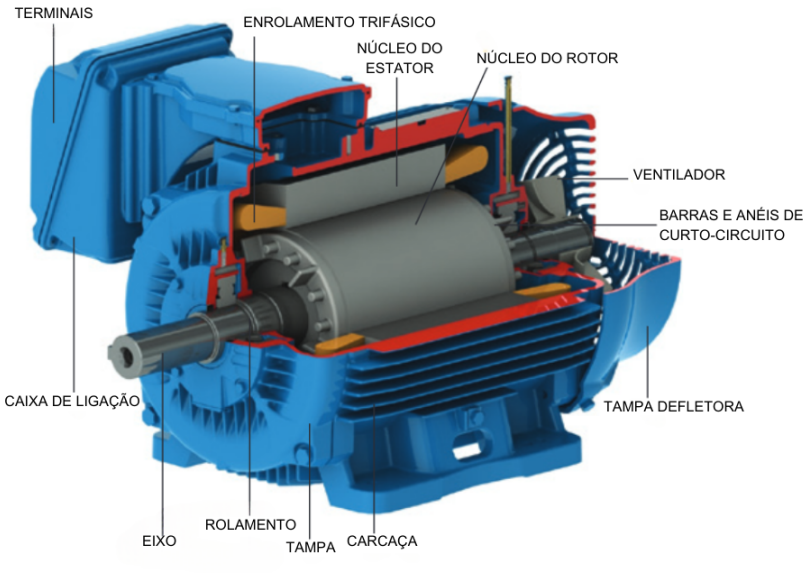
O motor elétrico é uma máquina cujo intuito é transformar energia elétrica em energia mecânica. Um MIT é composto por duas partes principais: o estator e o rotor. O estator, a parte fixa do motor, é formado por três componentes, enquanto o rotor, a parte móvel, também possui três componentes. A Figura 1 exibe um motor com vista em corte para ilustrar os componentes.

O motor de indução é caracterizado por ter apenas o estator conectado à rede de alimentação. O rotor não recebe alimentação externa; as correntes que circulam no rotor são induzidas eletromagneticamente pelo estator, o que justifica o nome "motor de indução" (WEG, 2024).

O princípio de funcionamento do motor de indução é simples: quando as bobinas do estator são percorridas por uma corrente elétrica, um campo magnético girante é criado ao redor do estator. Esse campo magnético induz uma corrente nas barras do rotor devido ao princípio da indução eletromagnética (Lei de Faraday). Como as barras do rotor estão em curto-circuito pelos anéis de extremidade, a corrente pode circular livremente nelas (WEG, 2024).

Essa corrente nas barras do rotor gera seu próprio campo magnético, com polaridade oposta ao campo girante. Campos magnéticos opostos se atraem, e como o campo girante do estator é rotativo, o rotor busca acompanhar a rotação desse campo, gerando o torque necessário para fazê-lo girar (WEG, 2024).

Figura 1 – Motor elétrico.



Fonte: Adaptado de WEG (2024).

2.1.1 Principais Falhas dos MITs

Há diferentes tipos de falhas que podem ocorrer em um MIT e essas falhas normalmente são identificadas conforme sua localização no motor. Segundo Suetake (2012), o *Institute of Eletrical and Electronics Engineers – IEEE* e *Electric Power Research Institute – EPRI* realizaram um levantamento para identificar, entre outras coisas, as características operacionais dos MITs. Por meio deste estudo, foi possível apontar os principais causadores de falhas nos motores, como pode ser visto na Figura 2.

Figura 2 – Distribuição percentual dos componentes com maior ocorrência de falhas.

(a) Levantamento IEEE. (b) Levantamento EPRI.

Fonte: Adaptado de Suetake (2012).

Ambas as pesquisas demonstram que o rolamento, o estator e o rotor são, res-

pectivamente, os componentes mais propensos a falhas. Quando qualquer uma dessas peças apresenta um defeito, independentemente do grupo a que pertença, o motor exibe características específicas como correntes e tensões desbalanceadas, aumento da vibração, aquecimento excessivo, entre outros (REIS, 2010).

No que tange os motivos para essas falhas, destaco na Tabela 1 as causas relacionadas aos defeitos de cada componente do motor.

Tabela 1 – Causas das falhas em rolamentos, estatores e rotores de MITs.

Componente	**Causas Prováveis**
Rolamento	Lubrificação inadequada Contaminação Desalinhamento Carga em excesso
Estator	Sobreaquecimento Curto-circuito Descargas elétricas Contaminação do óleo por umidade e poeira
Rotor	Variações de torque Oscilações de velocidade Vibrações e mudanças de componentes da frequência

Fonte: Adaptado de Reis (2010).

2.2 Procedimento de Diagnóstico de Falhas em MITs

Nesse sentido, de acordo com Reis (2010), o processo de detecção de falhas em MITs, utilizando sistemas de monitoramento, pode ser dividido em três partes:

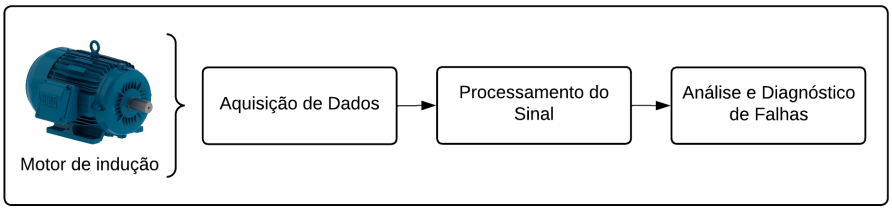
1. Obter dados relevantes do motor por meio de sensores apropriados para detecção de parâmetros importantes, por exemplo: sensores de tensão, corrente, vibração, temperatura, etc;

2. Realizar tratamento do sinal, ajustando-o para uma forma que possa ser mais facilmente interpretado. Há diversas técnicas para isso, tais como: Transformada Rápida de Fourrier (FFT), Transformada de Wavelet (WT), Métodos de Elementos Finitos (FEM), etc;

3. Identificar a falha, determinando o tipo, o impacto e a localização das falhas que possam estar afetando o motor. Os procedimentos para isto são realizados com auxílio de técnicas de Inteligência Artificial (IA), como por exemplo, redes neurais, lógica *fuzzy*, entre outros.

O fluxograma que descreve esse procedimento pode ser visualizado na Figura 3.

Figura 3 – Fluxograma básico de detecção de falhas em motores.



Fonte: Adaptado de Reis (2010).

2.3 Técnicas para Detecção de Falhas em MITs

Para identificar defeitos no motor, existem diversas técnicas, que se diferenciam pela quantidade e tipos de falhas que são capazes de abranger. Dentre elas, Reis (2010) destaca:

a) Análise de vibração;

b) Análise da assinatura elétrica;

c) Análise da temperatura;

d) Análise das flutuações de velocidade;

e) Análise por pulso de choque, entre outros.

Dentre os métodos mencionados, destaca-se a análise de vibração como uma das principais técnicas relacionadas à manutenção preditiva. Ela é fundamental para detectar erros de desbalanceamento, excentricidade de eixo, rolamentos com defeito, falhas nas engrenagens, eixos empenados, cavitação em bombas e desbalanceamento magnético em motores elétricos. Para realizar esse tipo de análise, utiliza-se sensores específicos que possam capturar os sinais de vibração, os quais são posteriormente processados por um software dedicado (NEPOMUCENO, 1989; PEREIRA, 2021).

2.3.1 Análise de Vibração Utilizando FFT

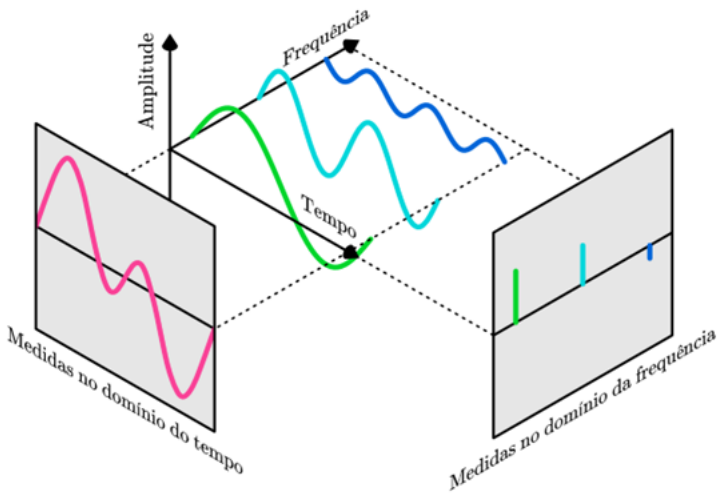
É importante entender que toda máquina rotativa produz vibrações, elas são conhecidas no projeto e consideradas aceitáveis para o funcionamento adequado.

Quando as vibrações são capturadas em um conjunto no tempo, elas geram formas de onda com padrões complexos. Para facilitar a compreensão e análise do sinal, podemos decompor essa forma de onda complexa em componentes individuais utilizando a Transformada Rápida de Fourier (FFT) (HOLANDA, 2016).

Ao realizar a análise com a FFT, é possível visualizar os sinais no domínio da frequência, onde cada componente do sinal original é claramente separado (HOLANDA,

2016). Enquanto que no domínio do tempo observamos a soma de diferentes sinais, o que dificulta a avaliação, como ilustrado na Figura 4.

Figura 4 – Sinais no domínio do tempo e da frequência.



Fonte: Weber (2024).

O espectro "padrão" representa o comportamento esperado do equipamento quando ele está em condição normal de operação (HOLANDA, 2016). Logo, quando o padrão de vibração aumenta em magnitude ou muda sua distribuição em frequências, pode-se caracterizar uma mudança em relação à condição de operação considerada normal. Além disso, cada tipo de defeito nos motores de indução gera um padrão distinto no domínio da frequência (TAVARES, 2021).

2.3.1.1 Falhas no Rolamento

Os defeitos em rolamentos possuem uma evolução lenta e geralmente apresentam sinais de alerta bem antes da falha ocorrer. A degradação de um rolamento pode começar na pista interna ou externa, em algum dos elementos rolantes (rolos ou esferas) ou na gaiola, propagando-se posteriormente para os outros componentes (HOLANDA, 2016).

Naturalmente o rolamento produz um sinal de baixa frequência, cuja frequência depende do número e do tamanho dos elementos rolantes, do ângulo de contato do rolamento e do diâmetro de passo do elemento rolante. Se os elementos rolantes encontrarem um defeito no rolamento, um sinal de alta frequência é gerado, resultando em um pico na amplitude do sinal. A frequência desses picos é influenciada pela rotação, pela posição do defeito no rolamento e pela geometria interna do rolamento (SKF, 2012).

A SKF, empresa especializada no desenvolvimento, projeto e fabricação de rolamentos, dispõe de um programa para o cálculo das frequências associadas a defeitos de rolamentos, o que possibilita a identificação de danos. Na ferramenta, de acordo com o rolamento escolhido e a rotação definida são apresentadas as frequências rotacionais características do rolamento, incluindo: a frequência do anel interno (f_i), a frequência do anel externo (f_e), a frequência do conjunto de elementos rolantes e gaiola (f_c), e a frequência do elemento rolante em seu eixo (f_r). Essas frequências representam os valores nos quais surgem os picos de amplitude durante o funcionamento normal do rolamento.

Além disso, na análise de sobrerolagem, são consideradas as frequências de falha para: o ponto no anel interno (f_{ip}), o ponto no anel externo (f_{ep}) e o elemento rolante (f_{rp}). Essas frequências correspondem aos picos de amplitude gerados quando ocorre um defeito específico no rolamento.

Nas Tabela 2 e Tabela 3 temos os valores de frequência correspondentes a 1200 e 1800 rotações por minuto, respectivamente, para o rolamento 6204-2Z. Enquanto nas Tabela 4 e Tabela 5 são referentes ao rolamento 6203-2Z.

Tabela 2 – Frequência de rotação e de falhas do rolamento 6204-2Z da SKF a 1200 RPM.

Frequência rotacional	f_i (Hz)	f_e (Hz)	f_c (Hz)	f_r (Hz)
	20 Hz	0 Hz	7,63	39,833
Frequência de sobrerolagem	f_{ip} (Hz)	f_{ep} (Hz)	f_{rp} (Hz)	
	98,956	61,044	79,665	

Fonte: Adaptado de SKF (2024).

Tabela 3 – Frequência de rotação e de falhas do rolamento 6204-2Z da SKF a 1800 RPM.

Frequência rotacional	f_i (Hz)	f_e (Hz)	f_c (Hz)	f_r (Hz)
	30 Hz	0 Hz	11,446	59,749
Frequência de sobrerolagem	f_{ip} (Hz)	f_{ep} (Hz)	f_{rp} (Hz)	
	148,435	91,565	119,498	

Fonte: Adaptado de SKF (2024).

Tabela 4 – Frequência de rotação e de falhas do rolamento 6203-2Z da SKF a 1200 RPM.

Frequência rotacional	f_i (Hz)	f_e (Hz)	f_c (Hz)	f_r (Hz)
	20 Hz	0 Hz	7,633	39,874
Frequência de sobrerolagem	f_{ip} (Hz)	f_{ep} (Hz)	f_{rp} (Hz)	
	98,939	61,061	79,747	

Fonte: Adaptado de SKF (2024).

Tabela 5 – Frequência de rotação e de falhas do rolamento 6203-2Z da SKF a 1800 RPM.

Frequência rotacional	f_i (Hz)	f_e (Hz)	f_c (Hz)	f_r (Hz)
	30 Hz	0 Hz	11,449	59,81
Frequência de sobrerolagem	f_{ip} (Hz)	f_{ep} (Hz)	f_{rp} (Hz)	
	148,408	91,592	119,621	

Fonte: Adaptado de SKF (2024).

2.3.1.2 Desbalanceamento Elétrico

Em um sistema equilibrado, as tensões e correntes devem ter a mesma magnitude e uma defasagem de 120º entre si. Qualquer desvio desses padrões resulta em desbalanceamento, que pode ser causado por cargas assimétricas, onde uma das fases carrega mais corrente do que as outras (TOYAMA, 2024).

O desbalanceamento de fases, que é um tipo de desbalanceamento elétrico, pode causar inúmeros problemas, incluindo vibrações excessivas. Como as correntes são desiguais, elas exercem forças desiguais no rotor, resultando em vibrações mecânicas que aceleram o desgaste dos rolamentos e de outros componentes mecânicos. Isso aumenta a necessidade de manutenção e o risco de falhas no sistema (TOYAMA, 2024).

2.4 Classificação de Falhas Utilizando Aprendizado de Máquina

O *Machine Learning* (ML), ou Aprendizado de Máquina, é um campo da inteligência artificial que investiga métodos computacionais para adquirir novos conhecimentos e habilidades (MITCHELL, 1997). As técnicas de ML são orientadas a dados, isso significa que o programa irá aprender automaticamente a partir de grandes conjuntos de dados (LUDERMIR, 2021).

Há três tipos principais de aprendizado de máquina: Supervisionado, Não Supervisionado e Por Reforço, e esses três grupos podem ser subdivididos em subgrupos. De maneira geral, o aprendizado supervisionado se caracteriza pelo treinamento do modelo ser realizado com um conjunto de dados em que as entradas estão associadas a saídas conhecidas. O objetivo do algoritmo é desenvolver um classificador capaz de identificar corretamente uma classe de novos exemplos que ainda não foram rotulados (LUDERMIR, 2021).

Enquanto no aprendizado não supervisionado, os dados submetidos ao algoritmo não possuem rótulos. Nesse treinamento, o algoritmo identifica padrões nos dados e verifica se eles podem ser agrupados de alguma forma. E no aprendizado por reforço, ao invés de rótulos para cada dado de entrada, o algoritmo recebe recompensas ou punições com base nas respostas que oferece (LUDERMIR, 2021).

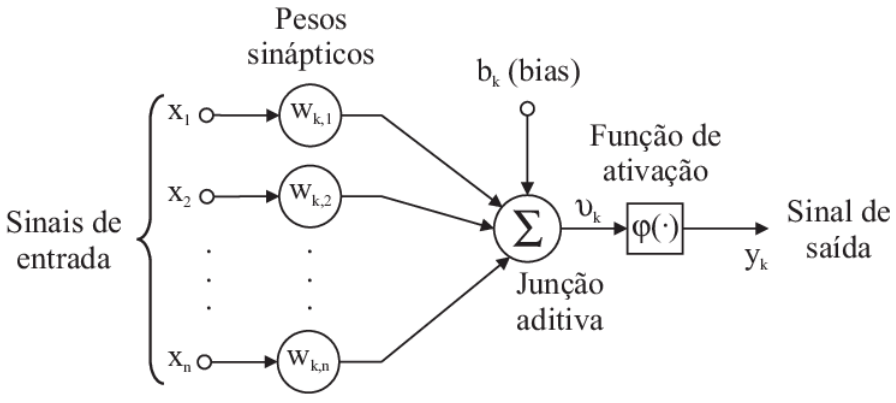
2.4.1 Redes Neurais Artificiais

Dentre as técnicas de Aprendizado de Máquina mais populares para resolver diversos tipos de problemas, destacam-se as Redes Neurais Artificiais. Uma rede neural é um sistema de processamento de informações composto por elementos interconectados, chamados neurônios artificiais ou unidades de processamento, que possuem a capacidade de armazenar conhecimento para realizar tarefas específicas (HAYKIN, 2001).

O Perceptron é a forma mais básica de rede neural, usado para tarefas de classificação e reconhecimento de padrões. Desenvolvido em 1958 por Frank Rosenblatt, o Perceptron foi influenciado pelos estudos de Walter Pitts e Warren Sturgis McCulloch. Essa arquitetura consiste em um único neurônio e apresenta um desempenho limitado a problemas linearmente separáveis (FARIAS; ROSSI, 2021).

A Rede Neural Artificial Perceptron de Múltiplas Camadas (PMC ou MLP — Multi Layer Perceptron) é uma arquitetura de rede que surge para contornar a restrição do Perceptron. A camada de entrada é composta por neurônios que apenas distribuem os sinais para a primeira camada oculta. As camadas ocultas (*hidden layers*), também chamadas de camadas intermediárias, são aquelas situadas entre a camada de entrada e a camada de saída da rede neural, elas possuem seus respectivos pesos e neurônios artificiais, que transferem os sinais de um neurônio para outro, obedecendo às funções de ativação (transferência) de cada neurônio. O treinamento da MLP consiste em ajustar os pesos sinápticos para que a resposta gerada pela rede se aproxime da resposta desejada (FARIAS; ROSSI, 2021). Na Figura 5 temos a representação de uma MLP.

Figura 5 – Modelo não-linear de um neurônio.



Fonte: Haykin (2001).

2.5 Sistemas Embarcados

Sistemas embarcados são sistemas computacionais compactos e de baixo custo, desenvolvidos para atender a necessidades específicas, sendo frequentemente baseados em microcontroladores. Os microcontroladores, por sua vez, são compostos por um único

circuito integrado que reúne um núcleo de processador, memórias voláteis e não voláteis, além de diversos periféricos de entrada e saída de dados. Em resumo, um microcontrolador é um computador de tamanho reduzido, projetado para executar tarefas específicas de forma eficiente e com baixo consumo de recursos (IEEE, 2020).

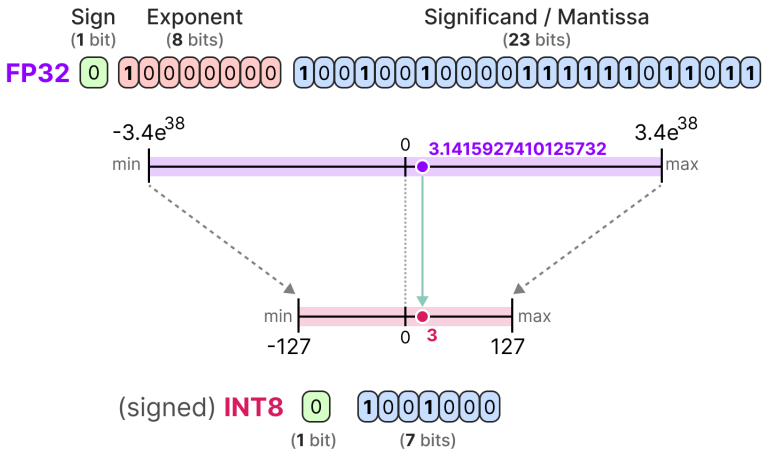
2.6 Quantização

Quando um modelo é treinado, ele gera um grande número de parâmetros, principalmente pesos, dependendo de sua complexidade. Isso pode resultar em altos custos de armazenamento. Durante a inferência, as saídas são calculadas a partir do produto entre a entrada e os pesos, o que também pode gerar números grandes (GROOTENDORST, 2024).

Normalmente, um valor é representado como um número de ponto flutuante (ou floats). Esses valores são representados por bits e quanto mais bits usamos para representar um valor, maior é a sua precisão, geralmente (GROOTENDORST, 2024). No entanto, em sistemas embarcados, é essencial aplicar otimizações para garantir que o hardware utilizado consiga suportar a operação de maneira eficiente. Nesse contexto, é preciso reduzir o número de bits que representam os valores, mantendo a precisão. Para isso, utiliza-se a quantização.

A quantização tem como objetivo reduzir a precisão dos parâmetros de um modelo, passando de larguras de bit maiores (como ponto flutuante de 32 bits) para larguras de bit menores (como inteiros de 8 bits)(GROOTENDORST, 2024). Na Figura 6 temos um exemplo de quantização do valor de π de float 32 bits para um inteiro de 8 bits.

Figura 6 – Exemplo de quantização de FP32 para INT8.



Fonte: Grootendorst (2024).

3 Desenvolvimento

A metodologia utilizada no desenvolvimento do projeto está de acordo com os seguintes passos:

- **Definição de Hardware**: Definir quais hardwares serão utilizados em cada etapa do desenvolvimento.
- **Coleta de Dados**: Coletar dados de vibração do motor em condições normais e defeituosas. Os dados coletados serão convertidos para o domínio da Frequência (FFT), adicionando ruído e padronizando-os para preparar a base de dados para o treinamento do modelo.
- **Análise e Desenvolvimento de Modelos de Redes Neurais**: Testar diferentes tipos de redes neurais, identificando suas vantagens e limitações com relação ao problema de detecção de defeitos. Com base nessa análise, desenvolver um modelo de rede neural quantizado adequado para implementação em microcontroladores.
- **Implementação no Microcontrolador ESP32-S3**: Adaptar o modelo quantizado utilizando TensorFlow Lite e integrá-lo ao microcontrolador ESP32-S3.
- **Eficiência do Modelo Quantizado**: Garantir que a eficiência do modelo quantizado não seja muito prejudicada pela quantização.

Neste capítulo, são detalhados os procedimentos de coleta dos dados, incluindo os dispositivos utilizados e as condições específicas do motor que foram monitoradas.

3.1 Definição dos Hardwares Utilizados no Desenvolvimento do Projeto

A bancada de testes utilizada no projeto possui uma interface de controle, equipada com botões para ligar e desligar o motor, definir sua velocidade, dentre outras coisas. Essa interface está conectada a um motor trifásico de indução com rotor em gaiola, como pode ser visto na Figura 7. As principais características do motor utilizado estão na Tabela 6.

A bancada está localizada no Laboratório de Controle e Acionamento de Sistemas do Departamento de Engenharia de Computação da UFRN.

O sensor utilizado para a coleta da vibração é o acelerômetro ADXL335, cujas características estão detalhadas na Tabela 7. A escolha desse sensor se deu pela sua

Figura 7 – Bancada de testes.

(a) Interface de controle. (b) Motor Trifásico de Indução Gaiola.

Fonte: Autora.

Tabela 6 – Características do Motor Trifásico de Indução WEG Gaiola.

Parâmetro	**Valor**
Potência	1,5 cv
Frequência de Operação	60 Hz
Tensão Nominal	220/380 V
Corrente Nominal	4,43/2,56 A
Fator de Serviço (FS)	1,15
Ip/IN	7,8
N de Polos	4

Fonte: Autora.

disponibilidade e adequação ao problema. Uma fotografia do sensor pode ser vista na Figura 8.

Para a coleta dos dados de vibração do motor, foi utilizado um ESP32 ESP-WROOM-32 DEVKit V1. Na Tabela 8 são apresentadas as especificações técnicas do ESP32, enquanto a sua imagem pode ser vista na Figura 9.

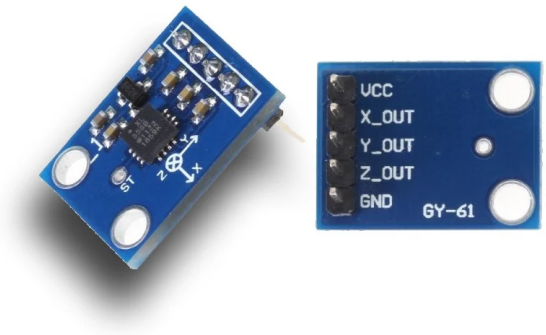
O hardware utilizado para executar as inferências da rede neural e classificar os defeitos foi o ESP32S3-BOX, cujas características estão detalhadas na Tabela 9. A escolha

Tabela 7 – Características do acelerômetro ADXL335.

Parâmetro	**Valor**
Faixa de medição	±3 g
Sensibilidade	330 mV/g
Corrente de fornecimento	400 μA
Resposta em frequência	550 a 1600 Hz
Temperatura de operação	-40 a +85 °C
Precisão	±10%

Fonte: Devices (2010)

Figura 8 – Sensor ADXL335.



Fonte: EletronicWings (2017)

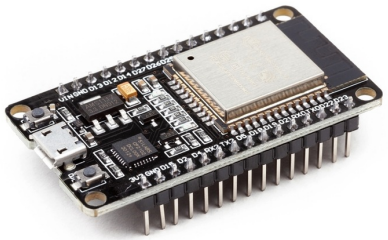
Tabela 8 – Características do Módulo ESP32-DEVKit-V1.

SoC	ESP32
Velocidade da CPU	240 MHz
Memória flash	4 MB
Memória SRAM	520 kB
Memória ROM	448 KB
WiFi	802.11 b/g/n/e/i
Bluetooth(R)	Bluetooth Low Energy v4.2 (BLE)
Alimentação	2,2V a 3,3V DC
GPIOs	34

Fonte: Espressif (2024a)

desse dispositivo se deve ao seu baixo custo aliado à sua alta capacidade de processamento e armazenamento. Na Figura 10 pode-se visualizar o ESP32S3-BOX.

Figura 9 – ESP32.



Fonte: Makiyama (2023)

Tabela 9 – Características do Hardware - ESP32S3-BOX.

SoC	ESP32S3
Velocidade da CPU	240 MHz
Memória flash	16 MB
Memória SRAM	512 kB
Memória PSRAM	8 MB
WiFi	IEEE 802.11 b/g/n
Bluetooth(R) LE	Bluetooth 5 e Bluetooth mesh
Alimentação	USB-C, 5V - 2A
GPIOs	34

Fonte: Espressif (2024b)

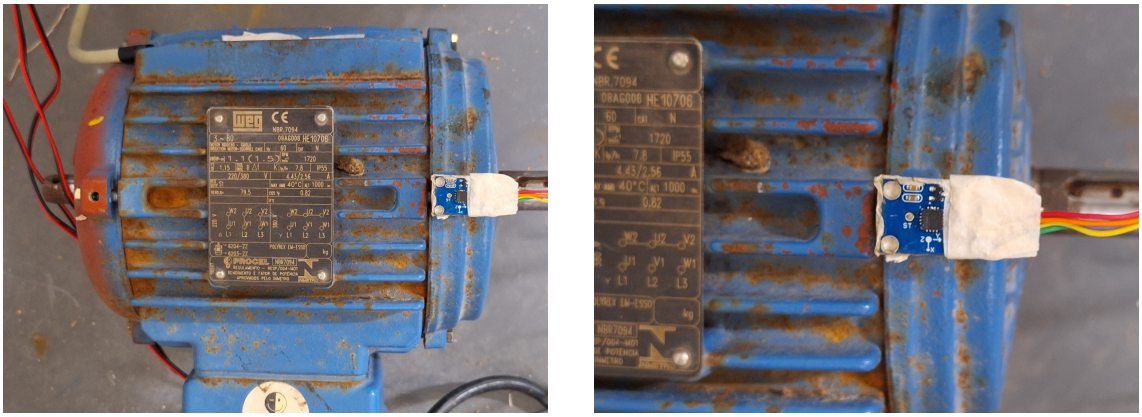
Figura 10 – ESP32-S3-BOX.

Fonte: Espressif (2022)

3.2 Coleta de Dados

O primeiro passo para realizar a coleta dos dados foi a definição do posicionamento do sensoriamento. O ADXL335 foi posicionado no motor conforme ilustrado na Figura 11. Ele foi instalado em um ponto recomendado pela norma ABNT NBR 10082, que estabelece – dentre outras diretrizes – os locais mais adequados para a aferição da vibração.

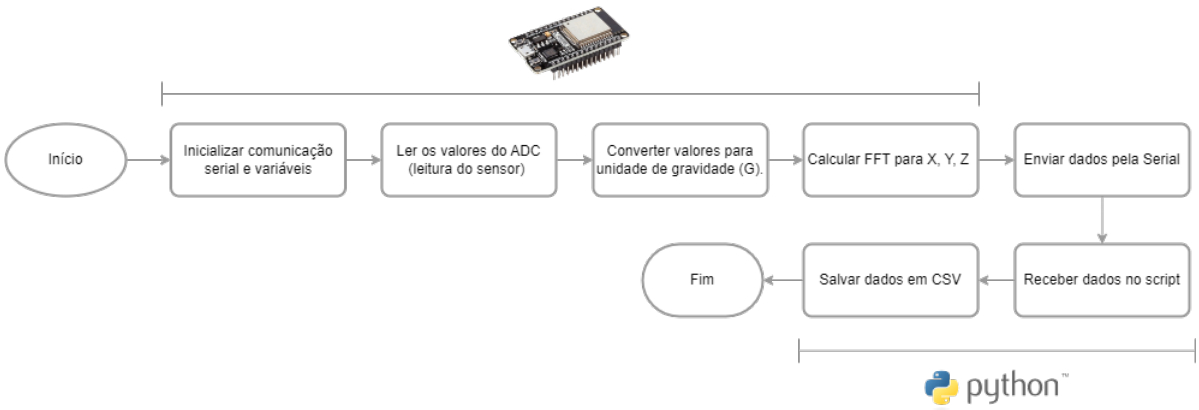
Figura 11 – Motor com sensor ADXL335.



Fonte: Autora.

Para a coleta dos dados de vibração do motor, foi utilizado um ESP32 programado em C utilizando a plataforma Arduino IDE. O código era responsável por capturar os sinais de vibração nos eixos x, y e z. Em seguida, esses dados eram convertidos para o domínio da frequência utilizando FFT diretamente no ESP32. Para salvar os valores, um *script* em Python acessava os sinais de FFT convertidos e os guardava em um arquivo .csv para uso posterior. O fluxograma que descreve o procedimento para aquisição de dados é apresentado na Figura 12.

Figura 12 – Esquema de aquisição de dados.



Fonte: Autora.

A coleta foi realizada em diferentes cenários operacionais, com o objetivo de capturar

uma variedade de comportamentos do motor a diferentes rotações, abrangendo tanto as condições normais quanto as situações de defeito.

3.3 Condições Monitoradas

Baseado no percentual dos componentes que mais apresentam falhas e na disponibilidade de material, foram estabelecidas as condições para o monitoramento do motor.

Foram analisadas três condições de operação: condição normal, falha de rolamento e desbalanceamento elétrico. Durante os testes, o número de rotações por minuto (RPM) do motor variou de 600 a 1800 RPM, com uma variação de aproximadamente 300 RPM.

Na Tabela 10 são apresentadas as condições monitoradas e as respectivas rotações por minuto (RPM) nas quais as amostras do motor foram coletadas.

Tabela 10 – Condições do motor coletadas.

Condição de Operação	**Rotações por Minuto**				
	600	900	1200	1500	1800
Operação Normal	✓	✓	✓	✓	x
Falha de Rolamento	x	x	✓	x	✓
Desbalanceamento Elétrico	✓	✓	✓	✓	x

Fonte: Autora.

A condição de operação normal foi estabelecida utilizando um motor que não apresentava nenhum tipo de problema ou defeito. As medições sob esta condição foram utilizadas como referência para identificar e comparar os espectros de frequência das demais condições operacionais. Já na condição de falha de rolamento, o rolamento foi substituído por outro repleto de desgastes e com pouca lubrificação. O objetivo era observar de forma clara a assinatura espectral do defeito. Na Figura 13, temos o rolamento utilizado para coletar dados do defeito.

Na condição de desbalanceamento elétrico, foi conectado aos terminais do estator um banco de resistores com uma carga ôhmica de 83,5 Ω em série com uma das bobinas do estator, com o intuito de provocar variações nas correntes. A distribuição desigual da carga elétrica no motor resultará em vibrações adicionais. Na Figura 14 temos o circuito utilizado nesta configuração.

De forma resumida, o fluxograma que descreve o processo de aquisição de dados realizado no trabalho é o mostrado na Figura 15. No primeiro bloco, temos a bancada de teste utilizada para coleta de dados. No segundo bloco, temos o sensor utilizado para

Figura 13 – Rolamento usado na condição de operação falha de rolamento.

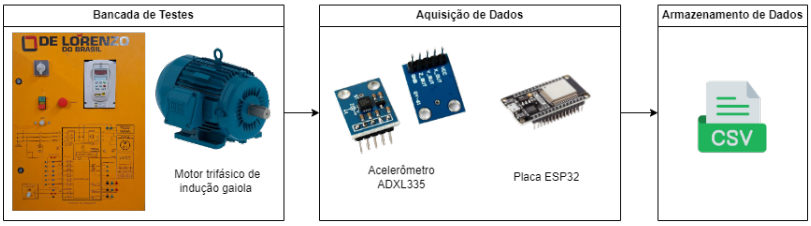
Fonte: (ARAúJO et al., 2024).

Figura 14 – Circuito para desbalanceamento elétrico.

Fonte: (ARAúJO et al., 2024).

medir aceleração e o ESP32 usado para coletar os dados do sensor. No último bloco, temos o arquivo CSV em que os dados são armazenados.

Figura 15 – Fluxograma de aquisição de dados.



Fonte: Autora.

3.4 Tratamento dos Dados

Considere uma amostra, os dados coletados do sensor ao se acionar o motor durante um período de 1 segundo. Assim, cada amostra contém um conjunto de 256 valores representando as amplitudes de aceleração, em unidades de gravidade (g), para cada um dos eixos (x, y e z). O *Dataset* que será utilizado para o treinamento da rede neural é baseado em um conjunto de amostras capturadas com o motor acionado em diferentes velocidades de rotação e em diferentes condições de operação (ver Tabela 10).

Como mencionado anteriormente, os MITs naturalmente vibram e isso pode ser causado por uma série de fatores, como variações na carga ou sua montagem. Nesse sentido, para possibilitar que o modelo aprenda padrões gerais e evitar que ele se ajuste excessivamente aos dados de treinamento (*overfitting*) foi adicionado ruído às amostras.

O ruído adicionado foi o Ruído Branco Gaussiano, que é caracterizado por adicionar valores aleatórios que são independentes e igualmente distribuídos. Nesse sentido, a distribuição dos valores de ruído segue a distribuição normal, com média (μ) de zero e desvio padrão (σ) especificado pelo projetista. No caso aqui descrito, será utilizado $\sigma = 1,6 \cdot 0,0025$ para determinar a amplitude dos valores do ruído ao redor de zero.

Um outro fator importante em coleta de dados experimentais é que a construção de um conjunto de dados com uma quantidade de dados necessários para o correto treinamento da rede pode levar muito tempo e ser muito custoso. Assim, neste trabalho será utilizada uma técnica de aumento de dados baseada no embaralhamento dos dados das amostras coletadas e da injeção de ruído Gaussiano nesses dados.

3.5 Arquitetura da Rede Neural

O ambiente de desenvolvimento escolhido foi o Google Colab, um serviço baseado no Jupyter Notebook hospedado, que não exige configuração para uso e oferece acesso gratuito a recursos de computação, como GPUs e TPUs. No Colab, foi utilizada a linguagem de programação Python que é amplamente empregada em projetos de aprendizado de máquina.

O treinamento do modelo foi realizado com o módulo Keras do pacote "tensorflow_model_optimization". Este pacote é utilizado para otimizar modelos treinados no TensorFlow, permitindo - entre outras coisas - a quantização dos modelos.

Para normalizar os dados de entrada da rede, foi utilizada a biblioteca "StandardScaler". A normalização é realizada em que cada característica (ou coluna) dos dados e o "StandardScaler" normaliza os dados através de

$$n = \frac{v - \mu}{\sigma}, \tag{3.1}$$

em que cada valor normalizado (n) é obtido a partir da subtração entre cada variável (v) do conjunto de dados e o valor médio (μ) dos dados dividido pelo desvio padrão (σ) desses dados, desta forma, ajustando os dados para que apresentem média zero e desvio padrão igual a um.

Para realizar a normalização dos dados testados no ESP32-S3, será salvo um arquivo contendo as médias e os desvios padrão calculados no Colab.

A definição da arquitetura da rede neural envolveu uma série de testes e ajustes para determinar a configuração mais adequada. Inicialmente, foi testada uma rede neural convolucional (CNN). No entanto, dada a simplicidade do problema, observou-se que o modelo convergia com um número muito pequeno de neurônios, tornando desnecessária a complexidade adicional de uma CNN.

Então foram realizados testes com uma rede perceptron de multicamadas. Para isso, diferentes configurações foram experimentadas, começando com 32 neurônios em cada uma das duas camadas intermediárias, em seguida 16, com 8 e, por último, com 4 neurônios.

Para preparar o modelo para o treinamento, foi utilizada a função de perda "categorical_crossentropy" que é empregada em problemas de classificação com múltiplas classes. Ela permite medir quanto a previsão do modelo está distante do rótulo verdadeiro. O algoritmo otimizador escolhido foi o Adam (*Adaptive Moment Estimation*), que é utilizado para ajustar os pesos do modelo durante o treinamento. E a métrica utilizada foi a acurácia.

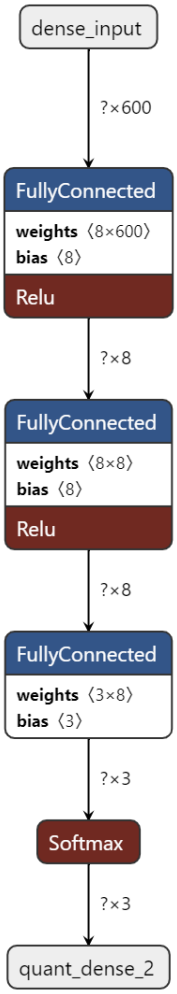
A matriz de confusão e a acurácia do modelo, serão calculadas utilizando funções da biblioteca "scikit-learn". A acurácia corresponde ao percentual de previsões corretas em relação ao total de previsões feitas. Enquanto a matriz de confusão é um recurso que facilita a interpretação das saídas da rede, para mostrá-la foi utilizada a biblioteca "Seaborn", que é baseada no "Matplotlib".

Para viabilizar a realização de inferências no ESP32-S3, foi necessário realizar a quantização do modelo e convertê-lo para TensorFlow Lite. O processo de quantização é feito para reduzir a precisão dos parâmetros da rede de ponto flutuante de 32 bits para inteiros de 8 bits.

Para quantização o TensorFlow conta com 2 técnicas. Na quantização pós-treinamento, é possível quantizar apenas os pesos ou tanto os pesos quanto as ativações. No entanto, esse tipo de quantização geralmente resulta em uma diminuição da acurácia do modelo. Enquanto a quantização consciente do treinamento (*quantization-aware training*) é uma técnica que simula os efeitos da quantização durante o processo de treinamento, o que permite ao modelo ajustar seus parâmetros minimizando os impactos negativos da quantização na acurácia.

Resumidamente, na Figura 16, é apresentado o diagrama da rede neural quantizada.

Figura 16 – Arquitetura da rede neural quantizada.

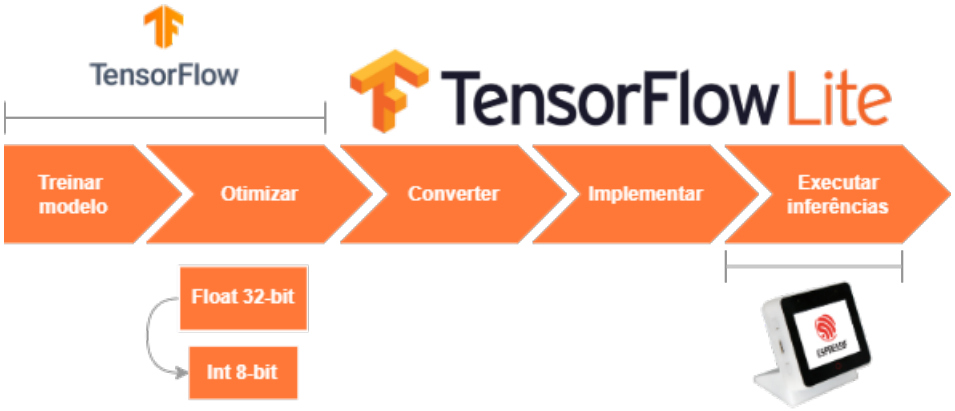


Fonte: Autora.

Após a quantização, o modelo é convertido para o formato TensorFlow Lite. O TensorFlow Lite é uma versão otimizada do TensorFlow, projetada para permitir a execução de modelos de aprendizado de máquina em microcontroladores. A conversão é realizada utilizando o recurso "tf.lite.TFLiteConverter". Após a conversão, o modelo pode ser salvo em um arquivo .h, para ser utilizado no ESP32-S3.

Em resumo, o fluxograma da Figura 17 descreve o processo de criação e implementação da rede neural utilizando o TensorFlow e o TensorFlow Lite.

Figura 17 – Fluxograma de implementação da rede com TensorFlowLite.



Fonte: Autora.

4 Resultados

Neste capítulo, são apresentados os detalhes da construção da base de dados, incluindo as etapas de adição de ruído e expansão do conjunto de dados, que visam melhorar a robustez dos dados do treinamento. Em seguida, é descrito o processo de treinamento da rede e, então, é realizada a avaliação do desempenho da rede neural, abordando as métricas de acurácia e a análise de seu comportamento com dados de teste.

Por fim, são discutidos os resultados, levando em conta os objetivos estabelecidos no início do trabalho e os resultados encontrados.

4.1 Construção da Base de Dados

A base de dados utilizada neste trabalho é composta de dados coletados através de ensaio em laboratório e de dados sintéticos criados por técnica de aumento de dados. As coletas dos dados foram realizadas em dias distintos devido ao acesso limitado ao laboratório e à necessidade de ajustes no código para a obtenção dos dados.

A Figura 18 ilustra o comportamento do motor sob condições normais de operação. Enquanto nas Figura 19 e Figura 20, vemos a FFT da vibração do motor em diferentes rotações por minuto, com desbalanceamento de corrente e com rolamento defeituoso, respectivamente. Nessas amostras, o ruído já foi adicionado.

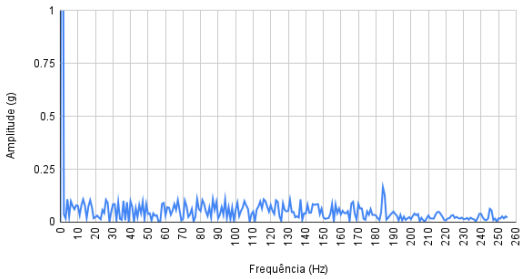
Após a coleta das amostras experimentais, foi necessário realizar o tratamento e a expansão dos dados. A partir das 30 amostras coletadas para cada rotação e eixo, aumentou-se o conjunto de dados de vibração do motor para 200 amostras para cada rotação e eixo. Cada nova amostra foi gerada selecionando aleatoriamente um bloco de 256 amostras dos dados existentes e adicionando ruído branco aos valores dos eixos x, y e z.

Como pode ser visualizado nas Figura 18, Figura 19 e Figura 20, a concentração de picos de amplitude ocorre entre 0 e 199 Hz. Portanto, para o treinamento e a validação, o sinal foi reduzido de 0 a 255 Hz para o intervalo de 0 a 199 Hz. O objetivo é otimizar o tamanho da rede e dos dados de inferência que serão utilizados no ESP32-S3.

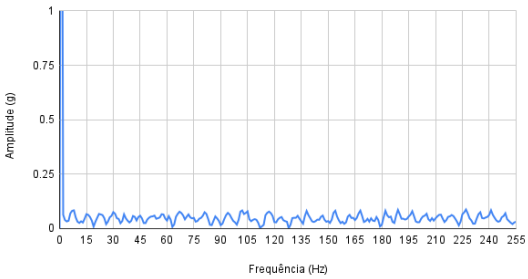
Assim, o conjunto de dados para o treinamento da rede possui 2000 amostras, totalizando 1.200.000 valores representando as amplitudes de aceleração nos eixos (x, y e z). Esses dados são divididos em tipo 1 e tipo 2, onde o tipo 1 será dividido em 80/20 para treinamento e validação, e o tipo 2 é somente para testes. Das 2000 amostras, o tipo 1 tem 1600, divididas em 1280 para treinamento e 320 para validação, enquanto o tipo 2 tem 400 amostras para teste.

Figura 18 – Vibração do motor em diferentes rotações por minuto na condição normal.

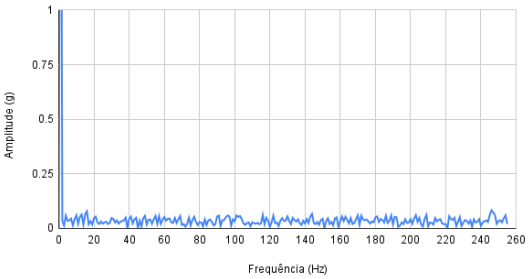
(a) Motor a 600 RPM - FFT do Eixo X



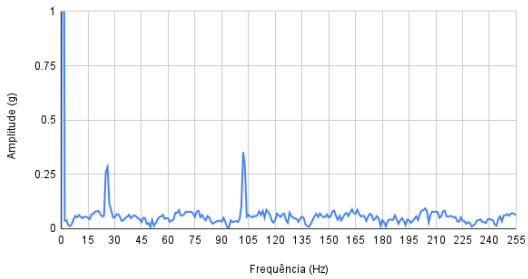
(b) Motor a 900 RPM - FFT do Eixo X



(c) Motor a 1200 RPM - FFT do Eixo X



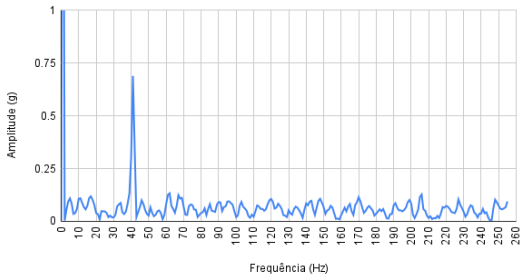
(d) Motor a 1500 RPM - FFT do Eixo X



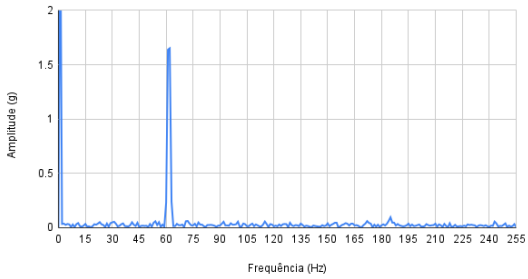
Fonte: Autora.

Figura 19 – Vibração do motor em diferentes rotações por minuto com desbalanceamento de corrente.

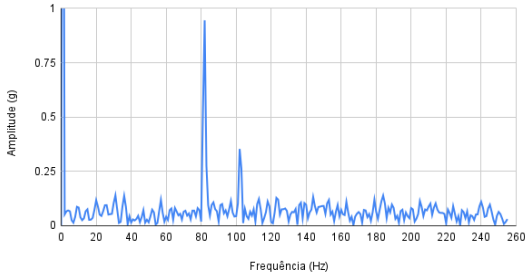
(a) Motor a 600 RPM - FFT do Eixo X



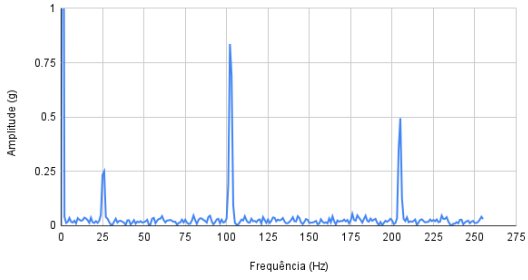
(b) Motor a 900 RPM - FFT do Eixo X



(c) Motor a 1200 RPM - FFT do Eixo X



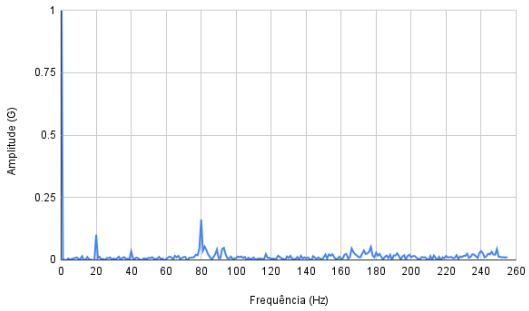
(d) Motor a 1500 RPM - FFT do Eixo X



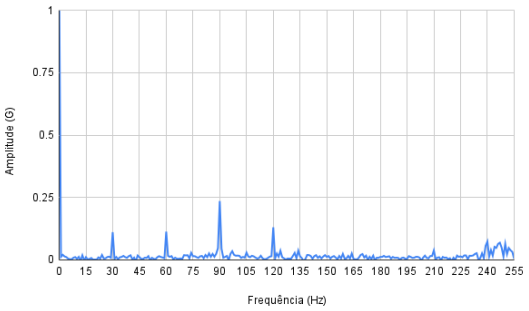
Fonte: Autora.

Figura 20 – Vibração do motor em diferentes rotações por minuto com rolamento defeituoso.

(a) Motor a 1200 RPM - FFT do Eixo X



(b) Motor a 1800 RPM - FFT do Eixo X



Fonte: Autora.

Para garantir a preservação da estrutura original dos dados, foram mantidas as cópias dos dados originais e adicionados novos blocos de dados com ruído ao conjunto de dados. Os dados foram divididos em conjuntos de treino (80%) e teste (20%), com rótulos adequadamente atribuídos a cada bloco.

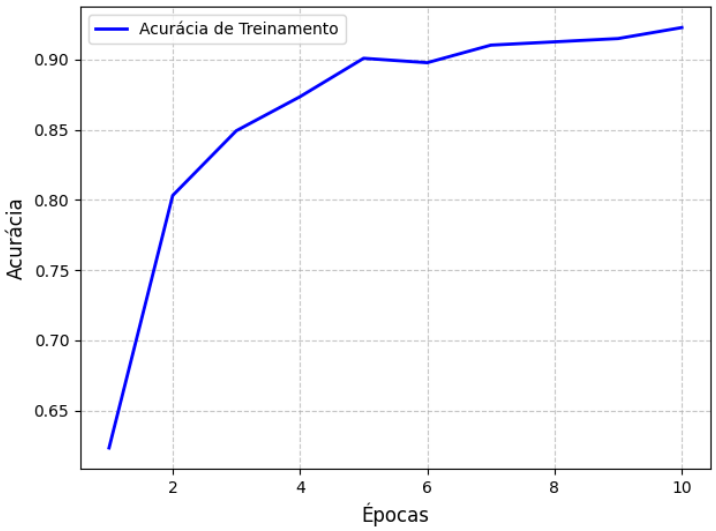
4.1.1 Treinamento da Rede Neural

Definiu-se que a rede recebe 600 entradas, correspondentes a 200 valores de x, 200 de y e 200 de z. A sua estrutura consiste em duas camadas intermediárias, cada uma com 8 neurônios. Essas camadas são "totalmente conectadas" (*fully connected*) e utilizam a função de ativação "ReLU" (*Rectified Linear Unit*). E uma camada de *dropout* com taxa de 0,50 foi utilizada para prevenir o *overfitting*. A camada de saída possui 3 neurônios, correspondentes às classes: normal, desbalanceamento e rolamento. A função de ativação "softmax" é utilizada na camada de saída e o modelo foi treinado por 10 épocas.

A Figura 21 mostra o gráfico da evolução da acurácia de treinamento, evidenciando uma tendência de melhoria contínua ao longo das épocas, com a acurácia alcançando 98% na última época, o que indica uma boa performance do modelo durante o processo de treinamento.

Considerando os benefícios mencionados anteriormente, optou-se pela técnica de quantização consciente do treinamento, que foi implementada utilizando a ferramenta "tensorflow_model_optimization". Após a quantização, o modelo foi convertido para o formato TensorFlow Lite e, em seguida, armazenado em um arquivo .h, para ser integrado e utilizado no ESP32-S3.

Figura 21 – Evolução da Acurácia de Treinamento por Época.



Fonte: Autora.

4.1.2 Desempenho da Rede Neural

Após o treinamento da rede, foram realizados dois testes: um utilizando a divisão 80/20 do conjunto de dados original e outro com um grupo de dados destinado somente para teste. O grupo 1 é composto por 320 amostras, sendo 128 da classe "normal", 135 da classe "desbalanceamento" e 57 da classe "rolamento", a quantidade foi selecionada aleatoriamente. Já o grupo 2 contém 400 amostras, distribuídas de forma equilibrada, com 160 amostras de "normal", 160 de "desbalanceamento" e 80 de "rolamento".

Para o primeiro conjunto de dados, foram obtidos resultados muito satisfatórios, como ilustrado na Figura 22. O modelo apresentou uma perda (*Test loss*) de 0,07147103548049927, indicando que o erro médio das previsões do modelo em relação aos valores reais é baixo. Além disso, a acurácia (*Test accuracy*) foi de 0,981249988079071, o que significa que o modelo acertou aproximadamente 98% das previsões para este conjunto de dados. Na matriz de confusão, foi observado que apenas 6 amostras da classe "normal" foram classificadas incorretamente como "desbalanceamento".

Com o segundo grupo de testes, o resultado foi semelhante, vide a Figura 23. Foi obtida uma perda de 0,07483244687318802 e uma acurácia de 0,9825000166893005, onde 7 amostras da classe "normal" foram classificadas incorretamente como "desbalanceamento".

Após a quantização, foi realizado um novo teste com o segundo grupo, e os resultados mostraram-se equivalentes ao modelo em *float* de 32 bits, como pode ser visto na Figura 24. A acurácia do modelo base foi de 98,25%, com uma perda de 0,0748. Já com o modelo quantizado, a acurácia manteve-se em 98,25%, com uma leve variação na perda, que passou para 0,0962.

Para fins de comparação de tamanho dos modelos, como mostrado na Figura 25,

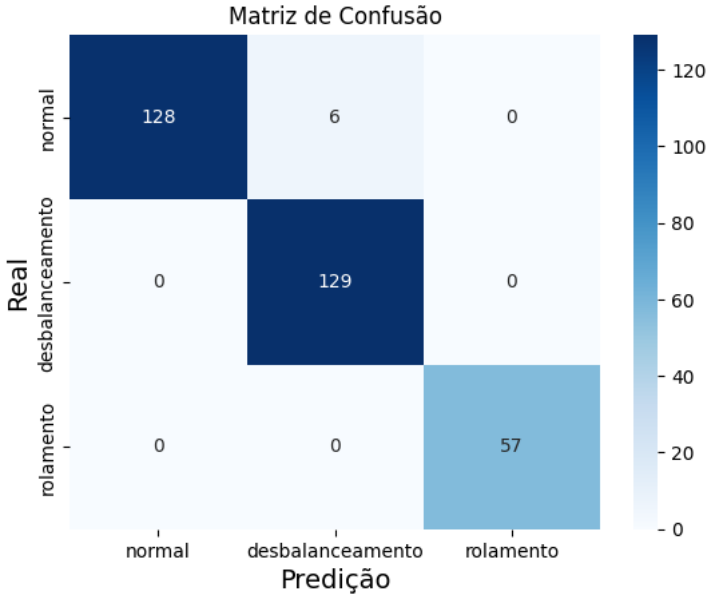
Figura 22 – Resultados da avaliação do modelo com grupo 1 de teste.

(a) Perda e acurácia do conjunto 1 de teste.

```
result = model.evaluate(X_test, y_test, verbose=0)
print("Test loss:", result[0])
print("Test accuracy:", result[1])
```

```
Test loss: 0.07147103548049927
Test accuracy: 0.981249988079071
```

(b) Matriz de confusão.



Fonte: Autora.

o modelo base convertido para TensorFlow Lite possui 21,7 KB, enquanto o modelo quantizado corresponde a apenas 7,8 KB.

4.2 Implementação da Rede Neural no ESP32-S3

Para realizar as inferências no ESP32-S3, foi utilizado o framework ESP-IDF. O processo de embarcar uma rede neural no microcontrolador exige que o código carregue o modelo, prepare os dados de entrada (incluindo normalização e quantização), execute a inferência e processe os resultados gerados. A inferência é realizada por meio de um interpretador do TensorFlow Lite, que aplica o modelo aos dados de entrada e gera as saídas correspondentes.

Dessa forma, inicialmente, é necessário incluir no código o modelo da rede neural treinado no formato '.h', além dos dados que serão avaliados e das médias e desvios padrão obtidos pelo "StandardScaler" para realizar a normalização. Juntamente das bibliotecas necessárias para o funcionamento do TensorFlow Lite.

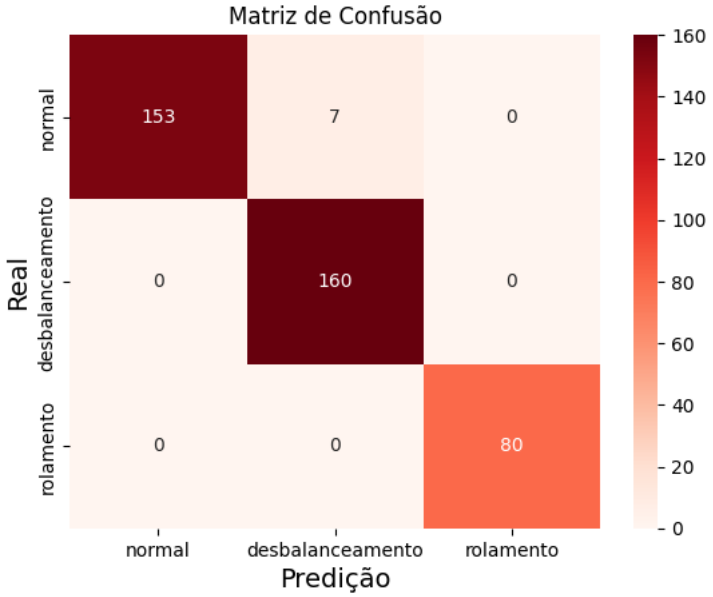
Figura 23 – Resultados da avaliação do modelo com grupo 2 de teste.

(a) Perda e acurácia do conjunto 2 de teste.

```
results = model.evaluate(X2_test, y2_test, verbose=0)
print("Test loss:", results[0])
print("Test accuracy:", results[1])
```

```
Test loss: 0.07483244687318802
Test accuracy: 0.9825000166893005
```

(b) Matriz de confusão.



Fonte: Autora.

Figura 24 – Comparação de acurácia entre o modelo base e o modelo quantizado.

```
baseline_model_accuracy = model.evaluate(
    X2_test, y2_test, verbose=0)

q_aware_model_accuracy = q_aware_model.evaluate(
   X2_test, y2_test, verbose=0)

print('Baseline test accuracy:', baseline_model_accuracy)
print('Quant test accuracy:', q_aware_model_accuracy)
```

```
Baseline test accuracy: [0.07483244687318802, 0.9825000166893005]
Quant test accuracy: [0.09619836509227753, 0.9825000166893005]
```

Fonte: Autora.

No bloco de configuração, que antecede a execução das inferências, é necessário carregar o modelo de rede neural e alocar o tamanho da arena de tensores na memória do ESP32-S3 de forma adequada. Se o tamanho da arena for inferior ao necessário, o programa não funcionará. Mas, é importante considerar a capacidade de memória do dispositivo para não alocar um espaço maior do que o disponível.

Figura 25 – Comparação do tamanho do modelo base e o modelo quantizado.

(a) Tamanho do modelo original.

```
len(tflite_model)
```

```
21708
```

(b) Tamanho do modelo quantizado.

```
len(tflite_qaware_model)
```

```
7768
```

Fonte: Autora.

Em seguida, é necessário identificar as operações que o intérprete deve suportar durante a inferência do modelo. No caso, são: *FullyConnected*, *Relu* e *Softmax*. E o último passo é criar e configurar o intérprete do modelo, além de alocar os tensores necessários para a inferência.

Para executar as inferências, é preciso normalizar e quantizar os dados de entrada. A normalização é realizada utilizando os parâmetros de média e desvio padrão. Em seguida, a quantização converte esses dados normalizados para o formato int8, que é o formato utilizado pelo modelo para inferência. O cálculo realizado para quantização é mostrado na Equação 4.1.

$$\frac{v_n}{v_s} + z_p, \tag{4.1}$$

em que v_n é o valor normalizado, v_s é a escala e z_p é o ponto zero.

Após a preparação dos dados, a inferência é realizada utilizando o modelo carregado na memória do dispositivo. O intérprete do TensorFlow Lite processa o modelo, aplicando as operações registradas sobre os dados de entrada. A saída do modelo é gerada no formato int8, e, em seguida, é desquantizada para converter os valores de volta para o formato de ponto flutuante. O cálculo realizado para desquantização é mostrado na Equação 4.2.

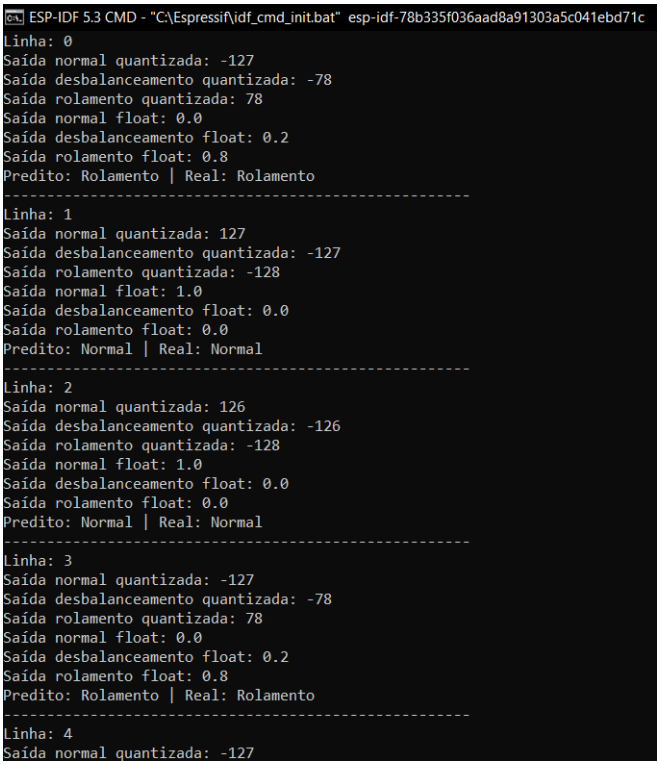
$$(q_s - z_p) \cdot v_s, \tag{4.2}$$

em que q_s é a pontuação da quantização.

Após esse processo, é possível calcular as probabilidades de a saída pertencer a cada uma das classes. Assim, a classe predita é determinada com base no maior percentual entre as classes. Por fim, os resultados são impressos, comparando a classe predita com o rótulo real dos dados de entrada, o que possibilita avaliar a precisão da inferência.

Os resultados das inferências podem ser visualizados na Figura 26. O objetivo foi testar o modelo com diversos dados de diferentes classes. No terminal, foram impressos os valores da saída quantizada, da saída convertida para o formato de ponto flutuante, e a comparação entre a classe predita e a classe real, com o intuito de analisar a saída do modelo e avaliar seu desempenho.

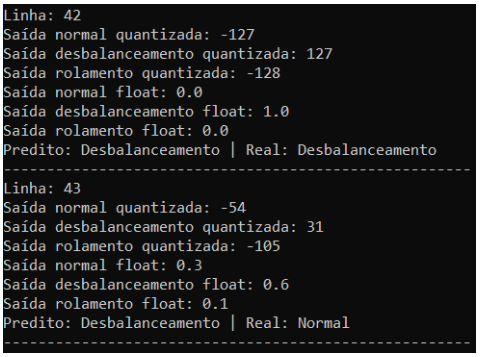
Figura 26 – Resultados da inferência no ESP32-S3.



Fonte: Autora.

Além disso, na Figura 27 é possível observar quando a inferência faz uma previsão incorreta, classificando como defeito de desbalanceamento quando, na verdade, trata-se da condição normal, seguindo o mesmo padrão observado na Figura 23.

Figura 27 – Resultado da inferência fazendo uma previsão incorreta.



Fonte: Autora.

4.3 Discussão dos Resultados

Com o objetivo de desenvolver um sistema de detecção de defeitos em MITs utilizando uma rede neural quantizada para implementação em um microcontrolador

ESP32-S3, serão avaliados os resultados obtidos em cada etapa do processo, desde a coleta de dados até a implementação e validação do modelo.

Os resultados da coleta de dados indicaram que a quantidade de amostras era insuficiente para treinar o modelo de forma adequada, sendo necessário expandir o número de amostras. Além disso, a introdução de ruído Gaussiano foi essencial para simular condições mais semelhantes às reais de operação e garantir um pouco mais de robustez do modelo. Para o treinamento, foi decidido que o tamanho da FFT deveria ser limitado de 0 a 199 Hz, em vez de 0 a 255 Hz, a fim de reduzir o tamanho dos dados e melhorar o desempenho no microcontrolador. A padronização dos dados também foi aplicada para garantir consistência nas entradas e melhorar a performance do modelo treinado. Após os ajustes, a quantidade de dados foi suficiente para um treinamento satisfatório da rede.

Com base no objetivo de testar diferentes tipos de redes neurais para verificar o mais adequado, diversas arquiteturas foram avaliadas. As redes neurais convolucionais (CNN) foram consideradas devido à sua reputação consolidada em diversos tipos de problemas. Mas, observou-se que o modelo convergia com um número muito pequeno de neurônios, tornando desnecessário um modelo dessa complexidade. Em seguida, foi testado um modelo perceptron multicamadas com diferentes arquiteturas, variando o número de neurônios nas camadas ocultas (4, 8, 16 e 32 neurônios). Observou-se que, com 8 neurônios em cada camada, o modelo já apresentava uma convergência adequada, alcançando uma acurácia em torno de 98%.

A partir dessa análise, um modelo foi treinado e quantizado, permitindo sua implementação em microcontroladores. A quantização resultou em uma redução significativa no tamanho do modelo, transformando tanto os pesos quanto as ativações em int8, mas mantendo a mesma acurácia. Manter a mesma eficiência do modelo quantizado em relação ao modelo base é um resultado positivo.

A implementação do modelo quantizado utilizando TensorFlow Lite no microcontrolador ESP32-S3 cumpriu o objetivo geral de criar um sistema embarcado capaz de identificar defeitos em MITs. Os resultados mostraram que o processo de integração foi bem-sucedido. Os testes de inferência realizados no microcontrolador ESP32-S3 demonstraram que o sistema de detecção de defeitos estava funcionando de acordo com o esperado. A validação dos resultados indicou que o modelo foi capaz de identificar defeitos com uma taxa de precisão superior a 90%.

5 Conclusões

Os motores de indução trifásicos (MITs) têm sido amplamente utilizados na indústria nas últimas décadas devido à sua robustez, versatilidade, baixo custo e alta eficiência. Para garantir um alto desempenho e evitar paradas não programadas, é essencial realizar o monitoramento regular das condições desses motores. Neste contexto, o presente trabalho apresentou um sistema de monitoramento das condições de motores de indução trifásicos, utilizando uma rede neural embarcada em um microcontrolador ESP32-S3 da Espressif, com o objetivo de detectar defeitos.

Para isso, foi feita a coleta de dados de vibração para treinamento e teste em uma bancada de testes. A aquisição foi feita utilizando o acelerômetro ADXL335, fixo em um ponto recomendado pela ABNT NBR 10082. As medições foram realizadas de 600, 900, 1200, 1500 e 1800 rotações por minuto do motor, considerando a operação normal e defeitos de desbalanceamento de corrente e falha de rolamento. Foram coletadas amostras das componentes X, Y e Z em um intervalo de 1 segundo. Os dados coletados no tempo foram convertidos para o domínio da frequência (FFT) e salvos em um arquivo CSV. Esses dados tiveram ruído adicionado e foram expandidos conforme a necessidade de um Dataset mais robusto.

A rede neural foi escolhida com base em testes com dois tipos de redes: CNN e perceptron de multicamadas. A segunda configuração foi testada com 32, 16, 8 e 4 neurônios na camada oculta, e a configuração com 8 neurônios foi escolhida por apresentar o desempenho adequado. O modelo treinado obteve uma acurácia de aproximadamente 98%, considerada satisfatória para o problema. Após o treinamento, a rede foi quantizada de float32 para int8, para viabilizar sua implementação no ESP32-S3. A quantização manteve a acurácia do modelo original em FP32. Ao testar o modelo no ESP32-S3, observou-se que ele funcionou conforme o esperado, com a inferência sendo realizada de maneira eficiente.

Diante disso, os resultados indicam que o sistema de monitoramento desenvolvido pode ser uma solução eficiente para a manutenção preditiva, contribuindo para a redução de paradas não programadas e contribuindo para a melhoria do desempenho operacional dos motores na indústria.

Para trabalhos futuros, sugerem-se algumas novas abordagens. Seria interessante realizar medições de uma maior variedade de defeitos para aumentar a abrangência do projeto. Além disso, recomenda-se ampliar as medições para além de vibrações, incluindo corrente, tensão e temperatura. Com a adição dessa complexidade, seria importante testar novamente o modelo CNN ou explorar outros modelos de inteligência artificial para avaliar se oferecem um desempenho superior.

Referências

ARAúJO, V. G. de et al. Monitoring and diagnosing faults in induction motors' three-phase systems using narx neural network. *Energies*, MDPI, v. 17, n. 18, p. 1–40, 2024. Citado na página 30.

DEVICES, A. *ADXL335 datasheet*. https://www.analog.com/media/en/technical-documentation/data-sheets/adxl335.pdf, 2010. Citado na página 26.

ELETRONICWINGS. *Complete Guide ADXL335 Accelerometer with Arduino Interfacing*. 2017. Disponível em: <https://www.electronicwings.com/sensors-modules/adxl335-accelerometer-module>. Acesso em: 21 nov. 2024. Citado na página 26.

ESPRESSIF. *ESP32-S3 Box*. 2022. Disponível em: <https://github.com/espressif/esp-box/blob/master/docs/_static/esp32_s3_box.png>. Acesso em: 21 nov. 2024. Citado na página 27.

ESPRESSIF. *ESP32 Series Datasheet Version 4.7*. 2024. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. Acesso em: 23 nov. 2024. Citado na página 26.

ESPRESSIF. *Hardware Overview*. 2024. Disponível em: <https://github.com/espressif/esp-box/blob/master/docs/hardware_overview/esp32_s3_box/hardware_overview_for_box.md>. Citado na página 27.

FARIAS, T. S.; ROSSI, R. G. *Estudo Comparativo de Arquiteturas de Redes Neurais em Analise de Sentimentos*. Dissertação (Mestrado) — Universidade Federal do Mato Grosso do Sul, 2021. Acesso em: 09 dez. 2024. Citado na página 22.

GGCE, G. C. de Conservação de E. E. *Terceiro Plano Anual de Aplicação de Recursos do Programa Nacional de Conservação de Energia Elétrica – PROCEL*. [S.l.], 2020. Citado na página 12.

GROOTENDORST, M. *A Visual Guide to Quantization*. 2024. Disponível em: <https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-quantization>. Acesso em: 09 dez. 2024. Citado na página 23.

HAYKIN, S. *Redes neurais: princípios e prática*. Porto Alegre, 2001. Citado na página 22.

HOLANDA, S. M. S. *Aplicação da Manutenção Preditiva por Análise de Vibrações em Equipamentos de Trens Urbanos com Plano de Manutenção Proposto*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2016. Disponível em: <https://repositorio.ufpe.br/bitstream/123456789/17606/1/Disserta%c3%a7%c3%a3o%20Sandra%20Holanda%20IMPRESS%c3%83O%20FINAL%2029ABR3.pdf>. Acesso em: 21 nov. 2024. Citado 2 vezes nas páginas 18 e 19.

IEEE. *O Que É Um Microcontrolador?* 2020. Disponível em: <https://edu.ieee.org/br-ufcgras/o-que-e-um-microcontrolador/>. Acesso em: 23 nov. 2024. Citado na página 23.

KARDEC, A.; NASCIF, J. *Manutenção: Função Estratégica*. 3. ed. [S.l.]: QualityMark, 2009. ISBN 978-85-7303-898-9. Citado na página 12.

KOLEHMAINEN, H. K. L. A. M. N. J. P. J. Technology comparison of induction motor and synchronous reluctance motor. *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017. Citado na página 12.

LUDERMIR, T. B. Inteligência artificial e aprendizado de máquina: estado atual e tendências. *ESTUDOS AVANÇADOS*, v. 35, 2021. Citado na página 21.

MAKIYAMA, M. *Placa ESP32: Descubra o que é, para que serve e muito mais!* 2023. Disponível em: <https://victorvision.com.br/blog/placa-esp32/>. Acesso em: 23 nov. 2024. Citado na página 27.

MITCHELL, T. *Machine Learning*. 1997. Citado na página 21.

NEPOMUCENO, L. X. *Técnicas de manutenção preditiva*. 1. ed. [S.l.]: Blucher, 1989. ISBN 9788521200925. Citado na página 18.

PEREIRA, K. Y. da S. *Diagnóstico de falhas elétricas em motores de indução trifásico utilizando método de análise de vibração e análise do espectro de corrente*. Dissertação (Mestrado) — Universidade Federal de Mato Grosso do Sul, 7 2021. Disponível em: <https://engeletrica.ufms.br/files/2021/08/TCC-LCL-DiagnosticoVibracao-Klariman-Pereira-2021.pdf>. Acesso em: 21 mai. 2024. Citado na página 18.

REIS, A. J. S. *Reconhecimento de Padrões de Falhas em Motores Trifásicos Utilizando Redes Neurais*. Dissertação (Mestrado) — Universidade Federal Do Rio Grande do Norte, 2 2010. Disponível em: <https://repositorio.ufrn.br/bitstream/123456789/15341/1/AdersonJSR_DISSERT.pdf>. Acesso em: 21 mai. 2024. Citado 2 vezes nas páginas 17 e 18.

SANTOS, S. T. C. A. dos. *Operação de Motor de Indução Trifásico Conectado à Rede por Meio de um Conversor de Nove Chaves com Fator de Potência Adiantado*. Dissertação (Mestrado) — Universidade De São Paulo, 2021. Disponível em: <https://www.teses.usp.br/teses/disponiveis/18/18153/tde-01032021-203910/publico/DissertSantosStefanThiagoCAdosCorrig.pdf>. Acesso em: 27 out. 2024. Citado na página 12.

SKF. *Manual de manutenção de rolamentos da SKF*. [S.l.], 2012. Citado na página 19.

SKF. *SKF Product select*. 2024. Disponível em: <https://productselect.skf.com/#/size-lubrication/single-bearing>. Acesso em: 21 nov. 2024. Citado 2 vezes nas páginas 20 e 21.

SUETAKE, M. *Sistemas inteligentes para monitoramento e diagnósticos de falhas em motores de indução trifásicos*. Tese (Doutorado) — Escola de Engenharia de São Carlos da Universidade de São Paulo, 2012. Citado na página 16.

TAVARES, L. H. P. *Diagnóstico de Falhas Em Máquinas Rotativas Utilizando Redes Neurais Artificiais*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, 2021. Acesso em: 21 nov. 2024. Citado 2 vezes nas páginas 12 e 19.

TOYAMA. *A importância de realizar o balanceamento de fases em um grupo gerador*. 2024. Disponível em: <https://toyama.com.br/blog/importancia-balanceamento-fases-grupo-gerador/>. Acesso em: 25 nov. 2024. Citado na página 21.

VITOR, A. L. D. O. et al. Detecção de falhas elétricas em motores de indução utilizando rede radial basis function. In: *Anais do XX Congresso Brasileiro de Automátican*. [S.l.: s.n.], 2014. v. 46, p. 37–42. Citado na página 12.

WEBER, L. *Introdução ao Software Octave e Modulação em Amplitude (AM)*. 2024. Disponível em: <https://revistaantenna.com.br/introducao-ao-software-octave-e-modulacao-em-amplitude-am/8/>. Acesso em: 21 nov. 2024. Citado na página 19.

WEG. *GUIA DE ESPECIFICAÇÃO MOTORES ELÉTRICOS*. 2024. Disponível em: <https://static.weg.net/medias/downloadcenter/h32/hc5/WEG-motores-eletricos-guia-de-especificacao-50032749-brochure-portuguese-web.pdf>. Acesso em: 20 nov. 2024. Citado 2 vezes nas páginas 15 e 16.