



UNIVERSIDADE FEDERAL DE ITAJUBÁ
INSTITUTO DE SISTEMAS ELÉTRICOS E ENERGIA - ISEE
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

PEDRO HENRIQUE CAMARGOS

**Sistemas Inteligentes para Monitoramento e Diagnóstico de Falhas em Motores de
Indução IEC 60034 / ISO 20816**

Itajubá

2021

PEDRO HENRIQUE CAMARGOS

**Sistemas Inteligentes para Monitoramento e Diagnóstico de Falhas em Motores de
Indução IEC 60034 / ISO 20816**

Dissertação apresentada ao Instituto de Sistemas Elétricos e Energia da Universidade Federal de Itajubá para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Engenharia Elétrica.

Área de concentração: Sistemas Elétricos de Potência

Orientador: Prof. Dr. Benedito Isaías Lima Fuly

Itajubá

2021

*A meu pai José Junio de Camargos e minha mãe Cleide Gonçalves,
por todo o apoio, força e dedicação em mim depositados. . .*

“Não importa o quanto você vá devagar desde que não pare.”

(Confúcio)

Resumo

Camargos, Pedro H. **Sistemas Inteligentes para Monitoramento e Diagnóstico de Falhas em Motores de Indução IEC 60034 / ISO 20816**. 2021. 137 f. Dissertação (Mestrado em Ciências) – Instituto de Sistemas Elétricos e Energia, Universidade Federal de Itajubá, Itajubá, 2021.

Nas plantas industriais os motores estão sujeitos a sofrerem falhas que podem ocasionar paradas na produção causando prejuízos econômicos. O monitoramento preditivo identifica estas falhas e reduz os custos com manutenção e operação. Neste trabalho, foram desenvolvidos sistemas de diagnóstico em motores de indução trifásicos baseados nas normas IEC 60034 e ISO 20816. O Sistema de Diagnóstico Regiões de Operação, através de uma rede neural de reconhecimento de padrões, classifica a região de operação do motor. O Sistema de Diagnóstico de Barra Quebrada do Rotor monitora o estado da máquina através de uma rede neural recorrente de memória longa de curto prazo, identificando o número de barras quebradas no rotor e a zona de operação conforme o ISO 20816. Por fim o Sistema de Diagnóstico *Fuzzy* utiliza um sistema neuro-*fuzzy* para prever o Fator de Redução do motor, prevenindo possíveis falhas que ocorrem devido ao desequilíbrio de tensão.

Palavras-chaves: Aprendizado de máquina. monitoramento. motor de indução. redes neurais artificiais.

Abstract

Camargos, Pedro H. **Intelligent Systems for Monitoring and Fault Diagnosis in Induction Motors IEC 60034 / ISO 20816**. 2021. 137 p. Dissertation (Master of Science) – Institute of Electrical Systems and Energy, Institute of Electrical Systems and Energy, Itajubá, 2021.

In industrial plants, engines are subject to failures that can cause production downtime causing economic losses. Predictive monitoring identifies these failures and reduces maintenance and operating costs. In this work, systems for monitoring and diagnosing faults in three-phase induction motors were developed based on the IEC 60034 and ISO 20816 standards. Through a pattern recognition neural network, the Operating Regions Diagnostic System classifies the region of operation of the motor. The Rotor Broken Bar Diagnostic System monitors the machine's state through a recurring neural network of short-term long memory, identifying the number of broken bars on the rotor and the operating zone according to ISO 20816. Finally, the Diagnosis *Fuzzy* uses a neuro-*fuzzy* system to predict the motor reduction factor, predicting possible failures that occur due to voltage imbalance.

Keywords: Machine learning. monitoring. induction motor. Artificial neural networks.

Lista de figuras

Figura 1 – Áreas da inteligência artificial	28
Figura 2 – Categorias aprendizado de máquina	29
Figura 3 – Técnicas do aprendizado de máquina clássico	31
Figura 4 – Método dos conjuntos	32
Figura 5 – Estrutura do processo de aprendizagem	34
Figura 6 – Exemplo aplicação da regressão linear (a) função hipótese (b) função de custo	34
Figura 7 – Aplicação do BA para a função de custo $J(w_1, w_2)$, noção intuitiva	35
Figura 8 – Noção intuitiva GD	36
Figura 9 – Interpretação geométrica da derivada direcional	37
Figura 10 – Aplicação do GD para a função de custo $J(w_1, w_2)$	38
Figura 11 – Oscilações GD	39
Figura 12 – Modelo do perceptron	45
Figura 13 – Funções localmente quadráticas (a) sigmóide (b) tangente hiperbólica	47
Figura 14 – Unidade lineares retificadas (a) ReLU (b) LReLU	48
Figura 15 – Modelo rede neural artificial	50
Figura 16 – Neurônio j na camada de saída	53
Figura 17 – Neurônio j na camada de oculta	53
Figura 18 – Rede neural <i>feedforward</i> e suas camadas	55
Figura 19 – Estrutura rede neural recorrente	55
Figura 20 – Célula LSTM (a) sem portão do esquecimento (b) com portão do esqueci- mento (c) com conexão <i>peephole</i>	57
Figura 21 – Exemplo de matriz de confusão	59
Figura 22 – Rede neural <i>fuzzy</i> (a) Arquitetura (b) Princípios de estimação	62
Figura 23 – Equivalência ANFIS	63
Figura 24 – Limites de tensão e frequência para motores	66
Figura 25 – Valores típicos de fatores de redução para motores da categoria N	67
Figura 26 – Pontos de medição de vibração para pequenas máquinas elétricas	68
Figura 27 – Faixas de valores típicos dos limites das zonas A/B, B/C e C/D	69
Figura 28 – Espectro das correntes de linha (a) motor saudável (b) motor com uma barra quebrada (c) variação da magnitude em função do número de barras quebradas	71

Figura 29 – Circuito elétrico do modelo da máquina assíncrona: eixo direto	73
Figura 30 – Circuito elétrico do modelo da máquina assíncrona: eixo de quadratura	73
Figura 31 – Circuito equivalente simplificado da máquina assíncrona	75
Figura 32 – Curvas torque <i>versus</i> velocidade (a) curva típica (b) categorias N, H e D	79
Figura 33 – Banco de dados dos motores de indução trifásicos, disponível em WEG (2021a)	82
Figura 34 – Bancada experimental	85
Figura 35 – Torque x escorregamento (a) Torque sequência positiva (b) Torque sequência negativa	86
Figura 36 – Corrente x escorregamento	87
Figura 37 – Etapas do desenvolvimento do SDRO IEC 60034-1	88
Figura 38 – Quantidade de motores coletados no banco de dados	89
Figura 39 – Diagrama de blocos, simulação e validação dos parâmetros	89
Figura 40 – Coordenadas de tensão e frequência do banco de dados	90
Figura 41 – Arquitetura da rede neural de reconhecimento de padrões SDRO IEC 60034-1	92
Figura 42 – Procedimento avaliação <i>overfitting</i>	93
Figura 43 – Sinais de vibração do motor com 4 barras quebradas	95
Figura 44 – Metodologia projeto SDBQ	96
Figura 45 – Espectro de corrente SDBQ	97
Figura 46 – Estrutura da rede neural recorrente LSTM SBDQ	99
Figura 47 – Diagrama de blocos SDBQ	99
Figura 48 – Diagrama de blocos SDF	100
Figura 49 – Diferenças nos valores eficazes e características RNA2	101
Figura 50 – Curvas de desempenho RNA2	102
Figura 51 – Conjunto de entradas, $ I_a - I_b $, $ I_b - I_c $ e $ I_c - I_a $ e saídas, f_D , do ANFIS1	103
Figura 52 – Fator de redução x Fator de Desequilíbrio	104
Figura 53 – Métricas do estudo comparativo RNA1 (a) A_{cc} (b) E_{sp} (c) F_1 (d) R_{ec}	106
Figura 54 – Variância e erro de viés RNA1	107
Figura 55 – Curvas de desempenho RNA1	108
Figura 56 – Matrizes de confusão com 1 correspondendo a região A e 2 equivalente à região B (a) etapa treinamento (b) etapa validação (c) etapa teste	109
Figura 57 – Curvas de desempenho LSTM	111
Figura 58 – Matriz de confusão LSTM conjunto (a) treinamento (b) validação (c) teste	112

Figura 59 – Distribuição da porcentagem dos sinais de vibração ISO 20816 (a) ponto <i>P</i> (b) ponto <i>R</i>	113
Figura 60 – RMSE treinamento/validação ANFIS1	114
Figura 61 – Resultados teste de precisão SDF 60034-26	115
Figura 62 – Circuitos elétricos nos eixos d e q	132
Figura 63 – Circuito elétrico nos eixo zero	132
Figura 64 – Velocidade de rotação	136
Figura 65 – Torque	137

Lista de algoritmos

Algoritmo 1 – Laço de repetição propagação direta	51
Algoritmo 2 – Algoritmo retropropagação	54
Algoritmo 3 – Laço de repetição RNA1	92

Lista de tabelas

Tabela 1 – Relação entre x_{s1} e x_{s2}	77
Tabela 2 – Valores mínimos do torque de partida (T_p/T_n), para motores, categoria N . . .	82
Tabela 3 – Valores mínimos do torque de partida (T_p/T_n), para motores, categoria H . . .	83
Tabela 4 – Critérios adotados na categorização dos motores de indução	84
Tabela 5 – Especificações MIT-V	84
Tabela 6 – Funções de treinamento do MATLAB/Simulink	94
Tabela 7 – Especificações motores de indução aplicação SDRO IEC 60034-1	94
Tabela 8 – Ensaio testes/verificação limitação SDRO IEC 60034-1, com tensão nominal V_n	95
Tabela 9 – Conjunto treinamento/validação/teste RNA2	102
Tabela 10 – Parâmetros sistema de inferência Sugeno ANFIS1	103
Tabela 11 – Desempenho RNA1	108
Tabela 12 – Métricas RNA1	109
Tabela 13 – Resultados, comparação ensaios e SDRO IEC 60034-1	110
Tabela 14 – Valores máximos e mínimos das entradas da RNA1	110
Tabela 15 – Desempenho LSTM	111
Tabela 16 – Métricas classificação LSTM	113
Tabela 17 – Equivalência entre os parâmetros elétricos e mecânicos ATP	134
Tabela 18 – Especificações das simulações	135

Lista de abreviaturas e siglas

ANFIS1	Subsistema Adaptativo de Inferência Neuro- <i>Fuzzy</i>
RNR	Redes neurais recorrentes
RNC	Redes neurais convolucionais
LSTM	<i>Long Short Term Memory</i>
IEC	<i>International Electrotechnical Commission</i>
ISO	<i>International Organization for Standardization</i>
IA	Inteligência artificial
SVM	Máquina de vetores de suporte
K-NN	K-vizinhos mais próximos
GD	Algoritmo do gradiente descendente
BA	Algoritmo de Busca Aleatória
GDML	Algoritmo de Gradiente Descendente em Minilotes
GDL	Algoritmo de Gradiente Descendente em Lote
GDE	Algoritmo de Gradiente Descendente Estocástico
GDM	Gradiente Descendente com Momento
SGD	Método Gradiente Conjugado em Escala
VN	Verdadeiros Negativos
VP	Verdadeiros Positivos
FP	Falsos Positivos
FN	Falsos Negativos
MCSA	Análise da Assinatura da Corrente
MVSA	Análise da Assinatura da Vibração

RNA1	Rede Neural Artificial de Reconhecimento de Padrões <i>feedforward</i> SDRO
RNA2	Rede neural Artificial Multicamadas SDRO
SDRO	Sistema de Diagnóstico Região de Operação SDF
SDF	Sistema de Diagnóstico <i>Fuzzy</i> IEC 60034-26
SDBQ	Sistema de Diagnóstico de Barra Quebrada do Rotor ISO 20816
EMTP	<i>Electromagnetic Transients Program</i>
MSE	Função de custo Erro Quadrático Médio
BFG	BFGS Quasi-Newton
CGB	Gradiente Conjugado com Reinicialização de Powell/Beale
CGP	Gradiente Conjugado Polak-Ribière
GDM	Gradiente Descendente com Momento
LM	Levenberg-Marquardt
RPR	Retropropagação Resiliente
SCG	Gradiente Conjugado em Escala
PA	Parada antecipada
RMSE	Raiz Quadrada Média do Erro
UM	Modelo de Máquina Universal

Lista de símbolos

h	Função hipótese
n_0	número de neurônios na camada oculta
J	Função de custo
V_{ab}, V_{bc}, V_{ca}	Tensões eficazes de linha
I_a, I_b, I_c	Corrente eficazes de linha na fase A, B e C
$a_{Iabk}, a_{Ibc_k}, a_{ca_k}$	Áreas das curvas de variação de corrente RNA2
f_n	Frequência nominal do motor
W_m	Velocidade de rotação
\mathbf{w}^i	Vetor de pesos, $i = 1, 2 \dots k$
α	Taxa de aprendizagem
w_i	Valor do peso
$D_{\mathbf{u}}$	Derivada direcional na direção de \mathbf{u}
r_{tan}	Reta tangente
C	Curva sobre a superfície V
V	Superfície em \mathbb{R}^3
∇J	Gradiente da função de custo J
β	Peso momentâneo (Gradiente Descendente com Momento)
β_{ang}	Diferença entre a posição de referência e a posição angular elétrica do rotor
\mathbf{d}^i	Vetor que representa a direção de descida, $i = 1, 2 \dots k$
\mathbf{A}	Matriz hessiana
γ	Fator matriz Hessiana método BFGS Quasi-Newton
\mathbf{p}^i	Vetor direção método gradiente conjugado, $i = 1, 2 \dots k$

β_X	Taxa de aprendizagem algoritmo gradiente conjugado onde $X = PR$ para o equacionamento proposto por Polak e Ribiere e $X = SGD$ para o equacionamento do método do Gradiente Conjugado em Escala
c_1, c_2, c_3	Constantes algoritmo Reinicialização Powell/Beale
J_a	Matriz jacobiana
η^+, η^-	Fatores algoritmo de Retropropagação Resiliente
Δ^k	Parâmetro de atualização
μ	Fator multiplicador Retropropagação Levenberg-Marquart
q	Função quadrática
σ	Função de ativação
$\sigma_{sig}, \sigma_{tan}$	Função sigmóide, tangente hiperbólica
σ_{Re}, σ_{RL}	Função unidade de retificação linear e função unidade linear retificada com vazamento
σ_{so}	Função <i>softmax</i>
b	Viés
\mathbf{x}	Vetor de entrada
y_i, \hat{y}_i	Valores saída conjunto treinamento e saída rede neural artificial
Δw_{ij}	Variação do peso no neurônio j e na conexão i
e_j	Erro do neurônio j
m	Número de elementos conjunto treinamento
S	Conjunto treinamento
$N_{treq}, N_{uw}, N_{trn}$	Número de equações de treinamento, pesos desconhecidos quantidade de elementos do conjunto de dados treinamento
L_v, L_t	Erro de validação e treinamento

I, H, O	Número de entradas, neurônios na camada oculta e classes de saída
E_p	Número da época
Δ	Vetor de diferenças entre os erro de teste e validação RNA1
$\mathbf{x}^{(i)}$	Vetor de características rede neural camada i
$\mathbf{W}^{[i]}$	Matriz de pesos na camada i
$\mathbf{Z}^{[i]}$	Matriz de pré-ativação na camada i
$\mathbf{A}^{[i]}$	Matriz de ativação na camada i
o_j	Variável de saída algoritmo retropropagação
$\mathbf{x}^{(t)}$	Entrada rede neural recorrente no tempo t
$\mathbf{h}^{(t)}$	Valores camada escondida rede neural recorrente no tempo t
$\mathbf{o}^{(t)}$	Saída rede neural recorrente no tempo t
$\mathbf{c}^{(t)}$	Vetor de estados da célula no tempo t
$\hat{\mathbf{c}}^{(t)}$	Vetor de estados da ativação função tangente hiperbólica no tempo t
$\mathbf{f}^{(t)}$	Vetor de estados no "portal do esquecimento" no tempo t
W^{hx}, W^{hh}	Matriz de pesos de entrada e oculta (rede neural recorrente)
W^i, W^o, W^c	Matriz de pesos de entrada, saída e estado
P^i, P^f, P^o	Matriz de pesos <i>peephole</i> de entrada, esquecimento e saída
A_{cc}	Acurácia
R_{cc}	Revocação
E_{sp}	Especificidade
F_1	Medida-F1
$\mu(x)$	Função de pertinência
f_D	Fator de Desequilíbrio

V_{max}, V_{avg}	Máxima tensão e tensão média considerando os valores eficazes nas três fases
f_b, f_{ss}	Frequência induzida no espectro de corrente no estator e da fonte de alimentação
s, s_{max}, s_{nl}	Escorregamento, máximo escorregamento e escorregamento na condição de operação sem carga
J_m	Momento de inércia
P_{out}, P_{in}	Potência ativa de saída, entrada
P_{j1}, P_{j2}	Perdas por efeito joule estator e rotor
P_{rot}	Perdas rotacionais
F	Coefficiente de atrito viscoso
ω_m	Velocidade de rotação eixo do motor
T_m, T_e	Torque eletromagnético e mecânico do eixo do motor
L'_{lr}, L_{ls}, L_m	reatâncias de dispersão do rotor, estator e magnetização
φ_s, φ'_r	Fluxos magnéticos rotor e estator
R_s, R'_r	Resistência estator e rotor
ω_r, ω_e	Velocidade angular referencial e elétrica
v_{qs}, v_{ds}	Tensão eixo d-q estator
v'_{qr}, v'_{dr}	Tensão eixo d-q rotor
$i_{qs}, i_{ds}, i'_{qr}, i'_{dr}$	Correntes no eixo d-q estator e rotor
i_{as}, i_{bs}, i_{cs}	Correntes ABC no estator
$i'_{ar}, i'_{br}, i'_{cr}$	Correntes ABC no rotor
v_{abs}, v_{bcs}	Tensões no sistema ABC no estator e rotor
v'_{abr}, v'_{bcr}	Tensões ABC no rotor
ω_r, ω_e	Velocidade angular referencial e elétrica

θ	Posição angular do referencial
r_1, r_2	Resistência dos enrolamentos do estator e rotor (circuito equivalente)
x_{s1}, x_{s2}	Reatâncias de dispersão nos enrolamentos do estator e rotor (circuito equivalente)
x_{mag}	Reatância de magnetização (circuito equivalente)
\dot{V}	Tensão por fase (circuito equivalente)
I_1, I_0, I_2	Corrente de entrada, magnetização e no ramo do rotor (circuito equivalente)
P_{ag}	Potência eletromagnética
Q_{in}	Potência reativa de entrada
a	Relação entre x_{s1} e x_{s2}
fp	Fator de potência
η	Rendimento nominal
T_n, T_{max}	Torque nominal, máximo e mínimo
T_{min}, T_p	Torque mínimo, partida
P_n	Potência nominal
F_r	Função Fator de Redução
$R_{a0}, R_{f0},$	Resistência de armadura, campo de sequência zero (UM)
$R_{aj1}, R_{fj1}, R_{fj2}$	Resistência de armadura, campo (UM)
L_{aj1}, L_{fj1}	Indutância de dispersão na armadura, campo (UM)
L_{a0}, L_{f0}	Indutância de dispersão na armadura, campo de sequência zero (UM)
λ_{mj}	Fluxo mútuo (UM)
v_{fj1}, v_{fj2}	Tensões de campo (UM)
i_{fj1}, i_{fj2}	Correntes de campo (UM)

$\lambda_{fj1}, \lambda_{fj2}$	Fluxos magnéticos de campo (UM)
v_{aj}, v_{a0}	Tensão na armadura e de sequência zero (UM)
i_{aj}, i_{a0}	Corrente na armadura e de sequência zero (UM)
v_{ω}	Tensão angular (UM)
ω	Velocidade angular (UM)
$\lambda_{a0}, \lambda_{ad}, \lambda_{aq}$	Fluxo magnético de sequência zero, direto e quadratura
β_{UM}	Posição do rotor (UM)

Sumário

1	Introdução	23
1.1	<i>Motivação e Relevância do Trabalho</i>	23
1.2	<i>Objetivos e Contribuições</i>	25
1.2.1	Objetivos	25
1.2.2	Contribuições	25
1.3	<i>Estrutura do Documento</i>	26
2	Aprendizado de Máquina	28
2.1	<i>Principais Categorias do Aprendizado de Máquina</i>	29
2.1.1	Aprendizado de Máquina Clássico	29
2.1.2	Aprendizado por Reforço	31
2.1.3	Método dos Conjuntos	32
2.1.4	Aprendizado profundo	32
2.2	<i>O Processo de Aprendizado</i>	33
3	Técnicas de Otimização	35
3.1	<i>Algoritmo de Busca Aleatória</i>	35
3.2	<i>Gradiente Descendente</i>	36
3.3	<i>Gradiente Descendente com Momento</i>	39
3.4	<i>Método de Newton</i>	40
3.4.1	BFGS Quasi-Newton	41
3.5	<i>Gradiente Conjugado</i>	41
3.5.1	Reinicialização de Powell/Beale	42
3.6	<i>Retropropagação Levenberg-Marquart</i>	43
3.7	<i>Retropropagação Resiliente</i>	43
4	Redes Neurais Artificiais	45
4.1	<i>Perceptron</i>	45
4.2	<i>Funções de Ativação</i>	45
4.2.1	Sigmóide	46
4.2.2	Tangente Hiperbólica	46

4.2.3	Unidade de Retificação Linear	47
4.2.4	Unidade Linear Retificada com Vazamento	47
4.2.5	Softmax	48
4.3	<i>Algoritmo de Retropropagação</i>	48
4.3.1	Propagação Direta	50
4.3.2	Retropropagação na Camada de Saída	52
4.3.3	Retropropagação na Camada Oculta	53
4.3.4	Etapas Algoritmo Retropropagação	54
4.4	<i>Redes Neurais Feedforward</i>	54
4.5	<i>Redes Neurais Recorrentes</i>	55
4.5.1	Redes Neurais Recorrentes de Memória Longa de Curto Prazo	56
4.6	<i>Matriz de Confusão</i>	58
4.7	<i>Métricas de Classificação</i>	60
4.7.1	Acurácia	60
4.7.2	Sensibilidade e Especificidade	60
4.7.3	Medida-F1	60
5	Sistema de Inferência Adaptativo Neuro-Fuzzy	62
6	Normas Internacionais	65
6.1	<i>IEC-60034 Máquinas Elétricas Rotativas</i>	65
6.1.1	IEC 60034-1 - Características e Desempenho	65
6.1.2	IEC 60034-26 Efeitos das Tensões Desequilibradas no Desempenho de Motores de Indução	66
6.2	<i>ISO 20816-1 Vibração Mecânica - Medição e Avaliação da Vibração da Máquina</i>	68
6.3	<i>ISO 13373-9 Monitoramento da Condição e Diagnóstico de Máquinas - Monitoramento de Condição de Vibração - Parte 9: Técnicas de Diagnóstico para Motores Elétricos</i>	70
7	Ferramentas Matemáticas e Computacionais	72
7.1	<i>Modelo da Máquina Assíncrona</i>	72
7.2	<i>Estimativa dos Parâmetros da Máquina de Indução a partir dos Dados do Fabricante</i>	74

7.3	<i>Categorias dos Motores de Indução Trifásicos</i>	79
8	Bancos de Dados	81
8.1	<i>MIT-IEC 60034-1</i>	81
8.2	<i>MIT-ISO 20816</i>	84
8.3	<i>MIT-IEC 60034-26</i>	85
9	Metodologia	88
9.1	<i>Sistema de Diagnóstico Região de Operação IEC 60034-1</i>	88
9.1.1	Preparação dos Dados	89
9.1.2	Processamento	91
9.1.3	Validação	91
9.2	<i>Sistema de Diagnóstico de Barra Quebrada do Rotor ISO 20816</i>	95
9.2.1	Preparação dos Dados	96
9.2.2	Processamento	97
9.2.3	Validação	100
9.3	<i>Sistema de Diagnóstico Fuzzy IEC 60034-26</i>	100
9.3.1	Subsistema Adaptativo de Inferência Neuro-Fuzzy	100
9.3.2	Detecção do Fator de Redução IEC 60034-26	103
9.3.3	Teste de Precisão SDF IEC 60034-26	104
10	Resultados e Discussões	106
10.1	<i>Sistema de Diagnóstico Região de Operação IEC 60034-1</i>	106
10.1.1	Comparação Algoritmos de Otimização	106
10.1.2	Avaliação de Desempenho	107
10.1.3	Limitações SDRO IEC 60034-1	109
10.2	<i>Sistema de Diagnóstico de Barra Quebrada do Rotor ISO 20816</i>	110
10.2.1	Limitações SDBQ ISO 20816	114
10.3	<i>Sistema de Diagnóstico Fuzzy IEC 60034-26</i>	114
10.3.1	Limitações SDF IEC 60034-26	115
11	Conclusões Gerais e Trabalhos Futuros	117
11.1	<i>Trabalhos Futuros</i>	118
11.2	<i>Publicações</i>	119

11.3	<i>Trabalhos Completos Publicados</i>	119
11.4	<i>Artigos Submetidos para Publicação</i>	119
11.5	<i>Informações Extras</i>	120
	REFERÊNCIAS	121
	Apêndice A – Modelo Máquina Assíncrona ATP	131
	Apêndice B – Partida Direta de um Motor de Indução Trifásico: Comparação entre os <i>softwares</i> ATP e MATLAB/Simulink	135

1 Introdução

1.1 Motivação e Relevância do Trabalho

Os motores de indução são cargas predominantes em diversos setores da indústria. Nas plantas industriais estas máquinas sofrem com perturbações mecânicas e elétricas que causam paradas na produção e prejuízos econômicos. Monitorar as grandezas elétricas e mecânicas contribui para diminuir a ocorrência destes distúrbios e reduz os custos com manutenção. Um sistema de monitoramento inteligente prevê a ocorrência de falhas automaticamente e pode ser desenvolvido utilizando-se algoritmos genéticos, redes neurais artificiais e/ou lógica *fuzzy* (HWANG; JEN; SHEN, 2009; RAO; Srinivasa Pai; NAGABHUSHANA, 2012; UNAL *et al.*, 2013).

Em 1943, o neurofisiologista Warren McCulloch e o matemático Walter Pitts desenvolveram o primeiro modelo matemático do neurônio artificial. De acordo com McCulloch e Pitts (1943), os eventos que ocorrem em uma rede neural podem ser tratados por suposições lógicas, culminando na formulação da “Teoria das Redes sem Círculos”. Hopfield (1982), avançando na fronteira do conhecimento, inventou a rede neural Hopfield. Com uma estrutura simples, esta rede simula a memória humana sendo usada no reconhecimento de padrões. A ideia da retropropagação surgiu na década de 60, mas foi somente em 1986 que Rumelhart, Hinton e Williams (1986) formalizaram a aplicação da retropropagação no treinamento das redes neurais artificiais.

No início dos anos 90, as redes neurais artificiais começaram a ser aplicadas em pesquisas de monitoramento da operação de motores de indução trifásicos. Em 1991, um trabalho pioneiro elaborado por Chow, Mangum e Yee (1991), diagnosticou falhas em um motor de indução de 0.75 hp utilizando redes neurais artificiais multicamadas (MLP). Chow, Mangum e Yee (1991) desenvolveram um detector de falhas de alta precisão utilizando redes neurais. No mesmo ano Chow e Yee (1991) propuseram um método de diagnóstico composto por um filtro de ruído e um detector de falhas com redes neurais artificiais. Nos anos de 1993 e 1994, um método de detecção de falhas que combinava redes neurais artificiais e lógica *fuzzy*, sistema conhecido como neuro-fuzzy, foi desenvolvido (CHOW; GOODE, 1993; GOODE; CHOW, 1994).

Na segunda metade da década de 90, Hyun e Nam (1995) mapearam os sinais de vibração durante a operação dos motores quando submetidos a falhas nos rolamentos. As características destes sinais alimentaram uma rede neural de regressão geral que diagnosticava falhas nas

engrenagens e nos rolamentos. [Elsimary \(1996\)](#) desenvolveu uma metodologia que combinava um algoritmo genético com um filtro de novidade, "*novelty filter*", diagnosticando falhas em motores de indução instalados em um parque industrial.

As redes neurais artificiais também compuseram ferramentas utilizadas na pesquisa e desenvolvimento de algoritmos de monitoramento inteligente nos primeiros anos do século XXI. Os sistemas de detecção de falhas baseados em redes neurais desenvolvidos por [Ye, Wu e Zargari \(2000\)](#), [Kim e Parlos \(2002\)](#), [Han et al. \(2006\)](#) estavam pautados na teoria de processamento de sinais, aplicando a Transformada de Wavelet na decomposição das ondas de correntes do estator.

[Samanta e Al-Balushi \(2003\)](#), [Yang e Kim \(2006\)](#) desenvolveram sistemas de monitoramento de falhas que identificavam rachaduras nos rolamentos. A vibração excessiva gerada por um rolamento defeituoso pode danificar os demais componentes do motor. [Samanta e Al-Balushi \(2003\)](#) compararam técnicas de pré-processamento para extração das características dos sinais de vibração (filtros passa-alta, passa-banda e Transformada Wavelet). Os autores constataram que o uso da Transformada Wavelet não melhorava significativamente os resultados de diagnóstico. [Yang e Kim \(2006\)](#) aplicaram a teoria de *Dempster-Shafer* no prognóstico de defeitos, resultando em um acompanhamento das condições da máquina em tempo real.

Nos sistemas preditivos desenvolvidos por [Khomfoi e Tolbert \(2007\)](#), as redes neurais perceptron multicamadas identificavam a categoria de falta, monofásica, bifásica ou trifásica e informava a bobina que sofreu o curto-circuito. Nos experimentos realizados por [Masrur et al. \(2007\)](#) os algoritmos informavam se o inversor estava operando com falha de circuito aberto.

Diferentes métodos de extração e seleção das características foram usados nas amostras do conjunto de treinamento/teste das redes neurais presentes nos sistemas de classificação de falhas. [Liang, Iwnicki e Zhao \(2013\)](#) aplicaram o espectro de potência e espectros de ordem superior na extração destas características. [Unal et al. \(2014\)](#), entretanto, deram preferência pela transformada de Hilbert e Transformada Rápida de Fourier.

Entre os anos de 2018 e 2020, diversos artigos científicos publicados descreveram métodos de monitoramento de falhas utilizando redes neurais recorrentes (RNR) e redes neurais convolucionais (RNC). Estas produções acadêmicas compõem o chamado "estado da arte". [Kerboua et al. \(2018\)](#) classificaram em tempo real a situação operativa do motor de indução. Os autores diagnosticaram falhas mecânicas e elétricas utilizando uma categoria de rede neural recorrente, conhecida como *Long Short-Term Memory* (LSTM). Estudos realizados por [Kerboua et al. \(2018\)](#) compararam a eficácia entre RNC e LSTM no diagnóstico de curtos-circuitos nos enrolamentos do estator. [Hao et al. \(2020\)](#), [Wang et al. \(2020b\)](#) analisaram a viabilidade das

LSTM na detecção de falhas nos rolamentos. As falhas nos rolamentos ocasionam cerca de 40% a 90% dos desligamentos inesperados dos motores de indução (TOMA; PROSVIRIN; KIM, 2020). Lee, Pack e Lee (2019), Xu *et al.* (2019) extraíram as características dos sinais de vibração com identificação dos danos nos rolamentos, feita através das RNC.

As normativas internacionais, IEC 60034 - *Rotating electrical machines* e ISO 20816 - *Mechanical vibration*, contêm critérios que, se adotados pelos responsáveis técnicos em uma indústria, minimizam a probabilidade de ocorrência de falhas nos motores de indução trifásicos. Os algoritmos de aprendizado de máquina são utilizados na resolução de problemas de previsão ou classificação. Nestas duas categorias de solução, é possível adotar critérios específicos na escolha das entradas e saídas do conjunto treinamento/validação/teste. Portanto, desenvolver sistemas de monitoramento preditivo com MLP, RNC ou RNR que contêm os critérios das normativas internacionais constitui um avanço no “estado da arte”. Esta temática foi pouco explorada na literatura acadêmica. Estes algoritmos poderão ser aplicados em um contexto prático, como por exemplo no projeto dos equipamentos de proteção de máquinas elétricas.

1.2 *Objetivos e Contribuições*

1.2.1 *Objetivos*

Os objetivos gerais desta pesquisa consistem na investigação, desenvolvimento e validação de arquiteturas de algoritmos inteligentes que efetuam o diagnóstico das falhas em motores de indução trifásicos. Neste contexto, segue os objetivos específicos:

- Desenvolver estratégias baseadas nas técnicas de aprendizado de máquina, especificamente redes neurais artificiais *feedforward* e redes neurais recorrentes.
- Utilizar as diretrizes e recomendações descritas nas normativas internacionais IEC 60034 e ISO 20816 no desenvolvimento dos sistemas de diagnóstico das condições operativas dos motores de indução trifásicos.

1.2.2 *Contribuições*

Desta forma, segue as principais contribuições da pesquisa:

- Integrar metodologias de reconhecimento de padrões, através do emprego de redes neurais artificiais, com as diretrizes empíricas descritas nas normativas internacionais IEC 60034 e ISO 20816.
- Disponibilizar um banco de dados com parâmetros do circuito equivalente de cerca de 1700 motores de indução trifásicos com rotor em gaiola.
- Desenvolvimento de um sistema neuro-*fuzzy* de previsão do Fator de Desequilíbrio baseado na norma IEC 60034-26 avançando a fronteira do conhecimento.

Durante a realização da pesquisa foram desenvolvidos os seguintes sistemas:

1. **Sistema de Diagnóstico Região de Operação IEC 60034-1:** algoritmo que monitora a região de operação de um motor de indução trifásico de acordo com as diretrizes e recomendações da norma IEC 60034-1: *Rotating electrical machines – Part 1: Rating and performance*.
2. **Sistema de Diagnóstico de Barra Quebrada ISO 20816:** informa ao usuário se o motor está operando com barras quebradas no rotor e identifica a zona operativa conforme as classificações da normativa ISO 20816: *Mechanical vibration — Measurement and evaluation of machine vibration — Part 1: General guidelines*.
3. **Sistema de Diagnóstico Fuzzy IEC 60034-26:** identifica automaticamente o Fator de Desequilíbrio da máquina e calcula o Fator de Redução conforme recomendação da norma IEC 60034-26: *Rotating electrical machines - Part 26: Effects of unbalanced voltages on the performance of three-phase cage induction motors*.

1.3 Estrutura do Documento

Esta tese é organizada em seções e subseções. O Capítulo 2 introduz alguns conceitos da teoria do aprendizado de máquina. No Capítulo 3 faz-se uma extensa investigação das técnicas de otimização utilizadas no treinamento de redes neurais artificiais. A representação matemática do algoritmo de retropropagação, e as métricas que avaliam a precisão e generalização das redes neurais de reconhecimento de padrões se encontram no Capítulo 4. A teoria do sistema de Inferência Neuro-*Fuzzy* Adaptativo é descrita no Capítulo 5. As diretrizes e recomendações das normativas internacionais IEC 60034-1, IEC 60034-26 e ISO 20816 utilizadas na elaboração dos algoritmos dos sistemas de diagnóstico estão no Capítulo 6. O Capítulo 7 e Capítulo 8 descrevem as ferramentas matemáticas e computacionais, e o banco de dados, respectivamente.

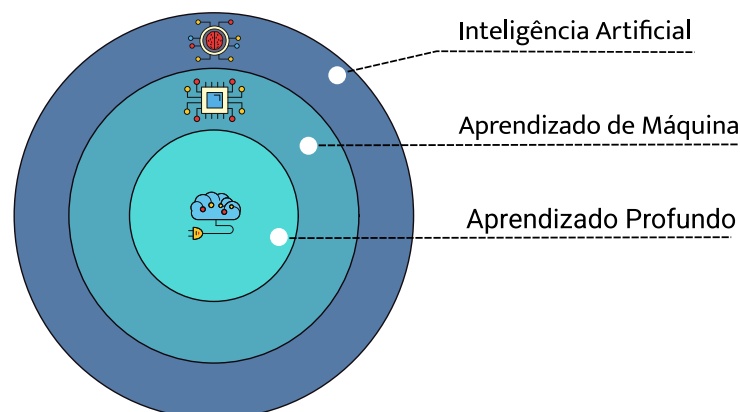
As metodologias usadas na elaboração dos sistemas de monitoramento podem ser consultadas no Capítulo 9. Discussões e resultados se encontram no Capítulo 10. Por fim, no Capítulo 11, são reportadas as conclusões, diretivas para trabalhos futuros e publicações aceitas e submetidas durante a realização da pesquisa.

2 Aprendizado de Máquina

Na segunda metade do século XX, precisamente em 1950, Alan Mathison Turing, matemático inglês, publicou um artigo intitulado "*Computing Machinery and Intelligence*" onde afirmou acreditar que em um futuro próximo às máquinas poderiam pensar e raciocinar. [Cope-land \(2004\)](#) descreve a trajetória do cientista precursor da consolidação do campo de estudos conhecido como "*Inteligência Artificial*" (IA). Etimologicamente, o termo *Inteligência Artificial* é composto por duas palavras: "*Inteligência*" e "*Artificial*". A *Inteligência* mensura a capacidade de entender, conhecer, raciocinar, planejar e interpretar. O adjetivo "*Artificial*" caracteriza algo que não é produzido pela natureza, ou seja, criado pelo homem. Portanto, podemos dizer que qualquer tecnologia ou algoritmo que permita à máquina imitar, desenvolver ou demonstrar a cognição ou comportamento humano é IA.

O aprendizado de máquina é uma subcampo da inteligência artificial, [Figura 1](#), também conhecido como "*Machine Learning*" e utiliza técnicas de análise preditiva. [Mitchell \(1997\)](#) apresenta uma definição para o aprendizado de máquina: "um programa de computador aprende com a experiência E no que diz respeito a alguma classe de tarefas T e medição de desempenho P se o seu desempenho nas tarefas em T , conforme medido por P , melhora com experiência E ". Imagine que se deseje que um computador aprenda a jogar damas. Inicialmente se fornece ao computador um conjunto de amostras que descrevem de forma numérica os resultados e jogadas feitas em diversos jogos de damas (experiência E). A classe de tarefas, T , é jogar um jogo de damas. Através de modelos matemáticos se mensura a probabilidade P de vencer jogos futuros.

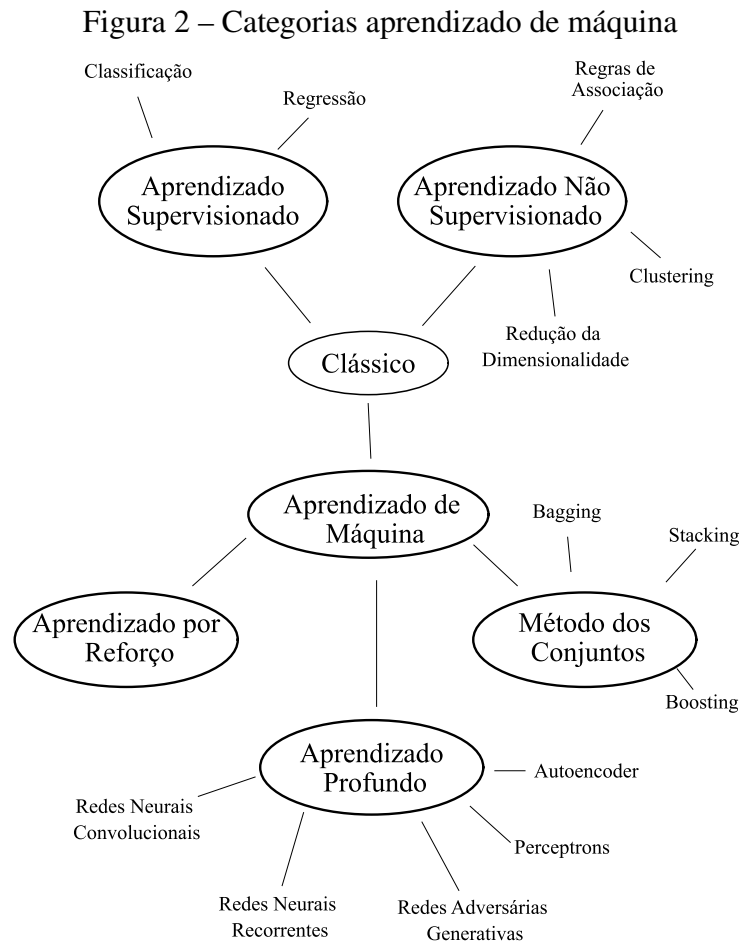
Figura 1 – Áreas da inteligência artificial



Fonte – Pedro H. Camargos, 2021

2.1 Principais Categorias do Aprendizado de Máquina

Existem muitas categorias de aprendizado que se diferenciam pela composição e finalidade a que se destina o algoritmo. As principais categorias são mostradas no diagrama da Figura 2.



Fonte – Adaptado de Zubarev (2019)

2.1.1 Aprendizado de Máquina Clássico

O aprendizado de máquina clássico é dividido em duas categorias: aprendizado supervisionado e não supervisionado (SINGH, 2020). No aprendizado supervisionado o algoritmo aprende a partir de um conjunto treinamento fornecido previamente pelo programador. Os dados deste conjunto são rotulados com valores numéricos ou rótulos do tipo *strings* (MIRJALILI; FARIS; ALJARA, 2019; BHATTACHARJEE, 2019). O aprendizado supervisionado é usado para resolver problemas de regressão e classificação (JOSEPHINE *et al.*, 2020; KUMAR *et al.*, 2018). Zubarev (2019) explica didaticamente as diferenças entre classificação e regressão:

algoritmos classificadores preveem o rótulo de uma entrada e trabalham com saídas discretas, enquanto a regressão prediz um número através de funções contínuas.

A regressão pode ser linear ou polinomial. Nos problemas de classificação, o conjunto treinamento é rotulado. Quando se filtra e-mails *spam*, por exemplo, o algoritmo de classificação recebe um conjunto treinamento com um excesso de e-mails rotulados. O algoritmo de classificação ao receber como entrada um novo endereço eletrônico, não pertencente ao conjunto treinamento, informa ao usuário se ele é ou não *spam*. Os algoritmos de classificação são: *Naive Bayes*, árvore de decisão, máquinas de vetores de suporte (SVM), regressão logística e k-vizinhos mais próximos (K-NN) (JOSHI, 2020).

O objetivo do aprendizado não supervisionado é encontrar padrões em dados sem rótulo (VERMEULEN, 2020). Três técnicas principais compõem o aprendizado não supervisionado: agrupamento (*Clustering*), redução de dimensionalidade e regras de associação. Para exemplificar, a aplicação do aprendizado não supervisionado, imagine que o departamento de engenharia de uma indústria possui 1000 relatórios que descrevem as falhas ocorridas nos motores de indução. Os algoritmos do aprendizado não supervisionado podem agrupar esses relatórios em grupos com características semelhantes. O conjunto treinamento composto por parte dos relatórios não são rotulados, ou seja, a informação sobre onde ocorreu a falha não está contida no conjunto treinamento. Por meio do *Clustering* os relatórios são divididos em grupos. Os relatórios que descrevem falhas nos rolamentos formariam o grupo *A*, os que descrevem as falhas nos enrolamentos do estator pertenceriam ao grupo *B* e assim sucessivamente.

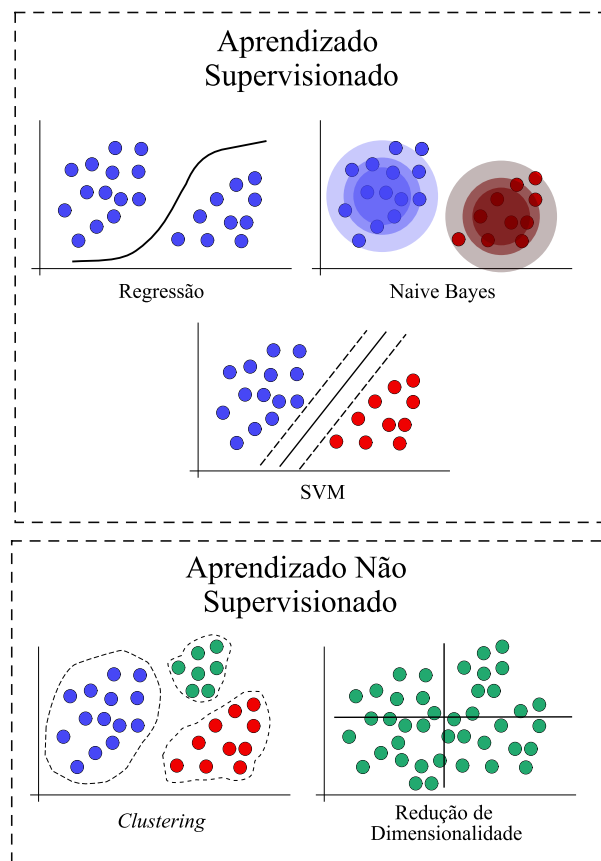
As dimensões do espaço são definidas de forma que cada uma das dimensões são perpendiculares entre si. Em um espaço tridimensional, formado pelos eixos x , y e z os planos xy , xz e yz são ortogonais. Na teoria do aprendizado de máquina, se uma entrada possui três características independentes elas podem ser consideradas ortogonais e de dimensão 3. Por exemplo, imagine que os dados de entrada são características de um motor de ímãs permanentes: comprimento longitudinal da carcaça, resistência dos enrolamentos do estator, peso dos ímãs internos e volume da máquina. Neste caso, o algoritmo terá um espaço de entrada de quatro dimensões. O método de redução de dimensionalidade é usado na otimização dos algoritmos de classificação. Reduzir a dimensão do espaço de características aumenta a precisão e rapidez computacional.

As principais técnicas do aprendizado clássico podem ser vistas na Figura 3. Regressão logística e SVM são modelos discriminativos, também chamados de condicionais. Estes modelos aprendem o limite de decisão (*decision boundary*) e operam com melhor desempenho quando

se fornece na entrada um grande número de amostras no conjunto treinamento. Por outro lado, os algoritmos classificadores de *Naive Bayes* são modelos gerativos. Matematicamente, estes aprendem a distribuição de probabilidade conjunta e requerem menor quantidade de amostras de treinamento se comparados aos modelos discriminativos.

Suponha um conjunto treinamento dado pelas entradas x e rótulos y . O modelo generativo aprende a distribuição de probabilidade conjunta $p(x, y)$ e faz as previsões usando as regras de Bayes para calcular $p(y/x)$ para escolher o rótulo. Nos modelos discriminativos $p(y/x)$ é calculada diretamente (NG; JORDAN, 2001). Ambos os métodos usam probabilidade condicional na categorização dos rótulos.

Figura 3 – Técnicas do aprendizado de máquina clássico



Fonte – Pedro H. Camargos, 2021

2.1.2 Aprendizado por Reforço

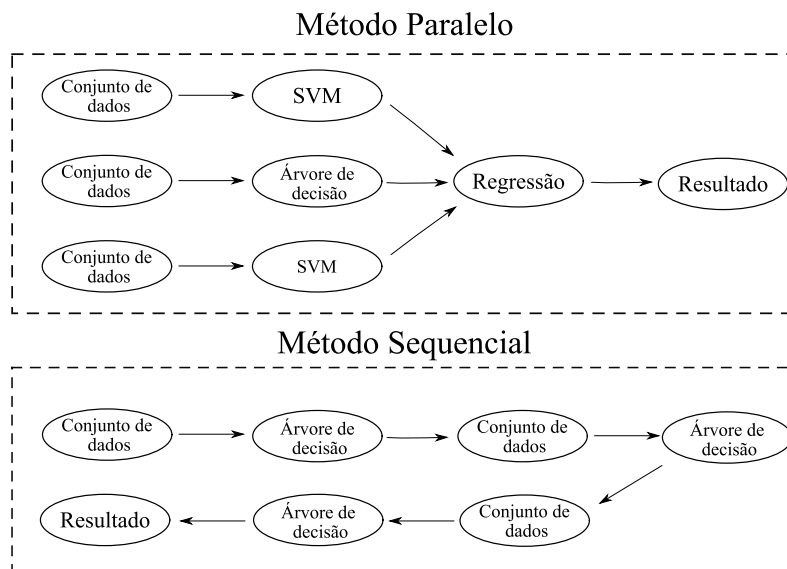
De acordo com Mueller e Massaron (2016), o aprendizado por reforço é como a aprendizagem por tentativa e erro. O algoritmo toma decisões baseadas na retroalimentação positiva ou negativa que recebe, aprendendo com o "ambiente". Nos métodos de aprendizagem por

reforço o sistema está continuamente interagindo com o ambiente em busca de produzir um comportamento desejado (JOSHI, 2020). Nos carros autônomos, esta forma de aprendizado é comumente utilizada uma vez que seria inviável o piloto automático memorizar todas as situações que podem ocorrer durante o tráfego do veículo. Portanto, para solucionar este problema se constrói cidades virtuais que permitem que o piloto automático aprenda com os erros e acertos durante os trajetos.

2.1.3 Método dos Conjuntos

O método dos conjuntos combina vários algoritmos de aprendizado de máquina com o objetivo de aprimorar os resultados. Estas técnicas produzem soluções mais precisas que os algoritmos individuais. O método dos conjuntos pode ser sequencial e paralelo, como mostra a Figura 4, (ZHOU; YU, 2005). Nos métodos sequenciais se explora a dependência entre os algoritmos de aprendizagem com treinamento sequencial. Nos métodos paralelos o erro é reduzido pelo cálculo da média ou aplicando um algoritmo de regressão.

Figura 4 – Método dos conjuntos



Fonte – Pedro H. Camargos, 2021

2.1.4 Aprendizado profundo

O aprendizado profundo é um subcampo do aprendizado de máquina com métodos de construção, treinamento e uso de redes neurais. Redes neurais convolucionais são frequentemente

utilizadas nos algoritmos de aprendizagem profunda (WANG *et al.*, 2020a). Dentre as inúmeras aplicações, se destacam a identificação de objetos em fotos e vídeos, reconhecimento e síntese de voz e processamento de imagem. Kulkarni e Satapathy (2020) afirmam que as técnicas de aprendizagem profunda possuem a capacidade de encontrar relações complexas entre um grande número de parâmetros de entradas.

2.2 O Processo de Aprendizado

Uma analogia pode perfeitamente representar o processo de aprendizagem de um algoritmo. Suponha que o algoritmo de aprendizagem seja representado por um estagiário de engenharia que trabalha em uma indústria. Imagine que este aluno não tenha cursado a disciplina de máquinas elétricas, logo, não consegue diferenciar um motor de corrente contínua de um motor de indução em gaiola.

Durante o primeiro mês de estágio o aluno necessita realizar a manutenção preditiva de uma máquina assíncrona. O seu supervisor mostra-lhe um número significativo de imagens de motores de indução com todas as características que tornam estas máquinas distinguíveis de outras categorias de motores. As características extraídas destas imagens são o conjunto treinamento. O estagiário (algoritmo) através de suas capacidades cognitivas constrói em sua mente o conceito e a imagem que define o motor de indução. Esta definição, que expressa as qualidades de uma máquina assíncrona, é análoga a uma função-alvo, também chamada de função hipótese.

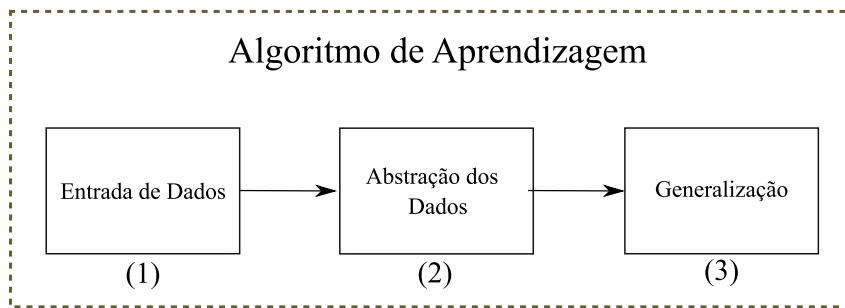
Para encontrar a melhor função-alvo o algoritmo passa pelo processo de otimização. Neste processo ele avalia muitas funções-alvo contidas no chamado espaço ou classe de hipóteses. A otimização ocorre no interior da mente do aluno. A função de custo, conhecida como função de perda ou função de erro conduz o processo de otimização mostrando quais são os ajustes nos parâmetros do algoritmo devem ser feitos para minimizar a função de custo (MUELLER; MASSARON, 2016).

A Figura 5 mostra as etapas do processo de aprendizagem de máquina: *Entrada de Dados*, *Abstração dos Dados* e *Generalização*. Imagine que uma imobiliária queira estimar o preço de uma casa de 750 m^2 . Inicialmente é feita a cotação dos preços de casas e suas respectivas áreas, como mostrado na Figura 6-a. Esse conjunto de dados (preço \times área) corresponde ao conjunto treinamento. O conjunto deve ser ajustado pelo programador, que elimina coordenadas

não coerentes (etapa 1). O algoritmo usado para estimar os preços de imóveis será do tipo aprendizagem supervisionada, uma regressão linear, cujo objetivo é obter uma função hipótese, $h(x) = w_1 + w_2x$.

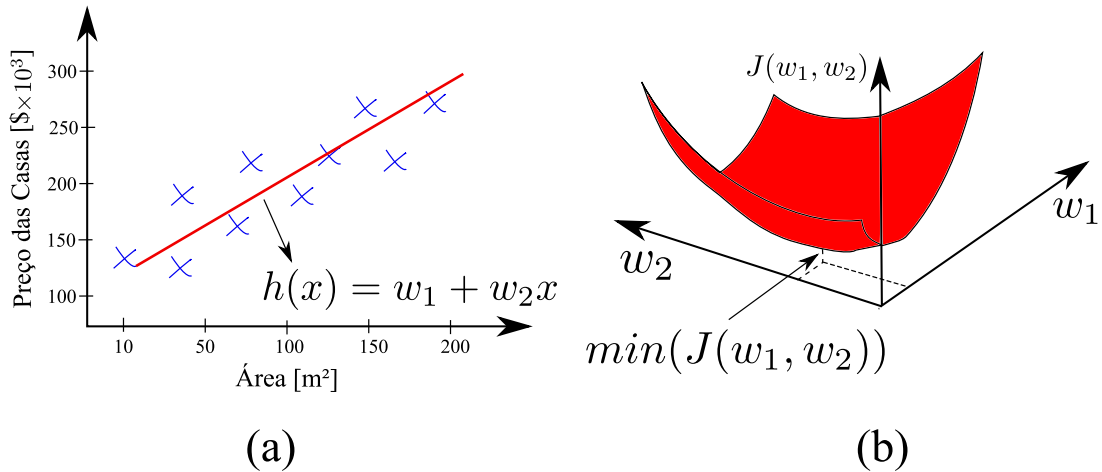
Na etapa de "Abstração dos Dados" estima-se os parâmetros w_1 e w_2 através da minimização da função de custo, $J(w_1, w_2)$, (Figura 6-b), encontrando valores específicos para w_1 e w_2 . Portanto, é possível prever o preço de uma casa que não está contida no conjunto treinamento. Esta etapa é chamada de "Generalização". Utilizando a função h a imobiliária será capaz de prever o preço da casa de 750 m^2 .

Figura 5 – Estrutura do processo de aprendizagem



Fonte – Pedro H. Camargos, 2021

Figura 6 – Exemplo aplicação da regressão linear (a) função hipótese (b) função de custo

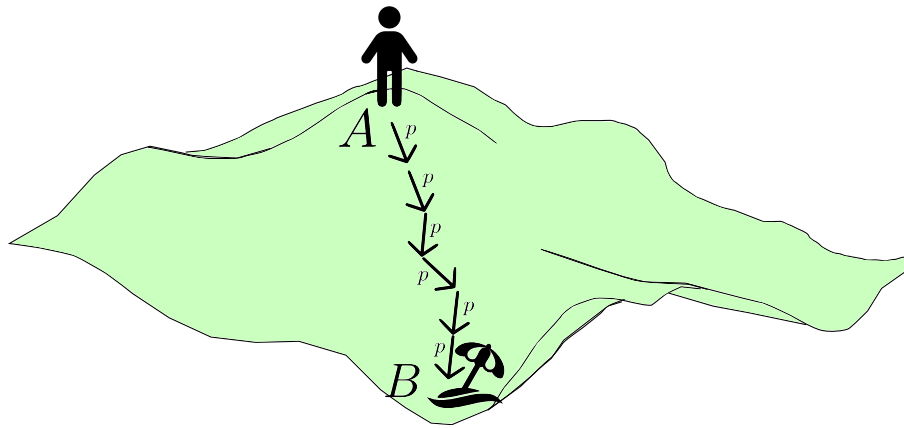


Fonte – Pedro H. Camargos, 2021

3.2 Gradiente Descendente

O Algoritmo do Gradiente Descendente (GD) consiste em encontrar os valores dos parâmetros que minimizam a função de custo. Este procedimento, similar ao que ocorre no BA, é realizado de forma iterativa. Para entender o funcionamento do GD, considere a situação mostrada na Figura 8. Um homem que está no ponto A , deseja atravessar o vale da montanha em direção à praia, ponto B . Cada seta representa a direção de descida mais íngreme. O indivíduo parte do ponto A e percorre o caminho até o ponto B , mínimo local. Cada trecho p percorrido representa uma iteração no GD.

Figura 8 – Noção intuitiva GD



Fonte – Pedro H. Camargos, 2021

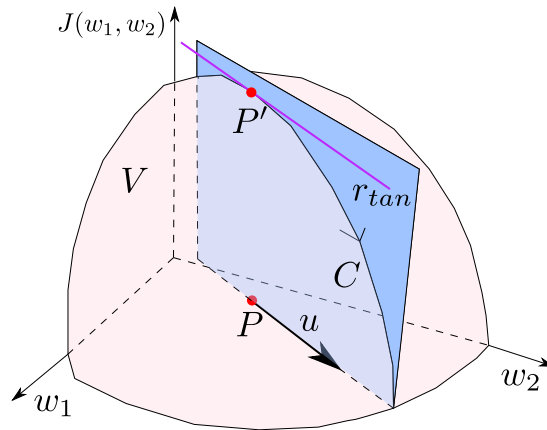
Para entender a formulação matemática do GD é preciso revisitar alguns conceitos do cálculo diferencial e integral de várias variáveis. Considere a superfície V , cuja representação se encontra na Figura 9. Imagine que se queira determinar a taxa de variação de $z = J(w_1, w_2)$, no ponto $P = (w_1^0, w_2^0, 0)$, na direção de um vetor unitário \mathbf{u} . Um plano vertical que passa por $P' = (w_1^0, w_2^0, J(w_1, w_2))$ intercepta V em uma curva C . A inclinação da reta tangente r_{tan} a C é a taxa de variação de $J(w_1, w_2)$ na direção do vetor unitário, conhecida como a derivada direcional de J na direção de \mathbf{u} no ponto P . Matematicamente a definição da derivada direcional é definida como:

Definição 1. (Derivada Direcional): Seja $J : D \rightarrow \mathfrak{R}$ uma função de n variáveis, isto é $D \subseteq \mathfrak{R}^n$. Considere um ponto P no interior de D e $\mathbf{u} \in \mathfrak{R}^n$ um vetor unitário. A derivada direcional de J em P na direção de \mathbf{u} é

$$D_{\mathbf{u}}J(P) = \lim_{h \rightarrow 0} \frac{J(P + h\mathbf{u}) - J(P)}{h} \quad (2)$$

se esse limite existir.

Figura 9 – Interpretação geométrica da derivada direcional



Fonte – Adaptado de Stewart (2013)

Stewart (2013) explica que a derivada direcional na direção de \mathbf{u} em um ponto $P = (w_1^0, w_2^0, w_3^0, \dots, w_n^0)$ é o produto escalar dos vetores gradiente e diretor \mathbf{u} . É possível demonstrar utilizando a Equação (3) que o valor máximo da derivada direcional em um ponto P ocorre quando \mathbf{u} possui o mesmo sentido do vetor gradiente. Em outras palavras o gradiente indica o máximo crescimento ou decrescimento da função J .

$$D_{\mathbf{u}}J(P) = \nabla J(P) \cdot \mathbf{u} \tag{3}$$

Na Equação (3), ∇J é o gradiente de J . Segue a definição do ∇J :

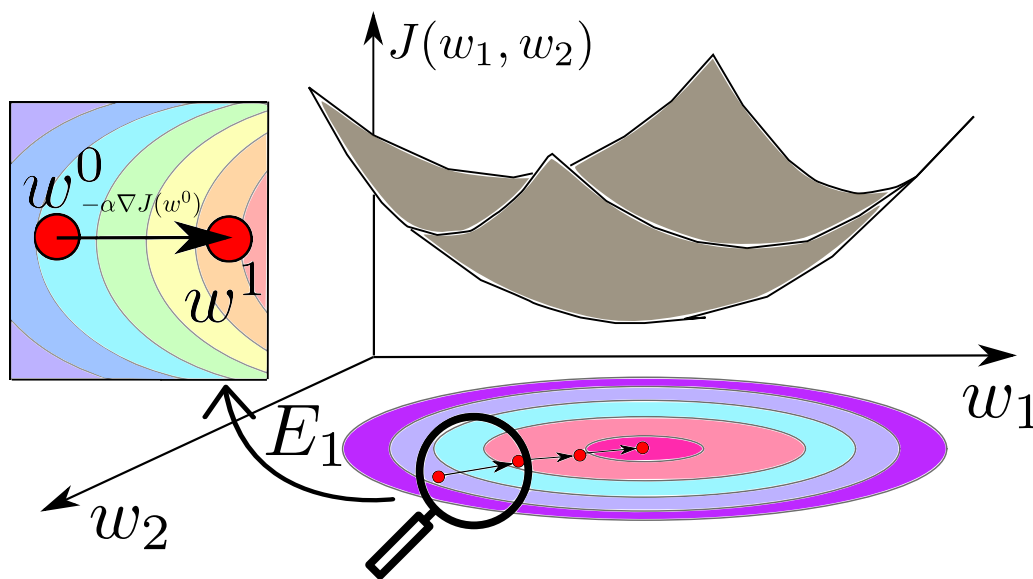
Definição 2. (Gradiente): Seja $J : D \rightarrow \Re$ uma função de n variáveis $(w_1, w_2, w_3, \dots, w_n)$ que possui derivadas parciais no ponto $P = (w_1^0, w_2^0, w_3^0, \dots, w_n^0)$. O gradiente de J em P denotado por ∇J , é o vetor

$$\nabla J(P) = \left[\frac{\partial J(P)}{\partial w_1}, \frac{\partial J(P)}{\partial w_2}, \frac{\partial J(P)}{\partial w_3}, \dots, \frac{\partial J(P)}{\partial w_n} \right]^T \tag{4}$$

O GD é um método de otimização local de primeira ordem. O ponto de mínimo local da função de custo é encontrado após k iterações (etapas), que atualizam os valores $\mathbf{w}^0, \mathbf{w}^1, \dots, \mathbf{w}^k$. Neste algoritmo, a direção do máximo decrescimento da função J é a mesma do vetor gradiente negativo. Na etapa E_1 , por exemplo, diferentemente do que ocorre no BA, a direção escolhida é a do vetor $-\alpha \nabla J(\mathbf{w}^0)$ indicado na Figura 10. Portanto, a derivada direcional que fornece a maior taxa de variação de J no ponto representado pelo vetor \mathbf{w}^0 possui o vetor diretor com a mesma direção e sentido de $\nabla J(\mathbf{w}^0)$. A formulação matemática de cada etapa é dada por:

$$\begin{aligned}
 \mathbf{w}^0 \\
 \mathbf{w}^1 &= \mathbf{w}^0 - \alpha \nabla J(\mathbf{w}^0) \\
 \mathbf{w}^2 &= \mathbf{w}^1 - \alpha \nabla J(\mathbf{w}^1) \\
 &\vdots \\
 \mathbf{w}^k &= \mathbf{w}^{k-1} - \alpha \nabla J(\mathbf{w}^{k-1})
 \end{aligned} \tag{5}$$

Figura 10 – Aplicação do GD para a função de custo $J(w_1, w_2)$



Fonte – Pedro H. Camargos, 2021

A escolha adequada da taxa de aprendizagem possibilita acelerar o processo de minimização da função de custo com um pequeno intervalo de tempo gasto no processamento computacional. Geralmente o desenvolvedor adota um valor fixo para α . Por meio da técnica de depuração, o programador verifica se o processo de otimização está operando adequadamente. O comportamento da função J é avaliado, na medida em que se aumenta o número de iterações. J deve diminuir com o aumento do número de iterações. Se α é um valor muito pequeno, o algoritmo demora para convergir. Por outro lado, para valores altos de α o algoritmo provavelmente irá divergir.

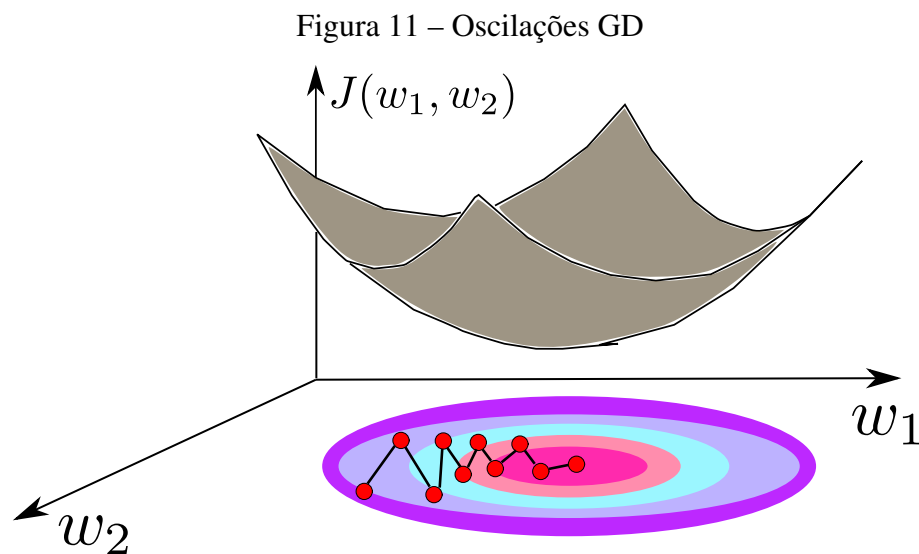
Existem três variações do GD: Algoritmo de Gradiente Descendente em Lote (GDL), Algoritmo de Gradiente Descendente Estocástico (GDE) e Algoritmo de Gradiente Descendente em Minilotes (GDML). No GDL, todas as amostras do conjunto treinamento são utilizadas em cada etapa, o que faz dele um algoritmo de fácil implementação. Entretanto, é necessário armazenar enorme quantidade de dados, o que acarreta em aumento do custo computacional.

Para um conjunto treinamento com um número grande de amostras, o GDE é o mais adequado. Cada iteração usa apenas uma amostra do conjunto treinamento (GÉRON, 2019; KHIRIRAT; FEYZMAHDAVIAN; JOHANSSON, 2017). Consequentemente, o tempo de processamento gasto para atingir a convergência é maior do que no GDL. Esta variação do GD é frequentemente utilizada no treinamento de redes neurais artificiais (MUSSO, 2020).

O GDML opera com m amostras por iteração. Isto significa que após percorrer um número limitado de exemplos do conjunto treinamento os parâmetros da função hipótese são atualizados. De acordo com Ng (2020), o GDML é mais eficiente que o GDE. A operação com mini lotes, , em que os parâmetros são atualizados a cada fração m do conjunto treinamento possibilita um processo de convergência mais estável que o algoritmo estocástico.

3.3 Gradiente Descendente com Momento

No processo de otimização o GD aplica passos infinitesimais. Oscilações no processo de atualização do erro podem ocorrer dependendo do valor estipulado para a taxa de aprendizagem, Figura 11. Para solucionar este problema, o ideal é adotar um alto valor para a taxa de aprendizagem que não resulte em oscilações. Uma maneira de aumentar α sem ocasionar oscilações, é modificar o GD incluindo uma constante que determina o efeito de mudanças passadas nos pesos na direção atual do movimento (LIU; GAO; YIN, 2020). Este novo algoritmo é conhecido como Gradiente Descendente com Momento (GDM).



Fonte – Pedro H. Camargos, 2021

Em geral, o método do GDM pode ser escrito como:

$$\mathbf{d}^{k-1} = \beta \mathbf{d}^{k-2} + (1 - \beta)(-\nabla J(\mathbf{w}^{k-1})) \quad (6)$$

$$\mathbf{w}^k = \mathbf{w}^{k-1} + \alpha \mathbf{d}^{k-1} \quad (7)$$

O "momento" no GDM se refere a nova direção de descida \mathbf{d}^{k-1} , atualizada de acordo com a Equação (6). O valor $\beta \in [0, 1]$ é chamado de "peso momentâneo". [Watt, Borhani e Katsaggelos \(2020\)](#) mencionam que na prática são usados valores de $\beta \in [0.7, 1]$.

3.4 Método de Newton

O método de Newton tem o objetivo de minimizar a função $J(\mathbf{w}^k)$ tal que $\mathbf{w} \in \mathbb{R}^n$. Suponha uma função quadrática $J(\mathbf{w}) = 0.5\mathbf{w}^T \mathbf{A}\mathbf{w} + \mathbf{d}^T \mathbf{w} + c$, onde \mathbf{A} é simétrica. O gradiente de J pode ser escrito como:

$$\nabla J(\mathbf{w}) = \mathbf{A}\mathbf{w} + \mathbf{d} \quad (8)$$

O princípio do método de Newton é localizar o ponto estacionário da aproximação da função de custo. Aplicando a expansão de segunda ordem da série de Taylor de J têm-se que:

$$J(\mathbf{w}^{k+1}) \approx J(\mathbf{w}^k) + \nabla J(\mathbf{w}^k)^T \Delta \mathbf{w}^k + 0.5(\Delta \mathbf{w}^k)^T \mathbf{A}_k \Delta \mathbf{w}^k \quad (9)$$

Utilizando a Equação (8) em (9) e igualando a zero obtêm-se a Equação (10) utilizada no método de Newton, onde \mathbf{A} é a matriz hessiana.

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \mathbf{A}_k^{-1} \nabla J(\mathbf{w}^k) \quad (10)$$

O método de Newton não opera satisfatoriamente quando se dispõe de um grande número de amostras no conjunto treinamento. Neste caso, o algoritmo necessita armazenar uma matriz hessiana que com uma enorme quantidade de elementos. Para solucionar este problema, os cientistas criaram os métodos Quasi-Newton que utiliza os mesmos passos do método de Newton com aproximações na matriz hessiana.

3.4.1 BFGS Quasi-Newton

Broyden (1970), Fletcher (1970), Goldfarb (1970) e Shanno (1970) publicaram artigos descrevendo técnicas que solucionam os problemas de otimização não linear. Uma das metodologias adotadas pelos autores, propõe uma aproximação para a matriz hessiana e ficou conhecida como método BFGS Quasi-Newton. Johnson (2019) detalha este método com a inversa da matriz hessiana, sendo dada por:

$$(\mathbf{A}^{k+1})^{-1} = (\mathbf{A}^k)^{-1} - \frac{1}{\gamma^T \Delta \mathbf{w}^k} \left[(\mathbf{A}^k)^{-1} \gamma (\Delta \mathbf{w}^k)^T + \Delta \mathbf{w}^k \gamma^T (\mathbf{A}^k)^{-1} \right. \quad (11)$$

$$\left. - \left(1 + \frac{\gamma^T (\mathbf{A}^k)^{-1} \gamma}{\gamma^T \Delta \mathbf{w}^k} \right) \Delta \mathbf{w}^k (\Delta \mathbf{w}^k)^T \right] \quad (12)$$

3.5 Gradiente Conjugado

No método do gradiente conjugado o vetor direção \mathbf{p}^k é mutualmente conjugado com a direção anterior, \mathbf{p}^{k-1} com respeito a matriz Hessiana, tal que

$$\Delta \mathbf{w}^k = (\mathbf{w}^{k+1} - \mathbf{w}^k) = \alpha \mathbf{p}^k \quad (13)$$

onde , com direção inicial $\mathbf{p}^0 = -\nabla J(\mathbf{w}^0)$, e as direções subsequentes são calculadas por:

$$\mathbf{p}^k = -\nabla J(\mathbf{w}^k) + \beta_X \mathbf{p}^{k-1} \quad (14)$$

Um conjunto de vetores, $C = \{\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^k, \dots, \mathbf{p}^n\}$, é mutualmente conjugado com a matriz hessiana \mathbf{A} , se e somente se cada vetor $\mathbf{p}^k \in C$ satisfazer a seguinte igualdade,

$$\mathbf{p}^{kT} \mathbf{A} \mathbf{p}^j = 0 \quad k \neq j \quad (15)$$

tal que a matriz \mathbf{A} seja um fator da função quadrática $J(\mathbf{w})$. Hagan *et al.* (2016) explicita a relação entre a variação do gradiente da função de custo e a matriz hessiana como sendo:

$$\Delta(\nabla J(\mathbf{w}^k)) = \nabla J(\mathbf{w}^{k+1}) - \nabla J(\mathbf{w}^k) = \mathbf{A} \Delta \mathbf{w}^k \quad (16)$$

O escalar β_X pode assumir diferentes formulações. Dentre elas se destaca a consideração matemática proposta por [Polak e Ribiere \(1969\)](#) expressa pela Equação (17).

$$\beta_{PR} = \frac{\Delta(\nabla J(\mathbf{w}^{k-1}))\nabla J(\mathbf{w}^k)}{\nabla J(\mathbf{w}^{k-1})^T \nabla J(\mathbf{w}^{k-1})} \quad (17)$$

O método do Gradiente Conjugado em Escala (SGD) foi descrito pela primeira vez por [Møller \(1993\)](#), e é baseado em direções conjugadas. Este algoritmo não executa uma pesquisa de linha em cada iteração e requer um mínimo de memória em cada iteração com atualização independente dos parâmetros ([MATHWORKS, 2020d](#); [BAGHIRLI, 2015](#)). No SGD o parâmetro β_X é calculado por:

$$\beta_{SGD} = \frac{|\nabla J(\mathbf{w}^{k+1})|^2 - \nabla J(\mathbf{w}^{k+1})^T \nabla J(\mathbf{w}^k)}{\nabla J(\mathbf{w}^k)^T \nabla J(\mathbf{w}^k)} \quad (18)$$

3.5.1 Reinicialização de Powell/Beale

Para os algoritmos de gradiente conjugado anteriormente citados, a direção de descida do gradiente é redefinida utilizando o negativo do gradiente. Neles o ponto de reinicialização padrão ocorre quando o número de iterações é igual ao número de parâmetros da rede (pesos e vieses). Porém, existem outros métodos de reinicialização que podem melhorar a eficiência do treinamento, a Reinicialização de Powell/Beale por exemplo. Essa técnica resolve o problema de pouca ortogonalidade entre o gradiente atual e o gradiente anterior. A Reinicialização de Powell/Beale é descrita a seguir:

- **Etapa 1:** Seja $\mathbf{w}^1 \in \mathfrak{R}^n$; $c_1, c_2 \in (0, 1)$, $c_3 \in (0, \infty)$, $\epsilon \in [0, 1]$
Faça $\mathbf{d}^1 = -\nabla J(\mathbf{w}^k)$; $k = t = 1$
- **Etapa 2:** Se $k = 1$, avance para a **etapa 5**;
Caso $k - t \geq n$ faça $t = k - 1$;
Caso contrário se $k \geq 2$ e $|\nabla J(\mathbf{w}^{k-1})^T \nabla J(\mathbf{w}^k)| > c_1 \|\nabla J(\mathbf{w}^k)\|^2$, faça $t = k - 1$;
- **Etapa 3:** Se $k > t + 1$, calcule \mathbf{d}^k como se segue:

$$\mathbf{d}^k = -\nabla J(\mathbf{w}^k) + \beta^k \mathbf{d}^{k-1} + \gamma^k \mathbf{d}^t \quad (19)$$

As informações das formulações de β^k e γ^k se encontram em [Powell \(1977\)](#) e [Dai e Yuan \(1998\)](#);

Caso $k = t + 1$ calcule \mathbf{d}^k adotando $\gamma^k = 0$;

- **Etapa 4:** Se a relação $-c_3 \|\nabla J(\mathbf{w}^k)\|^2 \leq \mathbf{d}^{kT} \nabla J(\mathbf{w}^k) \leq -c_2 \|\nabla J(\mathbf{w}^k)\|^2$ não for satisfeita, faça $t = k - 1$ e vá para a **etapa 3**;
 - **Etapa 5:** Procure uma nova direção fazendo $\mathbf{w}^{k+1} = \mathbf{w}^k + \alpha^k \mathbf{d}^k$;
 - **Etapa 6:** Se $\|\nabla J(\mathbf{w}^{k+1})\| \leq \epsilon$, pare;
- Caso contrário faça $k = k + 1$ e avance para a **etapa 2**;

3.6 Retropropagação Levenberg-Marquart

O algoritmo Retropropagação Levenberg-Marquart utiliza uma aproximação matriz hessiana no treinamento da rede, tal que,

$$\mathbf{A} = J_a(\mathbf{w}^k)^T J_a(\mathbf{w}^k) \quad (20)$$

onde J_a é a matriz jacobiana. Seja uma função $J(\mathbf{w}^k)$ composta pela soma de funções quadráticas dada por:

$$J(\mathbf{w}^k) = \sum_{i=1}^N q_i^2(\mathbf{w}^k) \quad (21)$$

Hagan e Menhaj (1994) mostram que $\nabla J(\mathbf{w}^k)$ é igual à multiplicação de J_a , pela função quadrática $q(\mathbf{w}^k)$, tal que:

$$\nabla J(\mathbf{w}^k) = J_a(\mathbf{w}^k)^T q(\mathbf{w}^k) \quad (22)$$

Portanto aplicando uma modificação Levenberg-Marquart na Equação (10) no método de Gauss-Newton têm-se que,

$$\Delta \mathbf{w}^k = [J_a(\mathbf{w}^k)^T J_a(\mathbf{w}^k) + \mu I]^{-1} J_a(\mathbf{w}^k)^T q(\mathbf{w}^k) \quad (23)$$

onde μ é um fator multiplicador. Este algoritmo parece ser o método mais rápido para treinar redes neurais *feedforward* de tamanho moderado com centenas de pesos (MATHWORKS, 2021d).

3.7 Retropropagação Resiliente

O algoritmo de Retropropagação Resiliente, sistematicamente detalhado por Riedmiller e Braun (1993), opera por meio da adaptação do parâmetro de atualização, Δ . Cada vez que a

derivada parcial correspondente ao peso w^k muda de sinal, o parâmetro de atualização Δ diminui de um fator η^- . Se a derivada mantém seu sinal Δ será proporcional a η^+ . Matematicamente têm-se que:

$$\Delta_{ij}^k = \begin{cases} \eta^+ \Delta^{k-1}, & \text{se } \nabla J(\mathbf{w}^{k-1}) \nabla J(\mathbf{w}^k) > 0 \\ \eta^- \Delta^{k-1}, & \text{se } \nabla J(\mathbf{w}^{k-1}) \nabla J(\mathbf{w}^k) < 0 \\ \Delta^{k-1}, & \text{senão} \end{cases} \quad (24)$$

Aplicada a regra de adaptação aos parâmetros de atualização, o peso é atualizado da seguinte forma,

$$\Delta \mathbf{w}^{k+1} = \mathbf{w}^k + \Delta \mathbf{w}^k \quad (25)$$

onde $\Delta \mathbf{w}^k$ é dada por:

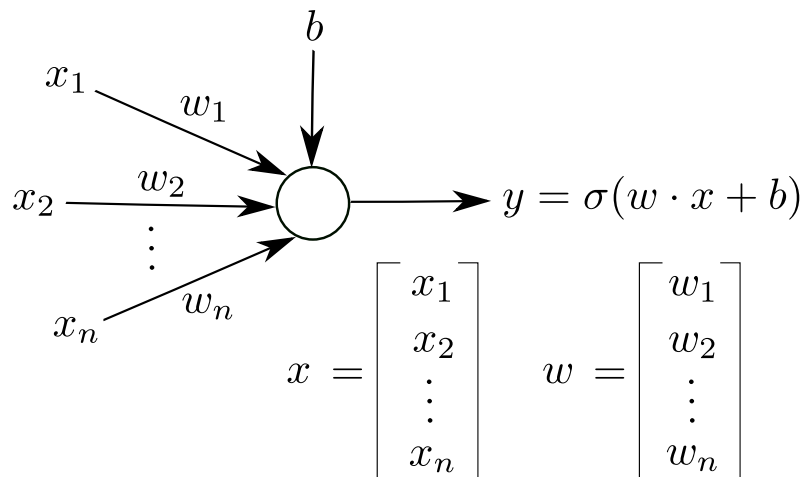
$$\Delta \mathbf{w}^k = \begin{cases} +\Delta^k, & \text{se } \nabla J(\mathbf{w}^k) > 0 \\ -\Delta^k, & \text{se } \nabla J(\mathbf{w}^k) < 0, \\ 0 & \text{senão} \end{cases} \quad (26)$$

4 Redes Neurais Artificiais

4.1 Perceptron

As redes neurais artificiais foram desenvolvidas com o objetivo de simular o funcionamento de uma rede de neurônios biológicos. O neurônio artificial ou nó é uma unidade lógica, alimentada por entradas, que computam uma saída. Em 1969, os cientistas da computação Marvin Lee Minsky e Seymour Aubrey Papert aprimoraram o modelo do neurônio artificial proposto por McCulloch e Pitts (1943). Minsky e Papert (1969) desenvolveram um novo modelo matemático para o neurônio artificial, conhecido como *perceptron*. Sua representação pode ser vista na Figura 12. O perceptron recebe os sinais de entrada, x e retorna uma saída $y = \sigma(w \cdot x + b)$. O vetor w e o viés (*bias*), b , são os responsáveis pelo armazenamento das informações na rede (PHIL, 2017). A função de ativação, σ determina o comportamento do neurônio.

Figura 12 – Modelo do perceptron



Fonte – Pedro H. Camargos, 2021

4.2 Funções de Ativação

A função de ativação desempenha um importante papel no treinamento das redes neurais artificiais. Estas funções matemáticas fornecem a não linearidade necessária ao modelo (RASA-MOELINA; ADJAILIA; SINČÁK, 2020). O objetivo das funções de ativação é converter o sinal de entrada em um sinal de saída. As funções de ativação são classificadas em duas categorias: função linear por partes e função localmente quadrática.

Definição 3. (*Função linear contínua por partes*): uma função $f : \mathbb{R} \rightarrow \mathbb{R}$ é uma função linear por partes se esta é contínua e existe um número de pontos de quebra $q_1 \leq q_2 \leq \dots \leq q_k \in \mathbb{R}$ com $K \in \mathbb{N}$ tal que $f'(q_k-) \neq f'(q_k+)$ para todo $k = 1, \dots, K$ e $f(x)$ é linear em $(-\infty, q_1], [q_1, q_2], \dots, [q_{k-1}, q_k], [q_k, \infty)$.

Definição 4. (*Função localmente quadrática*): uma função $f : \mathbb{R} \rightarrow \mathbb{R}$ é uma função localmente quadrática se existe um intervalo aberto $(a, b) \subset \mathbb{R}$ no qual f é três vezes continuamente diferenciável e se existe $t \in (a, b)$ tal que $f'(t) \neq 0$ e $f''(t) \neq 0$.

Uma função linear por partes é uma função composta por um número limitado de segmentos lineares. Cada segmento é uma subfunção que possui um intervalo do domínio da função principal. Qualquer função de ativação suave não linear com a segunda derivada diferente de zero é localmente quadrática.

4.2.1 Sigmóide

A função sigmóide, (Equação (27)), $\sigma_{sig} : \mathbb{R} \rightarrow [0, 1]$ é uma função suave e continuamente diferenciável, categorizada como uma função localmente quadrática. O gráfico de σ_{sig} é mostrado na Figura 13-a. A curva possui duas assíntotas paralelas. Esta função também é utilizada nos algoritmos de aprendizado de máquina que contêm modelos de previsão probabilística, como, por exemplo, a regressão logística (LEVER; KRZYWINSKI; ALTMAN, 2016).

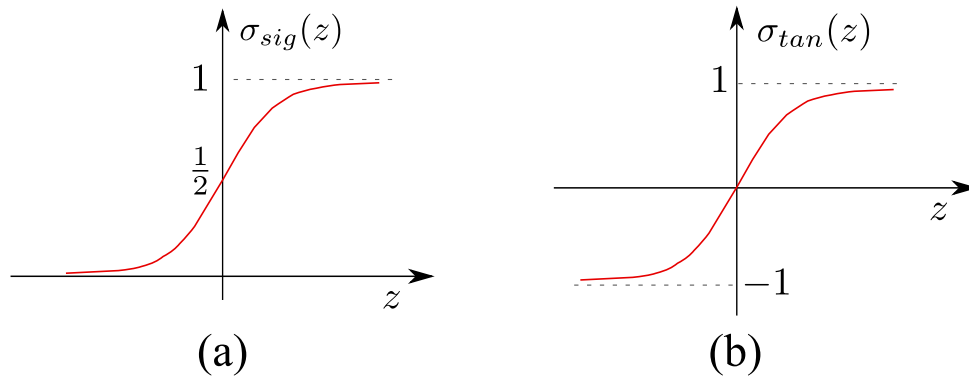
$$\sigma_{sig}(z) = \frac{1}{1 + e^{-z}} \quad (27)$$

4.2.2 Tangente Hiperbólica

A função tangente hiperbólica é localmente quadrática tal que $\sigma_{tan} : \mathbb{R} \rightarrow [-1, 1]$. É uma função simétrica, (Figura 13-b) sendo facilmente implementada nas arquiteturas de *hardware* (SHAKIBA; ZHOU, 2020). Matematicamente é definida como:

$$\sigma_{tan}(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (28)$$

Figura 13 – Funções localmente quadráticas (a) sigmóide (b) tangente hiperbólica



Fonte – Pedro H. Camargos, 2021

4.2.3 Unidade de Retificação Linear

A unidade de retificação linear (ReLU), Equação (29), é uma função linear amplamente utilizada nos problemas de aprendizado profundo. ReLU possui menor complexidade de implementação e maior desempenho no processo de convergência se comparada a função sigmóide. De acordo com [Ohn e Kim \(2019\)](#), o gráfico da ReLU (Figura 14-a) pode ser representado por uma combinação de funções por partes lineares.

$$\sigma_{Re}(z) = \max\{z, 0\} \quad (29)$$

4.2.4 Unidade Linear Retificada com Vazamento

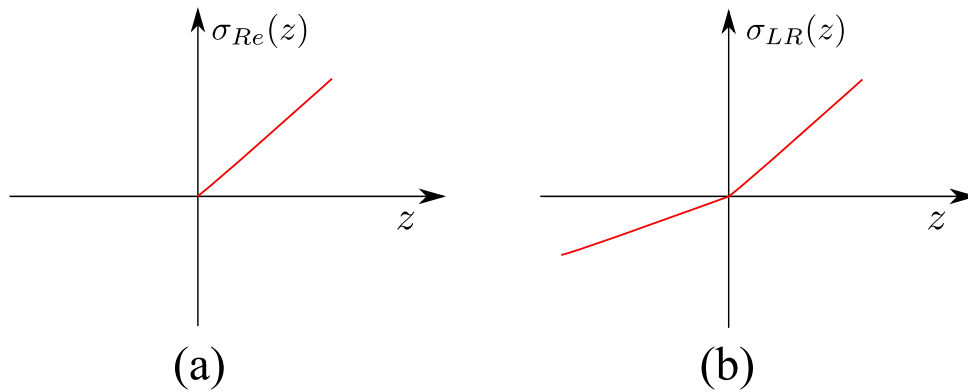
Proposta por [Maas, Hannun e Ng \(2013\)](#) a unidade linear retificada com vazamento ou função LReLU é uma variação da ReLU. O domínio da LReLU é composto por uma parte negativa, representada pela Equação (30), onde a é uma constante.

$$\sigma_{LR}(z) = \max\{z, az\} \text{ para } a \in (0, 1) \quad (30)$$

Conforme enunciado por [Pedamonti \(2018\)](#), duas dificuldades podem ser encontradas no treinamento das redes neurais artificiais profundas: desaparecimento e explosão de gradientes. Optar pela ReLU ao invés da sigmoide pode resolver estes problemas. Entretanto, o algoritmo pode sofrer com o “ReLU morrendo”, ou seja, os neurônios ReLU se tornam inativos e emitem apenas o valor “0” para qualquer entrada. Após a “morte” não ocorre a alteração dos pesos. Para

contornar este problema é preferível optar pela LReLU. Graficamente a LReLU é mostrada na Figura 14-b.

Figura 14 – Unidade lineares retificadas (a) ReLU (b) LReLU



Fonte – Pedro H. Camargos, 2021

4.2.5 Softmax

A função de ativação *Softmax* determina a probabilidade de um valor pertencer a uma classe. *Softmax* é usada nos problemas de classificação múltipla e aprendizado por reforço. As probabilidades calculadas estarão no intervalo de 0 a 1. Na construção das redes neurais a *Softmax* frequentemente compõe as camadas finais da rede. Sua representação matemática é dada por:

$$\sigma_{so}(z) = \frac{e^z}{\sum_{j=0}^k e^{z_j}} \quad (31)$$

4.3 Algoritmo de Retropropagação

Em 1974, o americano [Werbos \(1974\)](#) descreveu pela primeira vez o funcionamento do algoritmo de retropropagação (RP). A retropropagação é uma técnica utilizada no treinamento das redes neurais artificiais, cujo objetivo é atualizar os pesos minimizando a função de custo. Dois tipos principais de funções de custo são comumente utilizados no aprendizado supervisionado de uma rede neural: soma do erro quadrático médio (Equação (32)) e entropia cruzada (Equação

(33)). As variáveis y_i e \hat{y}_i correspondem as saídas fornecidas pelo conjunto treinamento e a saída prevista pela rede neural respectivamente.

$$J = \sum_{i=1}^m \frac{1}{2} (y_i - \hat{y}_i)^2 \quad (32)$$

$$J = - \sum_{i=1}^m [y_i \ln(\hat{y}_i) - (1 - y_i) \ln(1 - \hat{y}_i)] \quad (33)$$

De acordo com [Puig-Arnavat e Bruno \(2015\)](#), o RP é baseado na generalização da regra de aprendizado de Widrow-Hoff ou regra Delta. A regra Delta foi introduzida por [Widrow e Hoff \(1960\)](#). O algoritmo de retropropagação requer uma função de ativação diferenciável podendo ser implementado no modo "online" ou em lote. No modo "online", os parâmetros de rede são ajustados amostra a amostra. Os exemplos do conjunto treinamento percorrem a rede, um de cada vez, ajustando os pesos depois de cada exemplo. A atualização dos pesos na rede pode ser feita por meio da minimização da função erro quadrático médio aplicando o algoritmo do gradiente descendente. O fator de atualização do peso i é dado por:

$$\Delta w_{ij} = -\alpha \frac{\partial J}{\partial w_{ij}} \quad (34)$$

O Δw_{ij} é a variação do neurônio j na conexão i . Aplicando a regra da cadeia na Equação (34) têm-se que:

$$\Delta w_{ij} = -\alpha \frac{\partial J}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}} \quad (35)$$

Seja o neurônio j que possui k conexões da camada anterior. O parâmetro net_j está relacionado com todas as entradas recebidas, x_i , e com o viés b_j , tal que:

$$net_j = \sum_{i=0}^k x_i w_{ij} + b_j \quad (36)$$

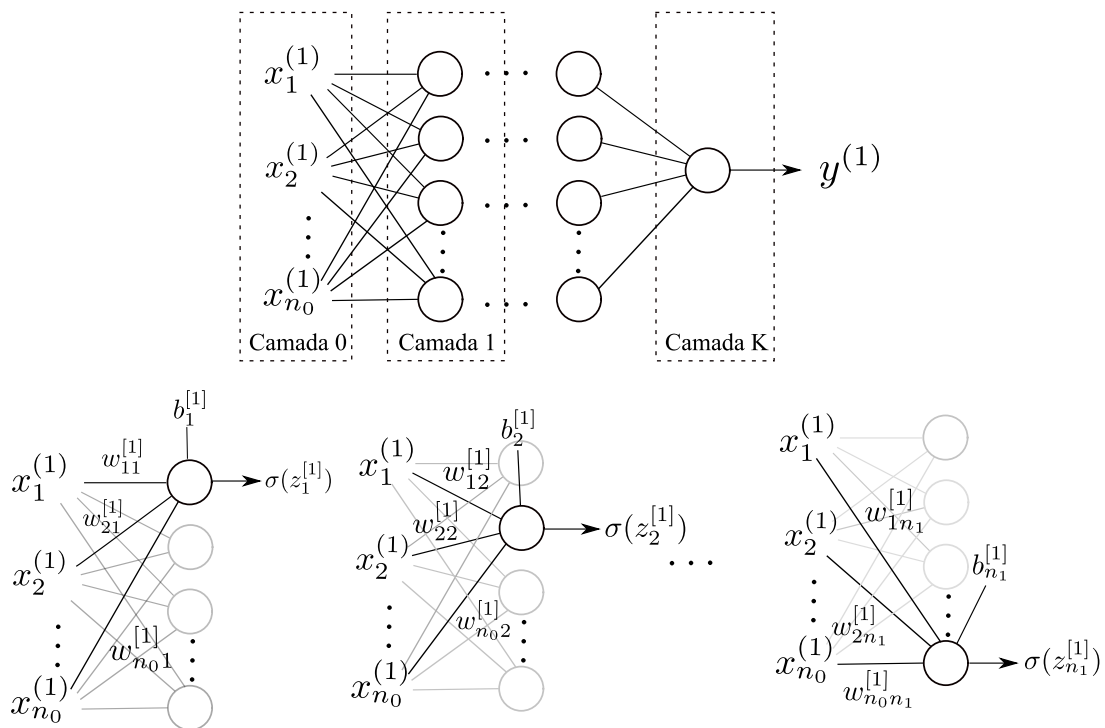
O erro e_j do neurônio j é calculado por meio da Equação (37). O valor do e_j varia conforme a categoria da camada a qual pertence o neurônio j : saída ou oculta. O RP pode ser dividido em quatro etapas: propagação direta, retropropagação na camada de saída, retropropagação nas camadas ocultas e atualização dos pesos.

$$e_j = -\frac{\partial J}{\partial net_j} = -\frac{\partial J}{\partial \sigma(net_j)} \frac{\partial \sigma(net_j)}{\partial net_j} \quad (37)$$

4.3.1 Propagação Direta

Seja uma rede neural formada por $k + 1$ camadas. As camadas são representadas pelo sobrescrito $[i]$, onde $i = 0, 1, \dots, k$. Portanto, as camadas ocultas possuem números de índices entre 1 e $k - 1$. As camadas são compostas por n_j quantidades de neurônios onde $j = 0, 1, \dots, k$ corresponde ao número da camada. Por exemplo, a camada oculta 4, possui n_4 neurônios. A rede neural é mostrada na Figura 15.

Figura 15 – Modelo rede neural artificial



Fonte – Pedro H. Camargos, 2021

Considere um conjunto treinamento S composto por m vetores de características, $\mathbf{x}^{(i)} \in \mathbb{R}^{n_0}$ para $i = 1, 2, \dots, m$. A rede neural é alimentada por um vetor de características $\mathbf{x}^{(1)}$. Os perceptrons da camada 1 recebem as entradas $x_1^{(1)}, x_2^{(1)}, \dots, x_{n_0}^{(1)}$ multiplicadas pelos respectivos pesos e calcula as saídas através da função de ativação σ . Os pesos relacionados a todos os nós da camada 1 são representados por:

$$\mathbf{W}_1^{[1]} = \begin{bmatrix} w_{11}^{[1]} \\ w_{21}^{[1]} \\ \vdots \\ w_{n_0 1}^{[1]} \end{bmatrix} \quad \mathbf{W}_2^{[1]} = \begin{bmatrix} w_{12}^{[1]} \\ w_{22}^{[1]} \\ \vdots \\ w_{n_0 2}^{[1]} \end{bmatrix} \quad \dots \quad \mathbf{W}_{n_1}^{[1]} = \begin{bmatrix} w_{1n_1}^{[1]} \\ w_{2n_1}^{[1]} \\ \vdots \\ w_{n_0 n_1}^{[1]} \end{bmatrix} \quad (38)$$

A Equação (39) calcula as variáveis de entrada das funções de ativação. A representação dos vetores é dada pela Equação (40). As saídas da camada 1 também são vetorizadas, cujo vetor pode ser visto na Equação (41).

$$\mathbf{Z}^{1} = \mathbf{W}^{[1]T} \mathbf{X}^{(1)} + \mathbf{b}^{[1]} \quad (39)$$

$$\mathbf{Z}^{1} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ \vdots \\ z_{n_1}^{[1]} \end{bmatrix} \quad \mathbf{W}^{[1]} = \begin{bmatrix} \mathbf{W}_1^{[1]} \\ \mathbf{W}_2^{[1]} \\ \vdots \\ \mathbf{W}_{n_1}^{[1]} \end{bmatrix} \quad \mathbf{X}^{(1)} = \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \vdots \\ x_{n_0}^{(1)} \end{bmatrix} \quad \mathbf{b}^{[1]} = \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ \vdots \\ b_{n_1}^{[1]} \end{bmatrix} \quad (40)$$

$$\mathbf{a}^{1} = \begin{bmatrix} \sigma(z_1^{[1]}) \\ \sigma(z_2^{[1]}) \\ \vdots \\ \sigma(z_{n_1}^{[1]}) \end{bmatrix} \quad (41)$$

O procedimento de vetorização descrito anteriormente pode ser estendido para todas as k camadas. O cálculo dos vetores de ativação para m exemplos em S é obtido pelo seguinte algoritmo:

Algoritmo 1 Laço de repetição propagação direta

- 1: **while** $i \leq m$ **do**
 - 2: $\mathbf{z}^{[1](i)} = \mathbf{W}^{[1]T} \mathbf{X}^{(i)} + \mathbf{b}^{[1]}$
 - 3: $\mathbf{a}^{[1](i)} = \sigma(\mathbf{z}^{[1](i)})$
 - 4: $\mathbf{z}^{[2](i)} = \mathbf{W}^{[2]T} \mathbf{X}^{(i)} + \mathbf{b}^{[2]}$
 - 5: $\mathbf{a}^{[2](i)} = \sigma(\mathbf{z}^{[2](i)})$
 - 6: \vdots
 - 7: $\mathbf{z}^{[k](i)} = \mathbf{W}^{[k]T} \mathbf{X}^{(i)} + \mathbf{b}^{[k]}$
 - 8: $\mathbf{a}^{[k](i)} = \sigma(\mathbf{z}^{[k](i)})$
-

As matrizes de ativação $A^{[1]}, A^{[2]}, \dots, A^{[k]}$ facilitam a implementação computacional do algoritmo de propagação direta. Através da transformação de vetores é possível obter as saídas de todos os m exemplos ao mesmo tempo, por meio das seguintes equações:

$$\mathbf{Z}^{[1]} = \mathbf{W}^{[1]} \mathbf{X} + \mathbf{b}^{[1]} \quad (42)$$

$$\mathbf{A}^{[1]} = \sigma(\mathbf{Z}^{[1]}) \quad (43)$$

$$\mathbf{Z}^{[2]} = \mathbf{W}^{[2]} \mathbf{X} + \mathbf{b}^{[2]} \quad (44)$$

$$\mathbf{A}^{[2]} = \sigma(\mathbf{Z}^{[2]}) \quad (45)$$

$$\vdots$$

$$\mathbf{Z}^{[k]} = \mathbf{W}^{[k]} \mathbf{X} + \mathbf{b}^{[k]} \quad (46)$$

$$\mathbf{A}^{[k]} = \sigma(\mathbf{Z}^{[k]}) \quad (47)$$

$$\mathbf{Z}^{[1]} = \begin{bmatrix} \mathbf{Z}^{1} & \mathbf{Z}^{[1](2)} & \dots & \mathbf{Z}^{[1](m)} \end{bmatrix} \quad (48)$$

$$\mathbf{Z}^{[2]} = \begin{bmatrix} \mathbf{Z}^{[2](1)} & \mathbf{Z}^{2} & \dots & \mathbf{Z}^{[2](m)} \end{bmatrix} \quad (49)$$

$$\vdots$$

$$\mathbf{Z}^{[k]} = \begin{bmatrix} \mathbf{Z}^{[k](1)} & \mathbf{Z}^{[k](2)} & \dots & \mathbf{Z}^{[k](m)} \end{bmatrix} \quad (50)$$

$$\mathbf{A}^{[1]} = \begin{bmatrix} \mathbf{a}^{1} & \mathbf{a}^{[1](2)} & \dots & \mathbf{a}^{[1](m)} \end{bmatrix} \quad (51)$$

$$\mathbf{A}^{[2]} = \begin{bmatrix} \mathbf{a}^{[2](1)} & \mathbf{a}^{2} & \dots & \mathbf{a}^{[2](m)} \end{bmatrix} \quad (52)$$

$$\vdots$$

$$\mathbf{A}^{[k]} = \begin{bmatrix} \mathbf{a}^{[k](1)} & \mathbf{a}^{[k](2)} & \dots & \mathbf{a}^{[k](m)} \end{bmatrix} \quad (53)$$

4.3.2 Retropropagação na Camada de Saída

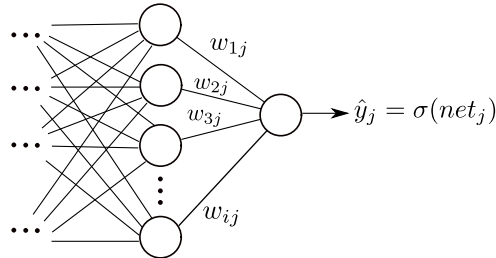
Considere o neurônio na camada de saída da rede neural representada na Figura 16 com função de ativação sigmoide. O valor do erro e_j para o neurônio j é:

$$e_j = -\frac{\partial J}{\partial \sigma(\text{net}_j)} \frac{\partial \sigma(\text{net}_j)}{\partial \text{net}_j} \quad (54)$$

$$e_j = -(y_j - \sigma(\text{net}_j)) \frac{\partial \sigma(\text{net}_j)}{\partial (\text{net}_j)} \quad (55)$$

$$e_j = \sigma(net_j)(y_j - \sigma(net_j))(1 - \sigma(net_j)) \tag{56}$$

Figura 16 – Neurônio j na camada de saída



Fonte – Pedro H. Camargos, 2021

Adotando a variável de saída como $o_j = \sigma(net_j)$, têm-se que:

$$e_j = o_j(y_j - o_j)(1 - o_j) \tag{57}$$

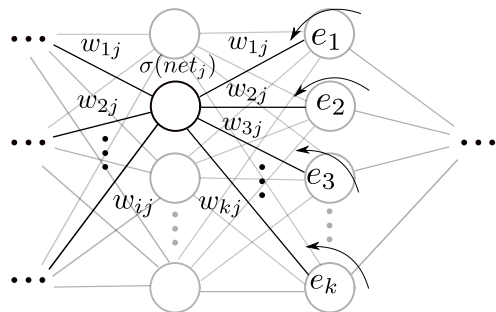
4.3.3 Retropropagação na Camada Oculta

A Figura 17 mostra um neurônio pertencente à camada oculta. O erro e_j é calculado por:

$$e_j = \frac{\partial J}{\partial \sigma(net_j)} \frac{\partial \sigma(net_j)}{\partial net_j} \tag{58}$$

$$e_j = \sum_{n=1}^k (y_n - \sigma(net_n)) \frac{\partial \sigma(net_n)}{\partial \sigma(net_j)} \frac{\partial \sigma(net_j)}{\partial net_j} \tag{59}$$

Figura 17 – Neurônio j na camada de oculta



Fonte – Pedro H. Camargos, 2021

Usando a regra da cadeia na Equação (59) pode se estabelecer que:

$$e_j = \sum_{n=1}^k (y_n - \sigma(net_n)) \frac{\partial \sigma(net_n)}{\partial net_n} \frac{\partial net_n}{\partial \sigma(net_j)} \frac{\partial \sigma(net_j)}{\partial net_j} \quad (60)$$

$$e_j = \sum_{n=1}^k \left(e_n w_{jn} \frac{\partial \sigma(net_j)}{\partial net_j} \right) \quad (61)$$

$$e_j = \sum_{n=1}^k w_{jn} e_n \sigma(net_j) (1 - \sigma(net_j)) \quad (62)$$

$$e_j = o_j (1 - o_j) \sum_{n=1}^k w_{jn} e_n \quad (63)$$

4.3.4 Etapas Algoritmo Retropropagação

A técnica de retropropagação otimizou o treinamento das redes com arquiteturas complexas diminuindo o custo computacional. O RP é um dos algoritmos mais importantes do aprendizado de máquina e pode ser implementado seguindo as etapas do Algoritmo 2.

Algoritmo 2 Algoritmo retropropagação

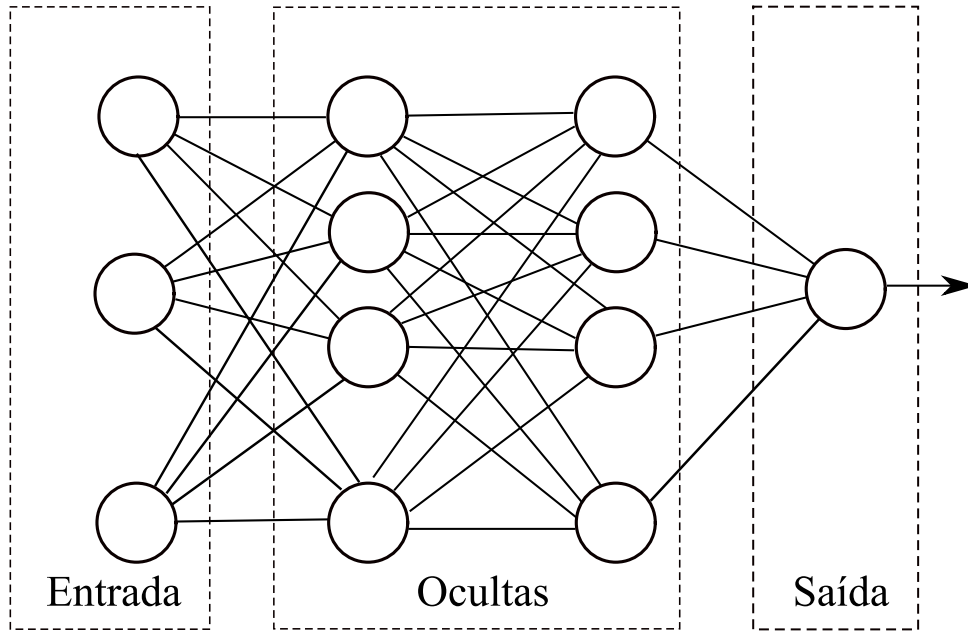
- 1: Inicializar todos os pesos com valores aleatórios
 - 2: Calcular as saídas da rede
 - 3: Para cada neurônio da saída calcular o erro: $e_j \leftarrow o_j(1 - o_j)(y_j - o_j)$
 - 4: Para cada neurônio presente na camada oculta calcular o erro: $e_j \leftarrow o_j(1 - o_j) \sum_{n=1}^k w_{jn} e_n$
 - 5: Ajustar os pesos da rede: $w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$
-

4.4 Redes Neurais Feedforward

Rede neural *feedforward*, também conhecida como perceptron multicamadas possui a arquitetura mais simples dentre as demais categorias de redes neurais artificiais (URSO *et al.*, 2018). A Figura 18 mostra a arquitetura de uma rede neural artificial *feedforward*. Cada camada é composta por um conjunto de perceptrons, que se conectam as outras unidades lógicas presentes nas camadas adjacentes. A saída da camada anterior fornece entrada para a próxima camada. A primeira camada é chamada de camada de entrada. Os dados de entrada da rede não são modificados nessa camada. Todas as variáveis de entrada que vieram da camada de entrada

são combinadas através de um ou mais nós na camada oculta. Redes neurais rasas possuem no máximo uma camada oculta. No entanto, redes neurais profundas possuem duas ou mais camadas ocultas. A camada de saída é a última camada de neurônios e fornece os resultados do algoritmo de aprendizagem.

Figura 18 – Rede neural *feedforward* e suas camadas

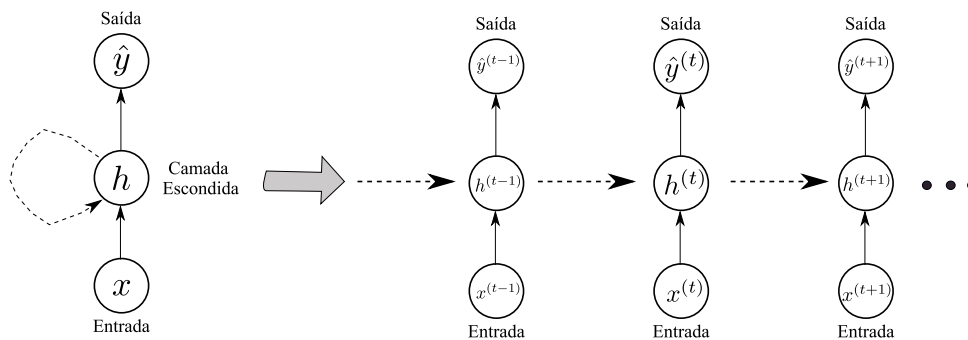


Fonte – Pedro H. Camargos, 2021

4.5 Redes Neurais Recorrentes

A rede neural recorrente usa as informações do passado para melhorar o seu desempenho com a inclusão de um ciclo que introduz a noção de tempo ao modelo. Estes ciclos são formados por conexões recorrentes que interligam etapas subsequentes de tempo, como mostrado na rede da Figura 19.

Figura 19 – Estrutura rede neural recorrente



Fonte – Pedro H. Camargos, 2021

A rede recebe uma entrada $\mathbf{x}^{(t)}$ e também os valores $\mathbf{h}^{(t-1)}$ oriundos do nó escondido no estado anterior de tempo, $t - 1$. A entrada neste estado influencia a saída no próximo estado de tempo, t , tal que

$$\hat{\mathbf{y}}^{(t)} = \sigma_{so}(W^{yh}\mathbf{h}^{(t)} + \mathbf{b}_y) \quad (64)$$

onde W^{yh} é uma matriz de pesos. Os valores de $\mathbf{h}^{(t)}$ são calculados utilizando a seguinte expressão matemática:

$$\mathbf{h}^{(t)} = \sigma_{sig}(W^{hx}\mathbf{x}^{(t)} + W^{hh}\mathbf{h}^{(t-1)} + \mathbf{b}_h) \quad (65)$$

As variáveis W^{hx} e W^{hh} correspondem as matrizes de pesos convencionais entre as entradas e pesos recorrentes entre a camada oculta, respectivamente. Os vetores \mathbf{b}_h e \mathbf{b}_y são os vieses. O algoritmo utilizado para treinar a RNR é chamado de retropropagação através do tempo ou *backpropagation through time* e foi desenvolvido por [Werbos \(1990\)](#). As redes neurais recorrentes são adequadas para resolver problemas com dados sequenciais de comprimento variável tais como: processamento de linguagem natural, classificação de sinais e análise de vídeos.

4.5.1 Redes Neurais Recorrentes de Memória Longa de Curto Prazo

[Hochreiter e Schmidhuber \(1997\)](#) desenvolveram o modelo das redes neurais recorrentes *Long Short-Term Memory* (LSTM), com o objetivo de solucionar o problema de desaparecimento do gradiente. Os autores melhoraram a capacidade de memória da célula recorrente padrão. As três estruturas de células LSTM são mostradas na Figura 20. O "portão do esquecimento" é uma função que decide se irá manter ou esquecer a informação. A Figura 20-a apresenta uma arquitetura sem esta função. As expressões matemáticas que modelam este tipo de célula são:

$$\mathbf{i}^{(t)} = \sigma_{sig}(W^{ih}\mathbf{h}^{(t-1)} + W^{ix}\mathbf{x}^{(t)} + \mathbf{b}_i) \quad (66)$$

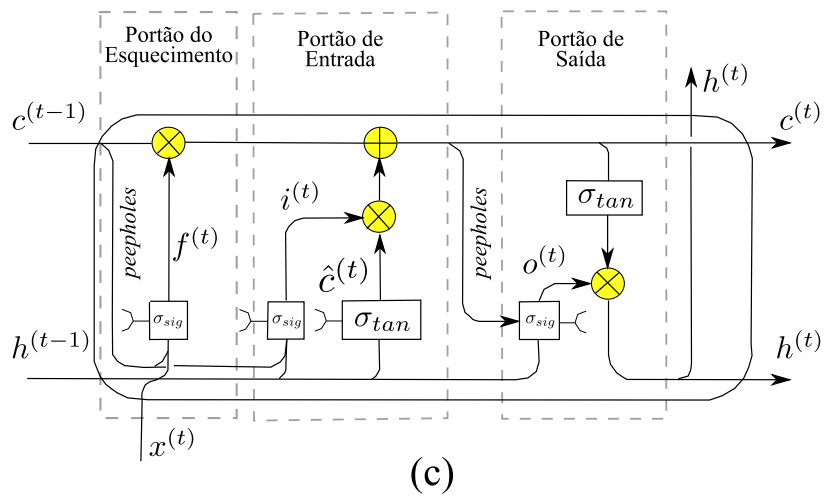
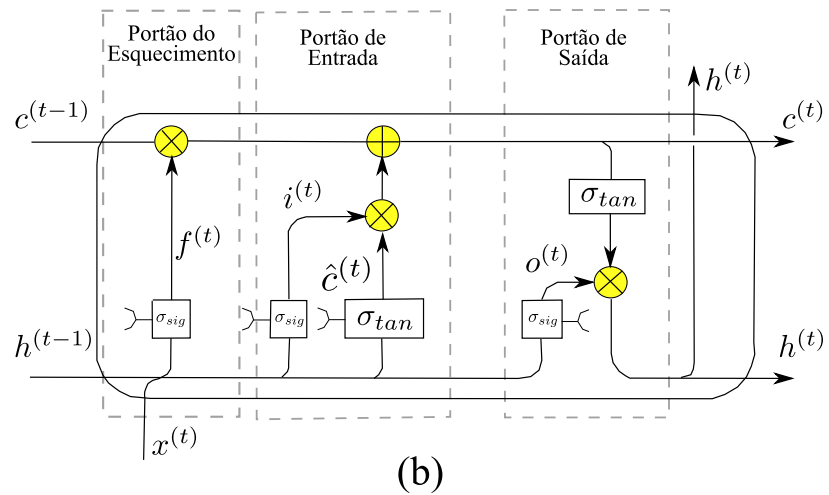
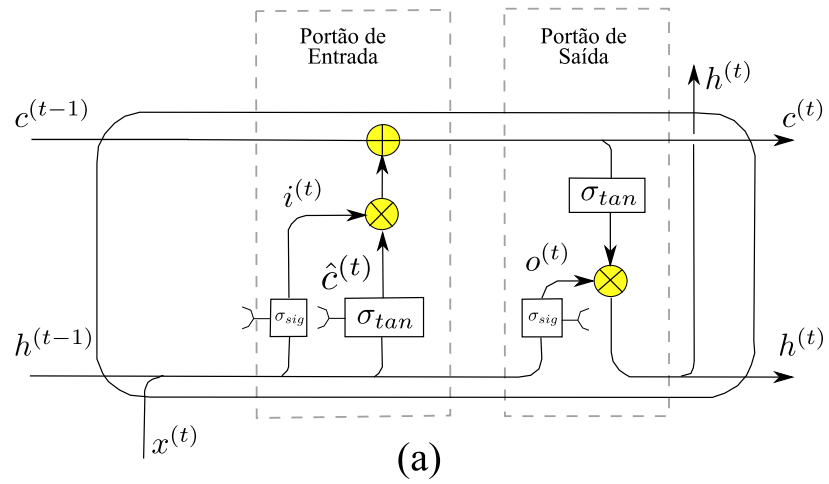
$$\hat{\mathbf{c}}^{(t)} = \sigma_{tan}(W^{\hat{c}h}\mathbf{h}^{(t-1)} + W^{\hat{c}x}\mathbf{x}^{(t)} + \mathbf{b}_{\hat{c}}) \quad (67)$$

$$\mathbf{c}^{(t)} = \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \odot \hat{\mathbf{c}}^{(t)} \quad (68)$$

$$\mathbf{o}^{(t)} = \sigma_{sig}(W^{oh}\mathbf{h}^{(t-1)} + W^{ox}\mathbf{x}^{(t)} + \mathbf{b}_o) \quad (69)$$

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \odot \sigma_{tan}(\mathbf{c}^t) \quad (70)$$

Figura 20 – Célula LSTM (a) sem portão do esquecimento (b) com portão do esquecimento (c) com conexão *peephole*



O parâmetro \mathbf{c}^t é chamado de estado da célula e W^i , $W^{\hat{c}}$ e W^o são as matrizes de pesos. Segundo Yu *et al.* (2019) quando o estado da célula é atualizado, a "porta de entrada" decide quais novas informações podem ser armazenadas no estado da célula, e a "porta de saída", quais serão as saídas. Gers, Schmidhuber e Cummins (1999) implementaram o "portão do esquecimento", mostrado na Figura 20-b, decide quais informações serão esquecidas pela célula. Matematicamente, têm-se que:

$$\mathbf{f}^{(t)} = \sigma_{sig}(W^{fh}\mathbf{h}^{(t-1)} + W^{fx}\mathbf{x}^{(t)} + \mathbf{b}_f) \quad (71)$$

$$\mathbf{i}^{(t)} = \sigma_{sig}(W^{ih}\mathbf{h}^{(t-1)} + W^{ix}\mathbf{x}^{(t)} + \mathbf{b}_i) \quad (72)$$

$$\hat{\mathbf{c}}^{(t)} = \sigma_{tan}(W^{\hat{c}h}\mathbf{h}^{(t-1)} + W^{\hat{c}x}\mathbf{x}^{(t)} + \mathbf{b}_{\hat{c}}) \quad (73)$$

$$\mathbf{c}^{(t)} = \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \odot \hat{\mathbf{c}}^{(t)} \quad (74)$$

$$\mathbf{o}^{(t)} = \sigma_{sig}(W^{oh}\mathbf{h}^{(t-1)} + W^{ox}\mathbf{x}^{(t)} + \mathbf{b}_o) \quad (75)$$

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \sigma_{tan}(\mathbf{c}^t) \quad (76)$$

Uma célula LSTM com conexão *peephole* é mostrada na Figura 20-c. As conexões *peephole* permitem que a célula LSTM inspecione seus estados internos atuais permitindo conexões diretas entre os mesmos. As matrizes P^f , P^i , P^o são chamadas de pesos *peephole*, com f , i e o denotando esquecimento, entrada e saída, respectivamente. O equacionamento desta célula LSTM é dado por:

$$\mathbf{f}^{(t)} = \sigma_{sig}(W^{fh}\mathbf{h}^{(t-1)} + W^{fx}\mathbf{x}^{(t)} + P^f \odot \mathbf{c}^{t-1} + \mathbf{b}_f) \quad (77)$$

$$\mathbf{i}^{(t)} = \sigma_{sig}(W^{ih}\mathbf{h}^{(t-1)} + W^{ix}\mathbf{x}^{(t)} + P^i \odot \mathbf{c}^{t-1} + \mathbf{b}_i) \quad (78)$$

$$\hat{\mathbf{c}}^{(t)} = \sigma_{tan}(W^{\hat{c}h}\mathbf{h}^{(t-1)} + W^{\hat{c}x}\mathbf{x}^{(t)} + \mathbf{b}_{\hat{c}}) \quad (79)$$

$$\mathbf{c}^{(t)} = \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \odot \hat{\mathbf{c}}^{(t)} \quad (80)$$

$$\mathbf{o}^{(t)} = \sigma_{sig}(W^{oh}\mathbf{h}^{(t-1)} + W^{ox}\mathbf{x}^{(t)} + P^o \odot \mathbf{c}^t + \mathbf{b}_o) \quad (81)$$

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \sigma_{tan}(\mathbf{c}^t) \quad (82)$$

4.6 Matriz de Confusão

A matriz de confusão é uma ferramenta usada na avaliação dos algoritmos de classificação. Conhecida também como matriz de erro, é uma tabela que facilita a interpretação do desempenho

dos algoritmos de classificação. Informações presentes na matriz são usadas na formulação matemática das métricas de classificação. A matriz é composta pelo número de verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN).

Em 1904, o estatístico britânico, [Pearson \(1904\)](#) publicou um trabalho intitulado "*On the theory of contingency and its relation to association and normal correlation*" que descreve a matriz de contingência. Durante a Segunda Guerra Mundial, 1939 a 1945, essas tabelas foram utilizadas em pesquisas que detectavam padrões do comportamento humano. Porém, somente em 1998 o termo matriz de confusão se consolidou na teoria do aprendizado de máquina.

A matriz de confusão é obtida após a etapa de treinamento do classificador, onde o algoritmo faz a previsão dos valores, categorizados conforme as classes previamente definidas. A Figura 21 mostra um exemplo de matriz de confusão de um problema de classificação binária. Baseado em algumas grandezas de entrada, tais como corrente, tensão ou temperatura, o algoritmo informa o estado do motor: defeito ou não defeito.

Os valores presentes na matriz de confusão indicam que 50 máquinas defeituosas foram classificadas corretamente (VP). Em 165 máquinas, 100 foram classificadas corretamente como não defeituosas (VN). Os FP e FN correspondem ao número de previsões incorretas. O algoritmo categorizou 10 motores que não continham nenhum defeito, como defeituosos. Raciocínio análogo é aplicado aos FN: 5 máquinas não defeituosas foram categorizadas como defeituosas. As previsões corretas são representadas ao longo da diagonal principal da matriz.

Figura 21 – Exemplo de matriz de confusão

		DEFEITO	NÃO DEFEITO		
		VP 50	FP 10		
Classes Previstas	DEFEITO				
	NÃO DEFEITO				
		Classes Verdadeiras			

Fonte – Pedro H. Camargos, 2021

4.7 Métricas de Classificação

4.7.1 Acurácia

A acurácia, A_{cc} , é a razão entre as previsões corretas e o número total de previsões. Esta métrica informa a taxa de acerto do classificador e é calculada na classificação binária por:

$$A_{cc} = \frac{VP + VN}{VP + VN + FP + FN} \quad (83)$$

Não é aconselhável avaliar unicamente a A_{cc} nos problemas de classificação que contêm classes desbalanceadas. Quando existe disparidade significativa entre o número de classes positivas e negativas esta métrica informa um desempenho equivocado do algoritmo. Dois classificadores podem produzir a mesma acurácia, mas possuem desempenhos diferentes, devido ao desbalanceamento de classes (THARWAT, 2020).

4.7.2 Sensibilidade e Especificidade

A sensibilidade e a especificidade, quantificam a taxa de acerto dos VP e VN respectivamente. A revocação ou sensibilidade, Equação (84), representa a proporção do número de previsões positivas que foram classificadas corretamente. Por outro lado, o cálculo da especificidade, E_{sp} , Equação (85) mede a porcentagem de acerto dos VN.

$$R_{ec} = \frac{VP}{VP + FN} \quad (84)$$

$$E_{sp} = \frac{VN}{VN + FP} \quad (85)$$

4.7.3 Medida-F1

A medida-F1, F_1 , é uma média harmônica desenvolvida em 1948 pelo cientista Dice (1945), sendo utilizada para avaliar algoritmos com rótulos múltiplos ou binários, onde as classes são altamente desequilibradas. Esta métrica ajuda a equilibrar as amostras positivas/negativas.

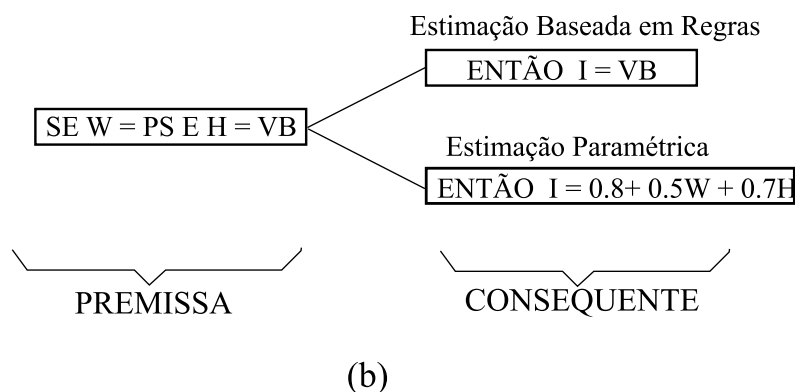
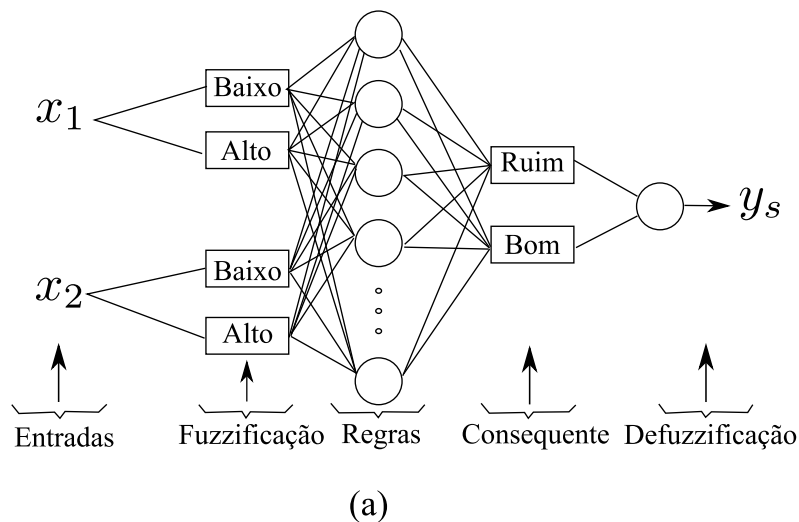
Nos projetos de classificadores que contêm desbalanceamento de classes, é recomendável adotar o F_1 ao invés da A_{cc} .

$$F_1 = \frac{2VP}{2VP + FP + FN} \quad (86)$$

5 Sistema de Inferência Adaptativo Neuro-Fuzzy

O Sistema Neuro-Fuzzy (SNF) combina duas técnicas: redes neurais artificiais e lógica *fuzzy*. As funções de pertinência associadas às variáveis de entrada são obtidas a partir de um conjunto de dados através da execução de um algoritmo de aprendizagem, retropropagação, por exemplo. O SNF gera automaticamente as regras do sistema de inferência *fuzzy*. A Figura 22-a mostra a arquitetura básica de um SNF. De acordo com Simões e Shaw (2007), a estrutura da rede *fuzzy* neural é construída considerando uma topologia na forma paramétrica, como mostrado na Figura 22-b onde as funções de pertinências são gaussianas.

Figura 22 – Rede neural *fuzzy* (a) Arquitetura (b) Princípios de estimação

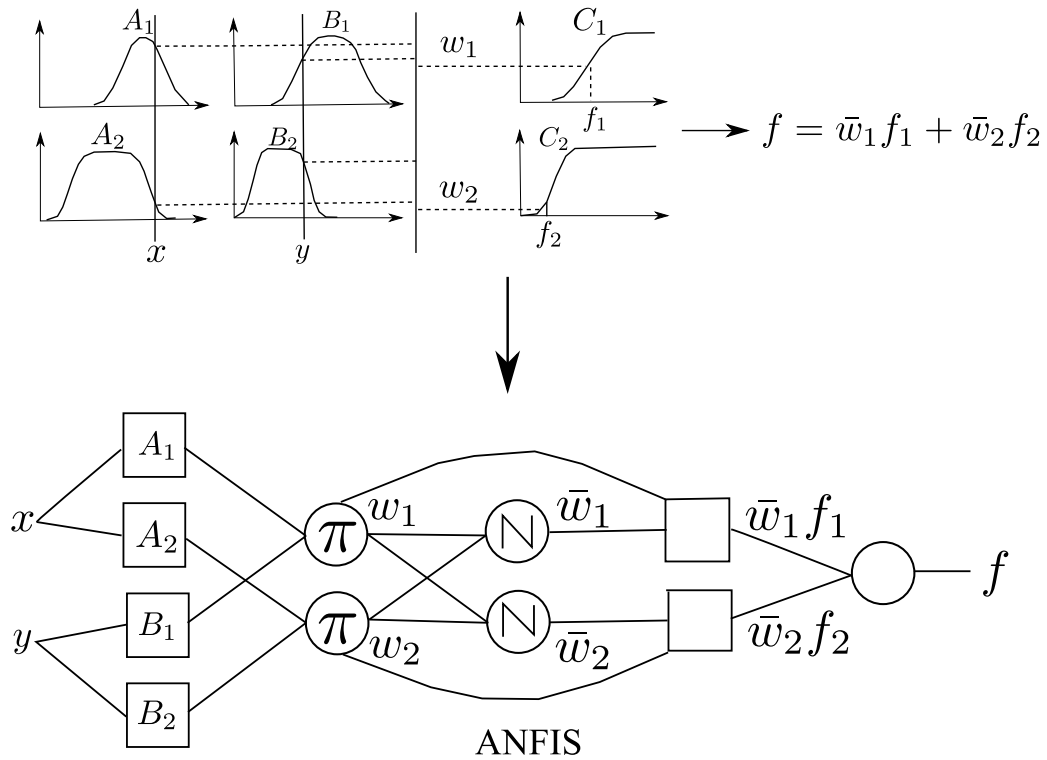


Fonte – Adaptado de Simões e Shaw (2007)

Uma das primeiras arquiteturas neuro-fuzzy desenvolvidas foi o Sistema de Inferência Adaptativo Neuro-Fuzzy, ou *Adaptative Neural Fuzzy Inference System* (ANFIS), proposto por Jang (1993). Esse sistema é ilustrado na Figura 23 e possui duas entradas, x , y , e uma saída z , com uma base de regras dada por:

1. Se x é A_1 e y é B_1 , então $f_1 = p_1x + q_1y + r_1$;
2. Se x é A_2 e y é B_2 , então $f_2 = p_2x + q_2y + r_2$;

Figura 23 – Equivalência ANFIS



Fonte – Adaptado de Jang (1993)

As funcionalidades de cada camadas são descritas a seguir:

- **Camada 1:** Cada nó i possui a função $O_i^1 = \mu_{A_i}(x)$, onde x é a entrada do nó e O_i^1 é a função de pertinência de A_i . Normalmente se adota $\mu_{A_i}(x)$ como sendo,

$$\mu_{A_i}(x) = e^{-\left(\frac{x-c_i}{a_i}\right)^2} \tag{87}$$

onde a_i, b_i, c_i são parâmetros predefinidos. A saída desta camada é igual aos graus de pertinência das entradas.

- **Camada 2:** Nesta camada, os nós são chamados de π . Eles possuem a função de multiplicar os sinais de entrada. Os pesos w_1 e w_2 são calculados por:

$$w_1 = \mu_{A_1}(x) \times \mu_{B_1}(x) \tag{88}$$

$$w_2 = \mu_{A_2}(x) \times \mu_{B_2}(x) \tag{89}$$

Cada saída de um nó representa o grau de pertinência ao qual é submetido o consequente de cada regra. Cada neurônio artificial executa a operação de T_{norm} e corresponde a uma regra.

- **Camada 3:** Esta camada realiza a normalização dos graus de ativação das regras utilizadas como um pré-processamento para a defuzzificação. Matematicamente, têm-se que:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2. \quad (90)$$

- **Camada 4:** As saídas da Camada 3 são multiplicadas pelas funções f_1 e f_2 , de tal forma que:

$$O_i^4 = \bar{w}_i f_i \quad i = 1, 2. \quad (91)$$

- **Camada 5:** A última camada possui apenas um nó, cuja saída é $O_1^5 = \sum \bar{w}_i f_i, i = 1, 2$. Esta etapa é denominada de defuzzificação.

6 Normas Internacionais

6.1 IEC-60034 Máquinas Elétricas Rotativas

A *International Electrotechnical Commission* (IEC) é uma organização internacional que promove a padronização das tecnologias elétricas e eletrônicas. A IEC publica normas internacionais que recomendam especificações técnicas para o projeto e operação das máquinas elétricas, fornecendo instruções, diretrizes e regras utilizadas para fabricar, instalar, testar, certificar e reparar dispositivos elétricos e eletrônicos.

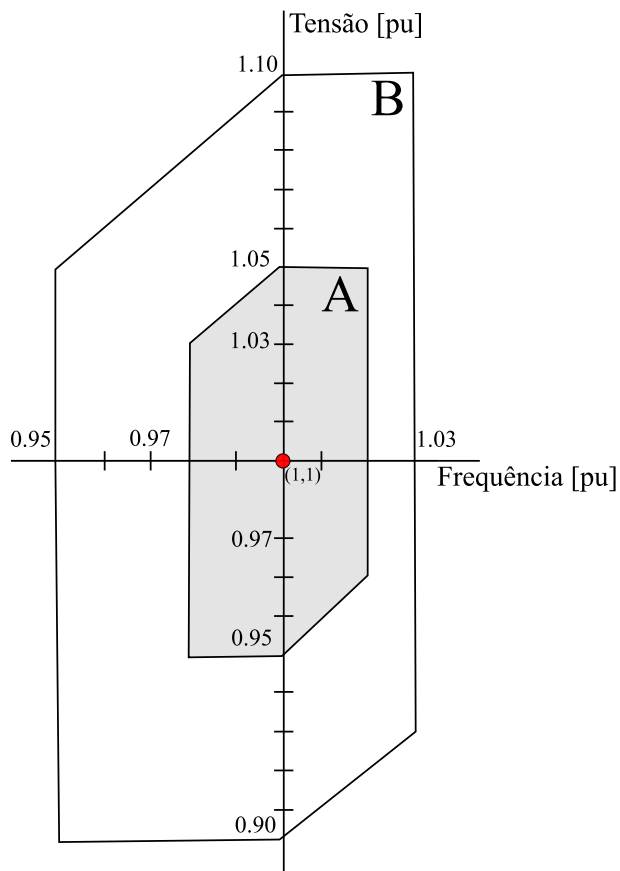
Os padrões IEC englobam uma vasta gama de tecnologias ligadas à engenharia elétrica e aos equipamentos instalados industriais. As normas IEC 60034-1: *Rating and performance* e IEC 60034-26: *Effects of unbalanced voltages on the performance of three-phase cage induction motors* definem critérios que afetam a segurança da operação das máquinas assíncronas. Desta forma, projetar e operar motores segundo estes critérios garante confiabilidade, interoperabilidade e eficiência no processo produtivo.

6.1.1 IEC 60034-1 - Características e Desempenho

A IEC 60034-1 especifica em quais condições os motores de indução com rotor gaiola de esquilo podem operar. De acordo com IEC (2017), as máquinas assíncronas devem ser capazes de entregar o torque nominal à carga, mesmo quando submetidos a pequenas variações de tensões e frequência. Os desvios de tensão e frequência são delimitados por duas regiões, como indicado na Figura 24. Estas variações nos valores nominais da fonte de alimentação podem comprometer o desempenho do motor.

Motores submetidos a valores de tensão e frequência na região A operam sem grandes desvios de temperatura, o que significa que a temperatura de operação se mantém praticamente a mesma da condição de operação nominal. Por outro lado, se as coordenadas estão no interior da região B, a máquina operará com maiores perdas por efeito joule se comparado à condição nominal. Nesta região, o motor está sujeito a um excessivo aumento em sua temperatura de operação podendo acarretar danos aos seus isolamentos.

Figura 24 – Limites de tensão e frequência para motores



Fonte – Adaptado de IEC (2017)

6.1.2 IEC 60034-26 Efeitos das Tensões Desequilibradas no Desempenho de Motores de Indução

O desequilíbrio de tensão em um sistema elétrico trifásico ocorre quando as fases apresentam diferentes valores de tensão em módulo ou ângulo. Tensões desequilibradas trazem consequências danosas a uma máquina assíncrona, comprometendo o seu desempenho e diminuindo sua vida útil. A norma IEC 60034-26, IEC (2006), descreve os efeitos das tensões desequilibradas no desempenho dos motores de indução trifásicos com rotor em gaiola.

A normativa IEC 60034-26 estabelece valores para o Fator de Redução, F_r , das máquinas assíncronas submetidas a condições de desequilíbrio de tensão. Os desequilíbrios de tensão produzem componentes de sequência negativa de tensão e corrente, responsáveis pelo aumento nas perdas por efeito joule. Correntes elevadas surgem no estator e rotor da máquina. O aumento da corrente de alimentação contribui para o sobreaquecimento e deterioração do dielétrico dos isolamentos. Em condições desequilibradas o motor está sujeito aos efeitos da componente de sequência negativa do campo magnético girante. Consequentemente, esforços mecânicos axiais

e radiais atuam sobre o eixo, desgastando os mancais. Também há o surgimento de vibrações, batimentos e ruídos. Em IEC (2006), menciona-se que o desequilíbrio de tensão afeta o torque desenvolvido pelo motor o que faz com que o torque eletromagnético resultante seja menor que aquele produzido em condições ideais.

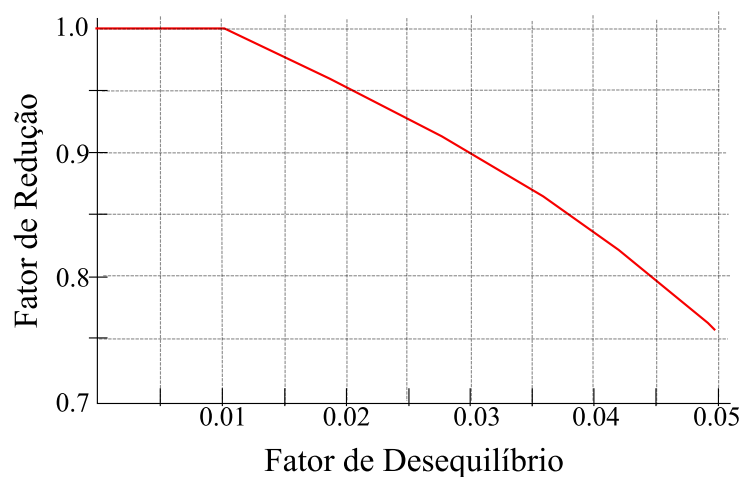
O Fator de Desequilíbrio quantifica o desequilíbrio de tensão. De acordo com IEC (2006), este parâmetro pode ser calculado por:

$$f_D = \left(\frac{V_{max} - V_{avg}}{V_{avg}} \right) \quad (92)$$

A Equação (92) é uma formulação proposta por IEC (2006) para ser usada em aplicações práticas na indústria. As variáveis V_{max} e V_{avg} correspondem a máxima e média tensão entre as três fases, respectivamente. A norma IEC 60034-26 não recomenda a operação dos motores de indução com $f_D > 0.05$.

Para compreender as diretrizes descritas na normativa é necessário definir o conceito do Fator de Redução, conhecido como *Derating Factor*. Correntes desequilibradas produzem um aumento de temperatura. Conseqüentemente haverá uma diminuição da potência entregue a carga, cujo valor é multiplicado por F_r . A normativa mostra graficamente, ver Figura 25, a relação entre o F_r e f_D para motores da categoria N.

Figura 25 – Valores típicos de fatores de redução para motores da categoria N



Fonte – Adaptado de IEC (2006)

Um Fator de Redução, $F_r = 1$, significa que o motor opera com desempenho satisfatório com 100% de sua potência nominal (LANE *et al.*, 2017; NEMA, 2003). Para compreender o conceito do F_r , imagine um motor com potência nominal, P_n , igual a 15 kW operando com

$f_D = 4\%$ com $F_r \approx 0.83$, obtido graficamente. A máxima potência de saída entregue a carga nesta condição, será igual 12.45 kW , sendo calculada por:

$$P'_n = F_r P_n \quad (93)$$

$$P'_n = (0.83)(15) \quad (94)$$

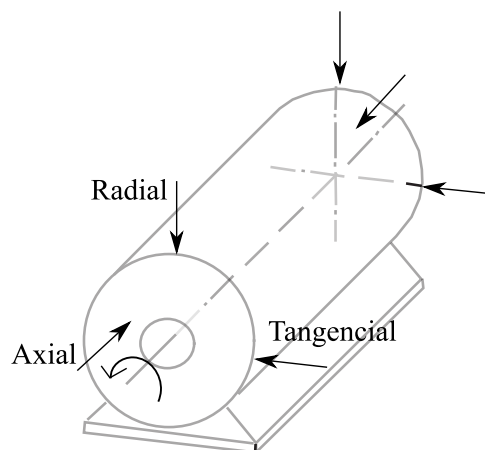
$$P'_n = 12.45 \text{ kW} \quad (95)$$

6.2 ISO 20816-1 Vibração Mecânica - Medição e Avaliação da Vibração da Máquina

Para monitorar as condições operativas dos equipamentos industriais, sistemas de medição de vibração devem ser instalados para medir continuamente os níveis de vibração dos motores elétricos instalados na planta industrial (VIBSENS, 2020). A “ISO 20816-1 *Mechanical vibration — Measurement and evaluation of machine vibration — Part 1: General guidelines*” é uma normativa internacional que fornece requisitos e diretrizes para a medição e avaliação da vibração em peças rotativas e não rotativas.

A norma ISO (2016) recomenda realizar as medições nas partes não rotativas na direção axial, radial ou tangencial. Os pontos de medição nas máquinas elétricas de pequeno porte são mostrados na Figura 26. Podem ser utilizados sensores de deslocamento, velocidade ou aceleração. A norma menciona que se deve medir os valores eficazes (*rms*) da velocidade, aceleração ou deslocamento.

Figura 26 – Pontos de medição de vibração para pequenas máquinas elétricas



Fonte – Adaptado de ISO (2016)

A [ISO \(2016\)](#) descreve o critério de avaliação da magnitude do sinal de vibração em peças não rotativas através da categorização das zonas A, B, C e D, classificando o estado de operação do motor de acordo com o nível de vibração. As zonas permitem uma avaliação qualitativa da saúde do motor, cujas definições são:

- **Zona A:** nível de vibração de máquinas recém comissionadas.
- **Zona B:** nível de vibração aceitável para uso irrestrito na operação de longo prazo.
- **Zona C:** nível de vibração considerado insatisfatório para operação contínua a longo prazo. A máquina pode ser operada por um período limitado nesta condição até que surja uma oportunidade adequada para uma ação corretiva.
- **Zona D:** valores de vibração causam danos à máquina.

As zonas são delimitadas pelo valor eficaz da velocidade de vibração, conforme ilustrado no diagrama da [Figura 27](#). Em geral, os limites da zona para máquinas de pequeno porte (até 15 kW) tendem a assumir os limites inferiores do diagrama ([JSME, 2021](#)).

Figura 27 – Faixas de valores típicos dos limites das zonas A/B, B/C e C/D

		Valor r.m.s Velocidade de Vibração [mm/s]			
0.28					0.28
0.45					0.45
0.71					0.71
1.12	Limite da Zona A\B 0.71 a 4.5				1.12
1.80					1.80
2.80		Limite da Zona B\C 1.8 a 9.3			2.80
4.50					4.50
7.10			Limite da Zona C\D 4.5 a 14.7		7.10
9.30					9.30
11.2					11.2
14.7					14.7
18.0					18.0
28.0					28.0
45.0					45.0

Fonte – Adaptado de [ISO \(2016\)](#)

O nível de banda larga mede o sinal de vibração não filtrado em uma ampla distribuição de frequências ([RESEARCH, 2021](#); [TROUT, 2021](#); [CORPORATION, 2021](#)). Para muitas máquinas, a vibração de banda larga consiste em um único componente de frequência. Neste caso, a norma recomenda adotar este valor de velocidade como critério de verificação da zona de operação.

6.3 ISO 13373-9 Monitoramento da Condição e Diagnóstico de Máquinas - Monitoramento de Condição de Vibração - Parte 9: Técnicas de Diagnóstico para Motores Elétricos

[Mohanty \(2014\)](#) menciona três componentes que devem ser considerados quando se analisa as falhas em um motor elétrico: rotor, estator e rolamentos. Caso o rolamento seja defeituoso, o entreferro torna-se excêntrico, acarretando o aparecimento de forças eletromagnéticas radiais pulsantes que levam a um aumento na vibração mecânica no motor. No estator, os enrolamentos estão sujeitos a degradação de seu isolamento quando expostos a situações de elevação extrema de temperatura. Esse sobreaquecimento poderá ser decorrente de quebras nas barras do rotor, em consequência do aumento das correntes que percorrem as barras adjacentes à barra danificada. De acordo com [Mehrjou *et al.* \(2015\)](#), [Alves \(2017\)](#), o aparecimento de rachaduras e quebras nas barras do rotor gaiola de esquilo é consequência dos esforços:

- **Térmicos:** caracterizados pelos desequilíbrios térmicos que geram pontos quentes no interior da máquina.
- **Magnéticos:** vibração excessiva causada por forças eletromagnéticas e tração magnética desequilibrada.
- **Dinâmicos:** forças centrífugas resultantes do torque de eixo.
- **Mecânicos:** consequência dos desgastes nas peças que compõem a gaiola, lâminas soltas e falha de rolamento.

A norma “ISO 13373-9 - *Condition monitoring and diagnostics of machines — Vibration condition monitoring — Part 9: Diagnostic techniques for electric motors*”, [ISO \(2017\)](#), estabelece procedimentos a serem considerados ao se realizar diagnósticos de falhas através da avaliação dos sinais de vibração dos motores de indução com rotor em gaiola. A normativa contém metodologias de diagnóstico baseadas na inspeção visual, magnitude da vibração, análise nos domínios do tempo e frequência.

Avaliar o espectro de frequência dos sinais de vibração e corrente de alimentação auxilia engenheiros na detecção de falhas nas barras do rotor. A Análise da Assinatura da Vibração (MVSA) e a Análise da Assinatura da Corrente (MCSA) são técnicas utilizadas na detecção de falhas nas barras do rotor.

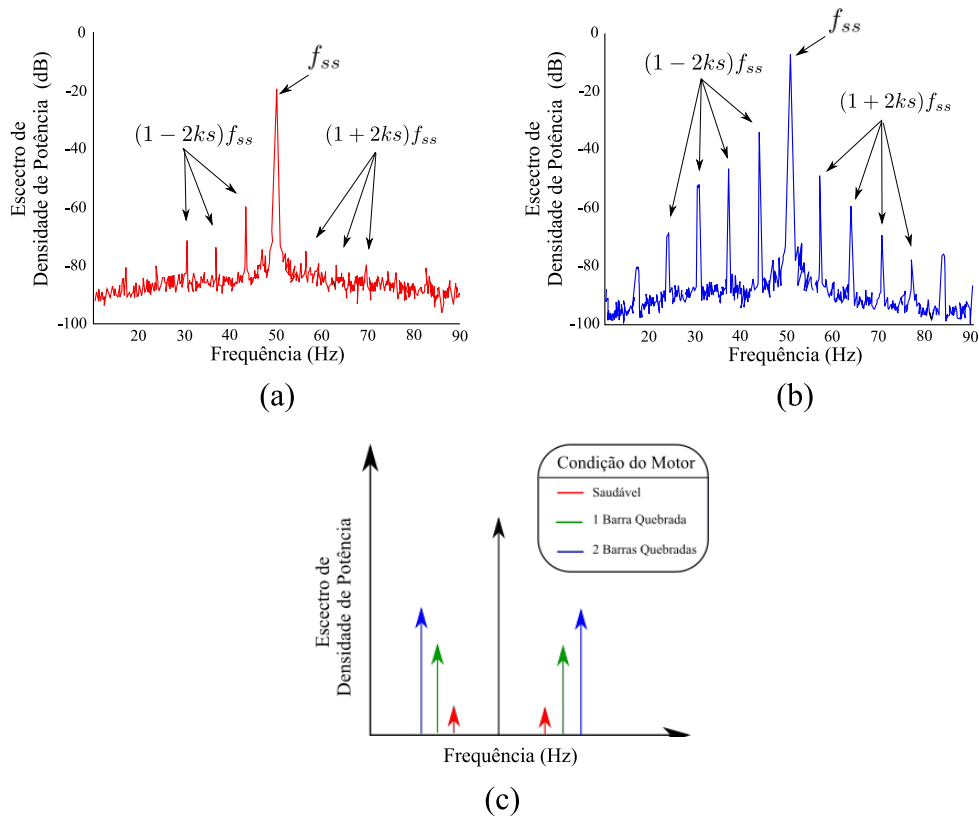
De acordo com a ISO 13373-9, as medições para detecção de barras quebradas necessitam ser realizadas alguns segundos após a partida da máquina [ISO \(2017\)](#). Na presença de barras quebradas há o surgimento de modulações na amplitude do sinal de vibração ou corrente (espectro

de frequência), seguidas do aparecimento dos componentes espectrais de bandas laterais ao redor do componente da frequência fundamental. Técnicas de monitoramento que detectam estas assimetrias na frequência de vibração são chamadas de MVSA. A técnica de detecção de falhas que avalia a decomposição espectral da corrente do estator é denominada MCSA (ALVES, 2017). Uma barra quebrada induz componentes no espectro de vibração ou corrente no estator com frequência dado por

$$f_b = (1 \pm 2ks)f_{ss}, \quad k = 1, 2, 3 \dots \quad (96)$$

onde k é um inteiro, s é o escorregamento e f_{ss} é a frequência da fonte (RANGEL-MAGDALENO *et al.*, 2009; MICELI *et al.*, 2014; CRUZ-VEGA *et al.*, 2017). Espectros de densidade de potência dos sinais de corrente são mostrados na Figura 28. A diferença nas amplitudes da banda fundamental, f_{ss} e inferior, $(1 - 2ks)f_{ss}$, é utilizada para distinguir motores saudáveis daqueles com barras quebradas. O aumento gradativo das amplitudes nas modulações laterais indica a gravidade da falha.

Figura 28 – Espectro das correntes de linha (a) motor saudável (b) motor com uma barra quebrada (c) variação da magnitude em função do número de barras quebradas



Fonte – Adaptado de Mehrjou *et al.* (2011)

7 Ferramentas Matemáticas e Computacionais

Ferramentas computacionais são indispensáveis quando se necessita desenvolver e testar técnicas vinculadas a inteligência artificial. O MATLAB/Simulink é um software que possui um ambiente de simulação amigável, com uma interface que possibilita ao usuário programar e realizar simulações de sistemas dinâmicos. Desenvolvido pela empresa norte-americana *MathWorks*, o software possui uma ferramenta gráfica que contém um conjunto personalizável de bibliotecas de blocos usados na modelagem de diversos equipamentos que compõem sistemas elétricos (MATHWORKS, 2021e). O bloco *Asynchronous Machine*, que modela uma máquina assíncrona, por exemplo, necessita dos parâmetros do circuito equivalente da máquina.

Este Capítulo 7, descreve detalhadamente o modelo da máquina assíncrona, usado nas simulações no MATLAB/Simulink que auxiliaram na obtenção do conjunto treinamento/validação do Sistema de Diagnóstico Região de Operação IEC 60034-1. A Seção 7.2 apresenta ao leitor a metodologia que estima os parâmetros do circuito equivalente de um motor de indução trifásico com rotor em gaiola. Este método foi utilizado para obter os dados de entrada necessários para modelar as máquinas na etapa de desenvolvimento do Sistema de Diagnóstico Região de Operação IEC 60034-1.

7.1 Modelo da Máquina Assíncrona

O software MATLAB/Simulink usa um método de espaço de estado para gerar as saídas das simulações de inicialização da máquina. Inicialmente a ferramenta computacional avalia a topologia da rede, separando blocos lineares e não lineares. Em seguida, calcula a matriz de espaço de estado usando o bloco *S-Function*. Modelos internos predefinidos simulam os elementos não lineares (MATHWORKS, 2020e).

O bloco representativo da máquina assíncrona trifásica no MATLAB/Simulink modela um motor com rotor gaiola de esquilo. Dois subsistemas representam a máquina: elétrico e mecânico. O momento de inércia do motor, J_m , e o coeficiente de atrito viscoso, F , são as entradas do subsistema mecânico. A velocidade de rotação do eixo do motor é a solução da Equação (97).

$$J_m \frac{d\omega_m}{dt} = T_e - F\omega_m - T_m \quad (97)$$

O equacionamento do bloco não considera os efeitos de saturação dos fluxos de dispersão (MATHWORKS, 2021a). Os circuitos elétricos do modelo dinâmico da máquina assíncrona são mostrados na Figura 29 e Figura 30. Os subscritos q e d correspondem ao eixo de quadratura e direto. As reatâncias de dispersão do rotor, L'_{lr} , estator, L_{ls} , e de magnetização, L_m , são utilizadas no cálculo dos fluxos magnéticos φ_s e φ'_r . As variáveis R_s e R'_r representam as resistências do estator e do rotor. O termo ω_r é a velocidade angular referencial e ω_e é a velocidade angular elétrica.

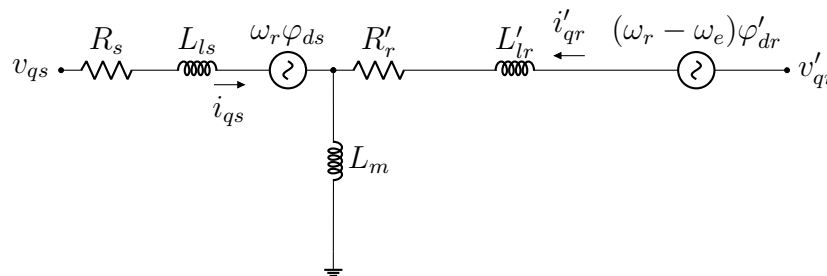
$$v_{qs} = R_s i_{qs} + \frac{d\varphi_{qs}}{dt} + \omega_r \varphi_{ds} \tag{98}$$

$$v_{ds} = R_s i_{ds} + \frac{d\varphi_{ds}}{dt} - \omega_r \varphi_{qs} \tag{99}$$

$$v'_{qr} = R'_r i'_{qr} + \frac{d\varphi'_{qr}}{dt} + (\omega_r - \omega_e) \varphi'_{dr} \tag{100}$$

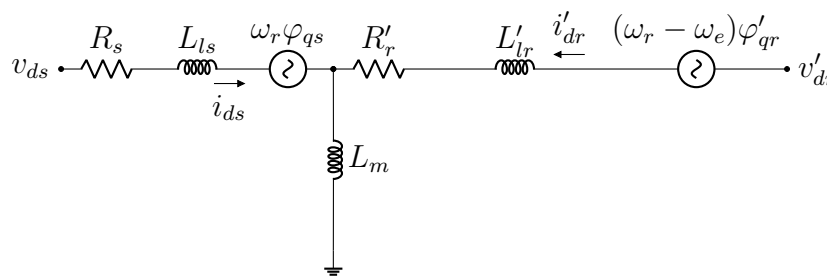
$$v'_{dr} = R'_r i'_{dr} + \frac{d\varphi'_{dr}}{dt} + (\omega_r - \omega_e) \varphi'_{qr} \tag{101}$$

Figura 29 – Circuito elétrico do modelo da máquina assíncrona: eixo direto



Fonte – MathWorks (2021a)

Figura 30 – Circuito elétrico do modelo da máquina assíncrona: eixo de quadratura



Fonte – MathWorks (2021a)

O modelo da máquina de indução possui três sistemas de referência: rotor, estacionário ou síncrono. É recomendável utilizar o sistema referencial do rotor nas simulações em que as

tensões do rotor são desequilibradas. De acordo com [MathWorks \(2021a\)](#), é necessário configurar o modelo para o referencial estacionário quando existe desequilíbrio nas tensões do estator. Se as tensões na parte estacionária e girante são balanceadas deve-se optar pelo referencial síncrono.

As relações a seguir descrevem as transformações das referências abc para dq aplicadas às tensões fase-fase do modelo do motor e correntes de linha:

$$\begin{bmatrix} v_{qs} \\ v_{ds} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 \cos \theta & \cos \theta + \sqrt{3} \sin \theta \\ 2 \sin \theta & \sin \theta + \sqrt{3} \cos \theta \end{bmatrix} \begin{bmatrix} v_{abs} \\ v_{bcs} \end{bmatrix} \quad (102)$$

$$\begin{bmatrix} v'_{qr} \\ v'_{dr} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 \cos \beta_{ang} & \cos \beta_{ang} + \sqrt{3} \sin \beta_{ang} \\ 2 \sin \beta_{ang} & \sin \beta_{ang} + \sqrt{3} \cos \beta_{ang} \end{bmatrix} \begin{bmatrix} v'_{abr} \\ v'_{bcr} \end{bmatrix} \quad (103)$$

onde θ é a posição angular do referencial. O termo β_{ang} é a diferença entre a posição de referência e a posição angular elétrica do rotor.

$$\begin{bmatrix} i_{as} \\ i_{bs} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ \frac{-\cos \theta + \sqrt{3} \sin \theta}{2} & \frac{-\sqrt{3} \cos \theta - \sin \theta}{2} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix} \quad (104)$$

$$\begin{bmatrix} i'_{ar} \\ i'_{br} \end{bmatrix} = \begin{bmatrix} \cos \beta_{ang} & \sin \beta_{ang} \\ \frac{-\cos \beta_{ang} + \sqrt{3} \sin \beta_{ang}}{2} & \frac{-\sqrt{3} \cos \beta_{ang} - \sin \beta_{ang}}{2} \end{bmatrix} \begin{bmatrix} i'_{qr} \\ i'_{dr} \end{bmatrix} \quad (105)$$

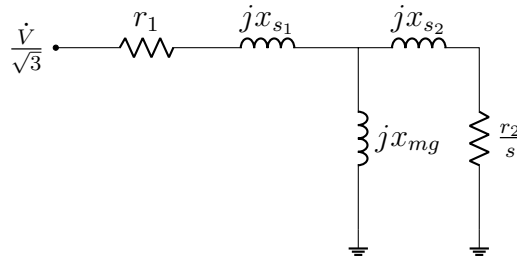
$$i_{cs} = -i_{as} - i_{bs} \quad (106)$$

$$i'_{cr} = -i'_{ar} - i'_{br} \quad (107)$$

7.2 Estimativa dos Parâmetros da Máquina de Indução a partir dos Dados do Fabricante

Os parâmetros que compõem o circuito equivalente da máquina assíncrona são usados para determinar as variáveis de entrada nos modelos dinâmicos da máquina assíncrona trifásica no MATLAB/Simulink. O circuito equivalente simplificado, desprezando as perdas joule no núcleo, é mostrado na Figura 31. As variáveis r_1 e x_{s1} representam a resistência dos enrolamentos e a reatância de dispersão do estator. O ramo de magnetização é composto apenas pela reatância de magnetização, x_{mg} . De maneira semelhante ao que ocorre no transformador, os parâmetros que modelam o rotor são referidos ao estator ([FITZGERALD et al., 2003](#)). As incógnitas do rotor, r_2 , x_{s2} , são a resistência e a reatância de dispersão, respectivamente. O escorregamento é representado pela letra s .

Figura 31 – Circuito equivalente simplificado da máquina assíncrona



Fonte – Pedro H. Camargos, 2021

Existem procedimentos de monitoramento das condições operativas da máquina que dependem dos parâmetros do circuito equivalente. Aumento da confiabilidade e redução de custos são algumas vantagens da implementação destes procedimentos (LU; HABELTLER; HARLEY, 2006; ARKAN; KOSTIC-PEROVIC; UNSWORTH, 2005; SINGH *et al.*, 2016). O método adotado para estimar os parâmetros do circuito equivalente, nos sistemas de diagnóstico de falhas descrito nas próximas seções, foi proposto por Amaral *et al.* (2020). Nesta metodologia não é necessário conhecer detalhadamente a geometria do motor. Os valores de r_1 , r_2 , x_{s1} , x_{s2} e x_{mg} são calculados a partir das informações presentes no catálogo do fabricante. Neste método as seguintes premissas são adotadas:

- O intervalo do escorregamento s deve ser $s \in [s_{max}, s_{nl}]$, onde s_{max} é o escorregamento quando o motor entrega à carga o torque eletromecânico máximo e s_{nl} é a condição de operação sem carga.
- A saturação magnética e o efeito *skin* são negligenciados.
- A soma das perdas por atrito e ventilação e perdas magnéticas é constante.

Amaral *et al.* (2020) aplicaram este método em 789 motores de 3 fabricantes diferentes e realizaram estudos comparando com outras metodologias propostos por Natarajan e Misra (1989) e Bortoni *et al.* (2019). Para a maioria dos motores analisados, as diferenças entre os parâmetros reais fornecidos pelo fabricante e os encontrados pela metodologia de Amaral *et al.* (2020) foram mínimas. O método proposto também apresentou erros menores se comparado com as metodologias de Natarajan e Misra (1989) e de Bortoni *et al.* (2019).

A potência de saída, P_{out} , potência de entrada P_{in} , perdas por efeito joule no estator, P_{j1} , e rotor, P_{j2} , e as perdas rotacionais, P_{rot} , pertencem ao equacionamento do balanço de potência ativa, ou seja:

$$P_{out} = P_{in} - P_{j1} - P_{j2} - P_{rot} \quad (108)$$

Após algumas manipulações algébricas descritas por [Amaral et al. \(2020\)](#) na Equação (108) obtêm-se:

$$P_{out} - (1 - s)P_{in} = -(1 - s)3r_1I_1^2 - P_{rot} \quad (109)$$

A corrente I_1 é o módulo da corrente injetada nos enrolamentos do estator. A folha de dados dos motores fornecidos pelos fabricantes contém dados referentes às condições operativas de 50%, 75% e 100% da carga nominal. O valor em módulo de I_1 para cada condição de operação foi calculado por,

$$I_{1i} = \frac{\sqrt{3}P_{outi}}{3V_n f p_i \eta_i} \quad \text{para } i = 50, 75 \text{ e } 100. \quad (110)$$

onde V_n é a tensão nominal de linha, $f p_i$ é o fator de potência e η_i é a eficiência. A variação do escorregamento foi considerada linear, ou seja, $s_{50} = 0.5s_{100}$, $s_{75} = 0.75s_{100}$ onde s_{100} é o escorregamento nominal. Para obter o parâmetro r_1 foi usada a seguinte expressão matemática:

$$\begin{bmatrix} P_{out_{50}} - (1 - s_{50})P_{in_{50}} \\ P_{out_{75}} - (1 - s_{75})P_{in_{75}} \\ P_{out_{100}} - (1 - s_{100})P_{in_{100}} \end{bmatrix} = \begin{bmatrix} -(1 - s_{50})3I_{150}^2 & -1 \\ -(1 - s_{75})3I_{175}^2 & -1 \\ -(1 - s_{100})3I_{1100}^2 & -1 \end{bmatrix} \begin{bmatrix} r_1 \\ P_{rot} \end{bmatrix} \quad (111)$$

A Equação (111) pode ser escrita na notação matricial. As variáveis a determinar, r_1 e P_{rot} , são as soluções da Equação (113).

$$\Psi_1 = \Lambda_1 \Theta_1 \quad (112)$$

$$\Theta_1 = (\Lambda_1^T \Lambda_1)^{-1} \Psi_1 \quad (113)$$

O equacionamento utilizado para estimar o valor da resistência r_2 foi obtido através da análise do balanço de potência ativa, tal que:

$$P_{out} + P_{rot} = (1 - s)P_{ag} \quad (114)$$

A potência eletromagnética, P_{ag} , é dada por:

$$P_{ag} = \frac{P_{j2}}{s} \quad (115)$$

Substituindo a Equação (115) em (114) têm-se que:

$$P_{j2} = \frac{s(P_{out} + P_{rot})}{(1-s)} \quad (116)$$

$$3r_2 I_2^2 = \frac{s(P_{out} + P_{rot})}{(1-s)} \quad (117)$$

$$3r_2(I_1^2 - I_0^2) = \frac{s(P_{out} + P_{rot})}{(1-s)} \quad (118)$$

$$3r_2 I_1^2 + K = \frac{s(P_{out} + P_{rot})}{(1-s)} \quad (119)$$

O modulo da corrente I_2 que percorre x_2 é expressa em função da corrente de magnetização, I_0 e I_1 . A variável $K = -3r_2 I_0^2$ é uma constante. Reescrevendo a Equação (119) para cada condição de carga tem-se que:

$$\begin{bmatrix} \frac{s_{50}(P_{out_{50}} + P_{rot})}{(1-s_{50})} \\ \frac{s_{75}(P_{out_{75}} + P_{rot})}{(1-s_{75})} \\ \frac{s_{100}(P_{out_{100}} + P_{rot})}{(1-s_{100})} \end{bmatrix} = \begin{bmatrix} 3I_{150}^2 & 1 \\ 3I_{175}^2 & 1 \\ 3I_{1100}^2 & 1 \end{bmatrix} \begin{bmatrix} r_2 \\ K \end{bmatrix} \quad (120)$$

Através de um procedimento análogo ao visto na estimativa de r_1 , o parâmetro r_2 pode ser obtido solucionando a seguinte expressão matemática:

$$\Theta_2 = (\Lambda_2^T \Lambda_2)^{-1} \Psi_2 \quad (121)$$

A relação, a , entre x_{s1} e x_{s2} , descrita nas normativas nacionais e internacionais, é mostrada na Tabela 1. Cabe ressaltar a importância de se conhecer a classe do motor, A , B , C e D categorias utilizadas no continente americano ou D , N e H na Europa.

Tabela 1 – Relação entre x_{s1} e x_{s2}

IEEE / NEMA	Categoria	IEC/NBR	Categoria
1	A e D	0.78	D
0.67	B	0.68	N
0.43	C	0.58	H

Fonte – Amaral *et al.* (2020)

Todos os cálculos utilizados na estimativa de x_{s_1} , x_{s_2} e x_{mg} foram realizados com as grandezas nominais. Os valores das correntes I_0 e I_2 são calculadas por:

$$I_0 = \sqrt{I_1^2 - \frac{s(P_{out} + P_{rot})}{3r_2(1-s)}} \quad (122)$$

$$I_2 = \sqrt{I_1^2 - I_0^2} \quad (123)$$

Assumindo que $r_{2s} = r_2/s$ e $x_{s_1} = ax_{s_2}$ e aplicando as leis de *Kirchhoff* no circuito equivalente do motor de indução obtêm-se o seguinte equacionamento:

$$x_{mg} = \frac{I_2}{I_0} \sqrt{r_{2s}^2 + x_{s_2}^2} \quad (124)$$

A potência reativa injetada na máquina é calculada pela Equação (125). A menor raiz da Equação (126) é o valor estimado de x_{s_2} . [Amaral et al. \(2020\)](#) descrevem detalhadamente o procedimento de obtenção desta formulação matemática.

$$Q_{in} = \frac{P_{out} \text{sen}(\cos^{-1}(fp))}{\eta fp} \quad (125)$$

$$A_q x_{s_2}^2 + B_q x_{s_2} + C_q = 0 \quad (126)$$

$$A_q = \frac{(K_1 + K_2)^2}{K_3^2} - K_4 \quad (127)$$

$$B_q = \frac{-2Q_{in}(K_1 + K_2)}{K_3^2} \quad (128)$$

$$C_q = \frac{Q_{in}^2}{K_3^2} - K_4 r_{2s} \quad (129)$$

As equações (130) a (133) são usadas no cálculo das constantes K_1 , K_2 , K_3 e K_4 . Com o valor de x_{s_2} previamente estimado, é possível calcular x_{s_1} usando os dados fornecidos pela Tabela 1. Substituindo o valor de x_{s_2} na Equação (124) obtêm-se x_{mg} .

$$K_1 = 3aI_1^2 \quad (130)$$

$$K_2 = 3(I_1^2 - I_0^2) \quad (131)$$

$$K_3 = 3I_0^2 \quad (132)$$

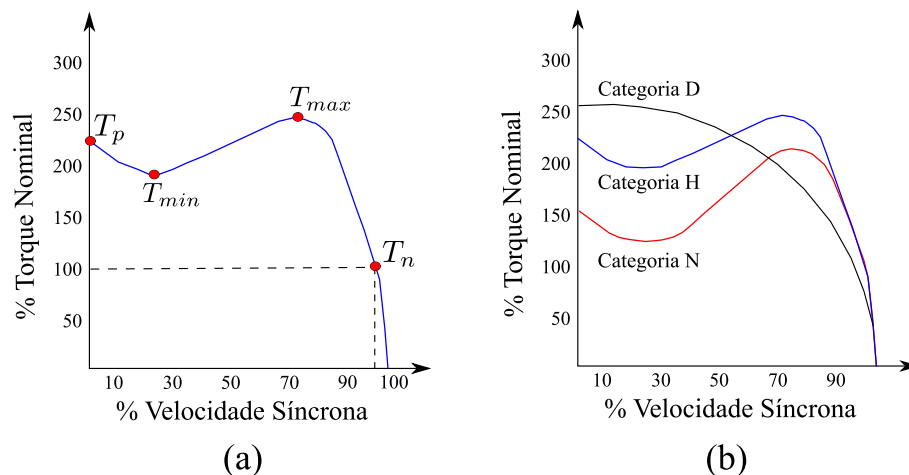
$$K_4 = \left(\frac{I_2}{I_0}\right)^2 \quad (133)$$

7.3 Categorias dos Motores de Indução Trifásicos

As máquinas assíncronas são oferecidas pelos fabricantes com uma ampla faixa de especificações de tensão, potência e velocidades padronizadas. As indústrias possuem a opção de escolher motores com curvas características de torque *versus* velocidade. A Figura 32-a mostra uma curva típica de torque de um motor de indução trifásico. No eixo das ordenadas o torque é expresso em porcentagem do torque nominal, T_n . Um motor que possui elevado valor de conjugado máximo, T_{max} , opera sem perda de velocidade quando submetido a quedas momentâneas na tensão de alimentação. O torque mínimo, T_{min} , é o menor conjugado desenvolvido pelo motor. Máquinas com baixos valores de T_{min} sofrem superaquecimento na partida (WEG, 2015).

As normas IEC 60034-12:2016 e NBR 17094-1:2008 classificam os motores de indução trifásicos com rotor de gaiola conforme o torque de partida, T_p , (ABNT, 2008; IEC, 2016). O T_p é desenvolvido com o rotor bloqueado sob tensão e frequência nominal. Motores projetados para operarem com partida direta pertencem a uma destas categorias: N, H ou D. As curvas de conjugado *versus* velocidade dos motores de categoria N, H e D podem ser vistas na Figura 32-b.

Figura 32 – Curvas torque *versus* velocidade (a) curva típica (b) categorias N, H e D



Fonte – WEG (2015)

Motores de categoria N possuem baixo escorregamento. De acordo com [ABNT \(2008\)](#), T_p não deve ser inferior a 1.3 vezes o conjugado resistente de uma carga que varia com o quadrado da velocidade. Os motores também devem ser capazes de suportar duas partidas consecutivas a frio e uma partida a quente após ter funcionado nas condições nominais.

Máquinas com altos torques e baixo escorregamento se enquadram na categoria H. Esta classe de motores também suporta duas partidas consecutivas a frio e uma partida a quente, sendo esses motores usados em aplicações que exigem maior conjugado na partida que os motores da categoria N.

Conjugado de partida alto e alto escorregamento (acima de 5%) são características de um motor de categoria D. Nestes motores, ocorre a diminuição no torque aplicado quando se aumenta a velocidade do motor. São máquinas que operam com cargas de inércia elevada como elevadores, guinchos e prensas excêntricas.

8 Bancos de Dados

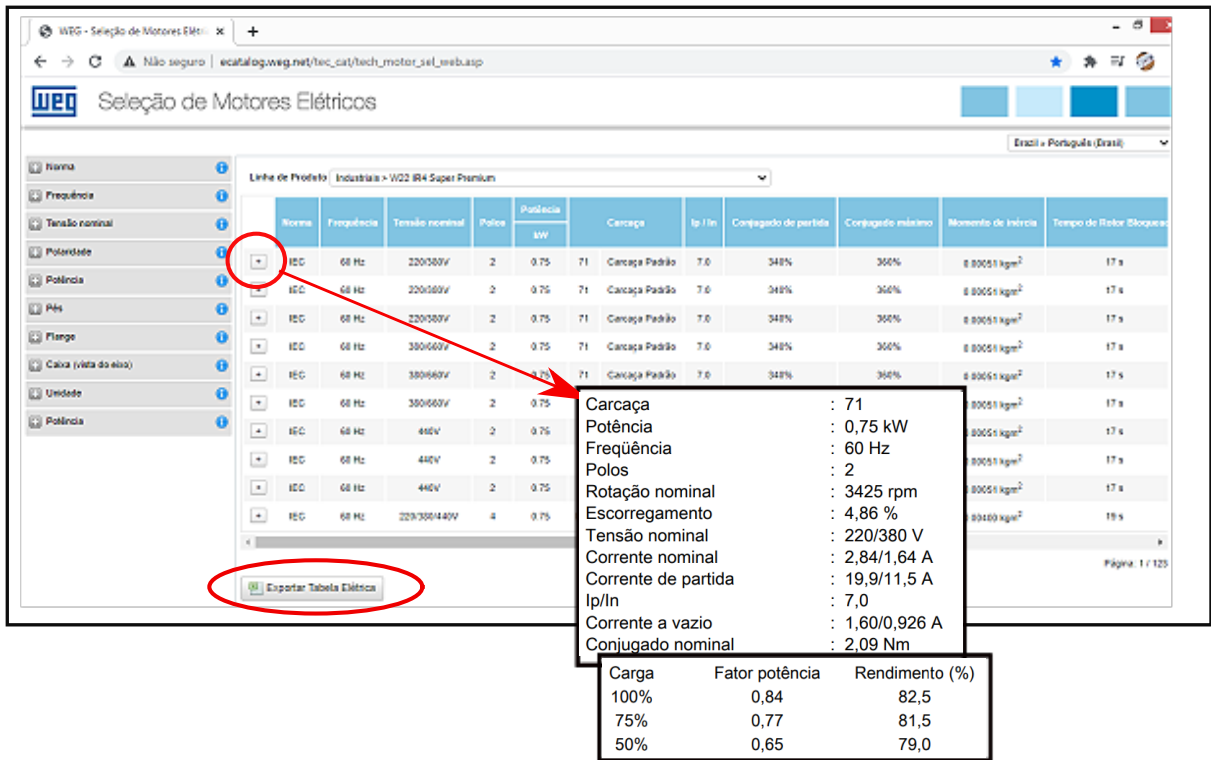
Os algoritmos dos sistemas de diagnóstico baseados nas normativas IEC 60034-1, IEC 60034-26 e ISO 20816 necessitam de conjuntos de dados para treinar, validar e testar a arquitetura de suas redes neurais artificiais. Três grandes conjuntos foram utilizados com esta finalidade. O primeiro conjunto, chamado de MIT-IEC 60034-1, é composto pelos parâmetros dos catálogos de motores de indução trifásicos industriais, sendo utilizado no desenvolvimento do Sistema de Diagnóstico Região de Operação IEC 60034-1. Dados experimentais contendo grandezas elétricas e mecânicas de um motor operando com barras quebradas no rotor compõem o segundo conjunto, denominado MIT-ISO 20816. Este banco de dados auxiliou na elaboração do Sistema de Diagnóstico de Barra Quebrada do Rotor ISO 201916. Por fim, amostras de corrente de linha de um motor de indução trifásico sob condições de tensão de alimentação desequilibrada fazem parte do conjunto MIT-IEC 60034-26, usado no treinamento da rede do Sistema de Diagnóstico Fuzzy IEC 60034-26. Descrições detalhadas de cada banco de dados se encontram neste capítulo.

8.1 MIT-IEC 60034-1

A WEG Sociedade Anônima é uma empresa multinacional brasileira fabricante de motores elétricos. Presente em mais de 12 países, a companhia fabrica cerca de 16 milhões de motores por ano (WEG, 2021b). São projetados e construídos motores trifásicos de baixa, média e alta tensão. A empresa disponibiliza em seu site um extenso banco de dados contendo as folhas de dados de uma variedade de motores de indução trifásicos, A Figura 33 contém a indicação do local de seleção no site do fabricante, em que se clica para se efetuar o *download* da planilha com os dados dos motores.

Os valores mínimos do conjugado de partida para as categorias N e H são mostrados na Tabela 2 e Tabela 3. As máquinas assíncronas do banco de dados foram classificadas segundo as categorias N, H ou D, cujos critérios adotados nesta categorização são mostrados na Tabela 4. Os limites de torque de partida mínimo e máximo em cada categoria são os menores valores da razão T_p/T_n encontrados na Tabela 2 e Tabela 3. Conhecer a categoria de cada motor garante que seja realizado a estimativa das reatâncias de dispersão, conforme descrito na Seção 7.2.

Figura 33 – Banco de dados dos motores de indução trifásicos, disponível em WEG (2021a)



Fonte – Pedro H. Camargos, 2021

Tabela 2 – Valores mínimos do torque de partida (T_p/T_n), para motores, categoria N

Faixa de Potência [kW]	Número de Pólos			
	2	4	6	8
-				
$0.37 \leq P_n \leq 0.63$	1.9	2.0	1.7	1.5
$0.63 < P_n \leq 1$	1.8	1.9	1.7	1.5
$1 < P_n \leq 1.6$	1.8	1.9	1.6	1.4
$1.6 < P_n \leq 2.5$	1.7	1.8	1.6	1.4
$2.5 < P_n \leq 4$	1.6	1.7	1.5	1.3
$4 < P_n \leq 6.3$	1.5	1.6	1.5	1.3
$6.3 < P_n \leq 10$	1.5	1.6	1.5	1.3
$10 < P_n \leq 16$	1.4	1.5	1.4	1.2

Continuação Tabela 2

$16 < P_n \leq 25$	1.3	1.4	1.4	1.2
$25 < P_n \leq 40$	1.2	1.3	1.3	1.2
$40 < P_n \leq 63$	1.1	1.2	1.2	1.1
$63 < P_n \leq 100$	1.0	1.1	1.1	1.0
$100 < P_n \leq 160$	0.9	1.0	1.0	0.9
$160 < P_n \leq 250$	0.8	0.9	0.9	0.9
$250 < P_n \leq 400$	0.8	0.8	0.8	0.8
$400 < P_n \leq 630$	0.7	0.7	0.7	0.7
$630 \leq P_n \leq 1600$	0.5	0.5	0.5	0.5
Mínimo (T_p/T_n)	0.5			

Fonte – ABNT (2008)

Tabela 3 – Valores mínimos do torque de partida (T_p/T_n), para motores, categoria H

Faixa de Potência [kW]	Número de Pólos		
	2	4	8
-	2	4	8
$0.37 \leq P_n \leq 0.63$	3.0	2.55	2.25
$0.63 < P_n \leq 1$	2.85	2.55	2.25
$1 < P_n \leq 1.6$	2.85	2.4	2.1
$1.6 < P_n \leq 2.5$	2.7	2.4	2.1
$2.5 < P_n \leq 4$	2.55	2.25	2.0
$4 < P_n \leq 6.3$	2.4	2.25	2.0
$6.3 < P_n \leq 10$	2.4	2.25	2.0
$10 < P_n \leq 16$	2.25	2.1	2.0
$16 < P_n \leq 25$	2.1	2.1	2.0
$25 < P_n \leq 40$	2.0	2.0	2.0

Continuação Tabela 3

$40 < P_n \leq 160$	2.0	2.0	2.0
Mínimo (T_p/T_n)	0.5		

Fonte – ABNT (2008)

Tabela 4 – Critérios adotados na categorização dos motores de indução

Categoria	Critério
N	$0.5T_n \leq T_p < 2T_n$
H	$2T_n \leq T_p < 2.75T_n$
D	$s > 0.05$ e $T_p \geq 2.75T_n$

Fonte – Pedro H. Camargos, 2021

8.2 MIT-ISO 20816

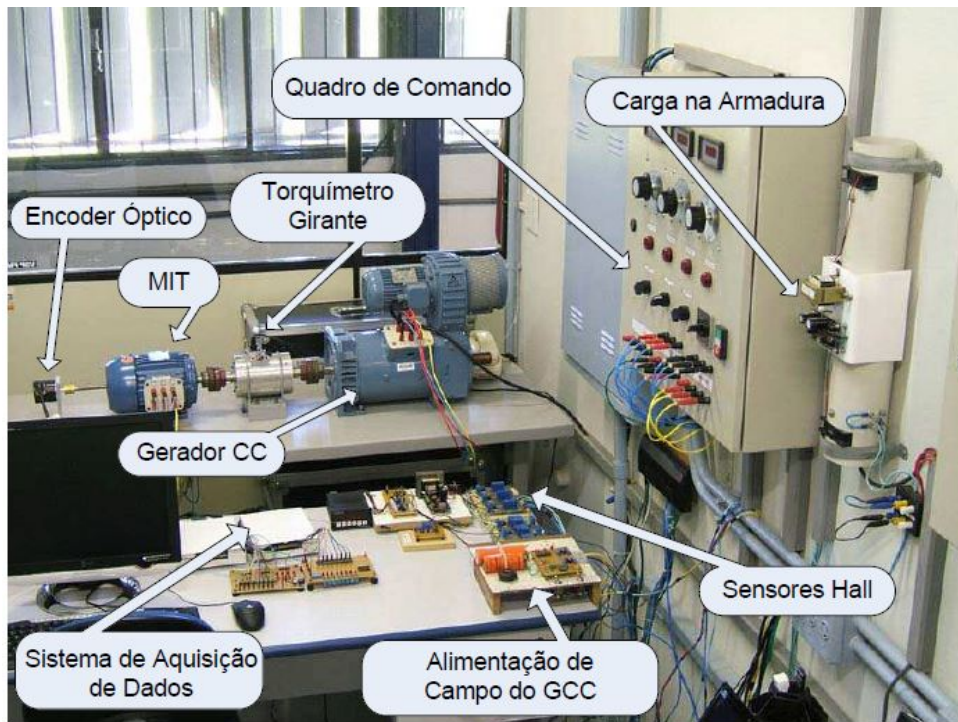
Este banco de dados é composto pelas formas de onda de corrente e tensão de linha e vibrações de um motor de indução com rotor em gaiola. As especificações do motor de indução (MIT-V) se encontram na Tabela 5. Os sinais de vibração foram obtidos através de experimentos realizados no Laboratório de Automação Inteligente de Processos e Sistemas (LAIPS) e no Laboratório de Inteligência e Controle de Máquinas Elétricas (LACIME) localizado na Escola de Engenharia da Universidade de São Paulo, São Carlos-SP. A bancada experimental é mostrada na Figura 34.

Tabela 5 – Especificações MIT-V

Parâmetro	Valor
Potência Nominal [HP]	1
Frequência [Hz]	60
Tensão Nominal [V]	220
Corrente Nominal [A]	3.02
Torque Nominal [Nm]	4.1
Rotação Nominal [rpm]	1715

Fonte – Pedro H Camargos, 2021

Figura 34 – Bancada experimental



Fonte – Suetake (2012), Liboni (2017)

No total, cinco acelerômetros foram usados simultaneamente, instalados no lado não acionado, lado acionado, na carcaça, na direção axial do motor e na mesa de suporte. Mediram-se as velocidades de vibração axial, tangencial e radial. As vibrações foram coletadas com o motor operando sob as cinco condições: sem falhas e com rotor contendo 1, 2, 3 ou 4 barras quebradas adjacentes. Em cada condição, torques de 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5 e 4.0 Nm foram aplicados no eixo da máquina. Suetake (2012) e Liboni (2017) fornecem mais informações sobre os procedimentos experimentais. Os arquivos deste banco de dados podem ser encontrados em Tremli *et al.* (2020), no formato de "struct" e extensão ".mat", ambos compatíveis com o software MATLAB/Simulink.

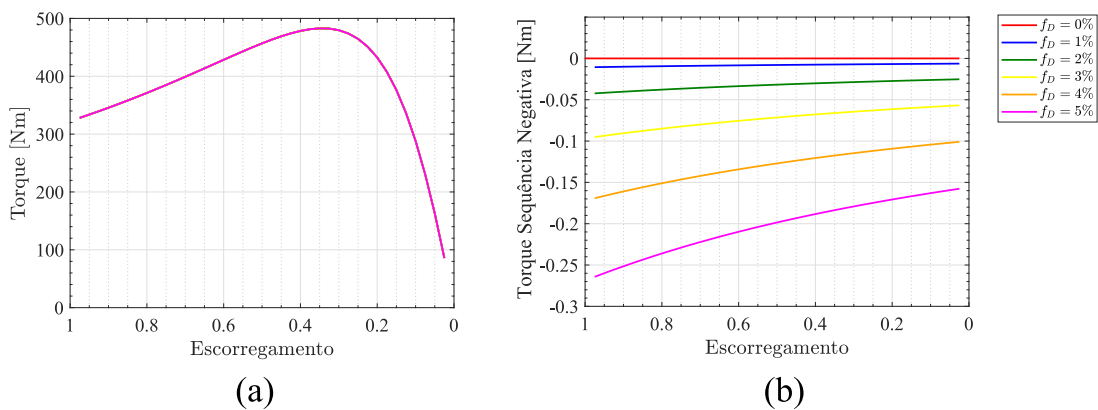
8.3 MIT-IEC 60034-26

Adekitan, Samuel e Amuta (2019) coletaram dados operacionais de um motor de indução operando sob tensões equilibradas e com Fator de Desequilíbrio de 1%, 2%, 3%, 4% e 5%. Este banco de dados é composto pelo torque de sequência positiva e negativa, potência eletromagnética, correntes no estator e no rotor, perdas nos enrolamentos e potências, ativa, reativa e do entreferro. Os ensaios para obtenção dos parâmetros foram realizados no *Department of*

Electrical and Information Engineering, Covenant University, Nigéria e foram publicados em um periódico *Data in brief*.

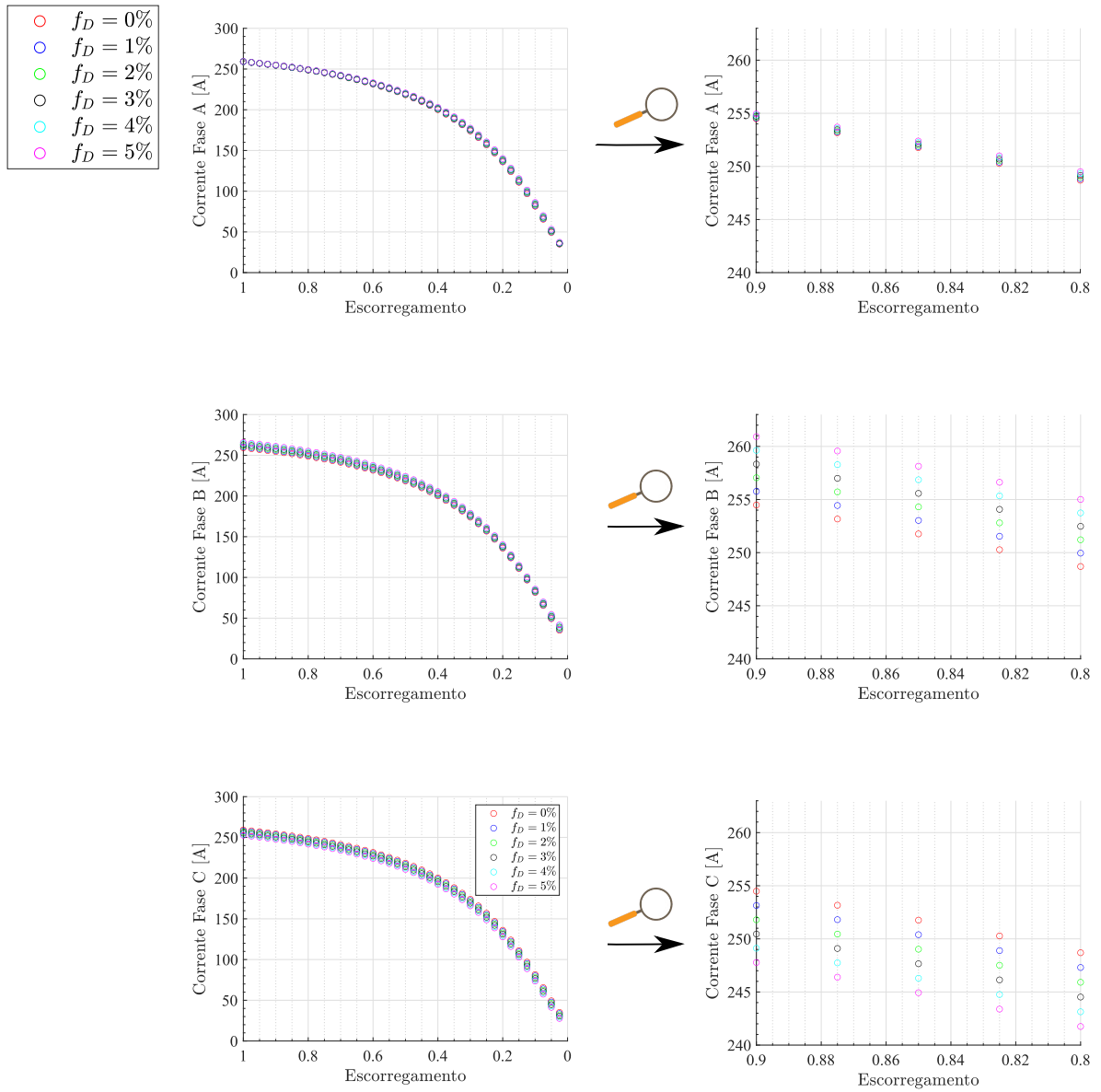
Neste trabalho, apenas os parâmetros de corrente elétrica foram utilizados. Entretanto, avaliou-se a confiabilidade dos dados disponibilizados por [Adekitan, Samuel e Amuta \(2019\)](#). Esta verificação foi realizada através da análise gráfica das curvas de corrente x escorregamento e torque x escorregamento (ver Figuras 35 36). Quanto maior o Fator de Desequilíbrio de tensão maior será o valor do torque de sequência negativa. O mesmo fenômeno não ocorre com o torque de sequência positiva.

Figura 35 – Torque x escorregamento (a) Torque sequência positiva (b) Torque sequência negativa



Fonte – Pedro H. Camargos, 2021

Figura 36 – Corrente x escorregamento



Fonte – Pedro H. Camargos, 2021

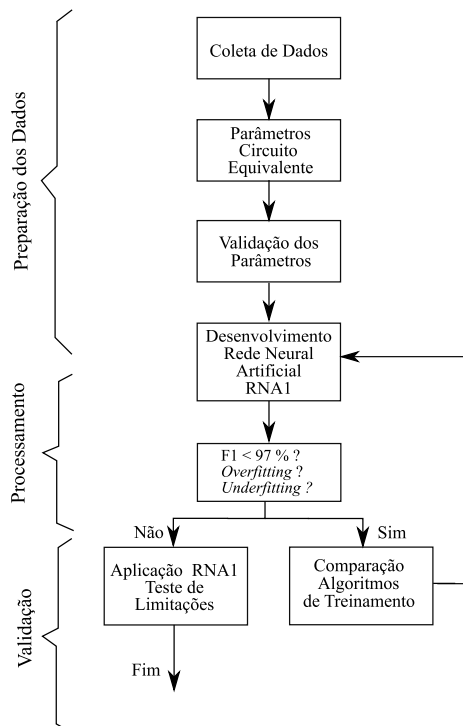
9 Metodologia

O desenvolvimento do Sistema de Diagnóstico Região de Operação IEC 60034-1, Sistema de Diagnóstico de Barra Quebrada do Rotor ISO 20816 e Sistema de Diagnóstico Fuzzy IEC 60034-26 se dividiu em três etapas: Preparação de Dados, Processamento e Validação. Os códigos de todos os sistemas foram elaborados utilizando bibliotecas nativas do software MATLAB/Simulink. Logo, este Capítulo 9, contém a descrição detalhada da metodologia utilizada em todo o processo de elaboração dos sistemas de diagnóstico e monitoramento dos motores de indução.

9.1 Sistema de Diagnóstico Região de Operação IEC 60034-1

O Sistema de Diagnóstico Região de Operação (SDRO) IEC 60034-1 tem o objetivo de classificar em qual região, A ou B, o motor esta operando (Seção 6.1.1). O SDRO IEC 60034-1 é composto por uma rede neural artificial de reconhecimento de padrões *feedforward* RNA1. Na classificação, as tensões de entrada do SDRO IEC 60034-1 possuem valores eficazes iguais. O fluxograma da Figura 37 mostra as etapas do desenvolvimento deste sistema.

Figura 37 – Etapas do desenvolvimento do SDRO IEC 60034-1

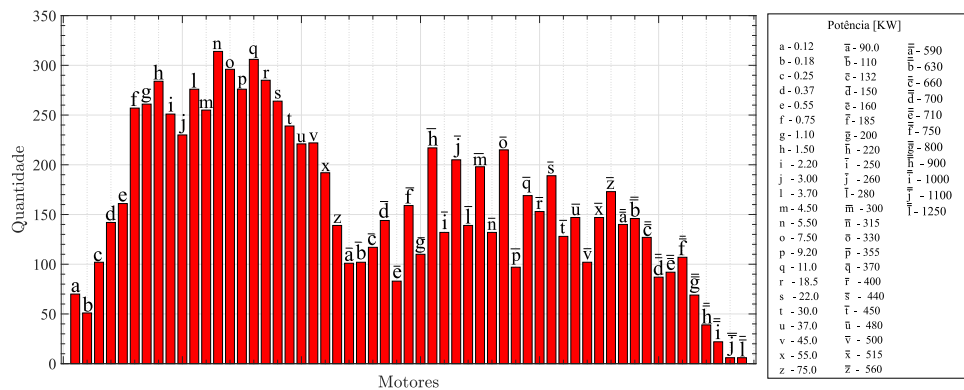


Fonte – Pedro H. Camargos, 2021

9.1.1 Preparação dos Dados

Os dados utilizados no SDR0 IEC 60034-1 foram obtidos no banco de dados da indústria WEG (veja Seção 8.1). O gráfico em barras da Figura 38 mostra as especificações nominais dos 9761 motores coletados.

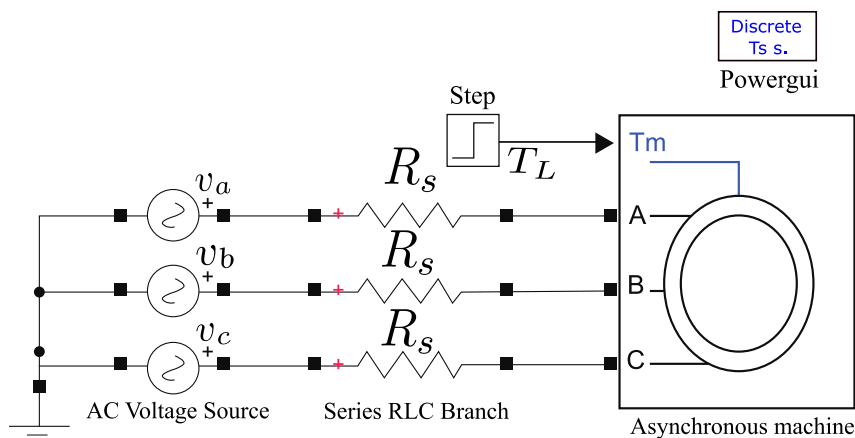
Figura 38 – Quantidade de motores coletados no banco de dados



Fonte – Pedro H. Camargos, 2021

A WEG não disponibiliza os parâmetros do circuito equivalente. Logo, através da metodologia descrita na Seção 7.2, os mesmos foram obtidos para todos os motores coletados. Possíveis erros deste método de cálculo foram detectados por simulações computacionais, caracterizando a etapa de validação dos parâmetros. A partida-direta em condições nominais de cada motor foi modelada no ambiente MATLAB/Simulink, como mostra a Figura 39.

Figura 39 – Diagrama de blocos, simulação e validação dos parâmetros



Fonte – Pedro H. Camargos, 2021

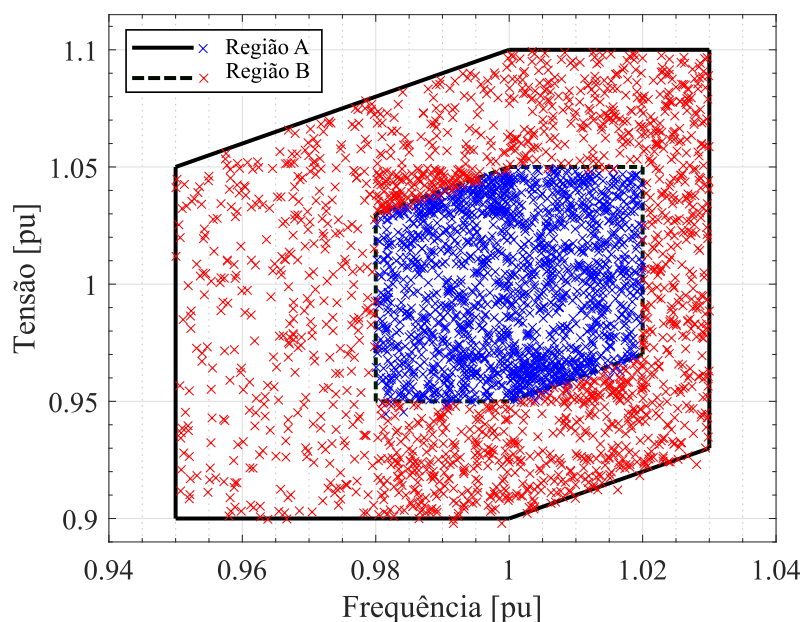
Os principais dados de entrada das simulações foram: tensões e frequência, valores nominais, parâmetros do circuito equivalente, momento de inércia e potência aparente nominal.

Tensões e correntes de linha nos enrolamentos, torque eletromagnético e velocidade de rotação constituem as variáveis de saída. Um algoritmo auxiliar calculou a potência ativa e reativa de entrada e o rendimento. Os motores que apresentaram variáveis de saída diferentes daquelas obtidas nos dados de catálogo, foram descartados.

As configurações das simulações de validação foram ajustadas considerando as análises realizadas no estudo comparativo de *software*. Neste estudo, foi comparado o desempenho do *Alternative Transient Program* (ATP) e MATLAB/Simulink nas simulações de partida-direta de um motor de indução. Diferenças mínimas foram obtidas nos resultados das simulações de ambos os *softwares*, demonstrando a confiabilidade dos ajustes de configuração feitos no MATLAB/Simulink. O Apêndice A descreve em detalhes o modelo da máquina assíncrona disponível no ATP. O estudo comparativo e as especificações dos componentes se encontram no Apêndice B.

Na etapa de validação dos parâmetros um algoritmo selecionou os motores usados na obtenção dos dados do conjunto treinamento/validação/teste da RNA1. Estes dados são formados pelas entradas e saídas das simulações de partida-direta, modelagem vista na Figura 39, com o *Asynchronous machine* alimentado com coordenadas de tensão e frequência pertencentes às regiões A ou B. As variáveis de entrada, tensão e frequência, foram geradas aleatoriamente e podem ser vistas na Figura 40.

Figura 40 – Coordenadas de tensão e frequência do banco de dados



Fonte – Pedro H. Camargos, 2021

9.1.2 Processamento

A etapa de processamento compreende os procedimentos adotados no desenvolvimento da arquitetura da RNA1. O SDRO IEC 60034-1 identifica a região de operação, A ou B. Entretanto, a máquina deve ser conectada diretamente a fonte de alimentação. A corrente de um dos terminais do estator, valor de tensão de linha, velocidade de rotação e potências ativas compõem as entradas da RNA1.

Foram usadas 3400 amostras, sendo metade categorizada na região A e o restante na região B. Dividiram-se as amostras em três subconjuntos: treinamento (50%), teste (25%) e validação (25%). Ng (2018) recomenda que se estabeleça uma única métrica de avaliação na etapa de otimização do algoritmo de aprendizagem de máquina. Fixando uma única métrica, o desenvolvedor consegue avaliar o desempenho de várias arquiteturas acelerando o processo de tomada de decisão. A métrica F_1 , (Equação (86)), foi escolhida para selecionar a melhor arquitetura de rede.

9.1.3 Validação

A arquitetura da rede neural do SDRO IEC 60034-1 foi definida de acordo com dois critérios de desempenho: $F_1 > 97\%$ e não ocorrência de *overfitting/underfitting*. A instrução de fluxo de controle *while*, descrita no Algoritmo 3, permitiu ajustar a melhor configuração de rede com o número mais adequado de neurônios da camada oculta, n_o . Considerou-se a não ocorrência de *overfitting/underfitting* nas redes que após treinamento/validação/teste apresentaram valores de erro de viés e variância menores que 1.6 %. Arquiteturas com todas as funções de ativação do tipo tangente hiperbólica e função de custo Erro Quadrático Médio (MSE) atenderam os critérios de desempenho. A Figura 41 mostra a representação da rede neural de reconhecimento de padrões *feedforward*.

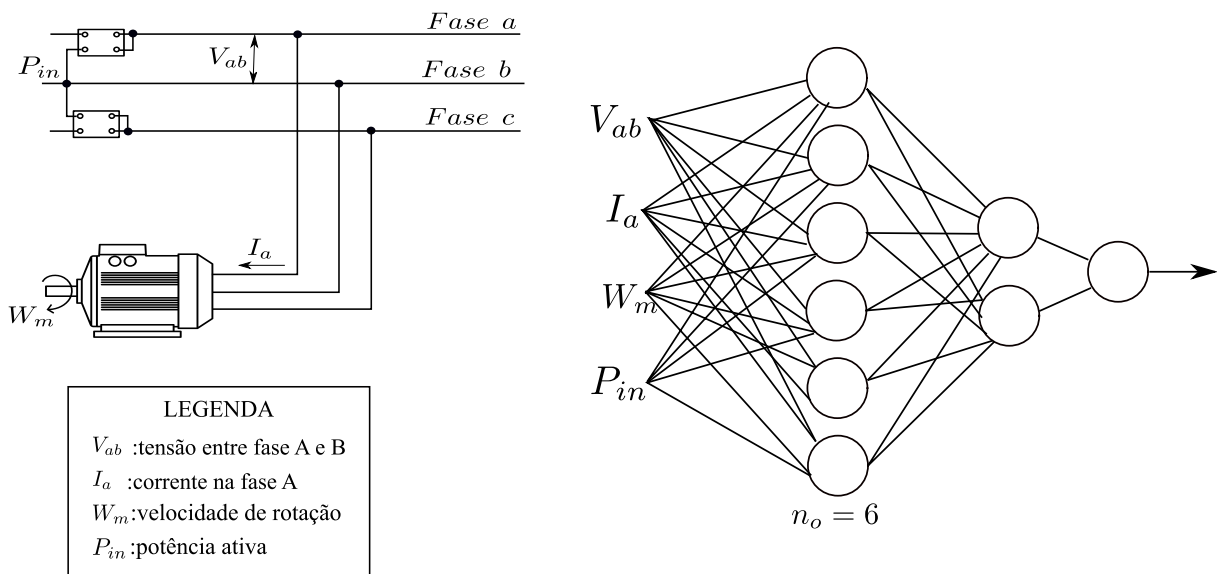
Algoritmo 3 Laço de repetição RNA1

```

1: while Overfitting e Underfitting do
2:    $n_o = 1$ 
3:    $\Delta n_o = 1$ 
4:   while  $F_1 \leq 0.97$  do
5:     Treina a rede
6:     Testa a rede
7:     Valida a rede
8:     Calcula  $F_1$ 
9:      $n_o = n_o + \Delta n_o$ 

```

Figura 41 – Arquitetura da rede neural de reconhecimento de padrões SDRO IEC 60034-1



Fonte – Pedro H. Camargos, 2021

Os gráficos das curvas de desempenho pós treinamento/validação/teste obtidos no MATLAB/Simulink podem ser vistos na Figura 42. Época é a medida do número de vezes que todos os vetores de treinamento são usados uma vez para atualizar os pesos na rede. A melhor época, E_{p_1} , ocorre quando o erro de validação atinge seu valor mínimo, L_{v_1} . Foram calculadas as diferenças entre os erros de teste e validação, vetor Δ . Redes neurais que tiveram algum valor de $\Delta_n > 10^{-2}$, para $n = 1, 2, \dots, k$ foram descartadas.

Se o número de parâmetros na rede for muito menor do que o número total de pontos no conjunto de treinamento, há pouca ou nenhuma chance de ter ocorrido *overfitting* (MATHWORKS, 2020a). Portanto, haverá *overfitting* se o número de equações de treinamento,

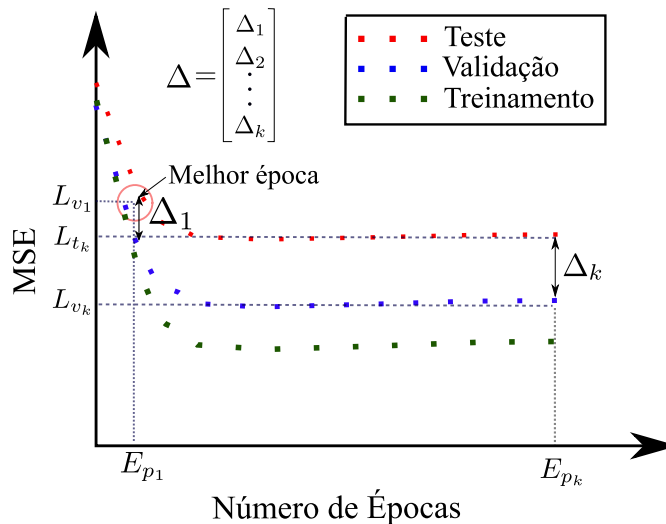
N_{treq} , for menor que o número de pesos desconhecidos, N_{uw} . Para uma rede de camada única, N_{uw} pode ser calculado por:

$$N_{uw} = (I + 1)H + (H + 1)O \quad (134)$$

As incógnitas I , H e O equivalem ao número de entradas, número de neurônios na camada oculta e número de classes de saída, respectivamente. N_{trn} é a quantidade de elementos do conjunto de dados usados para treinar a rede.

$$N_{treq} = N_{trn}O \quad (135)$$

Figura 42 – Procedimento avaliação *overfitting*



Fonte – Pedro H. Camargos, 2021

O MATLAB/Simulink disponibiliza diversos algoritmos de otimização (Tabela 6). Um estudo comparativo foi realizado com o intuito de avaliar se o Levenberg-Marquardt seria o algoritmo mais adequado para o projeto do SDRO IEC 60034-1. Realizou-se 14000 simulações de treinamento/validação/teste, 1000 para cada algoritmo e 7000 para cada arquitetura de rede vista na Figura 41. Cabe ressaltar que se adotou ajustes padronizados na etapa de treinamento pré-estabelecidos pelas sintaxes do MATLAB/Simulink, descritos por MathWorks (2020b). Obtiveram-se as métricas de classificação F_1 , A_{cc} , R_{ec} e E_{sp} ao final de cada simulação. Foram contabilizadas a quantidade de simulações com métricas pertencentes aos intervalos: 0 a 50%, 51% a 90% e acima de 91%.

Tabela 6 – Funções de treinamento do MATLAB/Simulink

Sigla	Algoritmo de Otimização	Função de Custo	Sintaxe MATLAB
BFG	BFGS Quasi-Newton	Entropia Cruzada	"trainbfg"
CGB	Gradiente Conjugado com Reinicialização de Powell / Beale	Entropia Cruzada	"traincgb"
CGP	Gradiente Conjugado Polak-Ribière	Entropia Cruzada	"traincgp"
GDM	Gradiente Descendente com Momento	Entropia Cruzada	"traingdm"
LM	Levenberg-Marquardt	Erro Quadrático Médio	"trainlm"
RPR	Retropropagação Resiliente	Entropia Cruzada	"trainrp"
SCG	Gradiente Conjugado em Escala	Entropia Cruzada	"trainscg"

Fonte – Adaptado de [MathWorks \(2020b\)](#)

Com o objetivo de testar e verificar as limitações da RNA1, o SDRO IEC 60034-1 foi utilizado na avaliação das condições operativas de alguns motores de indução que não constavam no banco de dados, cujas especificações se encontram na Tabela 7.

Tabela 7 – Especificações motores de indução aplicação SDRO IEC 60034-1

Motor	MIT-X	MIT-Y	MIT-Z
Potência Nominal [HP]	5	100	200
Frequência [Hz]	60	60	60
Tensão Nominal [V]	460	460	575
Corrente Nominal [A]	6.36	112	173
Momento de Inercia [kgm^2]	0.0057	0.0554	0.0634
Número de Polos	4	4	4
Rotação Nominal [rpm]	1750	1780	1785

Fonte – Adaptado de [MathWorks \(2021a\)](#)

As máquinas estiveram sujeitas aos ensaios descritos na Tabela 8. Em cada ensaio variou-se as tensões eficazes das fases AB, V_{ab} , BC, V_{bc} , e CA, V_{ca} que alimentam os enrolamentos do estator da máquina. A frequência da fonte também sofreu modificações em função da frequência nominal dos motores, f_n . Os valores de V_{ab} , V_{bc} e V_{ca} pertencem a uma região de operação, (ver Figura 24), que foi identificada manualmente em cada ensaio consultando a norma IEC 60034-1. Por exemplo, no ensaio II, a máquina está operando na região B. Os resultados encontrados nos ensaios foram comparados aos obtidos pelo SDRO IEC 60034-1, e se encontram na Seção 10.1.3.

Tabela 8 – Ensaio testes/verificação limitação SDRO IEC 60034-1, com tensão nominal V_n

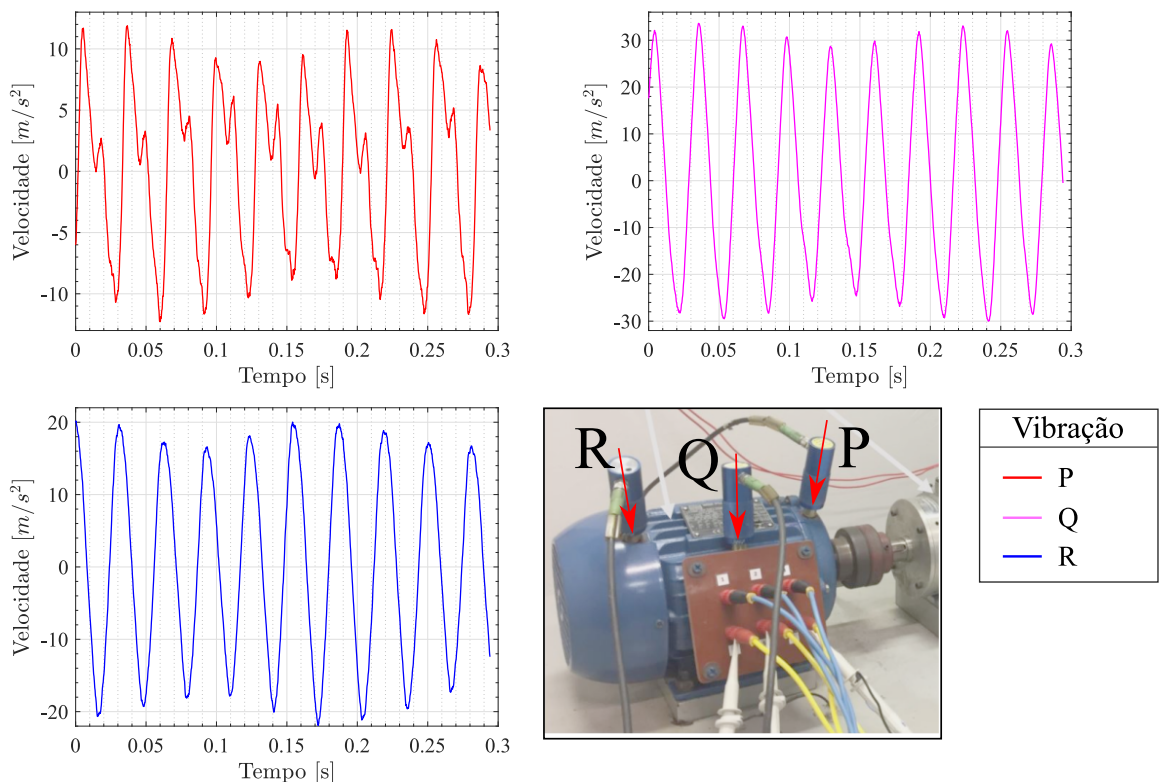
Motor	MIT-X			MIT-Y			MIT-Z		
	Ensaio	I	II	III	IV	V	VI	VII	VIII
V_{ab}, V_{bc}, V_{ca}	V_n	$1.06V_n$	$0.97V_n$	$0.9V_n$	$0.94V_n$	$1.1V_n$	$1.02V_n$	$0.99V_n$	$0.96V_n$
f_{ss}	f_n	$0.97f_n$	f_n	$0.97f_n$	$1.02f_n$	$1.03f_n$	$1.02f_n$	$0.97f_n$	$1.01f_n$

Fonte – Pedro H. Camargos 2021

9.2 Sistema de Diagnóstico de Barra Quebrada do Rotor ISO 20816

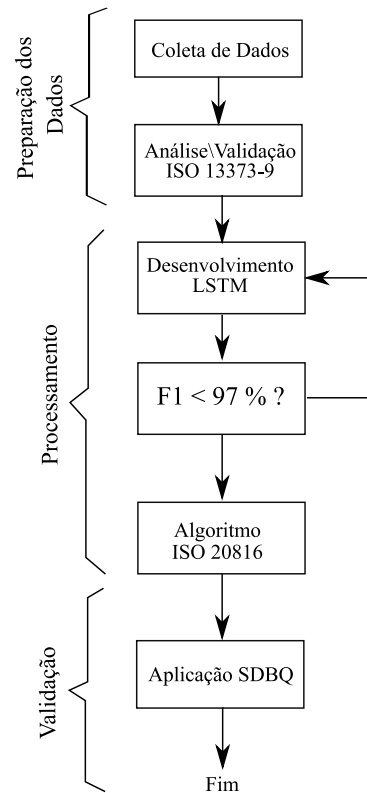
O Sistema de Diagnóstico de Barra Quebrada do Rotor (SDBQ) ISO 20816 foi desenvolvido com algoritmos que empregam redes neurais recorrentes LSTM. Este sistema detecta falhas nas barras do rotor do MIT-V. O SDBQ informa se o motor possui uma barra, duas, três ou quatro barras quebradas monitorando os sinais de vibração no lado acionado, ponto P , na carcaça, Q , e na tampa de ventilação da máquina, R como mostra a Figura 43. O SDBQ efetua o diagnóstico da zona de operação A, B, C ou D conforme a norma ISO 20816 (veja Seção 6.3). O projeto do SDBQ ISO 20816 foi dividido em três etapas: preparação, processamento e validação dos dados. O diagrama de desenvolvimento do SDBQ ISO 20816 é mostrado na Figura 44.

Figura 43 – Sinais de vibração do motor com 4 barras quebradas



Fonte – Pedro H. Camargos, 2021

Figura 44 – Metodologia projeto SDBQ



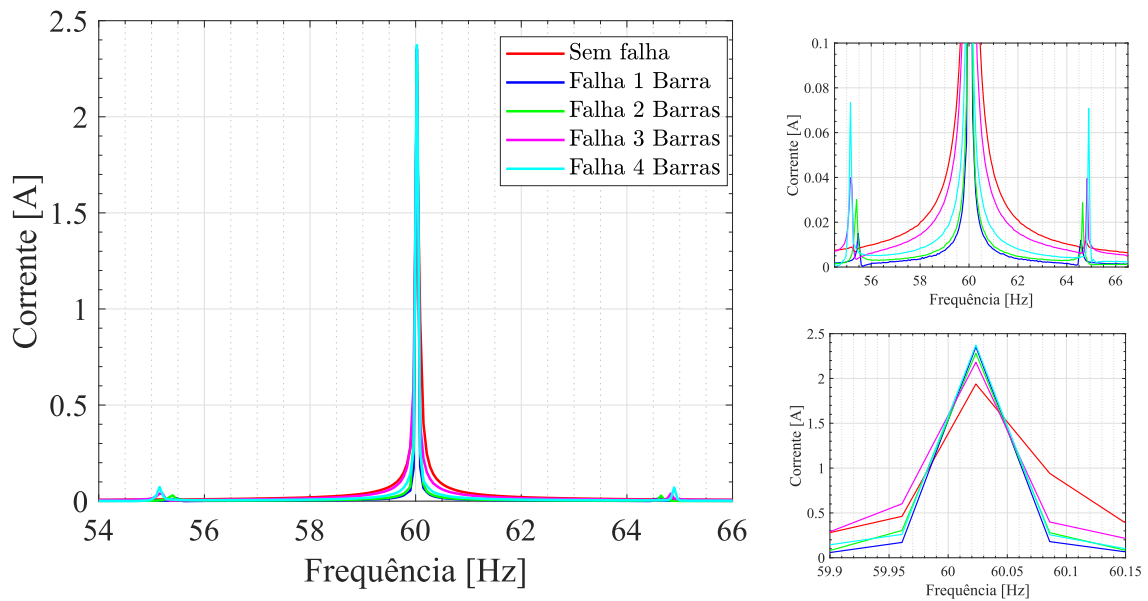
Fonte – Pedro H. Camargos, 2021

9.2.1 Preparação dos Dados

O conjunto de dados usados na elaboração do SDBQ foi previamente descrito na Seção 8.2. O conjunto treinamento/validação/teste da LSTM é composto pelos sinais de vibração no lado acionado, na carcaça e na tampa de ventilação, com amostragem de tempo de 0.3 segundos. Os parâmetros foram coletados com MIT-V operando em regime permanente e com torques de carga de 0.5 a 4 Nm, variando em intervalos de 0.5 Nm. Foram destinadas 50% das amostras deste conjunto para a etapa de treinamento, 25% para validação e 25% para teste. Em todas as etapas, a amostragem do conjunto foi aleatória.

A análise do conjunto de dados foi realizada avaliando o espectro de correntes através da técnica MCSA. O espectro de correntes do banco de dados mostrou que houve aumento na magnitude da corrente nas bandas laterais à medida em que se aumenta o número de barras danificadas, (Figura 45). Esta constatação reafirma a confiabilidade dos dados de vibração utilizados no projeto do SDBQ.

Figura 45 – Espectro de corrente SDBQ



Fonte – Pedro H. Camargos, 2021

9.2.2 Processamento

A LSTM é uma rede neural recorrente que classifica o estado de operação do motor. As classes são: saudável, quebra em uma barra, duas, três ou quatro barras. O ajuste do número exato de neurônios contidos na camada oculta bidirecional, *bilstmLayer*, foi realizado de acordo com a métrica F_1 . Caso F_1 fosse menor que 97% na etapa de treinamento para todas as classes, se incrementava o número de neurônios. A LSTM foi ajustada para conter 100 unidades na *bilstmLayer*. O Otimizador Adam, detalhado em MathWorks (2021h), foi o algoritmo adotado no treinamento da LSTM com função de custo entropia cruzada para k classes mutuamente exclusivas.

A Figura 46 mostra detalhadamente a composição das camadas da LSTM. Os principais parâmetros de ajuste implementados no *software* MATLAB estão destacados no esquemático. A configuração da LSTM possui seis camadas. A primeira é uma camada de entrada de sequência, *SequenceInputLayer*, que insere os dados de sequência na rede. As entradas da *SequenceInputLayer* são os vetores que correspondem aos sinais vibração coletados nos pontos P , Q e R (veja Figura 43). Na implementação desta camada o usuário pode escolher se irá aplicar a normalização nos dados. O parâmetro "auto" indica que os dados não foram normalizados (MATHWORKS, 2021g). A próxima camada é bidirecional, conhecida como *bilstmLayer*. Sua função é aprender as dependências bidirecionais de longo prazo entre etapas de tempo da série

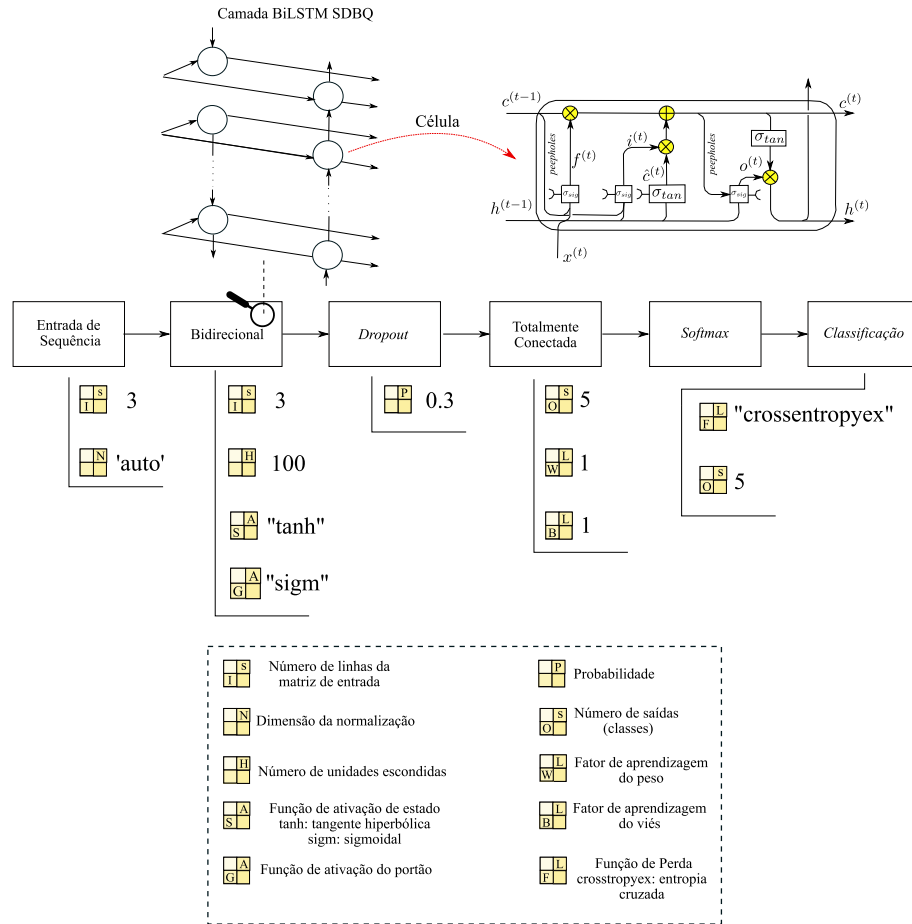
temporal. As funções de ativação das portas e de estado nesta camada são a sigmoide e tangente hiperbólica, respectivamente.

Para prevenir a ocorrência de *overfitting*, como demonstrado por [Srivastava et al. \(2014\)](#), uma camada de *dropout* foi conectada a *bilstmlayer*. Foi ajustado uma probabilidade de eliminar os elementos da entrada igual a 0.3 ou 30%. No momento do treinamento, é definido na *dropout* aleatoriamente quais os elementos de entrada serão iguais a zero ([MATHWORKS, 2021c](#)). Quando maior for a probabilidade ajustada, maior será o número de elementos descartados durante o treinamento.

Interligada a *dropout*, tem-se a camada totalmente conectada, *fullyConnectedLayer*, que multiplica as entradas por uma matriz de pesos e em seguida adiciona o vetor de vieses. No ato da montagem do algoritmo o programador possui a opção de ajustar os pesos e os vieses através da multiplicação pelo fator de aprendizagem de peso e de viés. Optou-se por não ajustar os pesos e vieses atribuindo o valor de 1 a estes multiplicadores. Em sequência, conectou-se a camada *softmax*. A última camada da LSTM é a de classificação, *classificationLayer*, cuja função é calcular a perda de entropia cruzada no processo de classificação ([MATHWORKS, 2021b](#)).

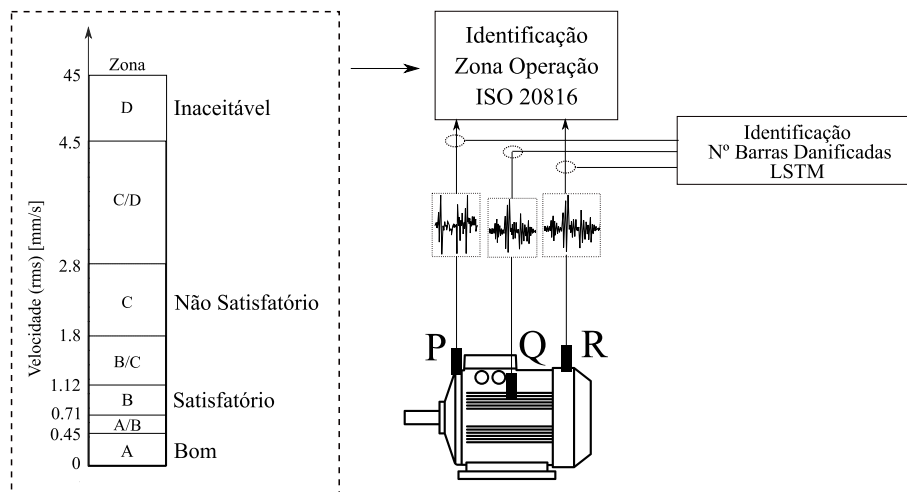
O algoritmo ISO 20816 informa a severidade do sinal de vibração conforme as diretrizes da norma [ISO \(2016\)](#). O algoritmo recebe como entrada os sinais de velocidade de vibração coletados nos pontos *P*, *Q* e *R*, medidos na máquina operando em regime permanente. Em seguida, os valores eficazes destes sinais são calculados. Os limites de vibração adotados são mostrados na Figura 47. Suponha que foram medidos valores de velocidade de vibração igual a 1.9 mm/s (ponto *P*), 1.5 mm/s e no ponto *R*. Apesar do sinal no ponto *R* estar entre as regiões *B/C*, a máquina será diagnosticada com nível de vibração insatisfatório para operação contínua a longo prazo.

Figura 46 – Estrutura da rede neural recorrente LSTM SBDQ



Fonte – Pedro H. Camargos, 2021

Figura 47 – Diagrama de blocos SDBQ



Fonte – Pedro H. Camargos, 2021

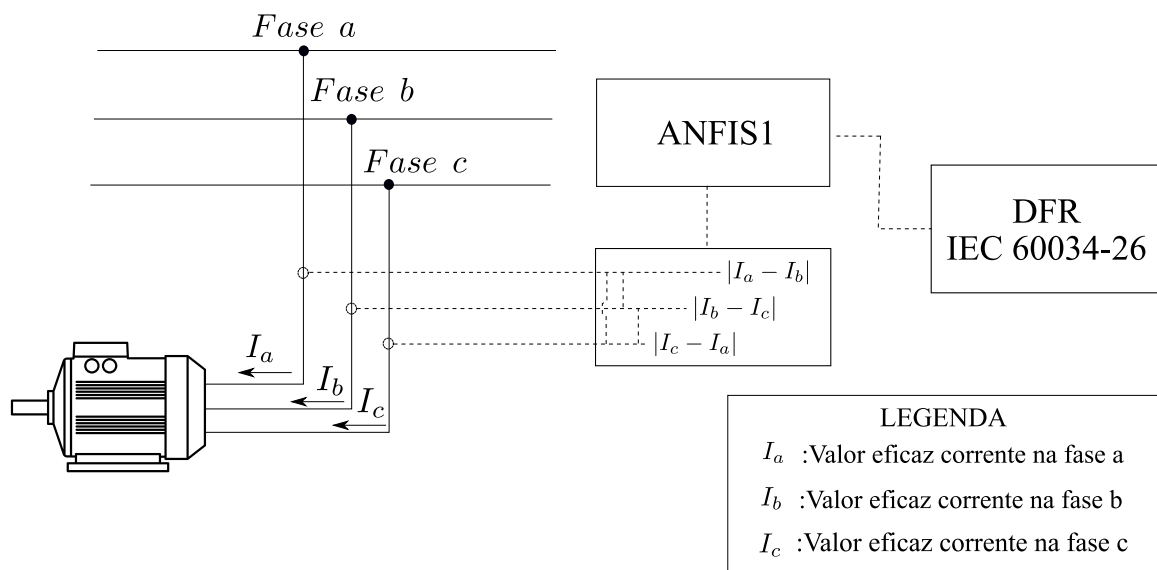
9.2.3 Validação

As amostras do conjunto de dados foram utilizadas na etapa de validação da LSTM. O SBDQ classificou todas as amostras dos sinais e as categorizou através do algoritmo ISO 20816. Foi realizada a análise das matrizes de confusão na verificação da confiabilidade da LSTM. Um levantamento analítico foi realizado com o intuito de avaliar as limitações operativas do SBDQ ISO 20816.

9.3 Sistema de Diagnóstico Fuzzy IEC 60034-26

O Sistema de Diagnóstico Fuzzy (SDF) IEC 60034-26 informa o Fator de Redução, F_r , do motor de acordo com as recomendações da norma IEC 60034-26, IEC (2006). O SDF IEC 60034-26 é composto pelo Subsistema Adaptativo de Inferência Neuro-Fuzzy (ANFIS1) e a Detecção do Fator de Redução IEC 60034-26 (DFR), conforme mostrado na Figura 48.

Figura 48 – Diagrama de blocos SDF



Fonte – Pedro H. Camargos, 2021

9.3.1 Subsistema Adaptativo de Inferência Neuro-Fuzzy

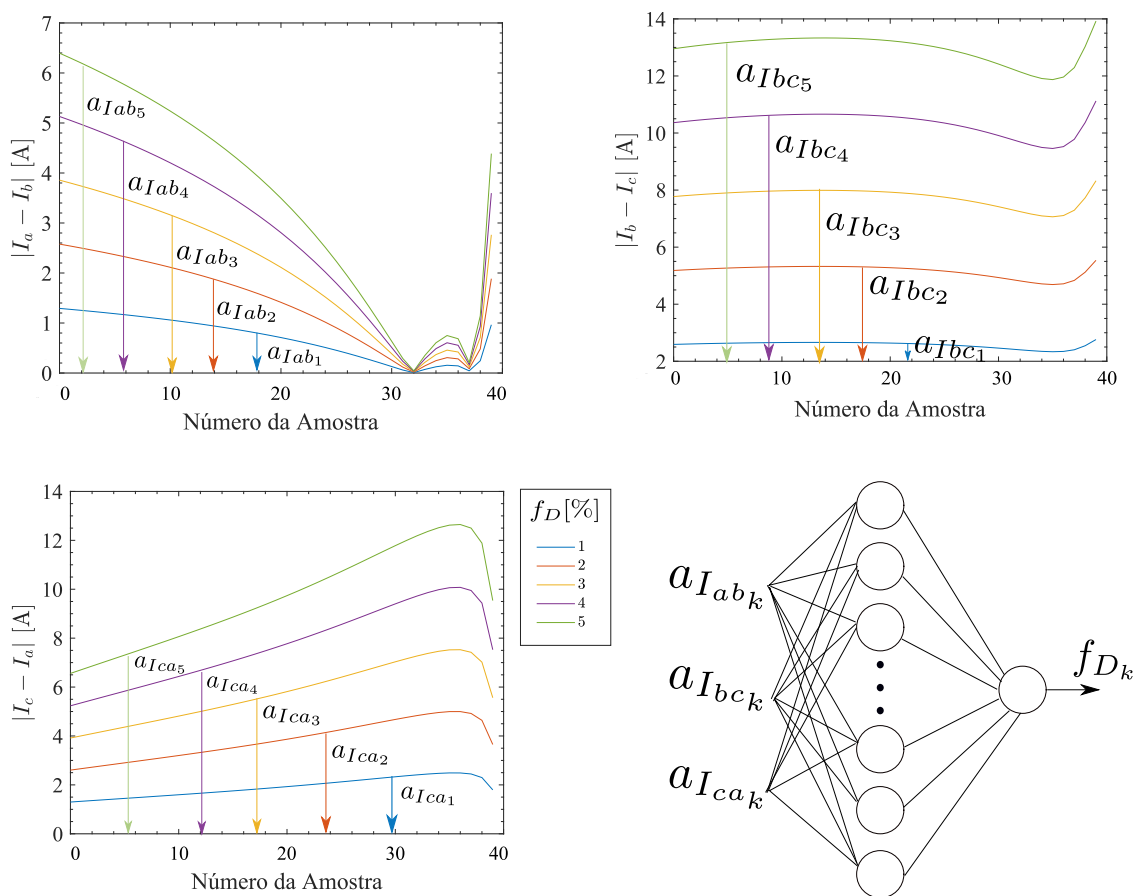
A Seção 8.3 descreveu as características do conjunto de dados usado no desenvolvimento do ANFIS1. Entretanto, foram coletados apenas os valores eficazes das correntes com o motor operando somente em cinco condições de desequilíbrio. Para complementação destes dados

se projetou uma rede neural artificial multicamadas, RNA2, que prevê o f_D quando o usuário fornece as variações de corrente, $|I_a - I_b|$, $|I_b - I_c|$ e $|I_c - I_a|$.

Os gráficos da Figura 49 demonstram que quanto maior o Fator de Desequilíbrio maior é a área abaixo da curva. Esta característica foi adotada na elaboração do conjunto treinamento, validação e teste da RNA2 (Tabela 9). Os dados de entrada da rede são as áreas a_{Iab_k} , a_{Ibc_k} e a_{Ica_k} , onde k corresponde à porcentagem do Fator de Desequilíbrio.

A RNA2 possui 10 neurônios na camada oculta. O algoritmo usado no treinamento da rede foi o SCG e a função de custo MSE. As curvas de desempenho da RNA2 se encontram na Figura 50. Valores muito baixos de MSE foram atingidos nas etapas de treinamento/validação/teste da RNA2 ao final de 6 épocas. Após projetada, a RNA2 foi usada na previsão de fatores de desequilíbrio diferentes dos coletados por Adekitan, Samuel e Amuta (2019). Na etapa de elaboração do ANFIS1, as entradas da rede são as áreas abaixo das curvas de variação de corrente com amostras k correspondentes a $1 < f_D < 2$, $2 < f_D < 3$, $3 < f_D < 4$ e $4 < f_D < 5$.

Figura 49 – Diferenças nos valores eficazes e características RNA2



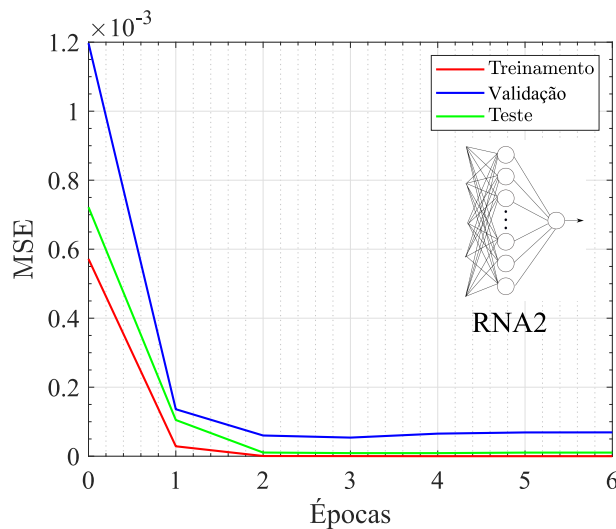
Fonte – Pedro H. Camargos, 2021

Tabela 9 – Conjunto treinamento/validação/teste RNA2

a_{Iab_1}	27.76	a_{Iab_2}	55.30	a_{Iab_3}	82.61	a_{Iab_4}	109.71	a_{Iab_5}	136.57
a_{Ibc_1}	102.99	a_{Ibc_2}	206.29	a_{Ibc_3}	309.88	a_{Ibc_4}	413.75	a_{Ibc_5}	517.88
a_{Ica_1}	76.33	a_{Ica_2}	153.16	a_{Ica_3}	230.49	a_{Ica_4}	308.30	a_{Ica_5}	386.59
f_{D_1}	0.01	f_{D_2}	0.02	f_{D_3}	0.03	f_{D_4}	0.04	f_{D_5}	0.05

Fonte – Pedro H. Camargos, 2021

Figura 50 – Curvas de desempenho RNA2



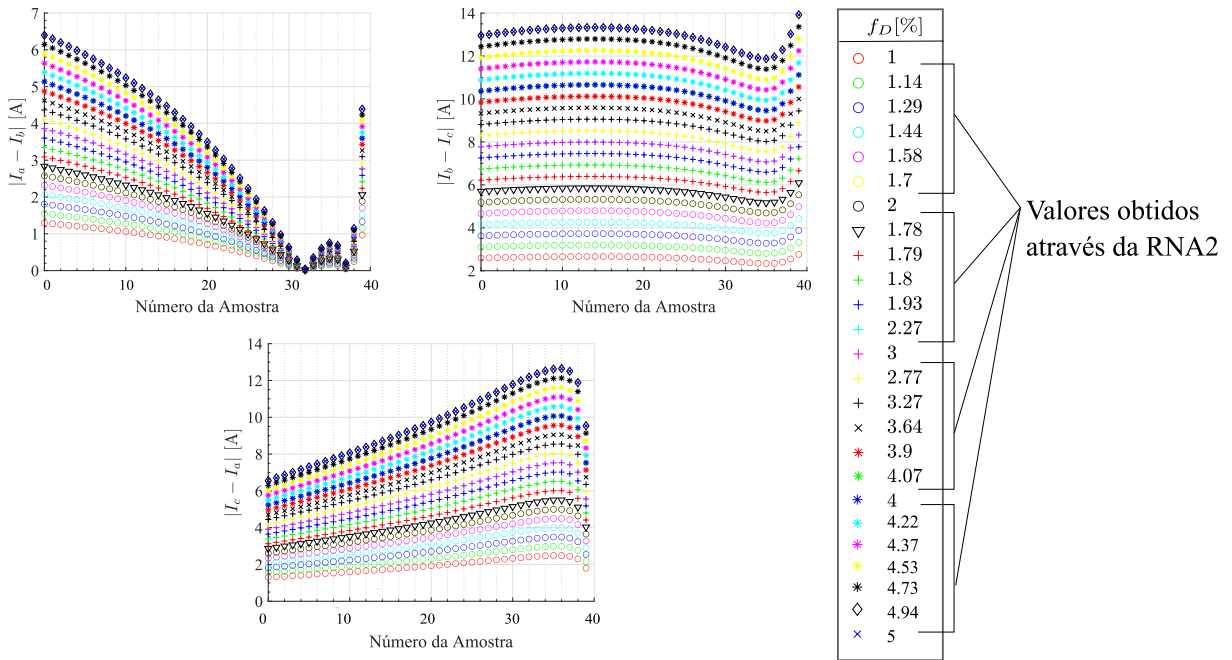
Fonte – Pedro H. Camargos, 2021

O ANFIS1 opera com aprendizagem neuro adaptativa. Isto significa que é possível aplicar a inferência *fuzzy* em um sistema em que o programador, a priori, não consegue discernir facilmente os parâmetros das funções de pertinência. O algoritmo trabalha com uma saída única, f_D , com o número de funções de pertinência sendo igual ao número de regras. Conforme especificado em [MathWorks \(2021f\)](#), a defuzzificação é realizada por média ponderada com peso unitário para cada regra.

As diferenças de corrente, valores discretos $|I_a - I_b|$, $|I_b - I_c|$ e $|I_c - I_a|$, são as entradas do ANFIS1 como mostra a Figura 51. O método neuro-adaptativo foi projetado com um número máximo de 40 épocas. Isto significa que quando se atinge este valor ocorre a parada no processo de treinamento. A retropropagação foi utilizada na otimização na etapa de treinamento. Foram escolhidos aleatoriamente 50% dos dados para treinamento, 25% para validação e 25 % para o teste de precisão (detalhado na Seção 9.3.3).

Finalizada a etapa de treinamento da rede neural, foi obtido um sistema de inferência *fuzzy* Sugeno. Os parâmetros deste sistema se encontram na Tabela 10. A média ponderada de todas as saídas de regra foi escolhida como método de defuzzificação, terminologia "wtaver".

Figura 51 – Conjunto de entradas, $|I_a - I_b|$, $|I_b - I_c|$ e $|I_c - I_a|$ e saídas, f_D , do ANFIS1



Fonte – Pedro H. Camargos, 2021

Tabela 10 – Parâmetros sistema de inferência Sugeno ANFIS1

Nº de entradas	3
Funções Pertinência por entrada	2
Nº de saídas	1
Nº de regras	8
Funções Pertinência / saída	8
Método de defuzzificação	"wtaver"

Fonte – Pedro H. Camargos, 2021

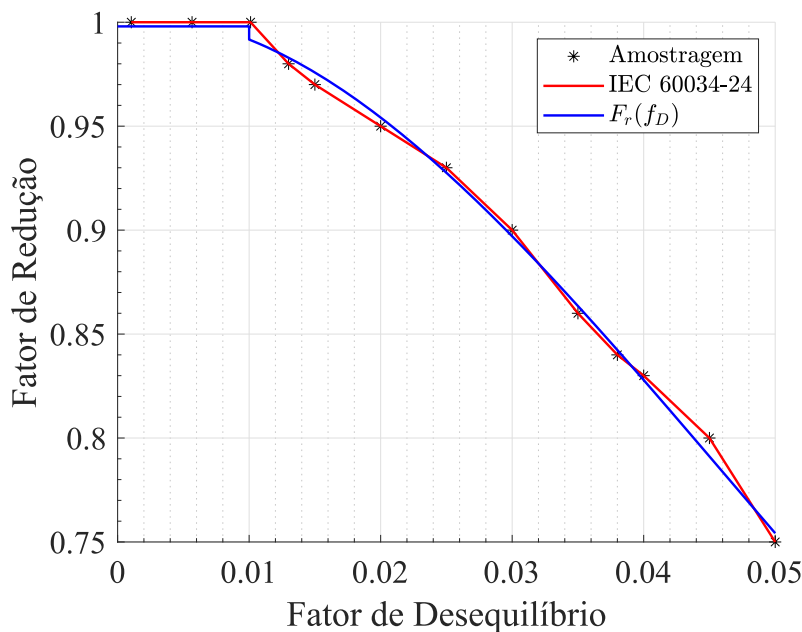
9.3.2 Detecção do Fator de Redução IEC 60034-26

O DFR 60034-26 calcula o F_r utilizando a função $F_r(f_D)$, Equação (136). Um método de ajuste de curvas polinomial usando a matriz *Vandermonde*, MathWorks (2020c), foi aplicado a uma amostragem de pontos da curva da Figura 25. O algoritmo de ajuste de curvas retorna os

coeficientes para um polinômio $F_r(f_D)$ de grau n que é o melhor ajuste (no sentido de mínimos quadrados) para os dados em y . Neste caso, y corresponde aos pontos que compõem o conjunto da amostragem. Os coeficientes em F_r estão em potências decedentes, e o comprimento de F_r é $n + 1$. Portanto, obteve-se a função aproximada $F_r(f_D)$. As curvas são mostradas na Figura 52.

$$F_r(f_D) = \begin{cases} 1 & f_D \leq 0.01 \\ 1259f_D^3 - 173.6f_D^2 + 0.5768f_D + 1.002 & f_D > 0.01 \end{cases} \quad (136)$$

Figura 52 – Fator de redução x Fator de Desequilíbrio



Fonte – Pedro H. Camargos, 2021

9.3.3 Teste de Precisão SDF IEC 60034-26

O teste de precisão avaliou a efetividade do SDF IEC 60034-26 na previsão dos fatores de redução. Dois conjuntos de elementos foram usados nesta etapa: D e A . O conjunto D que contém, vetores de corrente, $|I_a - I_b|$, $|I_b - I_c|$ e $|I_c - I_a|$ e seus respectivos Fatores de Redução, F_{rD} . Nenhum dos valores contidos em D foram usados no treinamento e validação do sistema neuro-*fuzzy*. O conjunto A é composto pelos mesmos valores de diferenças de corrente do conjunto D , porém os fatores de desequilíbrio, f_{DA} , e os fatores de redução, F_{rA} , foram obtidos

através do SDF 60034-26. O erro relativo, ΔE_{F_r} , para cada um dos 246 valores foi calculado por:

$$\Delta E_{F_r} = \frac{|F_{rD} - F_{rA}|}{F_{rD}} \quad (137)$$

10 Resultados e Discussões

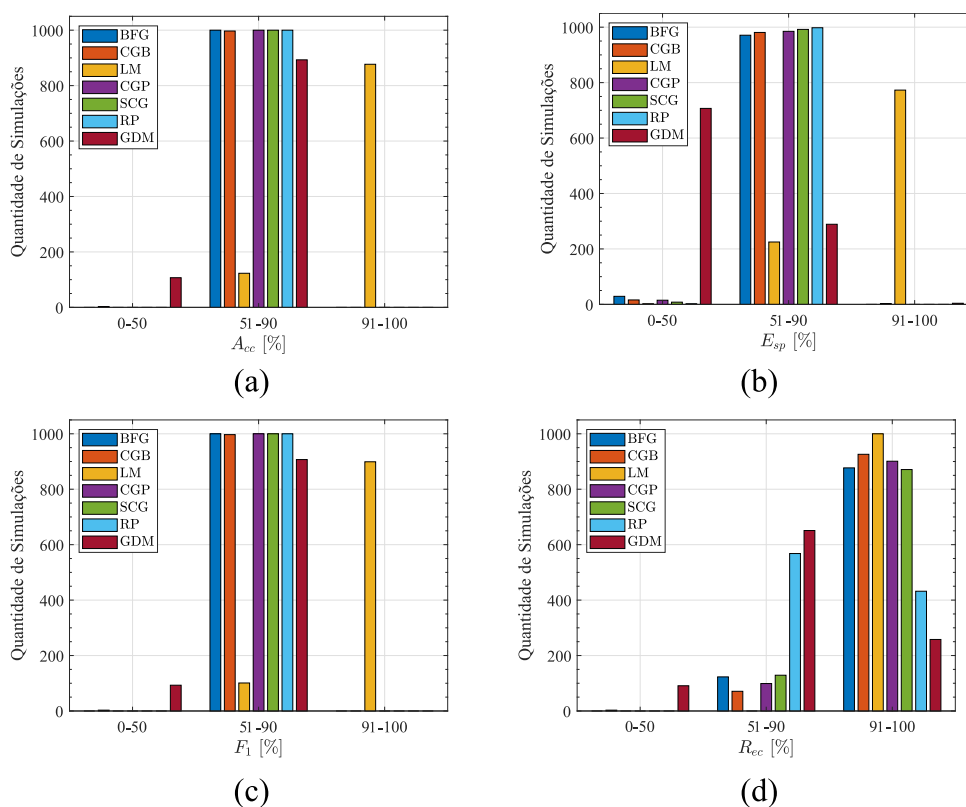
10.1 Sistema de Diagnóstico Região de Operação IEC 60034-1

10.1.1 Comparação Algoritmos de Otimização

Os algoritmos de otimização possuem diferentes características de desempenho tais como requisitos de memória, velocidade de processamento e precisão numérica. Os resultados gráficos do estudo comparativo para a RNA1 são mostrados na Figura 53 e Figura 54.

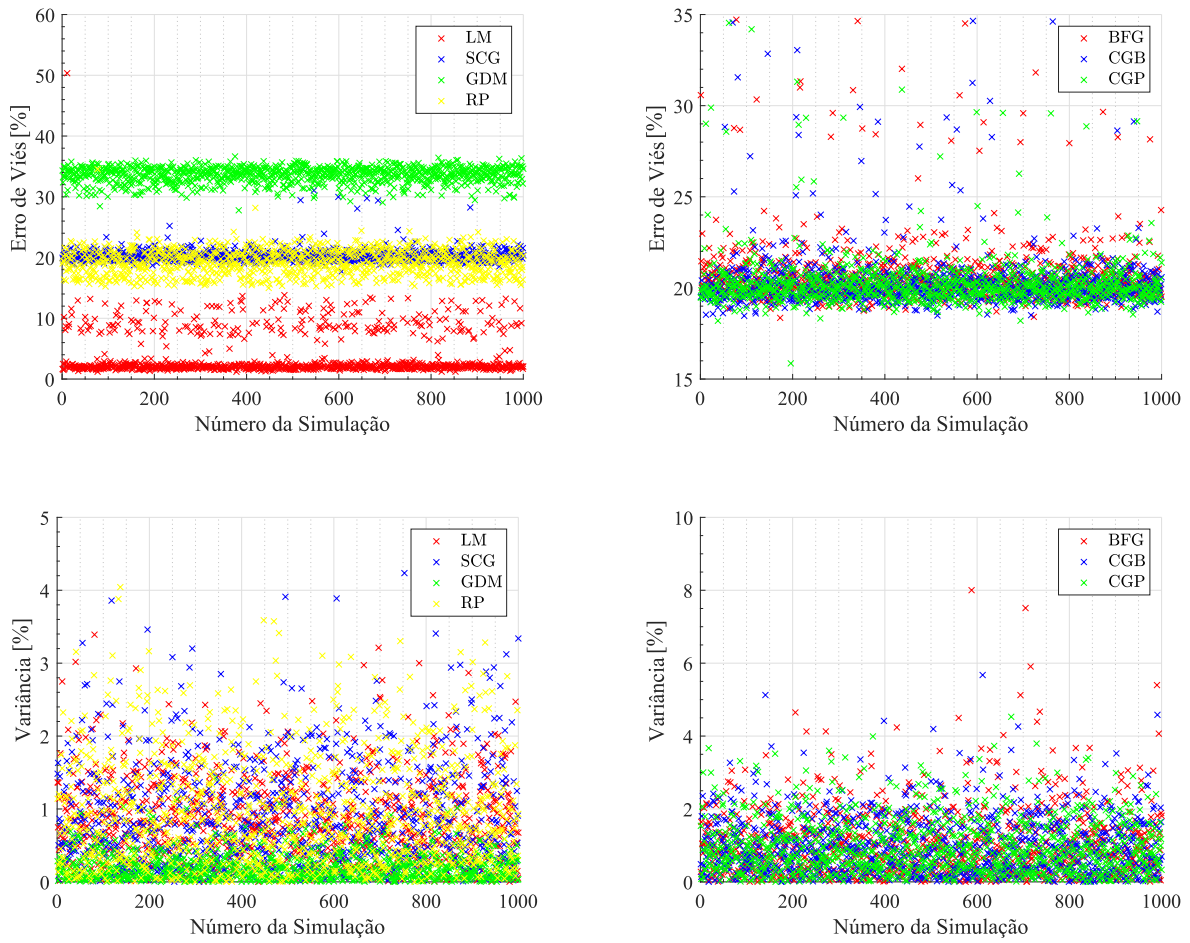
Todas as simulações realizadas com os algoritmos BFG, CGB, CGP, SCG e RP apresentaram valores A_{cc} e F_1 entre 51% e 90%. Entretanto, em mais de 87% delas foi encontrado um índice $R_{ec} > 91\%$. As redes treinadas com LM possuem a maior quantidade de simulações com todas as métricas acima de 91%. Apenas 10.1% das simulações realizadas com o LM tiveram $F_1 < 90\%$. Os erros de viés nas simulações com GDM foram superiores a 25% e as variâncias menores que 1%, indicando *undefitting*. O LM apresentou a menor porcentagem de simulações com erros de vieses menores que 10%.

Figura 53 – Métricas do estudo comparativo RNA1 (a) A_{cc} (b) E_{sp} (c) F_1 (d) R_{ec}



Fonte – Pedro H. Camargos, 2021

Figura 54 – Variância e erro de viés RNA1

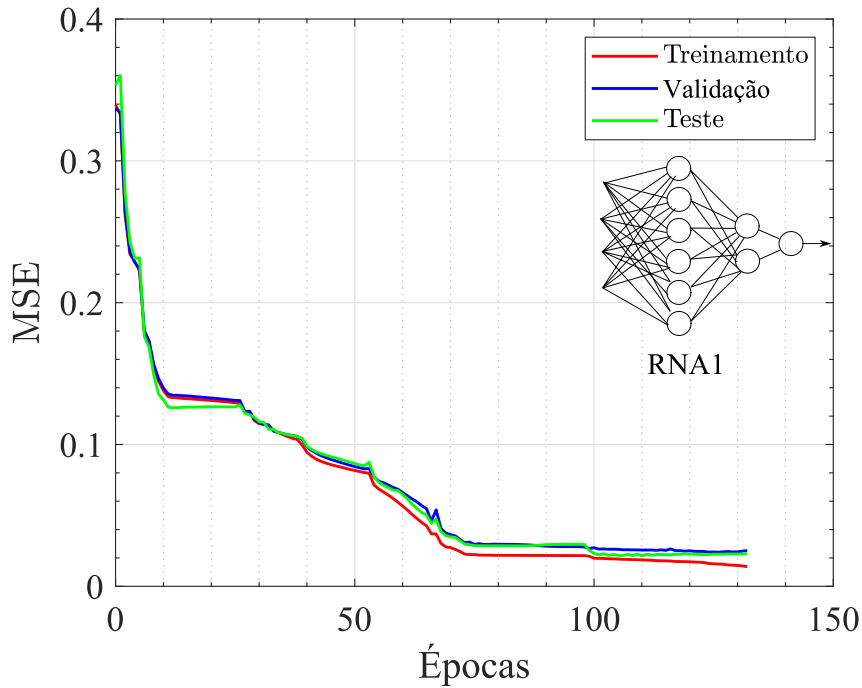


Fonte – Pedro H. Camargos, 2021

10.1.2 Avaliação de Desempenho

O algoritmo de treinamento/validação/teste da RNA1 foi configurado com a técnica de “Parada Antecipada”, PA, que melhora a generalização das redes neurais artificiais. O erro do conjunto de validação foi monitorado durante o processo de treinamento. Quando ocorre aumento significativo neste erro, a PA interrompe o treinamento/validação/teste após 6 iterações. Finalizado este processo, obteve-se os gráficos do MSE em função do número de épocas, Figura 55.

Figura 55 – Curvas de desempenho RNA1



Fonte – Pedro H. Camargos, 2021

As curvas de desempenho possuem comportamentos semelhantes e mostram que os erros de treinamento, validação e teste da RNA1 diminuíram até a época 132. Não houve crescimento significativo da curva de teste antes do aumento das curvas de validação e treinamento. O classificador RNA1 foi projetado com baixo valor de variância e erro de viés. Obteve-se $N_{treq} > N_{uw}$, com $N_{treq_{RNA1}} = 3400$ e $N_{uw_{RNA1}} = 44$ para RNA1. Os erros de treinamento, validação, teste são mostrados na Tabela 11. O erro total, considerando as 3400 entradas, foi igual a 1.95%.

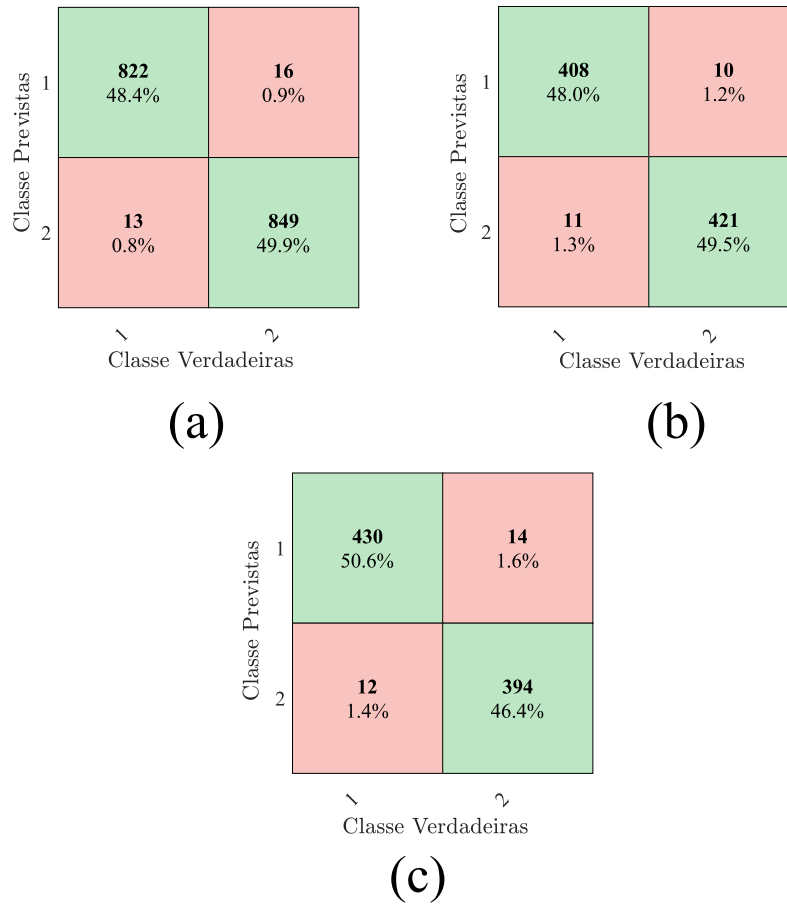
Tabela 11 – Desempenho RNA1

Erro [%]	Treinamento	Teste	Validação	Total	Erro de Viés [%]	Variância [%]
RNA1	1.56	2.26	2.40	1.95	1.56	0.69

Fonte – Pedro H. Camargos, 2021

As matrizes de confusão do conjunto treinamento/validação/teste são mostradas na Figura 56. Na etapa de treinamento, a RNA1 classificou 822 motores de indução operando na região A. Isso corresponde a 48.4% de todas as 1700 máquinas. Da mesma forma, 849 máquinas foram corretamente classificadas como trabalhando na região B, correspondendo a 49.9%. Raciocínio análogo pode ser feito nas análises das matrizes de confusão das etapas de validação e teste. Em todas as etapas, as classificações incorretas corresponderam a menos de 2%. As métricas se encontram na Tabela 12. Todas as métricas obtidas foram maiores que 97%.

Figura 56 – Matrizes de confusão com 1 correspondendo a região A e 2 equivalente à região B (a) etapa treinamento (b) etapa validação (c) etapa teste



Fonte – Pedro H. Camargos, 2021

Tabela 12 – Métricas RNA1

Métrica	Treinamento	Validação	Teste
F_1 [%]	98.26	97.49	97.06
E_{sp} [%]	98.49	97.04	97.45
R_{ec} [%]	98.09	97.60	96.84
A_{cc} [%]	98.29	97.53	96.94

Fonte – Pedro H. Camargos 2021

10.1.3 Limitações SDRO IEC 60034-1

Os resultados dos ensaios comparativos, Tabela 8, são mostrados na Tabela 13. Houve discrepâncias nas respostas IV e IX, indicando que o sistema de diagnóstico possui limitações.

Tabela 13 – Resultados, comparação ensaios e SDRO IEC 60034-1

Nº Ensaio	Parâmetro				Região	
	V_{ab}	I_a	P_{in}	W_m	Ensaio	SDRO IEC 60034-1
I	0.99	1.00	0.92	1.00	A	A
II	1.05	0.96	0.89	0.97	B	B
III	0.96	1.02	0.92	1.00	A	A
IV	0.89	1.10	0.90	1.00	B	A
V	0.93	1.07	0.91	1.02	B	B
VI	1.09	0.949	0.92	1.03	B	B
VII	1.01	1.007	0.84	1.02	A	A
VIII	0.98	0.985	0.80	0.97	B	B
IX	0.95	1.04	0.84	1.01	A	B

Fonte – Pedro H. Camargos 2021

A rede funcionará com alta taxa de confiabilidade apenas se forem fornecidas entradas pertencentes a intervalos pré-definidos. Os valores mínimos e máximos nesses intervalos se encontram na Tabela 14. Os dados de entrada dos conjuntos treinamento/validação/teste pertencem a estes intervalos, fornecendo limitações a RNA1. As bases adotadas foram: tensão de linha, corrente de linha, velocidade de rotação e potência de entrada, todos em seus respectivos valores nominais. Nos ensaios IV e IX os parâmetros I_a e P_{in} estão fora desses intervalos. As tensões nas três fases também devem ter o mesmo módulo para garantir a precisão do SDRO IEC 60034-1.

Tabela 14 – Valores máximos e mínimos das entradas da RNA1

Parâmetro	RNA1	
	Mínimo [pu]	Máximo [pu]
V_{ab}	0.90	1.10
I_a	0.93	1.08
W_m	0.95	1.04
P_{in}	0.86	1.01

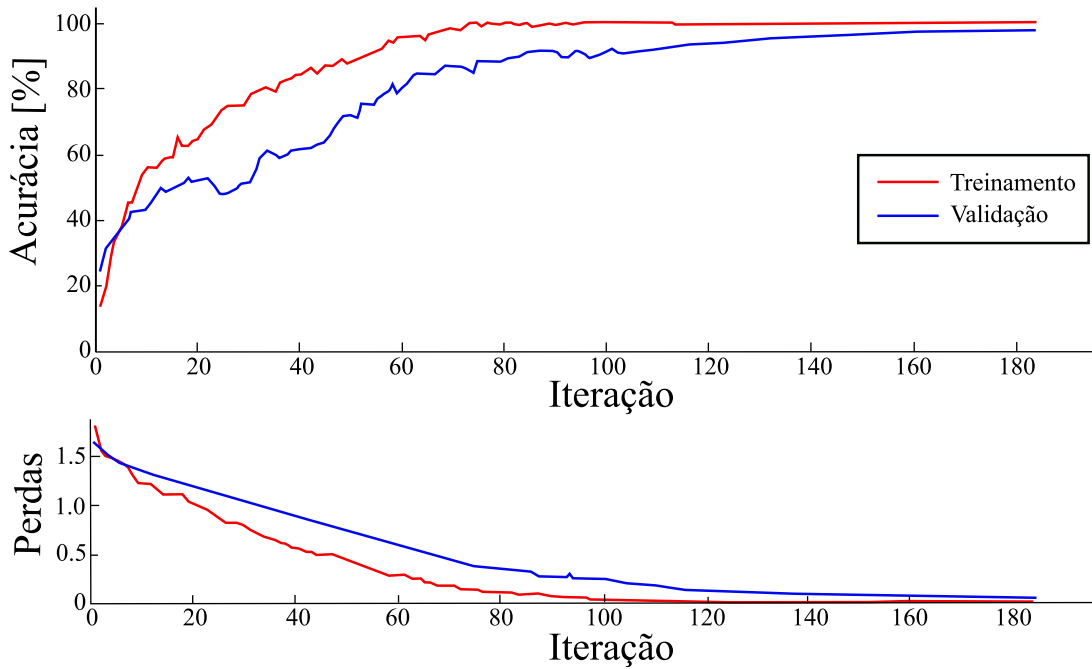
Fonte – Pedro H. Camargos 2021

10.2 Sistema de Diagnóstico de Barra Quebrada do Rotor ISO 20816

O algoritmo de treinamento/validação da LSTM utilizou a técnica de regularização de "abandono", também conhecida como "Dropout". Este é um método que melhora a generalização

das redes neurais profundas. Descarta-se os nós aleatoriamente (com suas conexões) temporariamente da rede durante o treinamento (BROWNLIE, 2018). Finalizado o procedimento de treinamento/validação, obteve-se os gráficos da acurácia e das perdas em função do número de iterações, mostrados na Figura 57.

Figura 57 – Curvas de desempenho LSTM



Fonte – Pedro H. Camargos, 2021

As curvas de desempenho possuem comportamentos semelhantes e mostram que os erros de treinamento, validação e teste da rede neural recorrente diminuíram até a iteração 180. O classificador LSTM foi projetado com baixo valor de variância e erro de viés atingindo um alto índice de precisão ao final do processo de treinamento. Os erros de treinamento, validação, teste são mostrados na Tabela 15. O erro total encontrado, considerando todo o conjunto treinamento/validação/teste, foi de 0.007 %.

Tabela 15 – Desempenho LSTM

Erro [%]	Treinamento	Teste	Validação	Total	Erro de Viés [%]	Variância [%]
LSTM	0.00	0.020	0.010	0.007	0.00	1.00

Fonte – Pedro H. Camargos, 2021

As matrizes de confusão do conjunto treinamento/validação/teste são mostradas na Figura 58. As células diagonais mostram o número e a porcentagem de classificações corretas. A categoria 0 indica que o motor está saudável. As classes 1, 2, 3 e 4 correspondem a uma, duas, três e quatro barras danificadas, respectivamente.

Todas as amostras do conjunto treinamento foram classificadas corretamente com 100 % de acurácia, como indicado na Tabela 16. Na etapa de validação, dois motores com defeito em apenas uma barra foram incorretamente categorizados nas classes 0 e 4. A menor métrica encontrada foi a revocação, indicando que somente 87.50 % dos motores com uma barra danificada foram classificados corretamente. Na testagem da rede, a classe 1 foi a única que apresentou uma métrica abaixo de 97%.

Figura 58 – Matriz de confusão LSTM conjunto (a) treinamento (b) validação (c) teste



Fonte – Pedro H. Camargos, 2021

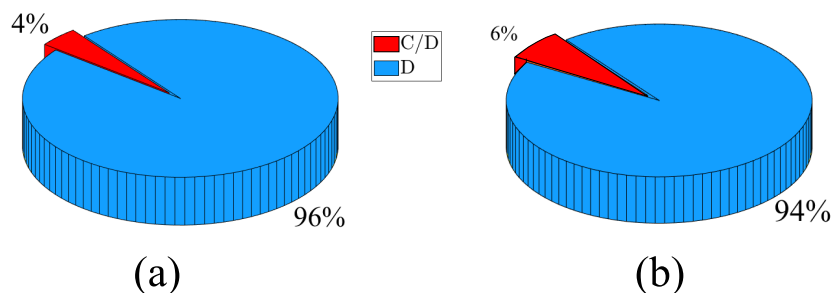
Tabela 16 – Métricas classificação LSTM

Treinamento					
Classe/Nº Barras Quebradas	0	1	2	3	4
F_1 [%]	100.00	100.00	100.00	100.00	100.00
E_{sp} [%]	100.00	100.00	100.00	100.00	100.00
R_{ec} [%]	100.00	100.00	100.00	100.00	100.00
A_{cc} [%]			100.00		
Validação					
Classe/Nº Barras Quebradas	0	1	2	3	4
F_1 [%]	98.11	93.33	100.00	100.00	97.44
E_{sp} [%]	98.65	100.00	100.00	100.00	98.77
R_{ec} [%]	100.00	87.50	100.00	100.00	100.00
A_{cc} [%]			98.00		
Teste					
Classe/Nº Barras Quebradas	0	1	2	3	4
F_1 [%]	100.00	97.30	100.00	100.00	97.78
E_{sp} [%]	100.00	100.00	100.00	100.00	98.72
R_{ec} [%]	100.00	94.74	100.00	100.00	100.00
A_{cc} [%]			99.00		

Fonte – Pedro H. Camargos, 2021

Os gráficos da Figura 59 mostram as porcentagens dos resultados do algoritmo ISO 20816. Cerca de 96 % dos sinais coletados no ponto P com máquina com rotor danificado foram categorizados na região D. Apenas 6% das velocidades adquiridas pelos acelerômetros em R pertenceram à condição C/D . Este é um indicativo de que caso o motor MIT-V possua alguma barra quebrada, ele deve ser imediatamente desligado.

Figura 59 – Distribuição da porcentagem dos sinais de vibração ISO 20816 (a) ponto P (b) ponto R



Fonte – Pedro H. Camargos, 2021

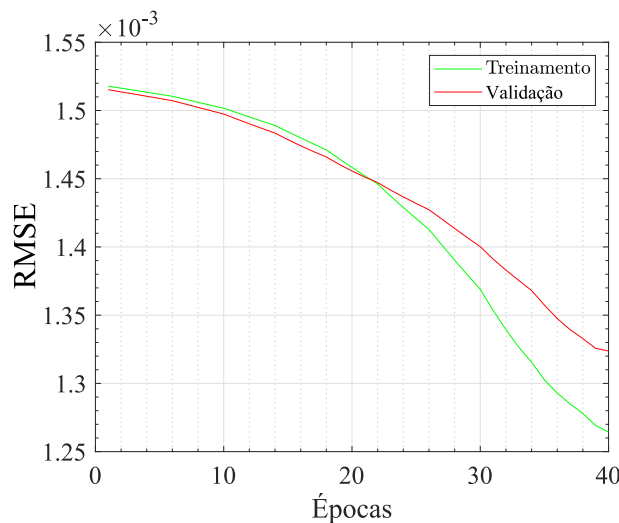
10.2.1 Limitações SDBQ ISO 20816

O SDBQ ISO 20816 possui a capacidade de detectar falhas nas barras do rotor apenas quando o MIT-V opera com torques constantes de $0.5 Nm$, $1.0 Nm$, $1.5 Nm$, $2.0 Nm$, $2.5 Nm$, $3.0 Nm$, $3.5 Nm$ e $4.0 Nm$ em regime permanente. Os dados de entrada da LSTM devem ser os sinais de vibração coletados nos pontos P , Q e R (veja Figura 43) e devem corresponder à velocidade de vibração que foi amostrado a cada 1.17×10^{-4} segundos. O algoritmo que compõe o SDBQ ISO 20818 se aplica apenas ao MIT-V. Uma generalização deste sistema poderá ser feita por meio da inserção de novos dados no conjunto treinamento/validação/teste, possibilitando o SDBQ ISO 20816 ser aplicado para outros motores de indução. Novos dados poderiam ser obtidos através de experimentos laboratoriais com outras máquinas ou através da modelagem com elementos finitos, onde se coletaria os sinais de vibração.

10.3 Sistema de Diagnóstico Fuzzy IEC 60034-26

Quarenta épocas foram contabilizadas durante a etapa de treinamento/validação do ANFIS1. A evolução da raiz quadrada média do erro (RMSE) é mostrada na Figura 60. Foram encontrados erros de treinamento e validação menores que 0.015.

Figura 60 – RMSE treinamento/validação ANFIS1

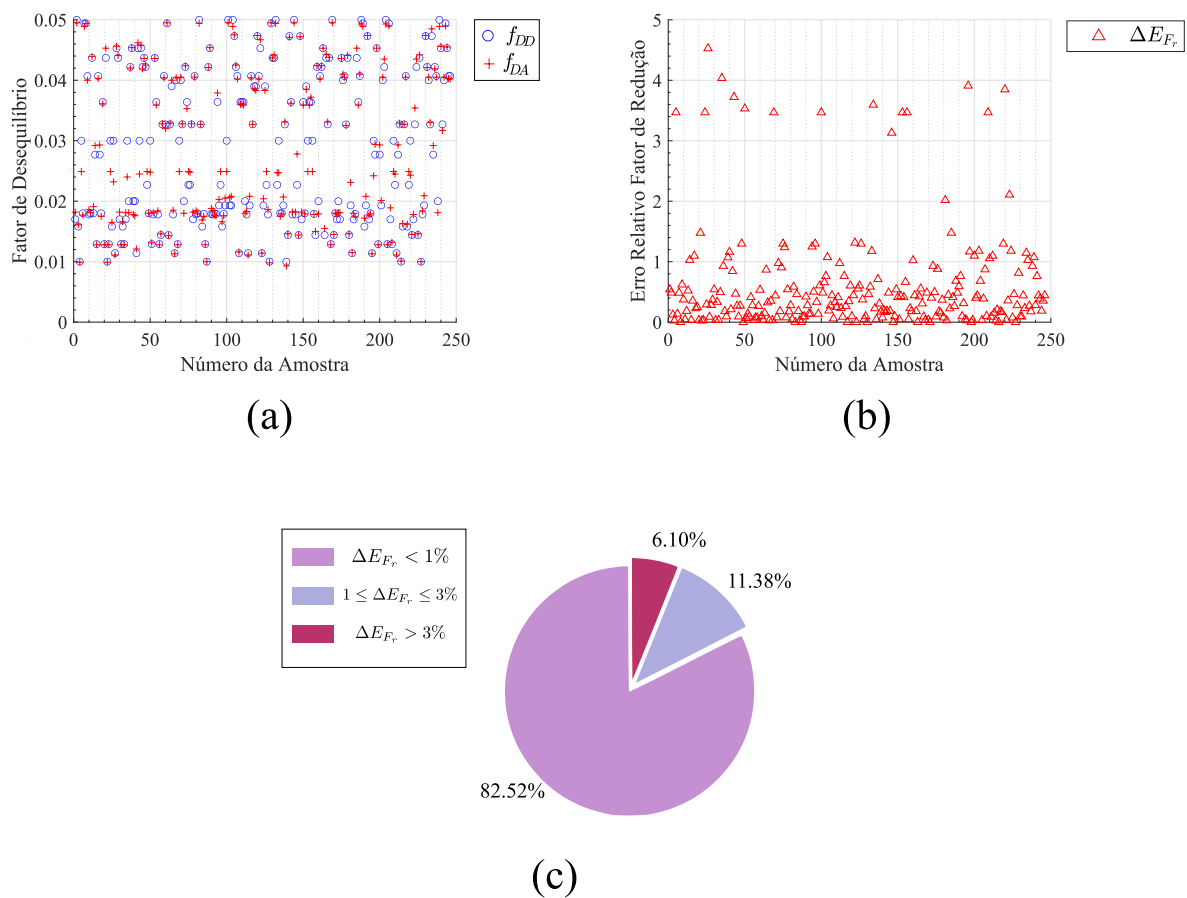


Fonte – Pedro H. Camargos, 2021

Os fatores de desequilíbrio utilizados no teste de precisão podem ser vistos na Figura 61-a. Para cada amostra do Fator de Desequilíbrio f_{DA} , se detectou F_r através da função $F_r(f_D)$, Equação (136), (para mais detalhes ver Seção 9.3.2). Os erros relativos, ΔE_{F_r} , entre

F_{rD} e F_{rA} , foram calculados, sendo mostrados na Figura 61-b. Aproximadamente 82.52 % dos erros estiveram abaixo de 1% (Figura 61-c). Apenas 6.10% das amostras estiveram com erros superiores a 3%. Cabe ressaltar que não foi encontrado no teste de precisão nenhum erro acima de 5%. Portanto, o SDF 60034-26 possui uma elevada precisão no diagnóstico dos fatores de redução.

Figura 61 – Resultados teste de precisão SDF 60034-26



Fonte – Pedro H. Camargos, 2021

10.3.1 Limitações SDF IEC 60034-26

O SDF IEC 60034-26 se limita a calcular fatores de redução de máquinas operando com $f_D < 5\%$. Uma parcela considerável do conjunto de dados de treinamento/validação da rede neural ANFIS1 é oriunda da previsão realizada pela RNA2. Portanto, para aprimorar a precisão e a confiabilidade do SDF IEC 60034-26, estes dados deveriam ser substituídos por dados coletados em ensaios laboratoriais. O SDF IEC 60034-26 foi projetado especificamente

para diagnosticar distúrbios no motor de indução trifásico ATLAS Y225 M descrito por [Adekitan, Samuel e Amuta \(2019\)](#).

11 Conclusões Gerais e Trabalhos Futuros

Neste trabalho, foram propostas metodologias inteligentes de identificação e diagnóstico de falhas em motores de indução trifásicos, baseadas nas normativas internacionais IEC 60034 e ISO 20816. A metodologia de obtenção dos parâmetros do circuito equivalente para a elaboração do SDRO IEC 60034-1 foi crucial na etapa de levantamento de dados. A validação através de simulações computacionais foi fundamental para garantir que o sistema operasse com alta precisão, classificando corretamente as zonas de operação.

O SDBQ ISO 20816 foi projetado utilizando dados experimentais. A precisão e confiabilidade dos resultados fornecidos pela LSTM garante fidelidade na elaboração da metodologia de detecção de falhas. A utilização das redes neurais recorrentes garantiu que dados sequenciais, sinais de vibração, fossem usados no desenvolvimento do SDBQ ISO 20816.

O SDF IEC 60034-26 combina duas técnicas: redes neurais artificiais e lógica *fuzzy*. O SDF IEC 60034-26 monitora o desequilíbrio de tensão por meio do Fator de Desequilíbrio e calcula o Fator de Redução. Sua metodologia poderá ser estendida para outros motores de indução com categoria N. Método pioneiro que poderia ser utilizado juntamente com outras técnicas preditivas no diagnóstico precoce de desequilíbrio de tensão e falhas em motores em aplicações indústrias.

Todas as metodologias com seus respectivos algoritmos descritos neste trabalho foram elaboradas com o objetivo de unir as aplicações práticas (empíricas), traduzidas na forma das normativas IEC 60034-1, IEC 60034-26, ISO 20816, com as técnicas de aprendizado de máquina, amplamente difundidas no universo acadêmico. Classificar e prever automaticamente saídas com excelente precisão são ações que podem e devem adentrar o universo dos sistemas de proteção, sobretudo aqueles vinculados aos motores de indução. Portanto os algoritmos dos sistemas de diagnóstico inteligente podem dar suporte a outros pesquisadores que desejam utilizar um ramo da inteligência artificial para desenvolver sistemas inteligentes de proteção e manutenção de máquinas elétricas.

Nesse contexto, pode-se pontuar que todos os objetivos específicos da pesquisa foram atingidos, ou seja:

- Desenvolveram-se estratégias baseadas nas técnicas de aprendizado de máquina para auxiliar na identificação de falhas em motores de indução trifásicos;

- Utilizou-se das diretrizes e recomendações descritas nas normativas internacionais IEC 60034 e ISO 20816 no projeto e desenvolvimento dos sistemas de diagnósticos inteligentes;

Esses objetivos específicos culminaram em um estudo e investigação das ferramentas de aprendizado profundo, tais como redes neurais *feedforward* e redes neurais recorrentes e suas métricas de avaliação. Os bancos de dados usados nesta pesquisa foram obtidos através de simulações computacionais e em laboratório, ou seja, ambientes controlados. Portanto, podem existir divergências caso se efetue medições das grandezas (tensões, correntes, potência de entrada, sinais de vibração e velocidade de rotação) em um ambiente industrial. Diante deste aspecto, aplicando as mesmas metodologias descritas na Seção 9, podem ser obtidas métricas de classificação com valores inferiores aos encontrados pelo pesquisador que realizou este trabalho.

11.1 Trabalhos Futuros

Durante o desenvolvimento deste trabalho, novas frentes de investigação surgiram são listadas a seguir:

- Aprofundar a investigação sobre outras categorias de redes neurais artificiais e suas aplicações em sistemas de diagnósticos de falhas.
- Explorar outras normativas, tais como IEC 60034-14: *Rotating electrical machines - Part 14: Mechanical vibration of certain machines with shaft heights 56 mm and higher - Measurement, evaluation and limits of vibration severity* e possíveis adequações de suas diretrizes em sistemas de monitoramento inteligentes de falhas em motores de indução.
- Desenvolver um sistema de diagnóstico de falhas utilizando redes neurais convolucionais e imagens termográficas com o objetivo de diagnosticar sobreaquecimento nos enrolamentos e carcaça do motor.
- Implementação do SDRO IEC 60034-1 em uma placa de circuito impresso para validação experimental e diagnóstico em tempo real.
- Implementação de um sistema de monitoramento que englobe avaliar o MSCA comparando o desempenho de diferentes arquiteturas de redes neurais artificiais.

11.2 Publicações

Durante a realização da pós-graduação, mestrado acadêmico em engenharia elétrica no Instituto de Sistemas Elétricos e Energia (ISEE) os seguintes trabalhos foram aceitos e submetidos em congressos e periódicos:

11.3 Trabalhos Completos Publicados

- PEREIRA, M. A. ; CAMARGOS, P. H. ; FULY, B. I. L. . [Determination of top oil and hot spot temperature of a power transformer using fuzzy logic](#). v. 11, p. 51361-51365, 2021.
- CAMARGOS,P.H; LOPES, B. I.; [Motor Starting Direct-on-line: Performance Analysis in ATP and MATLAB/SIMULINK Environments](#). IEEE 3rd of Global Power, Energy and Communication Conference. 2021.
- CAMARGOS,P.H.; CAETANO,R.E. [A Performance Study of a High-Torque Induction Motor Designed for Light Electric Vehicles Applications](#)..ELECTRICAL ENGINEERING, 2021.
- SANTOS,P.H.J; CAMARGOS,P.H.; CAETANO,R.E.[Analysis of the Losses of a 6 kW Induction Motor Designed for Light Electric Vehicle Applications](#) . 2021 IEEE International Conference on Environment and Electrical Engineering, 2021.
- SILVA, R.F.; CAMARGOS,P.H. [Avaliação da Impossibilidade de Atendimento dos Requisitos Técnicos Mínimos para Conexão de Centrais Eólicas: Fator de Potência Vs Magnitude de Tensão](#) . XIV Conferência Brasileira sobre Qualidade de Energia Elétrica, 2021.
- LUZ, C.M.A.; CAMARGOS, P.H.; RODRIGUES, W.B.; SILVA, A.; RIBEIRO,E.R..[Conversor SEPIC-Buck Modificado com Ampla Atenuação de Tensão e Maior Ciclo de Trabalho](#). Congresso Brasileiro de Automática 2020, 2020,Congresso Virtual. Anais do XXIII Congresso Brasileiro de Automática, 2020. v. 2.

11.4 Artigos Submetidos para Publicação

- CAMARGOS,P.H; LOPES, B. I.; . [A Monitoring Induction Motor System Based on Pattern Recognition Neural Network: A IEC 60034-1 Approach](#). IEEE Uruguay Conference. 2021.

- CAMARGOS,P.H.;CAETANO,R.E. Determining the Life Cycle Cost and Reliability of an Induction Motor Designed for Light Vehicle Applications: A Comparative Analysis..ELECTRICAL ENGINEERING, 2021.
- CAMARGOS,P.H.;RIBEIRO,F.P.;BELCHIOR, N.F;CARVALHO,C.T Time-Varying Harmonic Distortions in AC Drives.ELECTRICAL ENGINEERING, 2021.

11.5 *Informações Extras*

Os algoritmos de simulação juntamente com os arquivos que contêm todos os bancos de dados utilizados na elaboração desta pesquisa estão disponíveis. Para baixá-los, basta [clicar aqui](#) (*link via Google Drive*). Para mais informações e possíveis parcerias em pesquisas futuras, sinta-se a vontade para entrar em contato.

- Pedro Henrique Camargos
- E-mail: pedro.camargos.eng@gmail.com
- Currículo Lattes: <http://lattes.cnpq.br/3321340146708240>
- LinkedIn: <https://www.linkedin.com/in/camargosph/>

Referências

- ABNT. NBR 17094: Máquinas elétricas girantes - Motores de indução Parte 1: Trifásicos. *Associação Brasileira de Normas Técnicas*, p. 26–35, 2008. Citado 4 vezes nas páginas 79, 80, 83 e 84.
- ADEKITAN, A. I.; SAMUEL, I.; AMUTA, E. Dataset on the performance of a three phase induction motor under balanced and unbalanced supply voltage conditions. *Data in brief*, Elsevier, v. 24, 2019. Citado 4 vezes nas páginas 85, 86, 101 e 116.
- ALVES, D. A. Técnicas de detecção de falhas em barras do rotor nos motores de indução trifásicos. *Universidade Federal de São João del-Rei*, 2017. Citado 2 vezes nas páginas 70 e 71.
- AMARAL, G. F. V.; BACCARINI, J. M. R.; COELHO, F. C. R.; BACCARINI, L. M. R. A high precision method for induction machine parameters estimation from manufacturer data. *IEEE Transactions on Energy Conversion*, IEEE, 2020. Citado 4 vezes nas páginas 75, 76, 77 e 78.
- ARKAN, M.; KOSTIC-PEROVIC, D.; UNSWORTH, P. Modelling and simulation of induction motors with inter-turn faults for diagnostics. *Electric Power Systems Research*, Elsevier, v. 75, n. 1, p. 57–66, 2005. Citado na página 75.
- BAGHIRLI, O. *Comparison of Lavenberg-Marquardt, scaled conjugate gradient and Bayesian regularization backpropagation algorithms for multistep ahead wind speed forecasting using multilayer perceptron feedforward neural network*. 2015. Citado na página 42.
- BHATTACHARJEE, J. *Practical Machine Learning with Rust: Creating Intelligent Applications in Rust*. [S.l.]: Apress, 2019. Citado na página 29.
- BORTONI, E. C.; JR, J. V. B.; SILVA, P. V. da; FARIA, V. A.; VIEIRA, P. A. Evaluation of manufacturers strategies to obtain high-efficient induction motors. *Sustainable Energy Technologies and Assessments*, Elsevier, v. 31, p. 221–227, 2019. Citado na página 75.
- BROWNLEE, J. *A Gentle Introduction to Dropout for Regularizing Deep Neural Networks*. 2018. Disponível em: <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>. Citado na página 111.
- BROYDEN, C. G. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, Oxford University Press, v. 6, n. 1, p. 76–90, 1970. Citado na página 41.
- CHOW, M.-Y.; GOODE, P. The advantages and challenges of machine fault detection using artificial neural network and fuzzy logic technologies. In: IEEE. *Proceedings of 36th Midwest Symposium on Circuits and Systems*. [S.l.], 1993. p. 708–711. Citado na página 23.
- CHOW, M.-Y.; MANGUM, P. M.; YEE, S. O. A neural network approach to real-time condition monitoring of induction motors. *IEEE Transactions on industrial electronics*, IEEE, v. 38, n. 6, p. 448–453, 1991. Citado na página 23.
- CHOW, M.-y.; YEE, S. O. Methodology for on-line incipient fault detection in single-phase squirrel-cage induction motors using artificial neural networks. *IEEE Transactions on Energy Conversion*, IEEE, v. 6, n. 3, p. 536–545, 1991. Citado na página 23.
- COPELAND, B. J. *The essential turing*. [S.l.]: Clarendon Press, 2004. Citado na página 28.

- CORPORATION, N. *Vibration Control Definitions of Characteristics*. 2021. Disponível em: <https://www.newport.com/t/vibration-control-definitions-of-characteristics>. Citado na página 69.
- CRUZ-VEGA, I.; RANGEL-MAGDALENO, J.; RAMIREZ-CORTES, J.; PEREGRINA-BARRETO, H. Automatic progressive damage detection of rotor bar in induction motor using vibration analysis and multiple classifiers. *Journal of Mechanical Science and Technology*, Springer, v. 31, n. 6, p. 2651–2662, 2017. Citado na página 71.
- DAI, Y.; YUAN, Y. Convergence properties of beale-powell restart algorithm. *Science in China Series A: Mathematics*, Springer, v. 41, n. 11, p. 1142–1150, 1998. Citado na página 42.
- DICE, L. R. Measures of the amount of ecologic association between species. *Ecology*, Wiley Online Library, v. 26, n. 3, p. 297–302, 1945. Citado na página 60.
- DOMIJAN, A.; YIN, Y. Single Phase Induction Machine Simulation Using The Electromagnetic Transients Program: Theory And Test Cases. *IEEE Transactions on Energy Conversion*, v. 9, n. 3, p. 535–542, 1994. ISSN 15580059. Citado na página 131.
- DOMMEL, H. *Electromagnetic Transients Program. Reference Manual (EMTP Theory Book)*. Portland: Bonneville Power Administration, 1986. Citado na página 134.
- ELSIMARY, H. Implementation of neural network and genetic algorithms for novelty filters for fault detection. In: IEEE. *Proceedings of the 39th Midwest Symposium on Circuits and Systems*. [S.l.], 1996. v. 3, p. 1432–1435. Citado na página 24.
- FITZGERALD, A. E.; KINGSLEY, C.; UMANS, S. D.; JAMES, B. *Electric machinery*. [S.l.]: McGraw-Hill New York, 2003. v. 5. Citado na página 74.
- FLETCHER, R. A new approach to variable metric algorithms. *The computer journal*, Oxford University Press, v. 13, n. 3, p. 317–322, 1970. Citado na página 41.
- GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. [S.l.]: O'Reilly Media, 2019. Citado na página 39.
- GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. Learning to forget: Continual prediction with lstm. *IET*, 1999. Citado na página 58.
- GOLDFARB, D. A family of variable-metric methods derived by variational means. *Mathematics of computation*, v. 24, n. 109, p. 23–26, 1970. Citado na página 41.
- GOODE, P. V.; CHOW, M.-Y. A hybrid fuzzy/neural system used to extract heuristic knowledge from a fault detection problem. In: IEEE. *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*. [S.l.], 1994. p. 1731–1736. Citado na página 23.
- HAGAN, M.; DEMUTH, H.; BEALE, M.; JESUS, O. D. *Neural network design*, 2nd. Ed., Lexington, KY, 2016. Citado na página 41.
- HAGAN, M. T.; MENHAJ, M. B. Training feedforward networks with the marquardt algorithm. *IEEE transactions on Neural Networks*, IEEE, v. 5, n. 6, p. 989–993, 1994. Citado na página 43.
- HAN, T.; YANG, B.-S.; CHOI, W.-H.; KIM, J.-S. Fault diagnosis system of induction motors based on neural network and genetic algorithm using stator current signals. *International Journal of Rotating Machinery*, Hindawi, v. 2006, 2006. Citado na página 24.

- HAO, S.; GE, F.-X.; LI, Y.; JIANG, J. Multisensor bearing fault diagnosis based on one-dimensional convolutional long short-term memory networks. *Measurement*, Elsevier, p. 107802, 2020. Citado na página 24.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 56.
- HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 79, n. 8, p. 2554–2558, 1982. Citado na página 23.
- HWANG, Y. R.; JEN, K. K.; SHEN, Y. T. Application of cepstrum and neural network to bearing fault detection. *Journal of Mechanical Science and Technology*, v. 23, n. 10, p. 2730–2737, 2009. ISSN 1738494X. Citado na página 23.
- HYUN, B.-G.; NAM, K. Faults diagnoses of rotating machines by using neural nets: Grnn and bpn. In: IEEE. *Proceedings of IECON'95-21st Annual Conference on IEEE Industrial Electronics*. [S.l.], 1995. v. 2, p. 1456–1461. Citado na página 23.
- IEC, I. E. C. IEC 60034-26: Rotating electrical machines – Part 26: Effects of unbalanced voltages on the performance of three-phase cage induction motors. 2006. Citado 3 vezes nas páginas 66, 67 e 100.
- IEC, I. E. C. IEC 60034-12: Rotating electrical machines – Part 12: Starting performance of single-speed three-phase cage induction motors. 2016. Citado na página 79.
- IEC, I. E. C. IEC 60034-1: Rating and performance. 2017. Citado 2 vezes nas páginas 65 e 66.
- ISO. *ISO 20816-1: Mechanical vibration — Measurement and evaluation of machine vibration Part 1: General guidelines*. 2016. Citado 3 vezes nas páginas 68, 69 e 98.
- ISO. *ISO 13373-9: Condition monitoring and diagnostics of machines — Vibration condition monitoring — Part 9: Diagnostic techniques for electric motors*. 2017. Citado na página 70.
- JANG, J.-S. Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 23, n. 3, p. 665–685, 1993. Citado 2 vezes nas páginas 62 e 63.
- JOHNSON, S. G. Quasi-newton optimization: Origin of the bfgs update. *Notes for 18.335 at MIT - Massachusetts Institute of Technology*, 2019. Citado na página 41.
- JOSEPHINE, P. K.; PRAKASH, V.; DIVYA, K. *et al.* Supervised learning algorithms: A comparison. *Kristu Jayanti Journal of Computational Sciences (KJCS)*, p. 01–12, 2020. Citado na página 29.
- JOSHI, A. V. *Machine Learning and Artificial Intelligence*. [S.l.]: Springer, 2020. Citado 2 vezes nas páginas 30 e 32.
- JSME. *Revision of vibration-related ISO standards*. 2021. Disponível em: https://www.jsme.or.jp/conference/joutai/doc/kikaku/Alt_ISO10814_10816.pdf. Citado na página 69.
- KERBOUA, A.; METATLA, A.; KELAIAIA, R.; BATOUCHE, M. Real-time safety monitoring in the induction motor using deep hierarchic long short-term memory. *The International Journal of Advanced Manufacturing Technology*, Springer, v. 99, n. 9-12, p. 2245–2255, 2018. Citado na página 24.

- KHIRIRAT, S.; FEYZMAHDAVIAN, H. R.; JOHANSSON, M. Mini-batch gradient descent: Faster convergence under data sparsity. In: IEEE. *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. [S.l.], 2017. p. 2880–2887. Citado na página 39.
- KHOMFOI, S.; TOLBERT, L. M. Fault diagnostic system for a multilevel inverter using a neural network. *IEEE Transactions on Power Electronics*, IEEE, v. 22, n. 3, p. 1062–1069, 2007. Citado na página 24.
- KIM, K.; PARLOS, A. G. Induction motor fault diagnosis based on neuropredictors and wavelet signal processing. *IEEE/ASME Transactions on mechatronics*, IEEE, v. 7, n. 2, p. 201–219, 2002. Citado na página 24.
- KULKARNI, A. J.; SATAPATHY, S. C. *Optimization in Machine Learning and Applications*. [S.l.]: Springer, 2020. Citado na página 33.
- KUMAR, I.; DOGRA, K.; UTREJA, C.; YADAV, P. A comparative study of supervised machine learning algorithms for stock market trend prediction. In: IEEE. *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. [S.l.], 2018. p. 1003–1007. Citado na página 29.
- LANE, M.; SHAEBOUB, A.; GU, F.; BALL, A. Investigation of reductions in motor efficiency and power factor caused by stator faults when operated from an inverter drive under open loop and sensorless vector modes. *Systems Science & Control Engineering*, Taylor & Francis, v. 5, n. 1, p. 361–379, 2017. Citado na página 67.
- LAUW, H. K.; MEYER, W. S. Universal Machine Modeling for the Representation of Rotating Electric Machinery in an Electromagnetic Transients Program. *IEEE Transactions on Power Apparatus and Systems*, PAS-101, n. 6, p. 1342–1351, 1982. Citado na página 131.
- LEE, J.-H.; PACK, J.-H.; LEE, I.-S. Fault diagnosis of induction motor using convolutional neural network. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 9, n. 15, p. 2950, 2019. Citado na página 25.
- LEVER, J.; KRZYWINSKI, M.; ALTMAN, N. *Logistic regression*. [S.l.]: Nature Publishing Group, 2016. Citado na página 46.
- LIANG, B.; IWNICKI, S.; ZHAO, Y. Application of power spectrum, cepstrum, higher order spectrum and neural network analyses for induction motor fault diagnosis. *Mechanical Systems and Signal Processing*, Elsevier, v. 39, n. 1-2, p. 342–360, 2013. Citado na página 24.
- LIBONI, L. H. B. *Diagnóstico de falhas em motores de indução trifásicos baseado em decomposição em componentes ortogonais e aprendizagem de máquinas*. Tese (Doutorado) — Universidade de São Paulo, 2017. Citado na página 85.
- LIU, Y.; GAO, Y.; YIN, W. An improved analysis of stochastic gradient descent with momentum. *Computer Science, Mathematics - ArXiv*, 2020. Citado na página 39.
- LU, B.; HABETLER, T. G.; HARLEY, R. G. A survey of efficiency-estimation methods for in-service induction motors. *IEEE Transactions on Industry Applications*, IEEE, v. 42, n. 4, p. 924–933, 2006. Citado na página 75.
- MAAS, A. L.; HANNUN, A. Y.; NG, A. Y. Rectifier nonlinearities improve neural network acoustic models. In: CITESEER. *Proc. icml*. [S.l.], 2013. v. 30, n. 1, p. 3. Citado na página 47.

- MARTINEZ-VELASCO, J. A. *Transient Analysis of Power Systems*. [S.l.: s.n.], 2020. ISBN 9781119480532. Citado na página 133.
- MASRUR, M. A.; CHEN, Z.; ZHANG, B.; MURPHEY, Y. L. Model-based fault diagnosis in electric drive inverters using artificial neural network. In: IEEE. *2007 IEEE Power Engineering Society General Meeting*. [S.l.], 2007. p. 1–7. Citado na página 24.
- MATHWORKS. Improve shallow neural network generalization and avoid overfitting. 2020. Disponível em: <https://www.mathworks.com/help/deeplearning/ug/improve-neural-network-generalization-and-avoid-overfitting.html>. Citado na página 92.
- MATHWORKS. Patternnet. 2020. Disponível em: <https://www.mathworks.com/help/deeplearning/ref/patternnet.html>. Citado 2 vezes nas páginas 93 e 94.
- MATHWORKS. Polyfit. 2020. Disponível em: <https://www.mathworks.com/help/matlab/ref/polyfit.html>. Citado na página 103.
- MATHWORKS. Scaled conjugate gradient backpropagation. 2020. Disponível em: <https://www.mathworks.com/help/deeplearning/ref/trainscg.html>. Citado na página 42.
- MATHWORKS. Simscape Electrical User 's Guide (Specialized Power Systems). 2020. Disponível em: https://www.mathworks.com/help/releases/R2019b/pdf{_}doc/phymod/sps/powersys. Citado na página 72.
- MATHWORKS. Asynchronous Machine. 2021. Disponível em: <https://www.mathworks.com/help/phymod/sps/powersys/ref/asynchronousmachine.html>. Citado 3 vezes nas páginas 73, 74 e 94.
- MATHWORKS. classificationLayer. 2021. Disponível em: <https://www.mathworks.com/help/deeplearning/ref/classificationlayer.html>. Citado na página 98.
- MATHWORKS. dropoutLayer. 2021. Disponível em: <https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.dropoutlayer.html;jsessionid=4942085fddd3c413488469952a3d>. Citado na página 98.
- MATHWORKS. *Levenberg-Marquardt backpropagation*. 2021. Disponível em: <https://www.mathworks.com/help/deeplearning/ref/trainlm.html>. Citado na página 43.
- MATHWORKS. *MathWorks*. 2021. Disponível em: <https://www.mathworks.com/>. Citado na página 72.
- MATHWORKS. Neuro-Adaptive Learning and ANFIS. 2021. Disponível em: <https://www.mathworks.com/help/fuzzy/neuro-adaptive-learning-and-anfis.html>. Citado na página 102.
- MATHWORKS. sequenceInputLayer. 2021. Disponível em: <https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.sequenceinputlayer.html>. Citado na página 97.
- MATHWORKS. *TrainingOptionsADAM*. 2021. Disponível em: <https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.trainingoptionsadam.html>. Citado na página 97.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biology*, v. 5, p. 115–133, 1943. Citado 2 vezes nas páginas 23 e 45.

- MEHRJOU, M. R.; MARIUN, N.; KARAMI, M.; MISRON, N.; RADZI, M. A. M. Statistical features analysis of transient current signal for broken bars fault detection in Is-pmsms. In: IEEE. *2015 IEEE 3rd International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*. [S.l.], 2015. p. 1–6. Citado na página 70.
- MEHRJOU, M. R.; MARIUN, N.; MARHABAN, M. H.; MISRON, N. Rotor fault condition monitoring techniques for squirrel-cage induction machine—a review. *Mechanical Systems and Signal Processing*, Elsevier, v. 25, n. 8, p. 2827–2848, 2011. Citado na página 71.
- MEYER, W. ATP Rule Book. p. 4–7, 2002. Disponível em: <http://scholar.google.com/scholar?hl=en{&}btnG=Search{&}q=intitle:ATP+Rule+>. Citado 2 vezes nas páginas 132 e 133.
- MICELI, R.; GRITLI, Y.; TOMMASO, A. D.; FILIPPETTI, F.; ROSSI, C. Vibration signature analysis for monitoring rotor broken bar in double squirrel cage induction motors based on wavelet analysis. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, Emerald Group Publishing Limited, 2014. Citado na página 71.
- MINSKY, M.; PAPERT, S. A. *Perceptrons: An introduction to computational geometry*. [S.l.]: MIT press, 1969. Citado na página 45.
- MIRJALILI, S.; FARIS, H.; ALJARAHAH, I. *Evolutionary Machine Learning Techniques*. [S.l.]: Springer, 2019. Citado na página 29.
- MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997. Citado na página 28.
- MOHANTY, A. R. *Machinery condition monitoring: Principles and practices*. [S.l.]: CRC Press, 2014. Citado na página 70.
- MØLLER, M. F. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, Elsevier, v. 6, n. 4, p. 525–533, 1993. Citado na página 42.
- MUELLER, J. P.; MASSARON, L. *Machine learning for dummies*. [S.l.]: John Wiley & Sons, 2016. Citado 2 vezes nas páginas 31 e 33.
- MUSSO, D. Stochastic gradient descent with random learning rate. *Computer Science, Machine Learning - ArXiv*, 2020. Citado na página 39.
- NATARAJAN, R.; MISRA, V. Parameter estimation of induction motors using a spreadsheet program on a personal computer. *Electric power systems research*, Elsevier, v. 16, n. 2, p. 157–164, 1989. Citado na página 75.
- NEMA. *NEMA Standards Publication ANSI/NEMA MG 1-2003, Revision 1-2004: Motors and Generators*. [S.l.]: National Electrical Manufacturers Association, 2003. Citado na página 67.
- NG, A. Machine learning yearning: Technical strategy for ai engineers in the era of deep learning. 2018. Citado na página 91.
- NG, A. *CS229 Lecture notes*. 2020. Disponível em: <http://cs229.stanford.edu/syllabus.html>. Citado na página 39.
- NG, A.; JORDAN, M. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, v. 14, p. 841–848, 2001. Citado na página 31.

- OHN, I.; KIM, Y. Smooth function approximation by deep neural networks with general activation functions. *Entropy*, Multidisciplinary Digital Publishing Institute, v. 21, n. 7, p. 627, 2019. Citado na página 47.
- PEARSON, K. *On the theory of contingency and its relation to association and normal correlation*. [S.l.]: Dulau and Company, 1904. Citado na página 59.
- PEDAMONTI, D. Comparison of non-linear activation functions for deep neural networks on mnist classification task. *arXiv preprint arXiv:1804.02763*, 2018. Citado na página 47.
- PHIL, K. *Matlab deep learning with machine learning, neural networks and artificial intelligence*. Apress, 2017. Citado na página 45.
- POLAK, E.; RIBIERE, G. Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, v. 3, n. R1, p. 35–43, 1969. Citado na página 42.
- POWELL, M. J. D. Restart procedures for the conjugate gradient method. *Mathematical programming*, Springer, v. 12, n. 1, p. 241–254, 1977. Citado na página 42.
- PUIG-ARNAVAT, M.; BRUNO, J. C. Artificial neural networks for thermochemical conversion of biomass. In: *Recent advances in thermo-chemical conversion of biomass*. [S.l.]: Elsevier, 2015. p. 133–156. Citado na página 49.
- RANGEL-MAGDALENO, J. de J.; ROMERO-TRONCOSO, R. de J.; OSORNIO-RIOS, R. A.; CABAL-YEPEZ, E.; CONTRERAS-MEDINA, L. M. Novel methodology for online half-broken-bar detection on induction motors. *IEEE Transactions on Instrumentation and Measurement*, IEEE, v. 58, n. 5, p. 1690–1698, 2009. Citado na página 71.
- RAO, B. K.; Srinivasa Pai, P.; NAGABHUSHANA, T. N. Failure diagnosis and prognosis of rolling - Element bearings using artificial neural networks: A critical overview. *Journal of Physics: Conference Series*, v. 364, n. 1, p. 0–28, 2012. ISSN 17426596. Citado na página 23.
- RASAMOELINA, A. D.; ADJAILIA, F.; SINČÁK, P. A review of activation function for artificial neural network. In: IEEE. *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMII)*. [S.l.], 2020. p. 281–286. Citado na página 45.
- RESEARCH, V. *What is the PSD?* 2021. Disponível em: <https://vru.vibrationresearch.com/lesson/what-is-the-psd/>. Citado na página 69.
- RIEDMILLER, M.; BRAUN, H. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: IEEE. *IEEE international conference on neural networks*. [S.l.], 1993. p. 586–591. Citado na página 43.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986. Citado na página 23.
- SAMANTA, B.; AL-BALUSHI, K. Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. *Mechanical systems and signal processing*, Elsevier, v. 17, n. 2, p. 317–328, 2003. Citado na página 24.

- SHAKIBA, F. M.; ZHOU, M. Novel analog implementation of a hyperbolic tangent neuron in artificial neural networks. *IEEE Transactions on Industrial Electronics*, IEEE, 2020. Citado na página 46.
- SHANNO, D. F. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, v. 24, n. 111, p. 647–656, 1970. Citado na página 41.
- SIMÕES, M. G.; SHAW, I. S. *Controle e modelagem fuzzy*. [S.l.]: Editora Blucher, 2007. Citado na página 62.
- SINGH, A.; GRANT, B.; DEFOUR, R.; SHARMA, C.; BAHADOORSINGH, S. A review of induction motor fault modeling. *Electric Power Systems Research*, Elsevier, v. 133, p. 191–197, 2016. Citado na página 75.
- SINGH, A. M. P. *Learn tensorflow 2.0 - Implement Machine Learning and Deep Learning Models with Python*. [S.l.]: Apress, 2020. Citado na página 29.
- SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Citado na página 98.
- STEWART, J. *Cálculo volume 2*. [S.l.]: Cengage Learning, 2013. Citado na página 37.
- SUETAKE, M. *Sistemas inteligentes para monitoramento e diagnósticos de falhas em motores de indução trifásicos*. Tese (Doutorado) — Universidade de São Paulo, 2012. Citado na página 85.
- THARWAT, A. Classification assessment methods. *Applied Computing and Informatics*, Emerald Publishing Limited, 2020. Citado na página 60.
- TOMA, R. N.; PROSVIRIN, A. E.; KIM, J.-M. Bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 20, n. 7, p. 1884, 2020. Citado na página 25.
- TREML, A.; FLAUZINO, R.; SUETAKE, M.; MACIEJEWSKI, N. A. R. *Experimental database for detecting and diagnosing rotor broken bar in a three-phase induction motor*. IEEE Dataport, 2020. Disponível em: <https://dx.doi.org/10.21227/fmnm-bn95>. Citado na página 85.
- TROUT, J. *Vibration Analysis Explained*. 2021. Disponível em: <https://www.reliableplant.com/vibration-analysis-31569>. Citado na página 69.
- UNAL, M.; DEMETGUL, M.; ONAT, M.; KUCUK, H. Fault Diagnosis of Rolling Bearing Based on Feature Extraction and Neural Network Algorithm 2 Experimental Setup and Test. *Recent Advances in Telecommunications, Signals and Systems*, p. 179–185, 2013. Disponível em: www.wseas.us/e-library/conferences/2013/lemesos. Citado na página 23.
- UNAL, M.; ONAT, M.; DEMETGUL, M.; KUCUK, H. Fault diagnosis of rolling bearings using a genetic algorithm optimized neural network. *Measurement*, Elsevier, v. 58, p. 187–196, 2014. Citado na página 24.
- URSO, A.; FIANNACA, A.; ROSA, M. L.; RAVÌ, V.; RIZZO, R. Data mining: Classification and prediction. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, I, v. 3, p. 384–402, 2018. Citado na página 54.

VERMEULEN, A. F. *Industrial Machine Learning Using Artificial Intelligence as a Transformational Disruptor*. [S.l.]: Apress, 2020. Citado na página 30.

VIBSENS. *Why ISO 10816 Is Important For Vibration Measurement Engineers?* 2020. Disponível em: <http://www.vibsens.com/index.php/knowledge-base/iso10816-iso7919-charts/iso10816-charts>. Citado na página 68.

WANG, J.; ZHU, H.; WANG, S.-H.; ZHANG, Y.-D. A review of deep learning on medical image analysis. *Mobile Networks and Applications*, Springer, p. 1–30, 2020. Citado na página 33.

WANG, Z.; MA, H.; CHEN, H.; YAN, B.; CHU, X. Performance degradation assessment of rolling bearing based on convolutional neural network and deep long-short term memory network. *International Journal of Production Research*, Taylor & Francis, v. 58, n. 13, p. 3931–3943, 2020. Citado na página 24.

WATT, J.; BORHANI, R.; KATSAGGELOS, A. *Machine learning refined: foundations, algorithms, and applications*. [S.l.]: Cambridge University Press, 2020. Citado na página 40.

WEG. WEG Electric Motor Selection. 2021. Disponível em: http://ecatalog.weg.net/tec_cat/tech_motor_sel_web.asp. Citado 2 vezes nas páginas 7 e 82.

WEG. WEG in Numbers. 2021. Disponível em: <https://www.weg.net/institutional/BR/en/weg-in-numbers>. Citado na página 81.

WEG, S. Guia de especificação motores elétricos. 2015. Citado na página 79.

WERBOS, P. Beyond regression:”new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*, 1974. Citado na página 48.

WERBOS, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, IEEE, v. 78, n. 10, p. 1550–1560, 1990. Citado na página 56.

WIDROW, B.; HOFF, M. E. *Adaptive switching circuits*. [S.l.], 1960. Citado na página 49.

XU, G.; LIU, M.; JIANG, Z.; SÖFFKER, D.; SHEN, W. Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 19, n. 5, p. 1088, 2019. Citado na página 25.

YANG, B.-S.; KIM, K. J. Application of dempster–shafer theory in fault diagnosis of induction motors using vibration and current signals. *Mechanical Systems and Signal Processing*, Elsevier, v. 20, n. 2, p. 403–420, 2006. Citado na página 24.

YE, Z.; WU, B.; ZARGARI, N. Online mechanical fault diagnosis of induction motor by wavelet artificial neural network using stator current. In: IEEE. *2000 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies*. [S.l.], 2000. v. 2, p. 1183–1188. Citado na página 24.

YU, Y.; SI, X.; HU, C.; ZHANG, J. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, MIT Press, v. 31, n. 7, p. 1235–1270, 2019. Citado 2 vezes nas páginas 57 e 58.

ZHOU, Z.-H.; YU, Y. Ensembling local learners through multimodal perturbation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 35, n. 4, p. 725–735, 2005. Citado na página 32.

ZUBAREV, V. *Machine Learning for Everyone*. 2019. Disponível em: https://vas3k.com/blog/machine_learning/. Citado na página 29.

Apêndice A – Modelo Máquina Assíncrona ATP

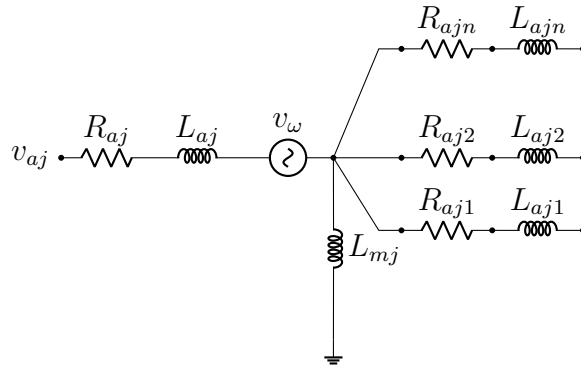
O ATP usa o Modelo de Máquina Universal (UM) implementado em 1980. Este modelo fornece as facilidades para estudar o comportamento transiente e estacionário de algumas máquinas elétricas tais como: máquinas síncronas, assíncronas e de corrente contínua (DOMIJAN; YIN, 1994). Três bobinas estacionárias, não acopladas magneticamente são representadas no UM: bobina d, bobina q e bobina zero. A representação dos enrolamentos de armadura da máquina possui no máximo três enrolamentos e o campo do rotor possui um número arbitrário de enrolamentos (LAUW; MEYER, 1982).

O método baseado em compensação é usado no ATP para gerar as saídas das simulações. Essa técnica é dividida em algumas etapas de cálculo. Cada circuito conectado ao terminal UM é substituído por um circuito de Thévenin, extraído as tensões de circuito aberto. Em seguida, os circuitos dos elementos de Thévenin (parâmetros de fase) são convertidos em quantidades d-q. Então, a corrente de armadura e de campo são obtidas resolvendo equações diferenciais. Através dos resultados obtidos, o torque eletromagnético é calculado. A velocidade do rotor é obtida através da solução de rede mecânica. A etapa final do cálculo converte as correntes d-q da armadura em quantidades de fase.

Parâmetros Elétricos

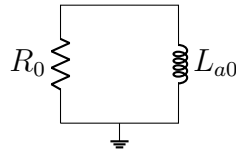
O circuito do UM é modelado com base no domínio zero, direto e quadratura (0-d-q). Os circuitos equivalentes do UM, representativos do motor de indução em gaiola de esquilo são mostrados na Figura 62 e 63. As variáveis do eixo d ou eixo q são resistência da armadura, R_{aj} , e indutância da armadura de dispersão, L_{aj} . A estrutura do enrolamento de campo possui um curto-circuito, com resistências R_{fj1} e indutância de fuga, L_{fj1} , referente ao lado da armadura. O j denota a quantidade do eixo d ou q. A indutância magnetizante é L_{mj} . Em máquinas de indução os valores d-q são idênticos. O circuito da Figura 63 possui os elementos do eixo zero, R_{a0} , resistência da armadura e indutância de dispersão, L_{a0} . As entradas do campo de sequência zero são resistência de campo, R_{f0} , e indutância, L_{f0} . A estrutura do enrolamento de campo é referida ao lado da armadura, representando um fluxo mútuo, λ_{mj} .

Figura 62 – Circuitos elétricos nos eixos d e q



Fonte – Meyer (2002)

Figura 63 – Circuito elétrico nos eixo zero



Fonte – Meyer (2002)

As tensões nos enrolamentos de armadura no domínio 0-dq são calculadas pelas equações (138) e (139), com v_{ω} , ω sendo tensão angular e velocidade angular, respectivamente. Os fluxos magnéticos zero, direto, quadratura, λ_{a0} , λ_{ad} , λ_{aq} podem ser calculados usando as equações (140) a (142).

$$v_{aj} = -R_{aj}i_{aj} - \frac{d\lambda_{aj}}{dt} + v_{\omega} \quad (138)$$

$$v_{a0} = -R_{a0}i_{a0} - \frac{d\lambda_{a0}}{dt} \quad (139)$$

$$v_{\omega} = \begin{cases} -\omega\lambda_{aq} & \text{if } j = d \\ +\omega\lambda_{ad} & \text{if } j = q \end{cases} \quad (140)$$

$$\lambda_{aj} = L_{aj}i_{aj} + \lambda_{mj} \quad (141)$$

$$\lambda_{a0} = L_{a0}i_{f0} \quad (142)$$

A estrutura dos enrolamentos de campo do UM, ao modelar uma máquina de indução, possui dois enrolamentos para representar as barras do rotor. Usando a Equação (143) é possível

calcular a tensão nos enrolamentos de campo. Neles, as variáveis λ_{fj1} e λ_{fj2} correspondem aos fluxos de campo magnético.

$$\begin{bmatrix} v_{fj1} \\ v_{fj2} \end{bmatrix} = - \begin{bmatrix} R_{fj1} \\ R_{fj2} \end{bmatrix} \begin{bmatrix} i_{fj1} \\ i_{fj2} \end{bmatrix} - \frac{d}{dt} \begin{bmatrix} \lambda_{fj1} \\ \lambda_{fj2} \end{bmatrix} \quad (143)$$

A matriz S transforma os parâmetros trifásicos no domínio ABC em elementos nos eixos direto e de quadratura. Através da transformação rotacional, P , as variáveis nos eixos d-q se tornam estacionárias quando vistas da estrutura do campo. O β_{UM} é a posição do rotor relacionada com a velocidade angular ω . Uma máquina de indução trifásica utiliza apenas a matriz P para transformar os parâmetros da armadura.

$$P = \begin{bmatrix} \cos \beta_{UM} & -\sin \beta_{UM} & 0 \\ \sin \beta_{UM} & \cos \beta_{UM} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (144)$$

$$S = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \quad (145)$$

Parâmetros Mecânicos

. O sistema mecânico da máquina é representado por um sistema elétrico equivalente através de um circuito elétrico externo. Neste circuito, um capacitor conectado à terra caracteriza a inércia da máquina. O torque aplicado no eixo do motor é modelado com uma fonte de corrente conectada ao nó de massa correspondente no bloco do motor. Uma resistência paralela precisa ser conectada ao capacitor para modelar o amortecimento. A Tabela 17 resume a equivalência entre os parâmetros mecânicos e elétricos (MARTINEZ-VELASCO, 2020; MEYER, 2002)

Tabela 17 – Equivalência entre os parâmetros elétricos e mecânicos ATP

Mecânico		Elétrico	
Torque	[Nm]	Corrente	[A]
Velocidade Angular	[rad/s]	Tensão	[V]
Posição Angular	[rad]	Carga no Capacitor	[C]
Momento de Inércia	kgm ²	Capacitância	[F]
Constante da Mola	[Nm/rad]	Indutância	[1/H]
Coefficiente de Amortecimento	[Nm/rad]	Condutância	[S]

Fonte – Adaptado de [Dommel \(1986\)](#)

Apêndice B – Partida Direta de um Motor de Indução Trifásico: Comparação entre os softwares ATP e MATLAB/Simulink

A Figura 64 mostra o diagrama de simulação modelado nos softwares ATP e MATLAB/Simulink. A máquina foi alimentada por uma fonte de tensão trifásica durante 1 segundo. Um torque nominal constante foi aplicado no eixo da máquina em 0.4 s. As especificações dos ajustes das simulações são mostradas na Tabela 18.

Tabela 18 – Especificações das simulações

	MATLAB/Simulink	ATP
Tipo de Simulação	Discreta	Discreta
Tempo de Amostragem [s]	10^{-6}	10^{-6}
Discrete Solver Model	Trapezoidal(asynchronous machine model) Tustin/Backward Euler (powergui model)	Trapezoidal

Fonte – Pedro H. Camargos, 2021

As especificações técnicas dos componentes usados nas simulações são descritas a seguir:

Fonte Trifásica

- Tensão: 400 [V] (V_s)
- Frequência: 50 [Hz]
- Resistência: 0.005 [Ω]

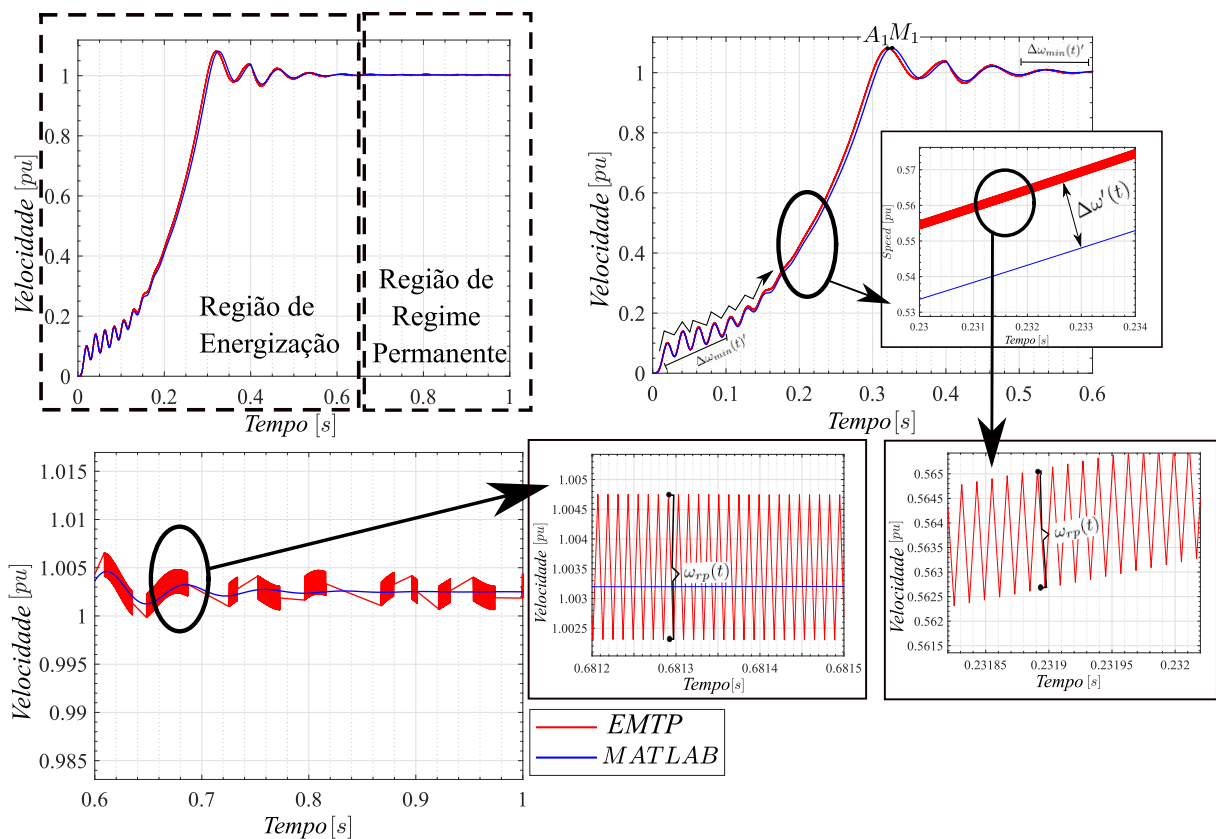
Máquina Assíncrona (Gaiola de Esquilo)

- Potência nominal: 15 [kW]
- Tensão nominal: 400 [V]
- Pares de polos: 2
- Velocidade nominal: 1475 [rpm]
- Frequência nominal: 50 [Hz]
- Corrente nominal: 49.6 [A]
- Escorregamento nominal: 1.67 [%]
- Resistência no estator: 0.1358 [Ω]
- Reatância de dispersão no estator: 0.3220 [Ω]
- Resistência no rotor: 0.1366 [Ω]
- Reatância de dispersão no rotor: 1.0724 [Ω]

- Reatância de magnetização: 16.524 [Ω]
- Momento de inércia: 0.18125 [kgm^2]
- Torque nominal: 97.2 [Nm]

Estudos comparativos dividiram as curvas em duas regiões: energização e regime permanente. As curvas de velocidade do rotor são mostradas na Figura 64. A diferença entre as curvas de velocidade nos dois softwares é $\Delta\omega'(t)$. O $\Delta\omega'(t)$ atinge o menor valor no tempo, $t \leq 0.15$ s. Os pontos A_1 e M_1 correspondem as maiores velocidades de rotação atingidas no ATP e MATLAB/Simulink, respectivamente. Os resultados mostraram uma diferença mínima entre os valores de pico, com velocidades em A_1 , $\omega_{A_1} = 1.082$ pu e M_1 , $\omega_{M_1} = 1.081$ pu. Em alguns trechos da curva de velocidade do ATP há uma pequena oscilação numérica dada por $\omega_{rp}(t)$.

Figura 64 – Velocidade de rotação

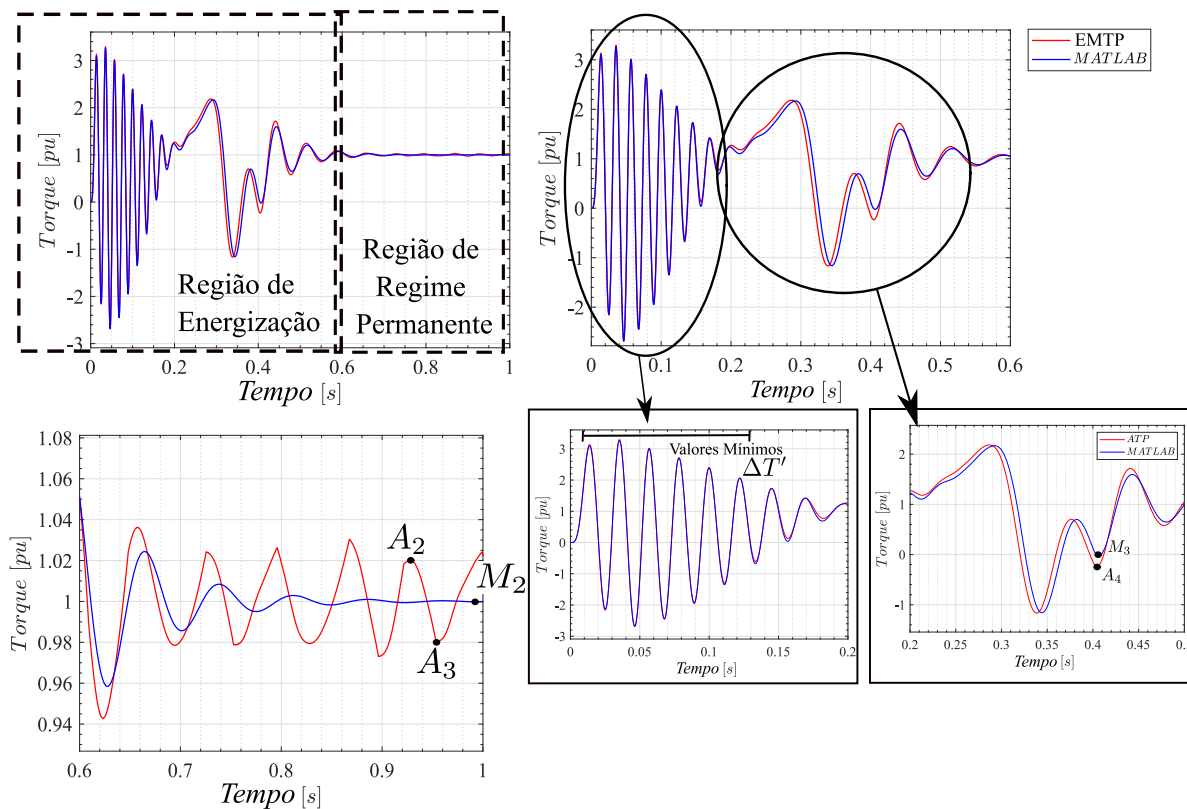


Fonte – Pedro H. Camargos, 2021

Na análise das curvas de torque (Figura 65), as diferenças de torque, $\Delta T'(t)$, foram mínimas para $t \leq 0,2$. Ainda na região de energização em $t \leq 0.4047$ s, $\Delta T'(t)$ tem seu valor máximo igual a 0.2187 pu. O ATP apresentou maiores oscilações de torque que o MA-

TLAB/Simulink na região de regime permanente, pontos $A_2(0.9557; 0.9803)$, $A_3(0.9357, 1.020)$. Em M_2 , o torque do MATLAB/Simulink já atingiu o valor nominal.

Figura 65 – Torque



Fonte – Pedro H. Camargos, 2021

Os resultados das simulações mostraram diferenças mínimas entre as curvas de torque e velocidade em ambos os *softwares*. No entanto, no regime permanente, as oscilações de torque do ATP foram significativas. O estudo comparativo auxiliou na escolha dos parâmetros configurados nas simulações no MATLAB/Simulink utilizadas nos sistemas de diagnósticos de falhas dos motores de indução.