

UNIVERSIDADE DE SÃO PAULO–USP
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
CURSO DE ENGENHARIA ELÉTRICA - ÊNFASE EM SISTEMAS DE ENERGIA E
AUTOMAÇÃO

Vitor do Nascimento Fontenele

**Aplicação de Sistemas Inteligentes para
Análise e Identificação de Falhas em
Motores de Indução Trifásicos**

São Carlos
2014

Vitor do Nascimento Fontenele

**Aplicação de Sistemas Inteligentes para
Análise e Identificação de Falhas em
Motores de Indução Trifásicos**

Trabalho de Conclusão de Curso apresentada ao Curso de Engenharia Elétrica - Ênfase em Sistemas de Energia e Automação da Escola de Engenharia de São Carlos como parte dos requisitos para a obtenção do título de Engenheiro Eletricista.

Área de concentração: Automação e Controle Inteligente de Máquinas Elétricas

Orientador: Prof. Dr. Marcelo Suetake

Coorientador: Prof. Dr. Ivan Nunes da Silva

São Carlos

2014

Trata-se da versão corrigida da monografia. A versão original se encontra disponível na EESC/USP que aloja o Curso de Engenharia Elétrica - Ênfase em Sistemas de Energia e Automação .

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

F683a Fontenele, Vitor do Nascimento
Aplicação de Sistemas Inteligentes para Análise e
Identificação de Falhas em Motores de Indução
Trifásicos / Vitor do Nascimento Fontenele; orientador
Marcelo Suetake; coorientador Ivan Nunes da Silva. São
Carlos, 2014.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Sistemas de Energia e Automação) -- Escola de
Engenharia de São Carlos da Universidade de São Paulo,
2014.

1. Motor de Indução Trifásico. 2. Sistemas
Inteligentes. 3. Redes Neurais Artificiais. 4.
Diagnóstico de Falhas. 5. Falhas Estatísticas. I.
Título.

FOLHA DE APROVAÇÃO

Nome: Vitor do Nascimento Fontenele

Título: "Aplicação de sistemas inteligentes para análise e identificação de falhas em motores de indução trifásicos"

Trabalho de Conclusão de Curso defendido e aprovado
em 28 / 11 / 2014,

com NOTA 9,5 (NOVE, CINCO), pela Comissão Julgadora:

Prof. Adjunto Marcelo Suetake - (UFSCar)

Prof. Dr. José Carlos de Melo Vieira Júnior - (SEL/EESC/USP)

Mestre Fabio Anderson Silva Borges - (Doutorando - SEL/EESC/USP)

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Homero Schiabel

*Este trabalho é para Clemente, Maria Francisca,
Vinícius, Vitória, Dalva e Géssyka, que nunca me
deixaram desamparado.*

Agradecimentos

Agradeço primeiramente a Deus, por sempre estar ao meu lado em todas as batalhas.

Agradeço também à minha família: a meu pai, Clemente, que sempre me incentivou a buscar o melhor de mim mesmo; à minha mãe, Maria Francisca, pelo seu esforço incansável por seus filhos, e pelo seu orgulho de vê-los no caminho certo; aos meus irmãos, Vinícius e Vitória, pelo seu amor, carinho e companheirismo; e à Dalva, minha segunda mãe, por ter me tratado também como filho.

À Gessyka Leônidas, minha companheira, pelo apoio, principalmente nos momentos mais difíceis: obrigado por fazer também seu um sonho que era só meu. Estendo meus agradecimentos também à sua família, e em especial ao seu irmão, Gérson Neto, por me ajudar sempre a cuidar de você.

Ao meu orientador Prof. Dr. Marcelo Suetake, pelas suas contribuições para a minha formação acadêmica e profissional, desde a sala de aula até este Trabalho, passando pela Iniciação Científica e pelas atividades de laboratório. Por estar sempre presente, e ter sido não só meu orientador, mas um bom amigo. Agradeço também ao meu co-orientador, o Prof. Dr. Ivan Nunes da Silva, pela forma bem-humorada e polida com que conduz o processo ensino-aprendizado, e que foi parte fundamental nesta caminhada.

Aos demais professores da Universidade de São Paulo (USP/SC), especialmente, o Prof. Dr. Luís Fernando Costa Alberto, pelo seu apoio em relação aos tópicos de máquinas elétricas; o Prof. Dr. Dennis Brandão, nos tópicos de Automação; o Prof. Dr. José Roberto B. de A. Monteiro, em Eletrônica de Potência; o Prof. Dr. Paulo R. Veronese, em Eletrônica Analógica; e o Prof. Dr. Danilo H. Spatti, em Eletrônica Digital.

Aos professores e colegas da UTFPR-CP, pelo acolhimento e apoio às minhas atividades em Cornélio Procopio: Prof. Dr. Alessandro Goedtel, Prof. Dr. Wagner Fontes Godoy e MSc. Clayton Graciola.

Aos colegas do Laboratório de Controle Inteligente de Máquinas Elétricas (LACIME) e do Laboratório de Automação Inteligente de Processos e Sistemas (LAIPS): MSc. Fabbio Borges e Eng^a. Luísa Liboni, pela ajuda nas etapas práticas do trabalho e ao Prof. Dr. Ricardo A. S. Fernandes, pelas orientações pontuais.

Aos meus colegas de curso e amigos, companheiros na Eletrônica e na Automação, pela paciência e cumplicidade durante os quatro anos que passamos juntos, em especial a Henrique Andrade, Rafael Fideles, Victor H.Y. Mathias, Lucas Garrigós, Luiz Desuó, Caio Fujiwara, Camila Ramalho, Gustavo Sanches, Juan Almeida, Amanda e Tamara Ferlin, Pedro Mosquera, Gabriel Rogatto, Aurélio Mizani, Lucas Barboza, Victor Esmeraldo, Diego Mansur, Gustavo Nicodemo e outros tantos.

Aos colegas de república, Gustavo Celso, José Inácio, Mateus Sobral, Felipe Bernardo e Guilherme Holanda, por terem sido ao mesmo tempo amigos e família.

Aos funcionários do Departamento de Elétrica e Computação (SEL/EESC/USP), em especial às secretárias de Graduação, Jussara Ramos Zoia e Aura Aparecido Brizolar.

Por fim, a todos os que contribuíram, direta ou indiretamente, para a realização deste trabalho.

*“Por que d’Ele e por meio d’Ele, e para Ele, são todas as coisas.
Glória, pois, a Ele eternamente. Amém.”
(Romanos, 11,36)*

*“Our virtues and our failings are inseparable, like force and matter.
When they separate, man is no more.”
(Nikola Tesla)*

Resumo

Fontenele, Vitor **Aplicação de Sistemas Inteligentes para Análise e Identificação de Falhas em Motores de Indução Trifásicos**. 84 p. Trabalho de Conclusão de Curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, 2014.

É constante e mundialmente difundida a busca por um sistema capaz de indicar, prever e/ou estimar a presença, teor e magnitude das falhas mais comuns apresentadas pelos Motores de Indução Trifásicos, que apresente fácil implementação e baixo custo computacional. Nesse sentido, este trabalho considerou as técnicas de análise da assinatura de corrente no domínio do tempo, investigando exaustivamente os atributos (média, variância, momentos estatísticos, curtose, obliquidade) que podem ser extraídos do sinal de corrente do motor, a fim de determinar uma topologia ótima de classificação neuronal, selecionando os atributos mais relevantes para a identificação das faltas de curto-circuito estatístico em estudo. A técnica desenvolvida foi então validada com ensaios de máquinas em diferentes condições de acionamento, carga e níveis de falha. Apesar de enfrentar dificuldades frente a variações de frequência quando treinada apenas com amostras de acionamento direto como previsto na literatura, a rede teve resultado excelente nos testes de generalização e frente a amostras ruidosas.

Palavras-chave: Motor de Indução Trifásico. Sistemas Inteligentes. Redes Neurais Artificiais. Diagnóstico de Falhas. Falhas Estatísticas..

Abstract

Fontenele, Vitor **Computational Intelligence Application in Analysis and Identification of Faults in Three-phase Induction Motors**. 84 p. Final Paper – São Carlos School of Engineering, University of São Paulo, 2014.

It is constant and widespread the research for a system capable of indicating, predicting or estimating the presence, type, and magnitude of the most common faults in three-phase induction motors with low computing cost and easy implementation. Accordingly, this study considered the Time-Domain Motor Current Signature Analysis, thoroughly investigating the attributes (mean, variance, statistical moments, kurtosis, skewness) which could be extracted from the motor current signal, in order to determine an optimal topology for a neural classifier, selecting the most relevant attributes for identifying the short-circuit stator faults in study. The technique developed was then validated against real machines tests in different driving, load, and fault conditions. Despite facing difficulties when facing frequency variations when trained only with direct driving samples as predicted by the literature review, the network excelled in generalization tests and noisy samples tests.

Keywords: Three phase induction motors. Intelligent Systems. Artificial Neural Networks. Fault Diagnosis. Stator Faults..

Lista de ilustrações

Figura 1	Consumo de Energia Elétrica por categoria no país em 2012 e a participação dos motores elétricos no consumo nacional	24
Figura 2	Visão geral da bancada de experimentos do LSI/UTFPR	30
Figura 3	MIRGE modificado para simulações de curto-circuito estatórico	31
Figura 4	Circuito equivalente do GCC	31
Figura 5	Torquímetro e visor de monitoramento	33
Figura 6	Placa de Aquisição NI USB-6221	34
Figura 7	Resposta em frequência do Filtro RC-série da aquisição	35
Figura 8	Sinal senoidal puro e sinais portadores de conteúdo harmônico	39
Figura 9	Distribuições com obliquidade negativa, nula e positiva	41
Figura 10	Histogramas para as distribuições correspondentes às funções da Figura 8	42
Figura 11	Abas do ambiente <code>Explorer</code> do WEKA	43
Figura 12	Exemplo de gráfico de dispersão gerado pelo WEKA	46
Figura 13	Desenho esquemático de um neurônio artificial	51
Figura 14	Desenho esquemático simplificado de uma RNA do tipo FFNN	52
Figura 15	FFNN e a notação para o algoritmo BP	52
Figura 16	Funções de ativação	55
Figura 17	Visualização de exemplo de FFNN para reconhecimento de padrões no MATLAB	57
Figura 18	Organograma simplificado dos testes realizados	61
Figura 19	Visualização da rede PMC gerada na GUI do WEKA	65
Figura 20	Gráfico de performance: PMC com topologia 15-8-1	68
Figura 21	Matriz de Confusão: PMC com topologia 15-8-1	69
Figura 22	Gráfico de eficiência (tempo de treinamento vs. taxa de acerto) para as topologias testadas	70

Figura 23	Gráfico de performance: generalização para curtos de 3% e 5%	72
Figura 24	Matriz de Confusão: generalização para curtos de 3% e 5%	73
Figura 25	Gráfico de performance: acionamento direto vs. inversor de frequência	74
Figura 26	Matriz de confusão: acionamento direto vs. inversor de frequência . . .	75
Figura 27	Matriz de confusão: diferentes níveis de ruído	76

Lista de tabelas

Tabela 1	Medidas estatísticas e símbolos correspondentes utilizados	38
Tabela 2	Comparação dos valores de pico, eficaz e do fator de pico entre os três sinais	40
Tabela 3	Comparação dos valores de curtose entre os três sinais	42
Tabela 4	Exemplo: O conjunto X, seus elementos e respectivas características . .	45
Tabela 5	Ranqueamento dos atributos (InfoGain + Ranker)	63
Tabela 6	Ranqueamento dos atributos (GainRatio + Ranker)	63
Tabela 7	Testes para determinação da melhor topologia do PMC	70
Tabela 8	Resultados de treinamento da rede com diferentes algoritmos	71

Lista de siglas

BP	<i>backpropagation</i> (algoritmo de treinamento)
DAQ	<i>Data Acquisition Board</i> — Placa de Aquisição de Dados
ELMNN	<i>Elman Neural Network</i> — Rede Neural do tipo <i>Elman</i>
EPE	Empresa de Pesquisa Energética
FFNN	<i>feed-forward neural network</i> — Rede Neural do tipo <i>feed-forward</i>
FFT	<i>Fast Fourier Transform</i> — Transformada Rápida de Fourier
FMM	<i>fuzzy min-max</i> (sistema de inferência)
GCC	Gerador de Corrente Contínua
GD	Gradiente Descendente (algoritmo de treinamento)
GDM	Gradiente Descendente com Momento (algoritmo de treinamento)
GUI	<i>Graphic User Interface</i> — Interface Gráfica de Usuário
LM	Levenberg-Maquardt (algoritmo de treinamento)
MCSA	<i>Motor Current Signature Analysis</i> — Análise da Assinatura de Corrente do Motor
MIRGE	Motor de Indução com Rotor em Gaiola de Esquilo
MIT	Motor de Indução Trifásico
MME	Ministério de Minas e Energia
MMQ	Método dos Mínimos Quadrados
PCA	<i>Principal Component Analysis</i> — Análise de Componentes Principais

- PMC** *Perceptron Multicamadas*
- PWM** *Pulse-Width Modulation* — Modulação em Largura de Pulso
- RBNN** *Radial Basis Function Neural Network* — Rede Neural do tipo Função de Base Radial
- RMS** *Root Mean Square* — Valor Quadrático Médio
- RNA** Rede Neural Artificial
- RPROP** *resilient propagation* (algoritmo de treinamento)
- SSA** *Singular Spectrum Analysis* — Análise de Espectro Singular
- SVM** *Support Vector Machine* — Máquina de Vetores de Suporte

Sumário

1	Introdução	23
1.1	Motivação e Relevância	23
1.2	Objetivos	27
1.3	Organização do Trabalho	27
2	Bancada de Experimentos	29
2.1	Visão Geral	29
2.2	O MIRGE Modificado	30
2.3	Simulador de Carga	31
2.4	Aparatos de Medição	33
2.4.1	Torque	33
2.4.2	Corrente/Tensão Trifásicas	33
2.5	Aparatos de Aquisição	33
3	Análises estatísticas da senoide e o software WEKA	37
3.1	Medidas estatísticas	37
3.1.1	Valor de pico (V_{pk})	38
3.1.2	Valor eficaz (V_{RMS})	38
3.1.3	Fator de pico (f_{pk})	39
3.1.4	Momento de 1ª ordem (m_1)	40
3.1.5	Momento de 2ª ordem (m_2)	40
3.1.6	Momentos de 3ª e 4ª ordem (m_3, m_4), Obliquidade (s_{kw}) e Curtose (k_{rt})	41
3.2	Mineração de dados com o software WEKA	42
3.2.1	Pré-processamento dos dados	42
3.2.2	Seleção de atributos	43
3.2.3	Visualização dos dados	45
3.2.4	Técnicas de classificação	46

3.3	Considerações finais	47
4	Redes Neurais Artificiais	49
4.1	O neurônio artificial	50
4.2	Redes <i>Perceptron</i> Multicamadas para problemas de classificação de padrões	51
4.2.1	Processos de treinamento	51
4.2.2	Generalização e <i>Overfitting</i>	54
4.2.3	Aspectos de aplicação	55
5	Desenvolvimento do Sistema Neural para Detecção de Curto-Circuito em MIRGE	59
5.1	Estrutura básica do sistema	59
5.2	Rotina de testes	59
5.3	Extração de características	61
5.4	Pré-processamento dos dados	62
5.5	Seleção dos atributos	63
5.6	Testes das topologias candidatas	64
6	Resultados Experimentais	67
6.1	Topologia ótima	68
6.2	Supressão de dados de curto-circuito	71
6.3	Acionamento Direto <i>versus</i> Inversor	73
6.4	Amostras ruidosas	74
7	Conclusões e Diretivas Futuras	77
	Referências	81

Introdução

1.1 Motivação e Relevância

A partir do desenvolvimento do conceito de sustentabilidade, durante a década de 1970, a utilização racional dos recursos naturais tornou-se uma responsabilidade governamental que foi repassada em grande medida às indústrias, historicamente os maiores exploradores de toda sorte de recursos primários. Seguiu-se um desenvolvimento das estruturas de fiscalização e das políticas de incentivo governamental, oferecendo às companhias que comprovadamente adotam uma maior eficiência em seus processos diversas vantagens econômicas. Ademais, o conceito de sustentabilidade adentrou ao panorama organizacional, e refletiu-se não apenas nas questões ambientais, mas também na racionalização do consumo de recursos energéticos e produtivos.

Assim, cresceu a procura por novas tecnologias que permitissem o melhor controle da performance dos processos produtivos. O setor industrial, segundo dados da Empresa de Pesquisa Energética (EPE), vinculada ao Ministério de Minas e Energia (MME), foi responsável por cerca de 40% do consumo de energia elétrica no Brasil no ano de 2012 (EPE, 2013). Desse percentual, uma quantia relevante deve-se aos motores elétricos: estudos apontam que esses equipamentos são responsáveis por 60 a 70% da eletricidade total absorvida pelas indústrias na União Europeia (BOGLIETTI et al., 2004), o que, refletido para a realidade brasileira representa um quarto da produção nacional, ou cerca de 115 milhões de MWh (vide Figura 1). Portanto, o estudo dessas máquinas é elemento indispensável em termos de eficiência e produtividade.

Os Motores de Indução Trifásicos (MIT) representam cerca de 90% do total de motores elétricos industriais (TRZYNADLOWSKI, 2001), sendo o seu maior expoente o Motor de Indução com Rotor em Gaiola de Esquilo (MIRGE), o que muito se deve à sua robustez, versatilidade e longa vida útil, que tornam seu custo bastante atraente (GOEDTEL, 2007; FITZGERALD; KINGSLEY JR.; UMANS, 2006). As falhas que apresentam maior prevalência em MIRGE incluem as avarias nos rolamentos (aproximadamente 40% do total), problemas no estator ou armadura (em torno de 30%), quebra de barras do rotor ou dos anéis

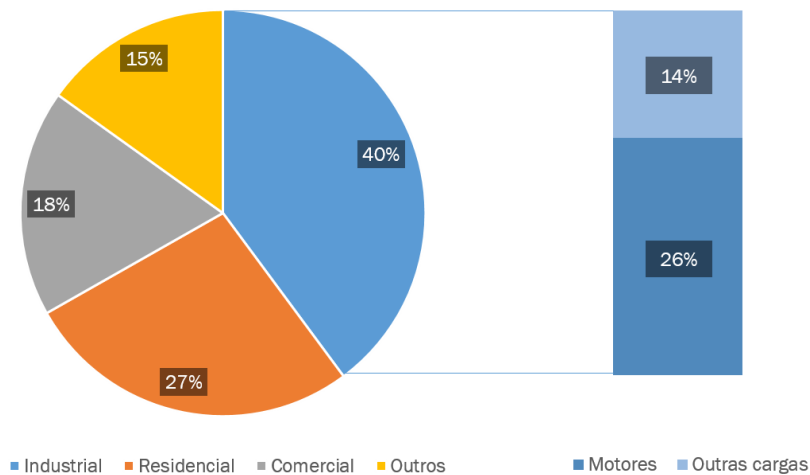


Figura 1 – Consumo de Energia Elétrica por categoria no país em 2012 e a participação dos motores elétricos no consumo nacional.

Fonte: EPE (2013) e Boglietti et al. (2004).

terminais (cerca de 10%) e falhas de excentricidade (NANDI; TOLIYAT; LI, 2005; YEH et al., 2007). Essas falhas são causadas principalmente pela grande variedade de ambientes aos quais esses motores encontram-se hoje expostos, que amplifica o processo natural de desgaste de qualquer máquina operante com o tempo (GHATE; DUDUL, 2010).

Nas indústrias, as paradas programadas são, em geral, o momento para a substituição de equipamentos. Na maioria dos casos, aplica-se o conceito de manutenção preventiva, efetuando-se a remoção dos motores para revisão em intervalos pré-definidos. Entretanto, erros na estimativa desses intervalos podem acarretar em paradas de emergência em caso de falha durante a operação, ou na remoção precoce da máquina, que ainda apresenta condições normais de funcionamento. Surge então a necessidade de se fazer um diagnóstico do motor, a fim de determinar seu estado atual e indicar o melhor momento para uma remoção para manutenção, de preferência sem interromper sua operação.

Conforme Pires et al. (2013), nos últimos anos, diversos métodos de diagnóstico para máquinas elétricas foram desenvolvidos e divulgados na literatura. Esses métodos realizam uma análise de diferentes tipos de sinais para identificação, tais como medida da temperatura, técnicas de reconhecimento do infravermelho (PICAZO-RÓDENAS et al., 2013), monitoramento de campo eletromagnético do motor, monitoramento de emissões acústicas e de radiofrequência, análise de vibração e de corrente/tensão nas linhas (PIRES et al., 2013).

Contudo, a simples aquisição de sinais do equipamento não garante sozinha a detecção de avarias. É preciso criar um sistema que aponte não apenas a falha existe, mas também seja capaz identificá-la, preferencialmente, a tempo de permitir sua correção antes que ela venha a comprometer o funcionamento do motor de forma significativa. Para isso, a técnica analítica mais usada atualmente é a análise de assinatura de corrente — em inglês, *Motor Current Signature Analysis* (MCSA) —, devido à sua versatilidade ao ser

capaz de detectar todos os principais problemas destacados (GYFTAKIS et al., 2013).

A análise do sinal de corrente do motor, seja no domínio do tempo ou no espectro de frequência é considerada hoje um padrão em manutenção preventiva, muito devido à facilidade de obtenção dos dados (GARDEL et al., 2012). Todavia, tais técnicas apresentam algumas limitações, principalmente quando se trabalha com frequências variáveis (HONG et al., 2012). O monitoramento do torque eletromagnético (GYFTAKIS et al., 2013) ou da indutância diferencial (HONG et al., 2012), a análise da impedância aparente complexa (DRIF; ESTIMA; CARDOSO, 2012) e mudanças do ângulo de impedância (MOOSAVI et al., 2012) são também aplicadas na determinação de barras quebradas, excentricidade, rotor e curtos-circuitos, respectivamente.

Observa-se na literatura que os Sistemas Computacionais Inteligentes (SCI) têm se mostrado de grande utilidade no controle de máquinas elétricas à medida em que conseguem fazer estimativas de parâmetros a partir de dados de simples obtenção, como sinais de tensão e corrente, evitando assim a interrupção do seu funcionamento para medição direta. Entre os SCI utilizados para análise dos parâmetros de detecção encontramos as Redes Neurais Artificiais, sistemas e controladores nebulosos (*fuzzy*), e outros sistemas híbridos.

Em Coelho e Medeiros (2013), um sistema baseado em mapas auto-organizáveis bi-dimensional é usado para monitorar um motor de indução e classificar curtos-circuitos incipientes em espiras do enrolamento, bem como mostrar a evolução da falha no tempo. Um inversor PWM foi utilizado na alimentação do motor e alguns componentes do espectro de frequência serviram de variáveis de entrada do sistema, que foram adquiridas a diferentes frequências, cargas e gravidades de falha, mostrando grande taxa de acerto.

Monaro (2013) propôs um novo método de proteção de geradores síncronos baseado em lógica *fuzzy*: um sistema integrado de software e hardware, baseado em simulações computacionais de um sistema elétrico e dados experimentais de faltas internas em máquinas síncronas foi utilizado em testes de tempo-real do algoritmo de proteção. Os resultados mostraram ser o novo sistema mais sensível e abrangente que a técnica tradicional de proteção.

Dentre os SCI, as RNA destacam-se pela sua capacidade de adaptação por experiência e de aprendizado, grande habilidade de generalização, tolerância a falhas e facilidade de prototipagem (SILVA; SPATTI; FLAUZINO, 2010), e ainda estão em pleno processo de desenvolvimento e aprimoração. Em Cheng, Ai e Xu (2013), utiliza-se uma técnica de otimização bioinspirada, o algoritmo de “colônia de formigas”, como alternativa ao *backpropagation* (BP) para o treinamento da RNA, que é mais largamente aplicada, apontando para uma significativa melhoria na detecção de anomalias em motores assíncronos ao evitar os problemas de quedas em mínimos locais da rede usual. Outro estudo associou a rede a um algoritmo BP melhorado, chamado de *resilient propagation* (RPROP), atingindo também melhores resultados (YANG; ZHU; XIE, 2012). Em Seera et al. (2012),

a MCSA é combinada com uma rede neural do tipo *fuzzy min-max* (FMM), numa clara aplicação dos sistemas híbridos inteligentes na detecção e monitoramento de falhas no estator e de excentricidade.

No caso de determinação de avarias nos rolamentos, um estudo envolvendo análise de espectro singular (SSA) e uma RNA mostrou-se “simples, tolerante a ruídos e eficiente” (MURUGANATHAM et al., 2013). Outro estudo tratou o sinal de vibração do motor através da transformada rápida de Fourier (FFT) e após comparar uma RNA do tipo *feed-forward* (FFNN) com outros três tipos de SCI na análise desse sinal tratado, a apontou como “apropriada como uma rede ótima no trabalho de classificação de falhas em rolamentos de motores de indução” (MAHAMAD; HIYAMA, 2011). Em Yadav e Wadhvani (2011), dez diferentes características do sinal de vibração, como valor quadrático médio (RMS), valor de pico, curtose e obliquidade, foram analisadas por uma RNA na identificação de três tipos de falhas em rolamentos atingindo taxa de acerto de 98,33%.

Para a detecção de curtos-circuitos entre espiras, vários tipos de SCI e de RNA, mais especificamente, podem ser utilizados. Em Asfani et al. (2012), uma FFNN, uma rede do tipo *Elman* (ELMNN) e uma de funções de base radial (RBFNN) foram comparadas, com os resultados mostrando a ELMNN como a rede mais simples a conseguir resolver o problema, porquanto todas tiveram desempenho satisfatório. Outro trabalho associou os

al ritmos de treinamento BP e Levenberg-Maquardt (LM), e nessas condições um PMC apresentou resultados superiores a RBFNN, com um número de épocas de treinamento 45 vezes menor (BRONIERA JR.; GOEDEL; GRACIOLA, 2013). A análise dos sinais de corrente e tensão simultaneamente também é utilizada com sucesso, sendo capaz de indicar a existência de curtos-circuitos no estator mesmo com a existência de desbalanço controlado entre as tensões de entrada, queda ou pico de tensão simétrica nas três fases e variação controlada no torque de carga (SARKHANLOO et al., 2011).

As referências supracitadas deixam claro, portanto, que é constante e mundialmente difundida a busca por um sistema capaz de indicar, prever e/ou estimar a presença, teor e magnitude das falhas mais comuns apresentadas pelos MIT, apresentando fácil implementação e baixo custo computacional. O desenvolvimento de modelos simples e robustos capazes de atender às necessidades de manutenção dessas máquinas em escala industrial, sem que haja a exigência de parada do processo produtivo é o objetivo de diversos trabalhos em centros de pesquisa de todo o planeta.

Para que se atinja esse ponto, considerando-se as técnicas de MCSA no domínio do tempo, ainda se faz necessária a investigação a fundo dos atributos (média, variância, momentos estatísticos, curtose, obliquidade) que podem ser extraídos do sinal de corrente do motor. A relevância da atividade aqui desenvolvida será então a melhoria de técnicas já implantadas anteriormente por autores como Gardel et al. (2012) e Ghate e Dudul (2010), através de um estudo exaustivo de tais atributos e das medidas estatísticas que

são utilizadas para sua obtenção. Além disso, será feita a seleção daqueles que são efetivamente os mais relevantes para a identificação das faltas em estudo, baseado em técnicas de ranqueamento, o que deverá proporcionar maior acurácia do sistema de classificação aproveitando-se de um menor esforço computacional.

1.2 Objetivos

Os principais objetivos deste trabalho consistiram na aquisição e processamento de amostras de sinais de corrente de MIRGE normais e com falhas de curto-circuito entre bobinas do enrolamento estático e a investigação de topologias ótimas de RNA para sua classificação. Objetivos específicos incluem:

- levantamento de dados experimentais dos MIRGE em acionamento direto e por inversor de frequência, em diferentes níveis de carga e curto-circuito;
- estudos exaustivos das variáveis estatísticas e atributos para identificação dessas falhas;
- investigação, desenvolvimento e validação de topologias de Redes Neurais Artificiais com dados experimentais para identificação das falhas.

1.3 Organização do Trabalho

O trabalho encontra-se dividido em sete capítulos. Após esta Introdução, no segundo capítulo, abordar-se-á o MIRGE modificado e a bancada de testes. No terceiro, as medidas estatísticas escolhidas e o *software* WEKA (HALL et al., 2009). A seguir, trata-se dos aspectos relativos às Redes Neurais Artificiais. No quinto capítulo, será detalhado o desenvolvimento do trabalho em si: aquisição dos dados, cálculo das estatísticas, seleção de atributos e criação da RNA. No capítulo seguinte, serão apresentados os resultados do sistema de detecção sob diferentes condições de treinamento e teste. E por fim, no último capítulo, serão apresentadas as conclusões deste trabalho e diretrizes para projetos futuros.

Bancada de Experimentos

2.1 Visão Geral

A bancada de experimentos utilizada neste trabalho localiza-se no Laboratório de Sistemas Inteligentes (LSI) da Universidade Tecnológica Federal do Paraná (UTFPR), e foi desenvolvida em consequência de um trabalho conjunto com a EESC-USP, através do Projeto Casadinho/Procad MCT/CNPq/MEC/CAPES - Ação Transversal nº 06/2011, coordenado pelo Prof. Dr. Alessandro Goedel, do Programa de Pós-Graduação em Engenharia Elétrica do Campus de Cornélio Procopio (PPGEE/CP).

Conforme ilustrado na Figura 2, o conjunto compõe-se basicamente de:

- ❑ um MIRGE modificado para testes com simulação de falhas estatísticas;
- ❑ um Gerador de Corrente Contínua (GCC), para simular diferentes condições de carga mecânica no eixo do motor em teste;
- ❑ um torquímetro digital, que permite o monitoramento da velocidade de rotação do conjunto motor-gerador e do torque aplicado no eixo de acoplamento;
- ❑ um conjunto de sensores *Hall* de corrente e tensão para medição de corrente e tensão trifásicas de alimentação do motor;
- ❑ uma placa de aquisição (DAQ) modelo *NI USB-6221*, fabricada pela *National Instruments*, para aquisição e conversão A/D dos sinais de corrente e tensão medidos;
- ❑ itens acessórios de comando e alimentação do conjunto: painel de comando com VARIAC trifásico e suporte a partida Δ -Y; inversor de frequência; uma resistência para servir de carga para o GCC; e um contator para acoplamento dessa carga ao GCC.

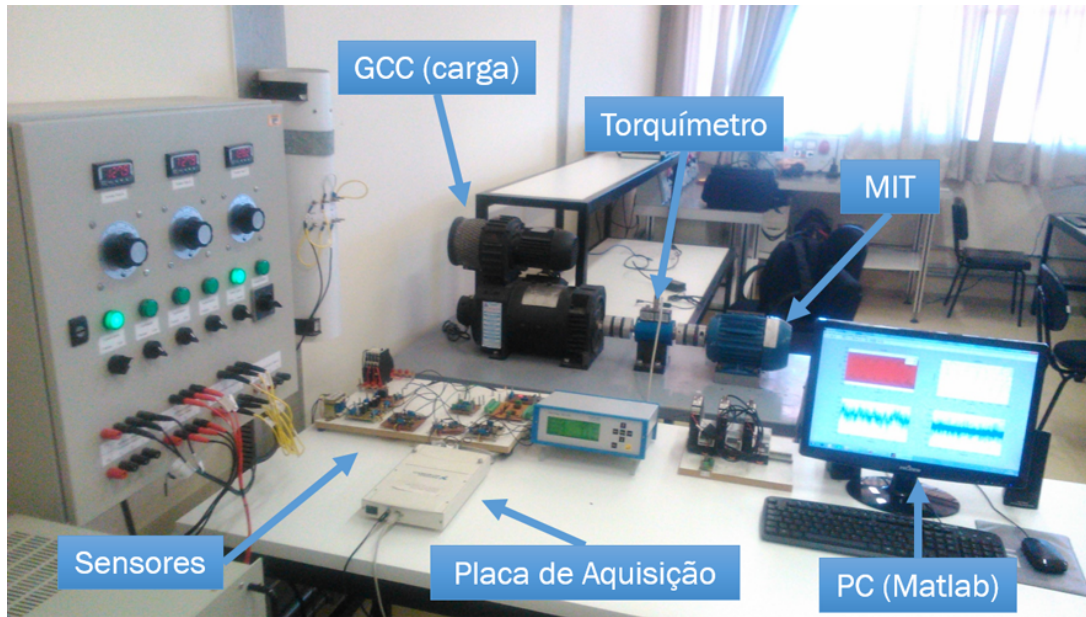


Figura 2 – Visão geral da bancada de experimentos do LSI/UTFPR.

Fonte: Autoria própria.

2.2 O MIRGE Modificado

O motor modificado, utilizado nos testes de simulação de curto-circuito no enrolamento de armadura através da perda do isolamento no fio componente da bobina do estator, foi desenvolvido por Gongora (2013), seguindo o proposto em Suetake (2012), onde esse procedimento é descrito em detalhes.

Numa situação real, uma vez comprometida a isolamento do esmalte que recobre o fio de cobre componente da bobina de armadura, há a possibilidade de curto-circuito entre bobinas de diferentes fases, entre a bobina e a carcaça do motor, ou mesmo entre diferentes espiras da mesma bobina. Aqueles que envolvem diferentes fases ou a carcaça do motor são em geral mais graves e podem levar a falhas permanentes no MIRGE, e portanto esses ensaios destrutivos não foram objetivo deste trabalho.

A Figura 3 ilustra o motor utilizado nos testes. Trata-se de um MIRGE comum de 1,0 cv (0,76 kW) de potência, IV pólos, marca WEG, modelo W21, que foi modificado conforme as instruções supracitadas, para apresentar *taps* de derivação dos enrolamentos. Na caixa lateral de ligação, portanto, além dos terminais para ligações em Δ ou Y, esse motor apresenta também uma *matriz de taps*, que permite então o curto-circuito entre espiras, tanto na mesma fase — em valores variando de 1% a 20%, conforme indicado pelas etiquetas afixadas — como entre fases diferentes.

Como dito anteriormente, porém, visando preservar a integridade do motor em teste, utilizou-se apenas dos curtos-circuitos entre espiras de mesma fase, nos valores de 1%, 3%, 5% e 10% permitidos pelo aspecto construtivo do projeto.



Figura 3 – MIRGE modificado para simulações de curto-circuito estático.

Fonte: Autoria própria.

2.3 Simulador de Carga

Para simular diferentes condições de operação do motor durante o teste, este foi acoplado a um GCC, que por sua vez foi conectado a uma resistência, que serviu como carga elétrica para este.

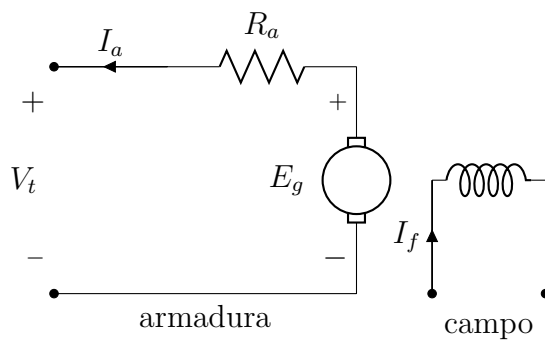


Figura 4 – Circuito equivalente do GCC.

Fonte: McPherson e Laramore (1990).

Conforme McPherson e Laramore (1990), através da análise do circuito equivalente da Figura 4, a tensão gerada pelo GCC deve ser dada por:

$$E_g = K_a \times \phi \times \omega_m \quad (1)$$

em que:

- $K_a \triangleq \frac{Z \cdot p}{2\pi \cdot a}$ é a constante de armadura da máquina, e depende apenas de fatores construtivos dela: o número total de condutores no enrolamento de armadura (Z),

o número de pólos (p) e o número de caminhos paralelos para a corrente em seu enrolamento (a);

- ϕ é o fluxo magnético gerado pelas bobinas do enrolamento de campo do GCC, que é proporcional à corrente I_f desse enrolamento (considerando que o núcleo do rotor opera na região linear, e não de saturação do ferro); e
- ω_m é a velocidade mecânica de rotação do eixo do GCC.

Dessa forma, pode-se afirmar que, ao conectar o GCC à carga resistiva, a tensão E_g passa a ser aplicada sobre a associação série entre a própria resistência do enrolamento de armadura e a resistência de carga. A corrente de armadura I_a gerada, portanto, pode ser dada pela Equação (2):

$$I_a = \frac{E_g}{R_a + R_L} \quad (2)$$

A potência elétrica P_{el} desenvolvida pelo GCC então é dada por:

$$P_{el} = (R_L + R_a) \times I_a^2 = \frac{E_g^2}{R_a + R_L} \quad (3)$$

Pode-se afirmar que, a menos das perdas mecânicas (essencialmente, perdas por atrito rotacional e acoplamento) e elétricas (que incluem a dissipação por efeito Joule no enrolamento de campo, queda de tensão nas escovas do comutador e efeitos de histerese e correntes de Foucault), a potência mecânica aplicada no eixo do motor é comparável a essa potência elétrica fornecida pela máquina. De fato, valores típicos de rendimento ($\eta \triangleq \frac{P_{mec}}{P_{el}}$) de GCC variam em torno de 85-95% (MCPHERSON; LARAMORE, 1990).

Assim, considerando novamente as Equações (1) e (3), e também que $\phi \propto I_f$, observa-se que a potência elétrica é função direta da corrente aplicada no enrolamento de campo da máquina de corrente contínua ($P_{el} \propto I_f$). Sabendo também que a potência mecânica é diretamente proporcional ao torque mecânico ($P_{mec} = \tau_{mec} \times \omega_m$), chega-se à conclusão de que:

$$\tau_{mec} \propto I_f \quad (4)$$

Dessa forma, mostra-se que é possível controlar o torque desenvolvido pelo MIRGE através da corrente de campo do GCC. Note que para a conclusão apresentada na Equação (4) foram desprezadas perdas elétricas e mecânicas e a não-linearidade do material magnético, conforme anteriormente discutido.

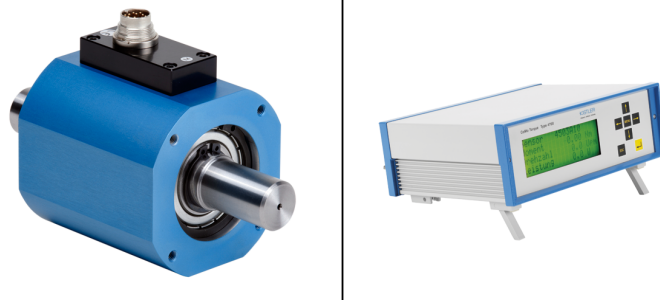


Figura 5 – Torquímetro *Kistler 4520A* e visor de monitoramento *Kistler 4700B*.

Fonte: Site do fabricante¹.

2.4 Aparatos de Medição

2.4.1 Torque

Para medição e acompanhamento do torque desenvolvido pelo motor, utilizou-se de um torquímetro digital da marca *Kistler*, modelo *4520A*, que se baseia em sensores *strain-gage* e retorna um sinal analógico na escala de 0-10 VDC, mais facilmente visualizado através do *Control Monitor CoMo Torque* modelo *4700B*, de mesmo fabricante (vide Figura 5).

Trata-se de um sensor robusto para medições de torque em eixos girantes, tanto para análises estáticas quanto dinâmicas, adequado para aplicações de controle de qualidade e medições laboratoriais. Através do visor, era possível regular a corrente de campo I_f do GCC através da variação de tensão de uma fonte CC e verificar seu efeito sobre a carga no eixo do MIRGE, ajustando-a para cada condição de teste.

2.4.2 Corrente/Tensão Trifásicas

Semelhante ao apresentado em Suetake (2012) e em Gongora (2013), utilizaram-se sensores Hall de corrente e tensão instalados em placas de condicionamento com amplificadores de instrumentação e correção de *offset*. Sua saída, modulada como sinal de tensão na faixa de -10 a +10 V_{CC} , pôde então ser ligada diretamente à placa de aquisição.

Por permitir isolamento galvânica entre as grandezas medidas e a DAQ, o sensor Hall torna a aplicação mais segura, protegendo a placa contra possíveis surtos na alimentação do motor. Vale lembrar que o circuito de condicionamento emprega amplificadores operacionais que saturam a saída dos sensores sob tais condições.

2.5 Aparatos de Aquisição

A placa de aquisição modelo *NI USB-6221* mostrada na Figura 6, fabricada pela *National Instruments*, dispõe de seis entradas analógicas (16-bit, 250 kS/s) e duas saídas

¹Disponíveis, respectivamente, em <http://www.kistler.com/exp/en/product/torque/4520A>. e <http://www.kistler.com/exp/en/product/monitors/4700B>. ; acesso em 13/10/2014.

analógicas (16-bit, 833 kS/s), além de 24 portas de entrada/saída digital (sendo oito delas sincronizadas com o *clock* da placa) e contadores de 32-bits.



Figura 6 – Placa de Aquisição *NI USB-6221*.

Fonte: Autoria própria.

Neste trabalho, utilizaram-se as seis entradas analógicas disponíveis para aquisição dos dados de corrente/tensão do MIRGE, conforme sinal gerado pelos sensores descritos na Seção 2.4.2. Como a placa de aquisição tinha taxa de amostragem definida para 5 kHz e o inversor de frequência para acionamento do motor a velocidades variáveis tinha uma frequência de chaveamento definida para 8 kHz, optou-se por colocar um filtro passa-baixa do tipo RC-série sintonizado em 1292 Hz nas entradas analógicas de corrente do motor na placa de aquisição, a fim de evitar que ruídos de chaveamento indesejados afetassem as grandezas a serem analisadas.

Essa frequência de sintonização do filtro foi escolhida por representar um valor em que a atenuação do sinal amostrado ocorre apenas para valores acima de 20 (vinte) vezes maiores que a frequência fundamental mais alta de acionamento (60 Hz) — vide Figura 7. Considerou-se, portanto, que a informação a ser extraída deveria estar contida, no máximo, dentro do conteúdo dos vinte primeiros harmônicos da amostra.

O controle e configuração da *NI USB-6221* foram feitos via *software* computacional MATLAB. Conhecida a bancada de testes, foi possível prosseguir para o estudo das técnicas de extração da informação contida nos sinais aquisitados.

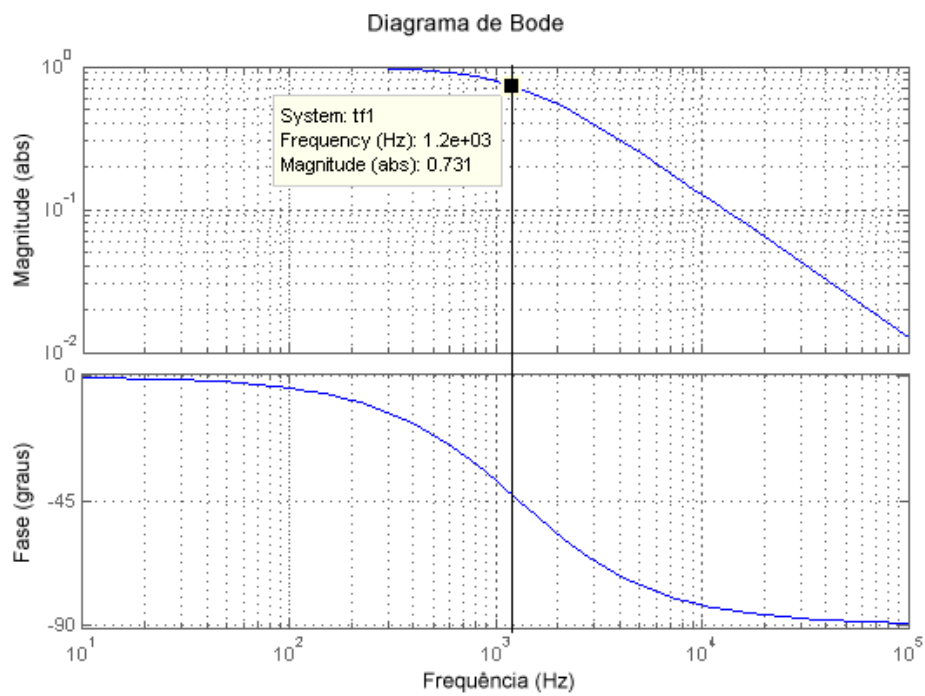


Figura 7 – Resposta em frequência do Filtro RC-série da aquisição.

Fonte: Autoria própria.

Análises estatísticas da senoide e o software WEKA

Para Witten, Frank e Hall (2005), “o trabalho do cientista (assim como o do bebê) é dar sentido aos dados, descobrir os padrões que governam o mundo físico e encapsulá-los em teorias que possam ser utilizadas para predição que irão acontecer em novas situações”. De fato, economistas, estatísticos, meteorologistas e engenheiros têm trabalhado com a ideia de obter, identificar, validar e utilizar automaticamente padrões em dados em atividades de predição. O crescimento das bases de dados em anos recentes com a informatização dos mais diversos sistemas trouxe a mineração de dados para a linha de frente das tecnologias de negócios (WITTEN; FRANK; HALL, 2005, p. 4).

A mineração de dados (ou *data mining*, do original em inglês) é definida como o processo de descobrir padrões em dados de forma automática, ou semi-automática (WITTEN; FRANK; HALL, 2005, p. 5). Neste capítulo, apresentar-se-á as técnicas utilizadas para caracterizar os padrões correspondentes aos motores testados e a utilização do software WEKA como ferramenta de auxílio nesse processo.

3.1 Medidas estatísticas

Baseado principalmente nos trabalhos de Gardel et al. (2012) e Ghate e Dudul (2010), este autor valeu-se uma série de medidas estatísticas na tentativa de extrair dados do sinal de corrente/tensão no domínio do tempo. No total, avaliou-se o efeito dos curtos-circuitos estatóricos no valor de 9 (nove) medidas estatísticas, conforme resumo na Tabela 1:

A seguir, será apresentada cada uma detalhadamente, em termos de sua definição matemática e das informações que permitem que se extraia do sinal adquirido.

Tabela 1 – Medidas estatísticas e símbolos correspondentes utilizados.

Núm.	Símbolo	Medida Estatística
1	V_{pk}	Valor de pico da onda (valor máximo absoluto)
2	V_{RMS}	Valor eficaz (RMS) da onda
3	f_{pk}	Fator de pico
4	m_1	Momento de 1ª ordem (média)
5	m_2	Momento de 2ª ordem (variância)
6	m_3	Momento de 3ª ordem
7	m_4	Momento de 4ª ordem
8	s_{kw}	Obliquidade (<i>skewness</i>)
9	k_{rt}	Curtose (<i>kurtosis</i>)

3.1.1 Valor de pico (V_{pk})

O valor de pico V_{pk} de um sinal $s(t)$, para as aplicações deste trabalho, pode ser definido como o máximo valor absoluto apresentado por este sinal dentro do conjunto ou subconjunto amostral avaliado, isto é, V_{pk} é numericamente igual à amostra de maior valor absoluto no subconjunto avaliado. Assim:

$$V_{pk} \triangleq \max\{|s(t)|\} \quad (5)$$

Para um sinal senoidal puro do tipo $s(t) = A \sin(2\pi f \cdot t + \varphi)$, V_{pk} fica sendo numericamente igual a A , amplitude da senoide. Considerando um conjunto amostral discreto $S(nT_a)$ para $s(t)$, temos que:

$$\lim_{T_a \rightarrow 0} \max\{|S(nT_a)|\} = \max\{|s(t)|\} = V_{pk} \quad (6)$$

Assim, considerando que no estudo aqui desenvolvido a taxa de amostragem T_a é de 5 kHz, para um sinal amostrado de frequência fundamental 60 Hz e máxima frequência de interesse em 1200 Hz, pode-se considerar que o valor de pico do sinal avaliado em tempo discreto é aproximadamente igual ao valor do sinal em tempo contínuo. Esta análise estende-se para todas as medidas seguintes, desenvolvidas em variável discreta.

Ainda considerando o estudo realizado, o interesse no valor máximo absoluto do sinal de corrente do motor tem relação direta com o fato de que o curto-circuito entre espiras provoca um desbalanço nas correntes de fase, com alteração, assim, do seu valor de pico.

3.1.2 Valor eficaz (V_{RMS})

O valor eficaz, ou valor quadrático médio, de uma amostra é uma medida de magnitude da grandeza avaliada, especialmente utilizada quando a variável medida oscila entre valores positivos e negativos, devido a sua componente quadrática. Tal valor é determinado por:

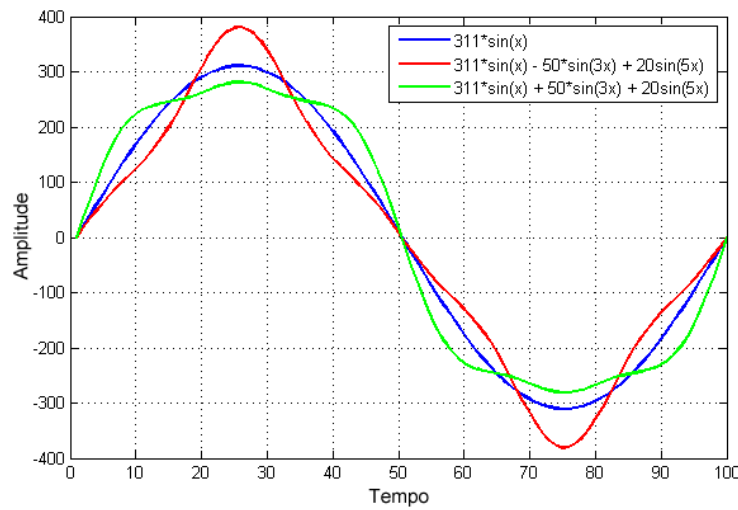


Figura 8 – Sinal senoidal puro e sinais portadores de conteúdo harmônico.

Fonte: Autoria própria.

$$V_{RMS} \triangleq \sqrt{\frac{1}{n} \sum [S(nT_a)]^2} \quad (7)$$

em que $S(nT_a)$ representa o sinal discretizado composto de n amostras do sinal original $s(t)$ a uma taxa de amostragem T_a .

Considerando-se as grandezas elétricas, o valor eficaz de uma corrente alternada é igual à corrente contínua que dissipa a mesma potência que esta quando aplicada em um elemento resistivo. Seu campo de interesse para o trabalho é bastante semelhante ao do valor de pico, por ser também o valor eficaz uma medida de magnitude.

3.1.3 Fator de pico (f_{pk})

Está definido como a relação entre o valor de pico e o valor eficaz de um sinal:

$$f_{pk} \triangleq \frac{V_{pk}}{V_{RMS}} \quad (8)$$

O fator de pico pode parecer, à primeira vista, de pouca relevância para ser adicionado a esta análise. No entanto, esta variável acrescenta informações de importância considerável, como ver-se-á a seguir.

Para uma senoide pura, o fator de pico é sempre igual a $\frac{A}{\frac{A}{\sqrt{2}}} = \sqrt{2} \approx 1,4142$, onde A é sua amplitude máxima. Senoides distorcidas por conteúdo harmônico de ordens superiores apresentam variações nesse valor. Pode-se fazer aqui uma rápida análise de sinais reais, que ajudará a compreender o valor dessa medida estatística. Tome-se para tal um sinal senoidal puro de frequência 60 Hz e dois outros de mesma frequência fundamental, porém portadores de conteúdo harmônico de 3^a e 5^a ordem (vide Figura 8).

Observa-se na Tabela 2 que, em termos de valor eficaz — que está diretamente relacionado com a potência — as ondas são praticamente equivalentes, e poderia não haver motivo para desconfiar que houvesse algum problema. No entanto, ao se avaliar o valor de pico para os 3 sinais, observa-se claramente uma discrepância significativa, que se reflete no fator de pico. A vantagem deste último reside, especialmente, em permitir a comparação de ondas de diferentes amplitudes, como as que serão verificadas, por exemplo, nas correntes do motor sob diferentes condições de carga.

Tabela 2 – Comparação dos valores de pico, eficaz e do fator de pico entre os três sinais.

Medida	Sinal Senoidal (azul)	Sinal 1 (vermelho)	Sinal 2 (verde)
V_{pk}	310,9609	380,8413	280,9546
V_{RMS}	218,8079	222,0639	222,0639
f_{pk}	1,4212	1,7150	1,2652

3.1.4 Momento de 1ª ordem (m_1)

Casella e Berger (2010) define o n -ésimo momento, ou momento de ordem n de uma variável aleatória X por:

$$\mu'_n \triangleq E[x^n] = \int_{-\infty}^{\infty} x^n f(x) dx \quad (9)$$

onde $f(x)$ é a função de distribuição de probabilidade associada à X . O momento central de X , m_n , por sua vez é definido como:

$$m_n \equiv \mu_n \triangleq E[X - \mu]^n = \int_{-\infty}^{\infty} (x - \mu)^n f(x) dx \quad (10)$$

onde $\mu = \mu'_1 = E[X]$ (média da distribuição).

O momento de 1ª ordem de uma distribuição representa, portanto, a sua média, e o momento central de 1ª ordem será sempre nulo, pois fazendo $n = 1$ na Equação (10), teremos $E[X - E[X]] = 0$. Neste trabalho, portanto, denotaremos m_1 sempre em referência à média $E[X]$.

3.1.5 Momento de 2ª ordem (m_2)

O momento central de 2ª ordem representa a variância de X :

$$\text{var}X = \sigma^2(X) \triangleq E[X - E[X]]^2 \equiv m_2 \quad (11)$$

A raiz quadrada positiva de m_2 (σ) representa o desvio padrão de X . A variância dá uma medida do grau de dispersão da variável em torno de seu valor esperado (média). Assim, quanto mais próxima a distribuição dos valores em X está de sua média, menor é o valor de m_2 , e vice-versa.

3.1.6 Momentos de 3^a e 4^a ordem (m_3, m_4), Obliquidade (s_{kw}) e Curtose (k_{rt})

Os momentos de 3^a e 4^a ordem são basicamente utilizados para definir os momentos padronizados $\frac{m_k}{\sigma^k}$.

O 3^o momento padronizado de uma distribuição representa a sua obliquidade:

$$s_{kw} \triangleq \frac{m_3}{(\sqrt{\sigma^2})^3} \quad (12)$$

A obliquidade é uma medida da assimetria de uma distribuição em torno de sua média. Uma distribuição com obliquidade negativa tem mais valores abaixo de sua média do que acima. Por outro lado, uma distribuição com obliquidade positiva exibe mais variedade de valores acima de sua média. A Figura 9 ajuda a compreender melhor esse conceito:

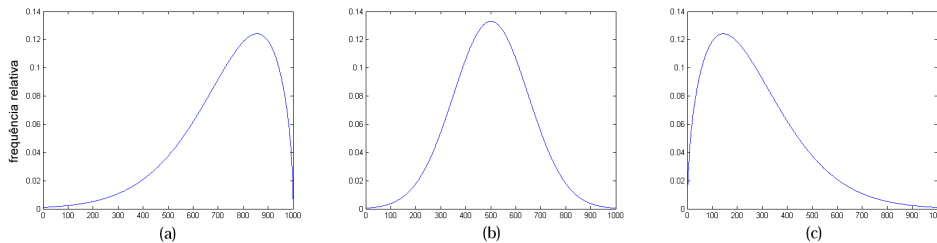


Figura 9 – Distribuições com obliquidade: (a) negativa, (b) nula e (c) positiva.

Fonte: Autoria própria.

Por sua vez, o momento padronizado de ordem 4 de uma distribuição representa a sua curtose:

$$k_{rt} \triangleq \frac{m_4}{(\sqrt{\sigma^2})^4} \quad (13)$$

A curtose compreende uma medida do quão “pontuda” é uma distribuição aleatória, isto é, do quanto a moda e os valores à sua vizinhança são representativos para a totalidade da distribuição. Também é descrita como o “tamanho dos ombros”, devido à sua pareidolia com a silhueta humana, por medir o quanto as “caudas” inferior e superior são relevantes em relação ao pico modal.

Distribuições com $k_{rt} = 0$ são ditas mesocúrticas; um exemplo é a distribuição normal ou gaussiana. Aquelas com $k_{rt} < 0$, por sua vez, são ditas platicúrticas — e seu pico é menos proeminente —, e as que apresentam $k_{rt} > 0$ são chamadas leptocúrticas — e apresentam uma maior concentração de valores em seu pico modal.

Voltando ao exemplo ilustrado na Figura 8, ao se contruir os histogramas do valor absoluto de cada um dos sinais (senoidal puro e ambos com conteúdo harmônico), encontram-se os gráficos representados na Figura 10, a seguir:

Observa-se então que, apesar de todos os sinais representarem distribuições leptocúrticas, em cada um deles uma característica diferente é exibida: ondas mais achatadas,

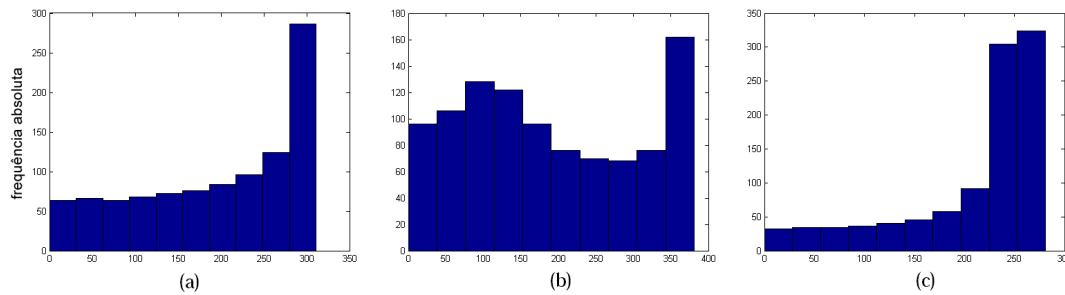


Figura 10 – Histogramas para as distribuições correspondentes às funções da Figura 8: (a) azul, (b) vermelho e (c) verde.

Fonte: Autoria própria.

como a da Figura 10c têm uma distribuição mais leptocúrtica, enquanto aquelas com formato de “chapéu de bruxa” (Figura 10b) apresentam um pico mais platicúrtica, quando comparadas à onda senoidal pura (Figura 10a).

Calculando-se a curtose para cada um destes sinais, chega-se ao resultado da Tabela 3, que confirma a análise qualitativa anteriormente desenvolvida. Repare no grande desvio no valor de curtose representado pelo “achatamento” do sinal senoidal mediante a injeção harmônica.

Tabela 3 – Comparação dos valores de curtose entre os três sinais.

Medida	Sinal Senoidal (azul)	Sinal 1 (vermelho)	Sinal 2 (verde)
k_{rt}	1,9311	1,7453	3,7495

3.2 Mineração de dados com o software WEKA

Acrônimo para “*The Waikato Environment for Knowledge Analysis*” (em inglês, “O Ambiente *Waikato* para Análise de Conhecimento”), WEKA é definido como “uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados” por seus idealizadores, o grupo de Aprendizagem de Máquina da Universidade de Waikato, em Hamilton, Nova Zelândia.

O software baseado na linguagem Java, se compõe de diversas ferramentas de mineração de dados, distribuídas em alguns “ambientes” para exploração e experimentação das diversas técnicas disponíveis. Neste trabalho, explorar-se-á apenas o ambiente **Explorer**, e suas abas **Preprocess**, **Classifier**, **Select Attributes** e **Visualize** (ver Figura 11), que terão suas aplicações melhor detalhadas na sequência.

3.2.1 Pré-processamento dos dados

A aba **Preprocess** é o ponto de início para a exploração do banco de dados. Além das ferramentas de arquivo, nessa aba o WEKA dispõe de um conjunto de ferramentas para

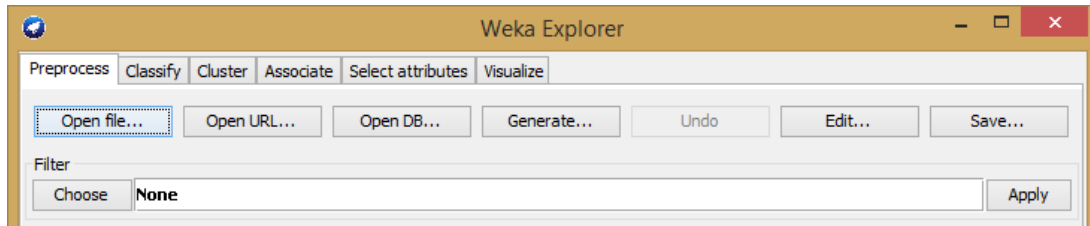


Figura 11 – Abas do ambiente Explorer do WEKA.

Fonte: Autoria própria.

filtragem, seleção e simples visualização dos atributos e de algumas de suas características básicas.

Aqui, portanto, pode-se carregar a amostra a ser analisada, e verificar o seu nome, o número de instâncias e uma lista com os atributos calculados para extração de informações, onde se pode selecioná-los e excluí-los. Ao selecionar um atributo específico, aparece uma segunda lista com informações sobre o mesmo, como seu máximo, mínimo, média e desvio padrão, além de um histograma de sua distribuição.

Todavia, o recurso mais relevante da aba **Preprocess** é a possibilidade de aplicação de uma variada gama de filtros supervisionados e não-supervisionados sobre o banco de dados. Os filtros disponíveis no WEKA incluem a discretização, padronização e inserção de ruído na amostra, além da remoção de atributos pouco relevantes para análise com base na sua variância.

3.2.2 Seleção de atributos

A aba **Select Attributes** permite a configuração e aplicação de técnicas de avaliação e pesquisa de atributos, visando à seleção daqueles que são mais relevantes dentro do conjunto de dados disponíveis para sua classificação. Os algoritmos disponíveis incluem técnicas de ranqueamento dos atributos, e ainda há a possibilidade de executá-los por diversas iterações através de uma técnica de validação cruzada a fim de garantir a validade da análise.

Dentre os métodos de avaliação dos atributos, destacam-se aqueles baseados no cálculo da entropia dos mesmos, como o **InfoGainAttributeEval** e o **GainRatioAttributeEval**. O primeiro calcula o ganho de informação dado por cada atributo ao processo de classificação, que é dado pela Equação (14):

$$IG(Y, X) = H(Y) - H(Y|X) \quad (14)$$

em que:

- $IG(Y, X)$ é o ganho de informação dado pelo atributo X para a classificação em relação à distribuição Y (varia entre um máximo de 1 e mínimo de 0);
- $H(Y)$ é a entropia da distribuição Y ;

- $H(Y|X)$ é a média ponderada das entropias das subclasses de Y para cada valor (ou classe de valores) do atributo X .

A entropia H , por sua vez, é dada por:

$$H(Y) = - \sum_{i=1}^N p_i \log_2 p_i \quad (15)$$

onde:

- p_i é a probabilidade da classe i dentro da distribuição Y ;
- N é o número total de classes distintas;
- $H(Y) \in [0; 1]$.

Quanto maior a entropia, portanto, maior a variedade dos elementos de uma distribuição em relação às classes componentes desta. Assim sendo, se a variedade do conjunto Y é total ($H(Y) = 1$), mas através de um teste X para um atributo é possível separar Y em conjuntos disjuntos, cada um apresentando somente uma de suas classes, temos $H(Y|X) = 0$, e portanto o ganho de informação para esse atributo é máximo ($IG(Y, X) = 1$).

O avaliador `GainRatioAttributeEval`, por sua vez, baseia-se no cálculo da razão do ganho de informação, como mostra a Equação (16):

$$IGR(Y, X) = \frac{H(Y) - H(Y|X)}{H(X, a)} \quad (16)$$

onde $H(X, a)$ representa a entropia da distribuição das a instâncias do atributo X . Também é chamada de informação intrínseca de um atributo e pode ser dada por:

$$H(X, a) \equiv IntI(X, a) = - \sum_{i=1}^a \frac{|X_i|}{|X|} \log \left(\frac{|X_i|}{|X|} \right) \quad (17)$$

Quanto maior a informação intrínseca de um atributo, mais amostras são necessárias para um processo de classificação, e portanto menor é a sua relevância (WITTEN; FRANK; HALL, 2005). Pode-se afirmar, portanto que o `GainRatioAttributeEval` é mais confiável em seu resultado que o `InfoGainAttributeEval`, haja vista a correção que ele apresenta em relação ao fator de $IntI$.

Um exemplo simples

Um certo conjunto X é composto de quatro elementos, $X = \{x_1, x_2, x_3, x_4\}$. Suponha-se, sem perda de generalidade, que os elementos x_i pertencem a duas classes distintas, aqui denominadas I e II, e que três atributos binários, A, B e C, podem ser associados a esses elementos. Um estudo mostrou então que $H(X) = 1$, de modo que as quatro amostras estão dispostas como na Tabela 4:

Tabela 4 – Exemplo: Conjunto A, seus elementos e respectivas características

Elem.	A	B	C	Classe
x_1	1	1	1	I
x_2	1	1	0	I
x_3	0	0	1	II
x_4	1	0	0	II

Faça-se então uma divisão do conjunto segundo o atributo A. Ter-se-á então dois subconjuntos: $X_0^A = \{x_3\}$, com $H(X_0^A) = 0$; e $X_1^A = \{x_1, x_2, x_4\}$, com $H(X_1^A) = 0,92$. O cálculo de IG para essa divisão segue:

$$IG(X, A) = H(X) - H(X|A) = 1 - \frac{3}{4}(0,92) - \frac{1}{4}(0) = 0,3112. \quad (18)$$

Procedendo de maneira análoga em relação a B, tem-se: $X_0^B = \{x_3, x_4\}$, com $H(X_0^B) = 0$; e $X_1^B = \{x_1, x_2\}$, com $H(X_1^B) = 0$. Segue:

$$IG(X, B) = H(X) - H(X|B) = 1 - \frac{2}{4}(0) - \frac{2}{4}(0) = 1. \quad (19)$$

Para C, tem-se: $X_0^C = \{x_2, x_4\}$, com $H(X_0^C) = 1$; e $X_1^C = \{x_1, x_3\}$, com $H(X_1^C) = 1$. Segue:

$$IG(X, C) = H(X) - H(X|C) = 1 - \frac{2}{4}(1) - \frac{2}{4}(1) = 0. \quad (20)$$

Isto posto, depreende-se da análise realizada que para separação do conjunto entre as classes I e II, o atributo que teve maior valor foi B, com ganho de informação 1 (total), isto é, com base em B é possível separar perfeitamente os elementos do conjunto X entre as classes I e II. Por outro lado, A tem valor intermediário, na medida em que seus subconjuntos-filhos consistem numa separação imperfeita das classes, e C teve o pior desempenho entre os três: utilizando-o para separar o conjunto X, não houve nenhum ganho em termos da classificação dos atributos.

3.2.3 Visualização dos dados

Na aba *Visualize*, são apresentados gráficos de dispersão 2-D ou 3-D relacionando as instâncias de 2 ou 3 atributos selecionados, e mostrando as diferentes classes de dados com cores diferentes.

Apesar de não interferir no resultado das técnicas de avaliação, a visualização do banco de dados é útil para confirmar qualitativamente a relação entre os mesmos e as classes dos dados. A Figura 12 mostra um exemplo desse recurso.

Nela, os valores de fator de pico da corrente nas fases A e B para cada teste estão distribuídos em um gráfico de dispersão, onde os pontos em azul representam os motores saudáveis, e em vermelho estão colocados aqueles relacionados aos motores com falha.

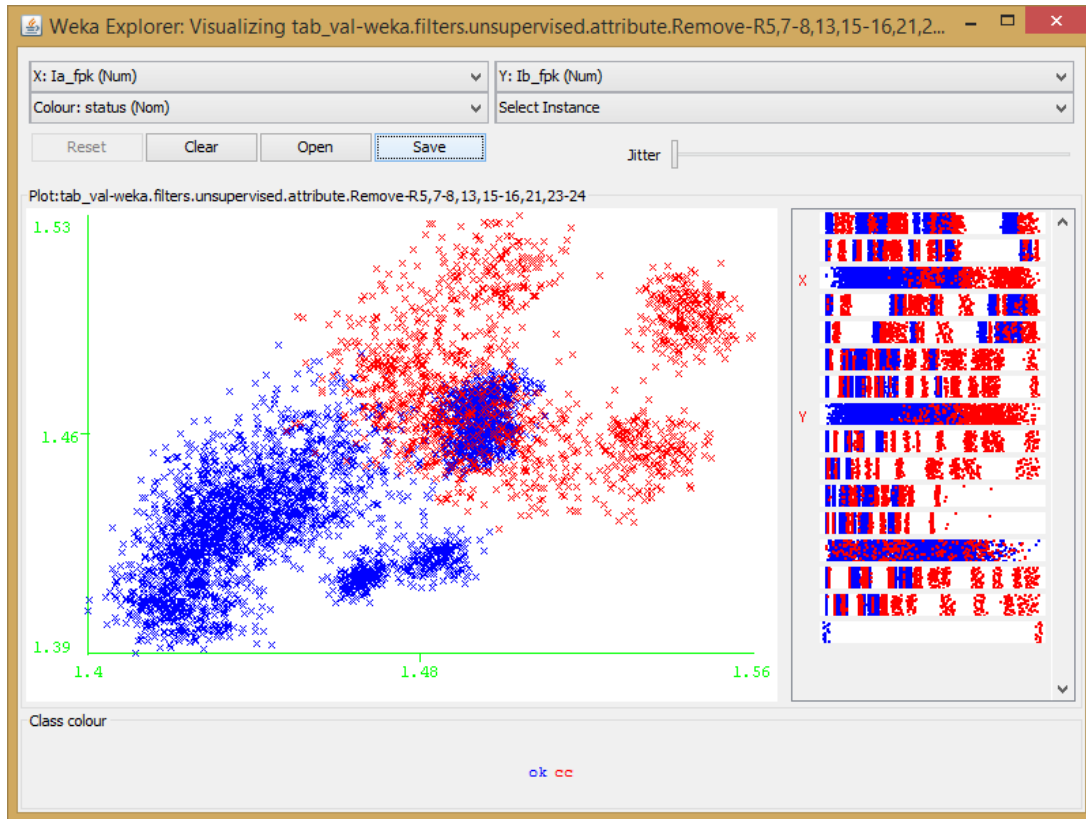


Figura 12 – Exemplo de gráfico de dispersão gerado pelo WEKA.

Fonte: Autoria própria.

De fato, é possível verificar visualmente que há alguma correlação entre esses fatores e o status do motor, e a magnitude dessa correlação poderá ser vista futuramente quando do desenvolvimento do sistema de classificação.

3.2.4 Técnicas de classificação

Por fim, na aba **Classifier**, o WEKA permite a configuração e execução de algoritmos classificadores sobre o banco de dados. Aqui também é possível utilizar a técnica de validação cruzada, e ainda utilizar um conjunto de teste diferente daquele usado para treinamento do algoritmo de classificação. É possível também verificar a matriz de confusão resultado do processo de classificação.

Os classificadores disponibilizados pelo WEKA incluem diversas técnicas como: RNA do tipo *Perceptron* Multicamadas (PMC) e função de base radial (RBFNN); *Support Vector Machine* (SVM); classificadores bayesianos e logísticos, etc. Vale ressaltar que, no caso de serem testados diferentes algoritmos nesta etapa, o programa não apaga os testes anteriores para fazer novos: do contrário, uma lista com o histórico de testes é gerada, de forma a permitir ao usuário a comparação entre eles.

Sedimentadas as bases teóricas e conhecendo-se as ferramentas do WEKA, foi possível prosseguir para o estudo da técnica de classificação através das Redes Neurais Artificiais

(RNA).

3.3 Considerações finais

Neste capítulo, discorreu-se sobre as estatísticas que podem ser aplicadas sobre os sinais de corrente trifásica que foram adquiridos na bancada de experimentos. Estes atributos aqui explicitados são candidatos a entradas do classificador neural, na medida em que se mostrarem efetivamente relevantes para a solução do problema, conforme a seleção de atributos aqui também discutida.

Passou-se então à investigação das RNA, a fim de estudar o seu funcionamento e principais características, que poderiam vir a consistir numa solução efetiva para o problema proposto, como ver-se-á a seguir.

Redes Neurais Artificiais

Segundo Silva, Spatti e Flauzino (2010), Redes Neurais Artificiais são “modelos computacionais inspirados nos mecanismos de aprendizagem do cérebro humano”, isto é, que tentam emular a forma como o cérebro resolve problemas. De fato, uma rede neural artificial pode ser considerada um modelo simplificado de uma porção do sistema nervoso de seres vivos.

Desde sua concepção, as RNA encontraram aplicações em diversas áreas do conhecimento, que podem ser divididas nas seguintes categorias principais:

1. **Aproximação de funções:** Mapeamento da relação entre as variáveis de um sistema, a partir de um conjunto conhecido de valores em seu domínio. Em geral, aplicam-se a processos cuja modelagem por técnicas convencionais são de difícil obtenção (problemas não-lineares);
2. **Controle de processos:** Identificação das ações de controle que devem ser aplicadas sobre um sistema em um determinado estado a fim de alcançar os requisitos de qualidade, eficiência e segurança do processo;
3. **Classificação de padrões:** Associação de uma amostra de entrada para uma das classes de saída previamente definidas, dentro de um conjunto discreto e conhecido;
4. **Agrupamento de dados (clusterização):** Identificação e detecção de similaridades e particularidades entre as diversas amostras do processo, objetivando-se agrupamento destas em grupos semelhantes;
5. **Sistemas de previsão:** Estimativa de valores futuros de uma série, a partir de medidas prévias;
6. **Otimização de sistemas:** Otimização (em ponto de mínimo ou máximo) de uma função objetivo, obedecendo-se às eventuais restrições impostas nas condições de contorno do problema;

7. **Memórias associativas:** recuperação de padrões corretos a partir de uma amostra incompleta ou distorcida.

Por tratar-se neste trabalho de um problema típico de classificação de padrões, escolheu-se para estudo as redes neurais, o que se deveu certamente a algumas de suas principais características, a saber (SILVA; SPATTI; FLAUZINO, 2010):

1. **Capacidade de aprendizado:** através de um algoritmo de treinamento, a RNA consegue relacionar as variáveis de entrada às saídas esperadas;
2. **Capacidade de generalização:** após o treinamento, a rede é capaz de extrapolar dados para valores não presentes na amostra inicial;
3. **Tolerância a falhas:** devido ao grande número de interconexões, uma falha localizada em geral não compromete o desempenho geral da rede;
4. **Facilidade de prototipagem:** a implementação da rede treinada em hardware é consideravelmente fácil, pois seus resultados são obtidos através de operações matemáticas bastante simples.

4.1 O neurônio artificial

Por ter como inspiração o modelo biológico para a solução dos problemas supracitados, também as redes neurais artificiais se compõem de unidades menores de processamento, os neurônios artificiais. Sua concepção, inclusive, envolveu o trabalho de Warren McCulloch e Walter Pitts, respectivamente um psiquiatra e neuroanatomista e um matemático (MCCULLOCH; PITTS, 1943). Já nesse modelo inicial, o neurônio artificial, assim como o biológico, consistia de uma unidade básica de processamento que gera suas respostas baseada na ponderação de suas entradas, assim que esse valor supera o seu potencial de ativação (HAYKIN, 1999).

No modelo de McCulloch e Pitts (1943), as entradas x_i , provenientes dos sinais adquiridos e/ou operações feitas sobre eles, análogas dos neurotransmissores, são ponderadas por seus respectivos pesos sinápticos w_i . Ao chegar à unidade de processamento (o “corpo celular”), é feita uma combinação linear (Σ) entre as entradas ponderadas, e caso elas ultrapassem o limiar de ativação θ , um sinal de saída (potencial de ativação u) é disparado no “axônio” do neurônio. Esse sinal é então modulado por uma função de ativação $f(\cdot)$, a fim de limitar sua resposta aos valores adequados à situação-problema. Um resumo visual desta operação pode ser visto na Figura 13.

Em termos matemáticos, tem-se:

$$u = \sum_i w_i \cdot x_i - \theta \quad (21)$$

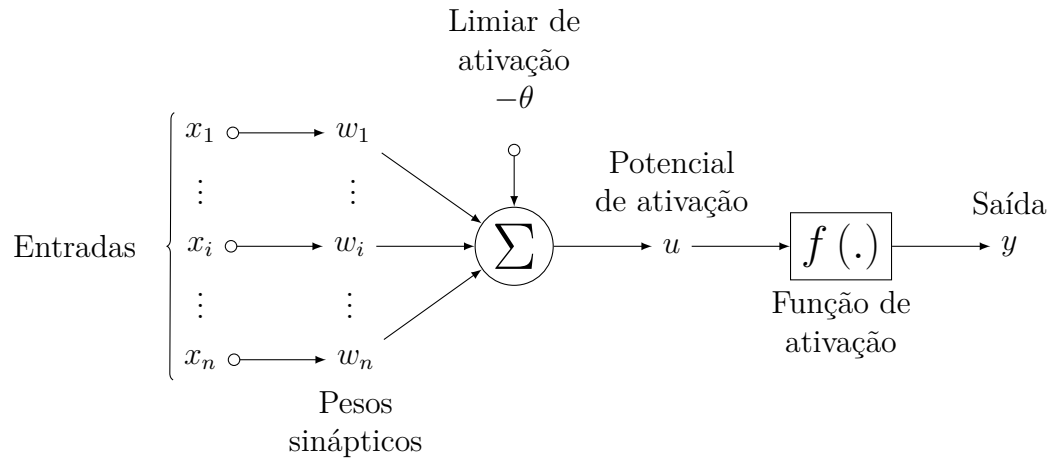


Figura 13 – Desenho esquemático de um neurônio artificial.

Fonte: Autoria própria.

$$y = f(u) \quad (22)$$

4.2 Redes *Perceptron* Multicamadas para problemas de classificação de padrões

Uma RNA divide-se em geral em dois tipos básicos: simples camada e múltiplas camadas. Sendo invariavelmente composta de neurônios, estes são categorizados em três conjuntos (camadas): o de entrada, responsável por receber os dados a serem processados; os ocultos, que realizam a extração da informação contida nesses dados; e o de saída, responsável pela apresentação dos resultados do processamentos das camadas anteriores (HAYKIN, 1999; SILVA; SPATTI; FLAUZINO, 2010).

Nas redes com arquitetura do tipo *feedforward* (FFNN), isto é, com alimentação dos dados sempre à frente, as saídas dos neurônios de cada camada são sempre repassadas como entradas dos neurônios das camadas seguintes. Um modelo simplificado desse tipo de estrutura pode ser visto na Figura 14. As redes PMC obedecem a esse tipo de arquitetura, possuindo pelo menos uma camada neural escondida (SILVA; SPATTI; FLAUZINO, 2010).

4.2.1 Processos de treinamento

O treinamento de redes PMC baseia-se essencialmente no algoritmo *backpropagation* (BP), que se compõe de duas fases. São elas:

1. A “**propagação adiante**” (passo *forward*), em que as entradas x_i de uma amostra do banco de dados de treinamento e as saídas de cada camada são sucessivamente calculadas. O objetivo desta fase é avaliar o estado atual da rede, observando como

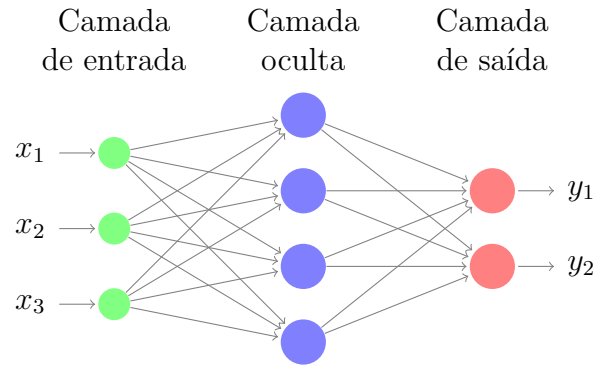


Figura 14 – Desenho esquemático simplificado de uma RNA do tipo FFNN.

Fonte: Autoria própria.

ela responde a uma entrada cuja resposta desejada é conhecida, e calculando os erros nas saídas;

2. A “**propagação reversa**” (passo *backward*), em que são feitos ajustes nos pesos sinápticos w_i e limiares de ativação θ_i de cada neurônio, camada a camada, desde a de saída em direção à de entrada.

As aplicações sucessivas dos dois passos para as amostras do conjunto de treinamento do banco de dados permitem um auto-ajuste dos pesos sinápticos e limiares a cada iteração, o que resulta na diminuição dos erros de resposta da rede (SILVA; SPATTI; FLAUZINO, 2010). Para melhor compreensão do algoritmo, defina-se um conjunto de variáveis auxiliares a partir do desenho esquemático anteriormente apresentado, conforme ilustra a Figura 15:

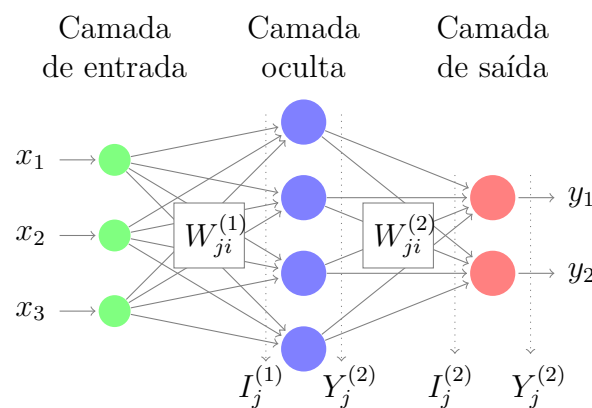


Figura 15 – FFNN e a notação para o algoritmo *backpropagation*.

Fonte: Autoria própria.

em que:

- $W_{ji}^{(L)}$ denota as matrizes que contêm os pesos sinápticos que conectam o j -ésimo neurônio da camada L ao i -ésimo neurônio da camada $L - 1$;

- $I_j^{(L)}$ são os vetores que contêm as entradas ponderadas $x_i \cdot w_i$ para o j -ésimo neurônio da camada L , dados por:

$$I_j^{(L)} \triangleq \sum_{i=0}^n W_{ji}^{(L)} \cdot Y_i^{(L)} \quad (Y_i^{(0)} = x_i) \quad (23)$$

- $Y_j^{(L)}$ denota os vetores que contêm as saídas y_i do j -ésimo neurônio da camada L . Sendo a função de ativação de cada neurônio dada por $f(\cdot)$, tem-se:

$$Y_j^{(L)} \triangleq f(I_j^{(L)}) \quad (24)$$

sendo $f(\cdot)$ uma função diferenciável em todo o seu domínio.

Em resumo ao complexo rigor matemático do BP, detalhado em Silva, Spatti e Flauzino (2010) e Haykin (1999), pode-se afirmar que no passo *forward*, são calculados os $I_j^{(L)}$ e $Y_j^{(L)}$, a partir das entradas da amostra e dos pesos atualmente existentes na rede. As saídas calculadas são então comparadas aos valores desejados para a amostra, e uma função erro é definida. Para que se possa determinar um mínimo local para essa função a fim de encontrar uma possível resposta para o problema, explora-se o gradiente da mesma, o que envolve a derivada $f'(\cdot)$ da função de ativação neuronal (daí a necessidade desta ser diferenciável). Esse passo também dá nome à técnica, chamada de Gradiente Descendente (GD). Por fim, a matriz $W_{ji}^{(L)}$ é ajustada no sentido de minimização do erro, o que resulta no algoritmo do passo *backward*:

$$W_{ji}^{(L)} \leftarrow W_{ji}^{(L)} + \eta \cdot \delta_j^{(L)} \cdot Y_i^{(L-1)} \quad (25)$$

em que:

- η é a taxa de aprendizagem do BP, que define o quanto cada iteração do processo de treinamento irá impactar no ajuste dos pesos sinápticos;
- $\delta_j^{(L)}$ é o gradiente local em relação ao j -ésimo neurônio da camada L , e portanto relaciona o ajuste local que deve ser feito em cada unidade de processamento no sentido de atingir o mínimo para o conjunto.

Desde sua concepção por Humelhart, Hinton e Williams (1986), o algoritmo de *back-propagation* aplicado ao aprendizado de RNA já sofreu diversas modificações com vistas à sua otimização, não apenas em termos da qualidade da resposta final, mas também com relação à eficiência no processo de convergência. A maioria desses aperfeiçoamentos referem-se à taxa de aprendizagem da rede e ao uso do gradiente local, além da inclusão de um termo de *momentum* (α).

O *momentum* funciona como um fator dinâmico na taxa de aprendizagem: à medida em que o sistema aproxima-se da resposta, o que é verificado pelo quanto os pesos foram

alterados entre iterações subsequentes, a taxa de aprendizagem é reduzida. Assim, a velocidade do algoritmo é aumentada quando ele está longe do resultado, e diminuída quando aproxima-se dele para fins de otimização. Esse método de treinamento é também chamado de Gradiente Descendente com Momento (GDM).

O *resilient propagation* (RPROP), por sua vez, foi proposto a fim de superar as desvantagens do GD puro, modificando a taxa de aprendizado não com base somente no módulo da função erro, mas sim enfatizando o seu sinal. Com isso, ele apresenta a vantagem de não sofrer influência de eventuais discrepâncias advindas do cálculo da derivada do erro (RIEDMILLER; BRAUN, 1993).

Outro algoritmo de treinamento que se destaca é o método de Levenberg-Marquardt (LM). Idealizado por Hagan e Menhaj (1994), ele foi inspirado no método de Newton-Raphson no sentido de estimar o ponto de mínimo da função erro quadrático — como no caso do Método dos Mínimos Quadrados (MMQ) —, envolvendo para tal o cálculo da Hessiana (matriz de derivadas parciais de segunda ordem) desta. Apesar de não ter uma velocidade de convergência tão grande quanto o RPROP, o algoritmo LM ainda é mais rápido que o BP convencional e apresenta uma grande capacidade de minimizar o erro da rede (SILVA; SPATTI; FLAUZINO, 2010).

4.2.2 Generalização e *Overfitting*

Como dito ainda no início deste Capítulo 4, ao final do processo de treinamento as Redes Neurais Artificiais devem ser capazes não apenas de resolver o problema a ela apresentado, mas também generalizar a solução encontrada para pontos inicialmente desconhecidos, isto é, não constantes da base de dados de treinamento. Todavia, há casos em que a rede responde de maneira excelente para as amostras com a qual ela foi treinada, e apresenta erros significativos para as demais. Nesse caso, diz-se que a rede sofreu memorização excessiva, ou *overfitting* (SILVA; SPATTI; FLAUZINO, 2010).

Para evitar que isso ocorra, dois procedimentos básicos podem ser aplicados. O primeiro, e mais simples deles, é diminuir a complexidade da RNA utilizada, reduzindo o número de camadas ocultas e/ou de seus neurônios, afinal, a complexidade da rede está diretamente ligada à complexidade dos problemas que ela é capaz de resolver (SILVA; SPATTI; FLAUZINO, 2010; SUETAKE, 2012). O simples aumento do número de neurônios, portanto, não garante um melhor desempenho: do contrário, a partir de certo ponto, a adição de novas unidades neuronais piora o resultado da técnica em termos de generalização.

Uma outra forma de combater o *overfitting* é utilizando o chamado *early-stopping*, que se soma às técnicas de validação cruzada durante o processo de treinamento do PMC. A validação cruzada consiste de utilizar amostras de conjuntos disjuntos para treinamento e validação: enquanto as amostras do primeiro conjunto são apresentadas à rede nos passos *forward* e *backward* do treinamento, as do segundo o são apenas no passo *forward*, para fins de monitoramento do erro de resposta (MORGAN; BOURLARD, 1989). O que se verifica

na prática é que com as crescentes iterações, o erro no conjunto de treinamento tende a decair, enquanto o do conjunto de validação o acompanha, a uma certa distância. A partir de determinado ponto, porém, o erro de validação começa a crescer, indicando o *overfitting*. Nesse momento então, encerra-se o processo de treinamento antes de atingir-se um erro mínimo — aqui configura-se o *early-stopping*.

4.2.3 Aspectos de aplicação

Nos problemas de classificação de padrões, o objetivo é, a partir de um conjunto de sinais e/ou medidas aquisitadas da planta em estudo, diferenciar amostras dentro de conjuntos previamente definidos, ou ainda, apontar a pertinência de uma amostra a uma ou mais classes discretas correspondentes a estados verificados no sistema real. Para tal, a estratégia mais difundida em redes neurais é a utilização de saídas binárias, em que cada neurônio de saída representa uma classe, e cada resposta indica o pertencimento ou não da amostra de entrada àquela classe (configuração *one of c-classes*) (SILVA; SPATTI; FLAUZINO, 2010).

Pelas suas características discretizantes, seria natural o uso de uma função de ativação do tipo degrau de Heavyside ou uma função sinal para os neurônios de saída, a fim de que eles possam definir de forma clara a superfície de delimitação entre as classes. No entanto, é importante destacar que a função de ativação deve necessariamente ser diferenciável em todo o seu domínio para que possa ocorrer o ajuste sináptico, e ambas as citadas apresentam descontinuidade para $x = 0$. Nesse caso, pode-se utilizar as funções logística e tangente hiperbólica, com efeito semelhante. Compare o resultado na Figura 16.

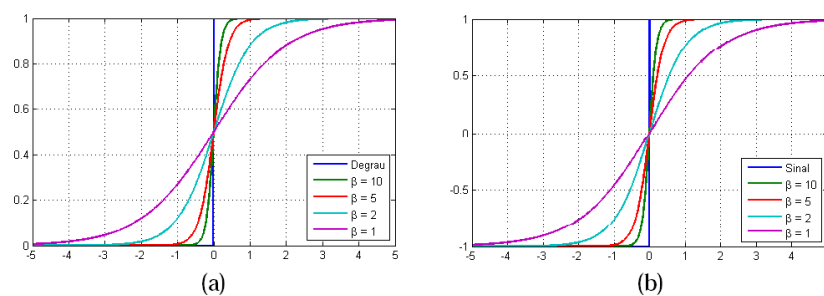


Figura 16 – Funções de ativação: (a) logística e degrau; (b) tangente hiperbólica e sinal.

Fonte: Autoria própria.

Observa-se que, variando-se os valores para β , parâmetro de ajuste da inclinação das funções logística e tangente hiperbólica no ponto de inflexão $x = 0$, pode-se recuperar uma função degrau ou sinal, conquanto elimina-se a descontinuidade na origem cartesiana.

Ademais, em problemas de classificação, essas funções de ativação também ganham valor nas camadas intermediárias do PMC. Além da já conhecida diferenciabilidade, elas apresentam uma faixa ativa, nas vizinhanças da origem, e uma de saturação, quando afastam-se dela. Enquanto os neurônios da camada de saída operam na faixa de saturação,

meramente incluindo ou excluindo a amostra desta ou daquela categoria, as unidades neuronais das camadas antecedentes trabalham na faixa ativa, isto é, aproveitam-se dos valores intermediários para aprimorar a sua resposta. Caso todas as camadas operassem na faixa de saturação, a estrutura neuronal meramente se assemelharia à de uma árvore de decisões.

Outra vantagem dessas funções diferenciáveis é a sua limitação entre valores sempre inferiores a 1 em módulo, o que permite à rede trabalhar sempre com valores pequenos e evitar que entradas totalmente discrepantes provoquem instabilidade nos cálculos internos a ela.

De fato, o ambiente computacional MATLAB, em sua caixa de ferramentas de redes neurais (Neural Network ToolboxTM), já apresenta ao usuário a possibilidade de trabalhar com uma RNA otimizada para problemas de reconhecimento de padrões, através do comando `patternnet(hiddenSizes,trainFcn)`. O comando cria um objeto do tipo “rede” na interface de usuário, na configuração FFNN, com as seguintes características básicas, por padrão:

1. **Função de ativação:** os neurônios de todas as camadas apresentam função de ativação do tipo `tansig` (tangente sigmóide);
2. **Funções de processamento:** as entradas são mapeadas para valores no intervalo $[-1; 1]$ pela função `mapminmax`, a fim de evitar a saturação dos neurônios das primeiras camadas. Além disso, a função `removeconstantrows` remove do banco de dados de treinamento e validação as amostras com valores constantes, evitando duplicidade e valores negativos;
3. **Configuração das camadas ocultas:** é dada pelo parâmetro `hiddenSizes`, passado ao comando. Pode-se definir já de início qual será o tamanho e a quantidade de camadas ocultas da rede;
4. **Algoritmo de treinamento:** é indicado através do parâmetro `trainFcn`. O MATLAB dispõe de diversos algoritmos já implementados, com parâmetros alteráveis pelo usuário, incluindo o GD (`traingd`), o GDM (`traingdm`), o RPROP (`trainrp`) e o de Levenberg-Maquardt (`trainlm`).

Após a criação e configuração de uma `patternnet` na interface de comando, é possível visualizar graficamente a rede criada através do comando `view`. Como forma de exemplo, configurou-se uma rede para três entradas e duas saídas, com quatro neurônios em uma camada escondida, análoga ao exemplo da Figura 14. O resultado do comando `view` está ilustrado na Figura 17.

De posse de uma bancada de testes apropriada para realização dos ensaios e aquisição dos dados, conhecendo as técnicas de extração da informação contida nesses sinais e as

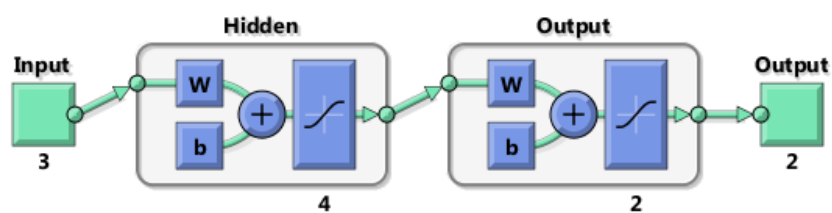


Figura 17 – Visualização de exemplo de FFNN para reconhecimento de padrões no MATLAB.

Fonte: Autoria própria.

de classificação dos motores em teste através da utilização das Rede Neural Artificial, prosseguiu-se para a etapa de desenvolvimento do projeto.

Desenvolvimento do Sistema Neural para Detecção de Curto-Circuito em MIRGE

5.1 Estrutura básica do sistema

O sistema de detecção de falhas aqui proposto compõe-se basicamente de três etapas sequenciais, a saber:

1. **Amostragem:** serão realizados testes em bancada, em diferentes condições de acionamento e carregamento, tendo em vista a obtenção das amostras de dados, em especial, as correntes trifásicas de linha do motor;
2. **Extração de características:** será feita a extração da informação contida no sinal de corrente, através do cálculo de um conjunto de estatísticas pré-definidos sobre o sinal;
3. **Pré-processamento:** nesta etapa será analisada a relevância de cada característica para a classificação das amostras, e subsequente seleção dos atributos mais relevantes para a próxima etapa;
4. **Teste das topologias candidatas:** por último, os dados pré-processados serão disponibilizados para treinamento e teste das topologias de RNA candidatas, a fim de que se possa encontrar uma solução ótima para o problema, em termos de acurácia e tempo de processamento.

5.2 Rotina de testes

Na etapa inicial desta fase do trabalho, o foco foi a montagem de uma extensa base de dados que permitisse a simulação em laboratório das diversas condições de uso e aplicação

às quais esse tipo de máquina encontra-se submetido. Para tal, ligou-se o motor em teste sob diferentes condições de:

- **Curto-circuito:** foram testados um motor normal, em condições originais de fábrica, bem como o MIRGE modificado conforme descrito na Seção 2.2, em condições de curto-circuito estático de 1, 3, 5 e 10% entre bobinas de mesma fase;
- **Acionamento:** os motores foram acionados tanto diretamente da rede, em 220V e 60Hz, quanto através de um inversor de frequência, nas velocidades de 40, 45, 50, 55 e 60Hz, com a tensão obedecendo a relação constante V/f a fim de manter também constante o fluxo e o torque no rotor, conforme configuração padrão do inversor;
- **Carga:** variando-se a corrente de campo I_f do GCC, reproduziram-se diferentes condições de conjugado para o motor, desde a subcarga até a sobrecarga. O MIRGE em teste tinha torque nominal de 4,0 N.m, e foi operado em 0,5, 1,0, 1,5, 2,0, 2,5, 3,0, 3,5, 4,0 e 5,0 N.m, varrendo desde 12,5% até 125% da condição nominal.

Todas as combinações possíveis das condições supracitadas foram testadas no motor modificado, num total de 270 ensaios. Para o motor original, onde não era possível realizar os testes de curto-circuito, foram feitos mais 54 ensaios. No organograma da Figura 18 é possível ver um resumo dessa operação.

Todos os 324 ensaios foram realizados quando a máquina já se apresentava em regime permanente, não tendo sido alvo deste trabalho a observação do comportamento em regime transitório. Nos testes em condições normais (sem curto-circuito), o tempo de amostragem de cada aquisição foi de 10 s (dez segundos). Nas situações em que foi simulado a falha, o tempo de amostragem foi reduzido para 2 s (dois segundos), a fim de preservar a integridade da máquina.

Os ensaios eram controlados via PC através do MATLAB, utilizando o código desenvolvido por Santos (2012) e revisado por Gongora (2013) e Broniera Jr., Goedel e Graciola (2013), e disponível na bancada de testes do Laboratório de Sistemas Inteligentes (LSI) da Universidade Tecnológica Federal do Paraná Campus Cornélio Procopio (UTFPR/CP). Por ele, é possível controlar a placa de aquisição *NI USB-6221*, em termos de suas entradas e saídas digitais e analógicas, taxa de amostragem, *offset* e ganho de cada canal. Ao se executar o código, um contator era acionado para fechar o contato do curto-circuito na bobina e o registro da aquisição era então iniciado imediatamente, pelo tempo assinalado.

O resultado do ensaio era exibido na interface de usuário por meio de gráficos que permitiam confirmar visualmente se o teste havia sido feito com sucesso, exibindo: tensões trifásicas, correntes trifásicas, o torque instantâneo e a velocidade do motor. Todo o resultado era então exportado para arquivos a fim de compor o banco de dados, devidamente identificado para posterior controle.

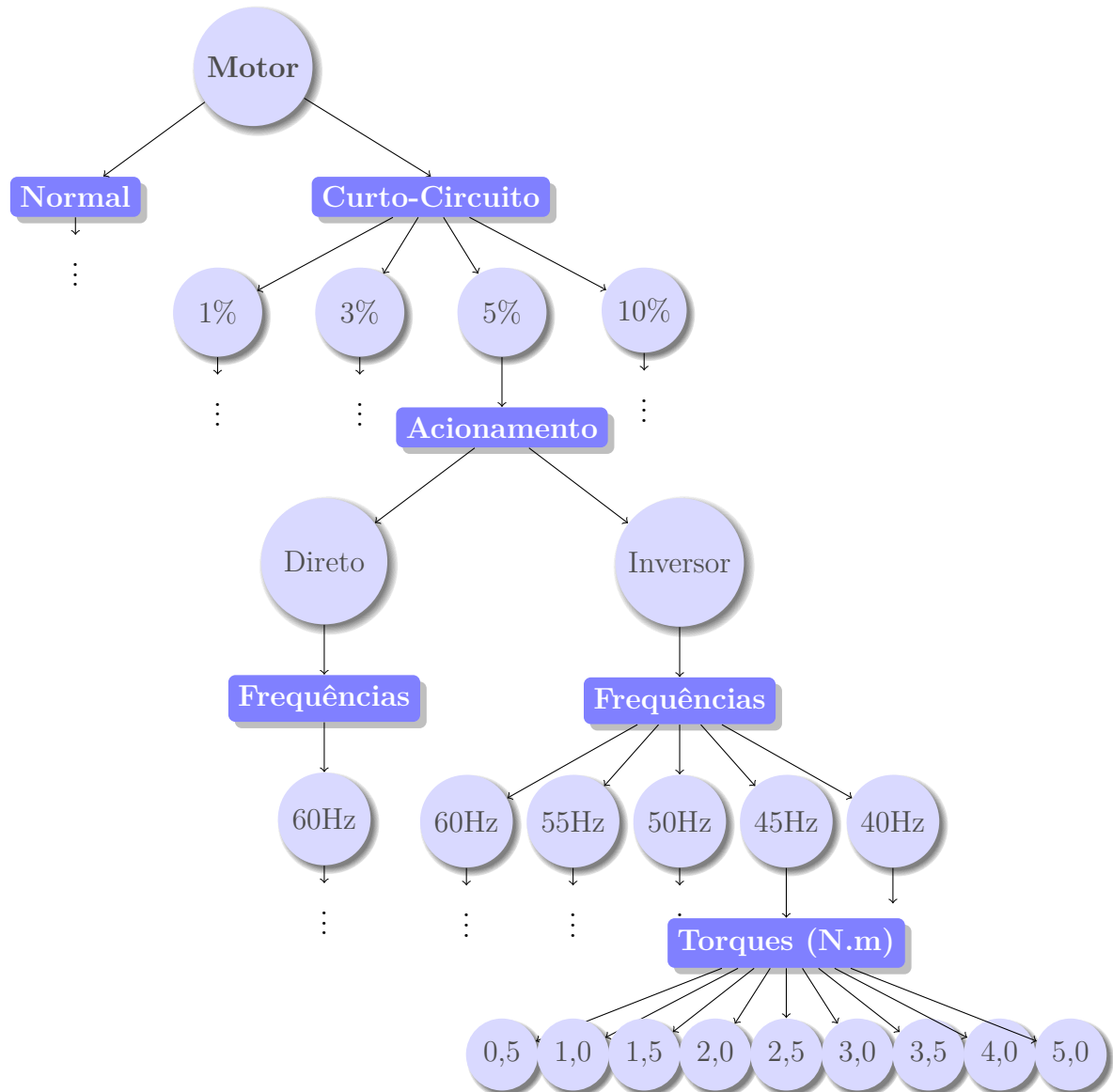


Figura 18 – Organograma simplificado dos testes realizados.

Fonte: Autoria própria.

5.3 Extração de características

A seguir, os arquivos gerados na etapa anterior foram novamente importados e lidos na interface MATLAB. O objetivo desta etapa foi o cálculo dos atributos dos sinais aquisitados, a fim de promover a extração das suas características e gerar as entradas que alimentarão o sistema de classificação neural.

Para tal, foi criada uma rotina que faz a leitura do banco de dados, importando os arquivos correspondentes às entradas de um pequeno menu, com vistas a facilitar a obtenção de bases de dados contendo conjuntos selecionados, por exemplo, para selecionar todas as amostras de atributos em que os motores foram acionados diretamente da rede elétrica.

O sistema de aquisição gera, para cada ensaio, um conjunto contendo as tensões e

correntes trifásicas, o torque instantâneo e a velocidade do motor. No entanto, apenas as correntes foram utilizadas, para que sua assinatura fosse descrita, numa técnica de MCSA no domínio do tempo. Separou-se então as correntes de linha, e para cada uma delas foram obtidos os pontos de cruzamento com a origem das ordenadas, isto é, os seus zeros.

O objetivo desse procedimento foi o cálculo da frequência instantânea real da corrente. De posse da frequência, foi possível normalizar os valores de corrente para as diferentes frequências de acionamento quando se utiliza do inversor, já que há o controle de V/f . Isso também permitiu a determinação o tamanho da janela de cálculo dos atributos, iniciando-a e terminando-a sempre ao final de um número inteiro de períodos do sinal, precaução que garante que a extração das informações fique menos sujeita às variações normais advindas da discretização inserida no problema pela DAQ.

A partir daí, definiu-se uma janela de três períodos (6π rad) do sinal, também visando a diminuir a influência da discretização, além de ampliar a quantidade de amostras disponíveis para treinamento e aumentar a robustez da rede neural. Em cada janela, o cálculo das estatísticas envolve a análise de no mínimo 250 pontos, para os casos em que a frequência de acionamento era de 60 Hz, chegando a 375 pontos em 40 Hz.

Os atributos são armazenados em uma matriz, juntamente com uma *flag* que identifica o status do motor em estudo, isto é, se ele encontra-se em curto-circuito ou não, já preparando o banco de dados para o treinamento da RNA. Observe-se que o nível do curto não foi indicado: o sistema deve ser capaz apenas de indicar a falha, não havendo prioridade na determinação de sua magnitude. A algoritmo gera então um banco de dados em formato `.arff`, pronto para a importação e análise no ambiente WEKA.

5.4 Pré-processamento dos dados

Na aba `Preprocess` do ambiente `Explorer` do WEKA, realizou-se a importação do banco de dados gerado pelo MATLAB, e a aplicação de filtros sobre o mesmo. O primeiro desses filtros foi o `RemoveUseless`, de caráter não-supervisionado e que age sobre os atributos. Sua função é remover do banco de dados aqueles atributos que não têm serventia para o estudo do problema, por terem uma variância muito grande ou nula. Utilizou-se o filtro na sua configuração padrão, em que ele elimina variâncias percentuais acima de 99% e as nulas. Já nessa etapa, a média (m_1) foi descartada, por permanecer sempre nula, reduzindo a quantidade de atributos a serem repassados para o sistema de classificação.

Além do `RemoveUseless`, outros filtros foram utilizados. Dentre eles, o `Normalize`, que normaliza todos os valores numéricos no *dataset* para um intervalo definido. No caso, utilizou-se o intervalo $[-1; 1]$, o mesmo da função de ativação tangente hiperbólica dos neurônios da RNA, a fim de evitar a saturação destes. Em alguns testes, utilizou-se também o filtro `AddNoise` para adicionar um ruído branco a parte da amostra, para fins

de teste da robustez do sistema de classificação.

5.5 Seleção dos atributos

Após a conveniente filtragem dos dados, realizou-se um ranqueamento dos atributos visando uma posterior seleção dos mesmos, na aba `Select Attributes` do WEKA. Para tanto, utilizou-se o método de pesquisa `Ranker` em conjunto com os avaliadores de atributos `GainRatioAttributeEval` e `InfoGainAttributeEval` disponíveis na interface do programa, conforme visto na Seção 3.2.2.

Nesta etapa foram classificados os 24 atributos que ainda compunham o *dataset* — são oito estatísticas para cada uma das correntes de linha, se considerarmos que a média já foi eliminada. Os resultados para ambos os métodos foram consideravelmente semelhantes, e podem ser vistos nas Tabelas 5 e 6. Os números associados a cada um dos atributos indicam a sua relevância calculada pelo algoritmo de seleção (WITTEN; FRANK; HALL, 2005).

Tabela 5 – Ranqueamento dos atributos (`InfoGainAttributeEval` + `Ranker`)

		Fase A	Fase B	Fase C	Média
1.	f_{pk}	0,19	0,11	0,03	0,11
2.	m_2	0,05	0,07	0,07	0,06
3.	m_4	0,05	0,07	0,06	0,06
4.	V_{RMS}	0,05	0,06	0,05	0,05
5.	k_{rt}	0,05	0,03	0,04	0,04
6.	s_{kw}	0,03	0,04	0,04	0,04
7.	V_{pk}	0,03	0,05	0,02	0,03
8.	m_3	0,03	0,03	0,02	0,03

Tabela 6 – Ranqueamento dos atributos (`GainRatioAttributeEval` + `Ranker`)

		Fase A	Fase B	Fase C	Média
1.	f_{pk}	0,19	0,06	0,03	0,09
2.	m_4	0,05	0,11	0,06	0,07
3.	m_2	0,05	0,07	0,07	0,06
4.	V_{RMS}	0,05	0,06	0,05	0,06
5.	V_{pk}	0,05	0,05	0,03	0,04
6.	s_{kw}	0,03	0,04	0,04	0,04
7.	k_{rt}	0,04	0,03	0,03	0,03
8.	m_3	0,02	0,03	0,02	0,02

A análise provou então a grande relevância do fator de pico na detecção da falha, além dos momentos de 2ª e 4ª ordem, que também determinam a curtose. Na sequência aparecem o valor de pico e o valor RMS, utilizados no cálculo do f_{pk} . No entanto, a

obliquidade e o momento de 3ª ordem se mostram de pouca valia nesta análise. Verifica-se também que na fase em curto-circuito (Fase A no exemplo acima), a relevância do fator de pico fica ainda maior, enquanto as fases vizinhas sofrem uma variação mais relevante em m_4 , fator ligado à curtose.

A importância desta etapa do trabalho encontra-se na redução da dimensionalidade do problema, o que permite diminuir o esforço desempenhado na obtenção dos atributos, bem como o gasto no treinamento do classificador neural. Os resultados fornecidos pelo WEKA permitiram então a eliminação das estatísticas menos relevantes, que não seguiram para os testes de topologia, a saber, a curtose, a obliquidade e o momento de 3ª ordem.

Os três atributos preteridos nesta etapa exigem cálculos adicionais a serem feitos sobre o sinal, enquanto o valor eficaz e de pico, apesar de não serem tão relevantes quanto f_{pk} , m_2 e m_4 , já fazem parte da rotina de cálculo do sistema para obtenção do fator de pico, que segundo os algoritmos de seleção é o atributo mais relevante para a classificação. Assim, em termos de esforço computacional, não haveria grande economia em se deixar V_{RMS} e V_{pk} de fora do conjunto de entradas do sistema.

Com isso, as 24 entradas da RNA foram reduzidas para 15, o que já representa uma diferença significativa, de cerca de 44%. Após a seleção, o banco de dados foi enxugado em termos dos atributos preteridos nesta etapa para que se pudesse passar para a seguinte, de testes das topologias de classificação.

5.6 Testes das topologias candidatas

Definidos os atributos de entrada e montado o novo banco de dados no WEKA, passou-se à aba **Classifier** do ambiente **Explorer** do aplicativo. Lá, é possível selecionar o classificador **MultilayerPerceptron**, nada mais do que uma rede do tipo PMC com nós sigmoidais e algoritmo de treinamento do tipo GDM.

Como teste inicial para comprovação da validade da utilização da técnica neural, executou-se o algoritmo PMC do WEKA com os parâmetros seguintes:

- ❑ **hiddenlayers** = 'a': número de neurônios na camada oculta dados automaticamente pela média entre o número de entradas e a quantidade de classes de saída;
- ❑ **learningRate** = 0.3: taxa de aprendizado do algoritmo;
- ❑ **momentum** = 0.2: fator de *momentum* do treinamento;

No início da rotina de execução, a interface gráfica de usuário (GUI) do WEKA exibiu o resultado ilustrado na Figura 19:

Observa-se que o WEKA trabalha na configuração *one of c-classes*, com um neurônio para cada classe na camada de saída. Ao pressionar-se o **start**, o programa inicia um

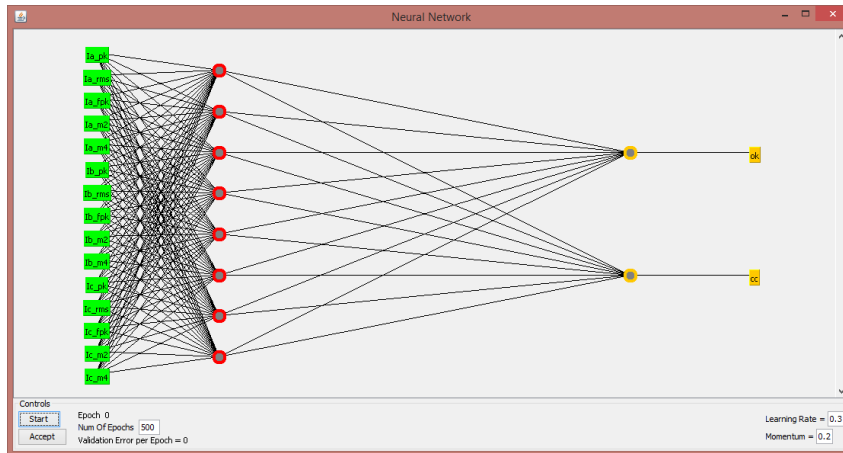


Figura 19 – Visualização da rede PMC gerada na GUI do WEKA.

Fonte: Autoria própria.

processo de treinamento com validação cruzada de 10 iterações. Ao fim do processo, o resultado é exibido na tela de saída do classificador:

```
=== Stratified cross-validation ===
```

```
=== Summary ===
```

Correctly Classified Instances	29464	99.4868 %
Incorrectly Classified Instances	152	0.5132 %
Kappa statistic	0.9891	
Relative absolute error	1.9723 %	
Root relative squared error	13.5221 %	
Total Number of Instances	29616	

```
=== Confusion Matrix ===
```

```

a      b      <-- classified as
18419   57 |      a = ok
 95    11045 |      b = cc
```

A média de épocas de treinamento foi de 140 durante o processo, e a taxa de acerto ficou em 99,49%, com cerca de 0,31% de falsos positivos e 0,85% de falsos negativos, segundo os valores indicados na matriz de confusão. Nela, 'ok' representa a classe dos motores saudáveis, e 'cc', a dos motores em curto.

É válido ressaltar que o MultilayerPerceptron do WEKA apresenta um pequeno número de parâmetros configuráveis, apenas um algoritmo de treinamento (o GDM) e serve apenas como ponto de partida e referência para o trabalho. De posse da informação que um PMC de 8 neurônios em sua camada oculta é capaz de resolver o problema-alvo,

buscou-se determinar qual a mais simples rede possível que também o fosse. Ademais, com a ajuda do MATLAB, é possível também fazer outros testes, como a validação e/ou treinamento com conjuntos de amostras selecionados e a verificação do comportamento da rede frente a amostras ruidosas.

Resultados Experimentais

Neste capítulo, são apresentados os resultados obtidos para os diferentes testes realizados com classificadores neurais para a detecção de curto-circuito estatórico em Motores de Indução com Rotor em Gaiola de Esquilo (MIRGE).

Para realização dos testes, a extensa base de dados foi dividida em subconjuntos convenientes para a verificação das condições de generalização e robustez do sistema de classificação. Conforme citado anteriormente, a base contém: amostras de motores acionados tanto diretamente da rede elétrica quanto a partir de um inversor de frequência PWM; diferentes condições de carga, desde o quase-vazio (12,5% da nominal) até a sobrecarga (125% do valor de tabela); diferentes níveis de curto-circuito, desde 1% até 10% entre espiras de bobinas de mesma fase; e ainda, dois diferentes motores, um deles modificado para o teste de curto-circuito, e o outro, na configuração original de fábrica para conjunto de controle.

Aproveitando-se de toda essa variedade de ensaios, foi possível conduzir uma série de estudos do classificador, como segue:

1. **Topologia ótima:** a primeira análise conduzida foi em busca de uma topologia para o PMC capaz de derivar a melhor acurácia possível com mínimo esforço computacional, partindo do resultado apontado pelo WEKA. Neste estudo, a base de dados foi integralmente utilizada;
2. **Supressão de dados de curto-circuito:** aqui, uma rede de topologia análoga à encontrada no estudo anterior foi treinada apenas com dados de motores saudáveis e com curtos-circuitos de 1% e 10%, e depois validada com dados de motores com curtos de 3% e 5%, a fim de verificar sua capacidade de generalização;
3. **Acionamento Direto *versus* Inversor:** desta vez, o classificador foi treinado apenas com as amostras em que o acionamento foi feito diretamente pela rede, e testada com aquelas associadas ao inversor trifásico;

4. **Amostras ruidosas:** por último, foram feitos testes adicionando-se um ruído branco às amostras de validação, com vistas a verificar a robustez do sistema.

6.1 Topologia ótima

Na busca pela topologia mais simples que pudesse resolver o problema de classificação, partiu-se do exemplo gerado pelo algoritmo do WEKA. Nele, havia uma configuração do tipo 15-8-2, com quinze entradas (cinco por fase), oito neurônios na camada oculta, e dois na camada de saída, na configuração *one of c-classes*. O primeiro passo foi investigar a possibilidade de solução do problema na configuração 15-8-1, com apenas um neurônio na camada de saída e resposta binária.

O banco de dados completo foi então importado para a interface do MATLAB, e lá uma rede `patternnet` foi treinada através do algoritmo `trainlm`, conforme visto no Capítulo 4. A Figura 20 mostra o erro de saída para os conjuntos de treinamento e validação durante o processo de treinamento da rede. Observa-se que o erro para as amostras efetivamente informadas à rede sempre decresce, enquanto as amostras de validação apresentam queda até a época 31, quando começa a crescer novamente. O *early-stopping* é então feito neste ponto.

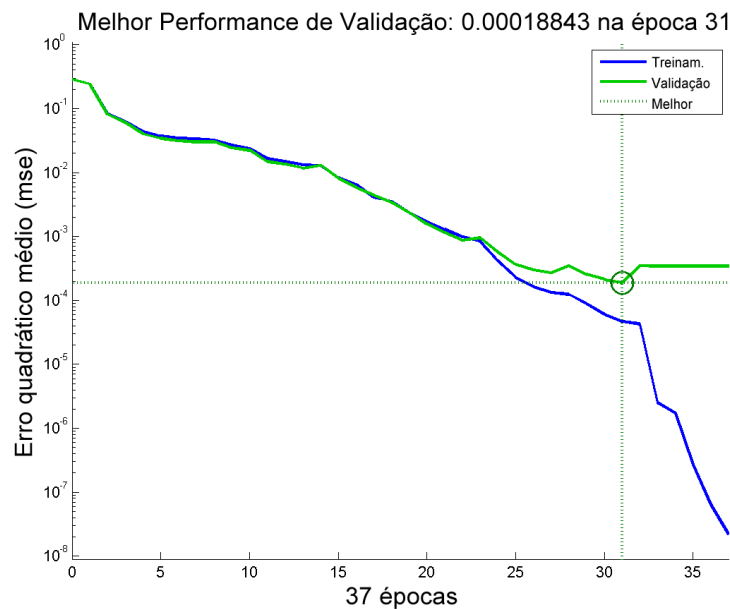


Figura 20 – Gráfico de performance do PMC com topologia 15-8-1.

Fonte: Autoria própria.

Ressalte-se que o número de épocas de treinamento foi bastante reduzido em comparação com os resultados preliminares do WEKA. A matriz de confusão da Figura 21 mostra também que o resultado atingido foi mais acurado, com apenas um falso-positivo registrado para o conjunto de treinamento, e um para o de validação, somando-se dois

em um total de 29.361 amostras. Na matriz, a classe-alvo indica o status do máquina sob avaliação: “1” representa motor em curto-circuito, e “0”, motor normal; enquanto a classe de saída indica qual foi a resposta do classificador frente à amostra apresentada. Em verde foram destacados os acertos, em vermelho, os erros, e as células cinzas e azuis representam os totais por classe e geral.

		Conjunto de Treinamento			Conjunto de Validação			Total		
Classe de Saída	0	14767 62.9%	0 0.0%	100% 0.0%	3707 63.1%	0 0.0%	100% 0.0%	18474 62.9%	0 0.0%	100% 0.0%
	1	1 0.0%	8721 37.1%	100.0% 0.0%	1 0.0%	2164 36.9%	100.0% 0.0%	2 0.0%	10885 37.1%	100.0% 0.0%
		0	1	100.0% 0.0%	0	1	100.0% 0.0%	0	1	100.0% 0.0%
		Classe Alvo			Classe Alvo			Classe Alvo		

Figura 21 – Matriz de Confusão do PMC com topologia 15-8-1.

Fonte: Autoria própria.

O teste apresentou resultados satisfatórios: de fato, a redução da camada de saída não representou um problema para o processo de classificação. Do contrário, a matriz de confusão acima explicita um resultado ainda melhor do que o observado no primeiro algoritmo, indicado pelo WEKA: dentre o total de amostras, apenas duas foram erroneamente identificadas, configurando dois falsos positivos, isto é, trata-se de informações provenientes de um motor saudável que foram apontadas como advindas de um equipamento defeituoso. O percentual de erros total neste teste, portanto, foi inferior aos 0,007%.

Comprovada a eficácia da nova configuração de apresentação das saídas, passou-se à redução da camada oculta, em busca da menor quantidade de neurônios que ainda permitisse a solução do problema. Esse número foi então sendo progressivamente reduzido, realizando-se para cada nova configuração 10 diferentes treinamentos, a fim de encontrar um melhor resultado para cada uma delas, haja vista que a inicialização randômica das matrizes de pesos e a divisão também aleatória dos conjuntos de treinamento e validação praticamente garante a unicidade dessas iterações. Dentre os 10 testes, escolheu-se aquele em que o erro quadrático para o conjunto de validação foi mínimo. Os resultados foram registrados na Tabela 7.

As configurações 15-3-1 e 15-2-1 do *Perceptron* Multicamadas, e a última, um *Perceptron* de simples camada, apresentaram resultados excelentes, mesmo quando comparados a redes de topologia bem mais complexa. A topologia 15-3-1, por exemplo, só teve seu desempenho superado pela 15-7-1, que apresenta o dobro de nós neuronais. E ainda com apenas um neurônio, o classificador atingiu uma taxa de acerto superior a 98%, o que mostra a quase-linearidade do problema diante da relevância da informação trazida pelas medidas estatísticas.

Tabela 7 – Testes para determinação da melhor topologia do PMC.

	Configuração do PMC						
	15-7-1	15-6-1	15-5-1	15-4-1	15-3-1	15-2-1	15-1
Núm. épocas	62	29	37	46	38	33	63
Tempo (s)	28,132	15,057	20,141	20,573	15,664	12,954	21,731
FP ¹ [tr] ²	0	2	1	3	2	12	145
FN ³ [tr]	0	18	3	5	3	13	259
FP [val] ⁴	2	2	1	0	2	4	27
FN [val]	0	1	3	2	1	6	66
FP total	2	4	2	3	4	16	172
FN total	0	19	6	7	4	19	325
Erros totais	2	23	8	10	8	35	497
Erro (%)	0,007	0,078	0,027	0,034	0,027	0,119	1,693

Todavia, a observação do número de épocas de treinamento e o tempo despendido pelo algoritmo de Levenberg-Maquardt mostra que a maior eficiência foi atingida para a configuração 15-2-1, conforme verifica-se na Figura 22, onde a eficiência foi calculada como a relação entre esses dois valores. Ora, a diferença entre 12 e 15 segundos de treinamento pode ser considerada desprezível em muitos casos, no entanto é necessário lembrar que os testes aqui apresentados foram feitos em apenas duas máquinas de um mesmo tipo, sob condições semelhantes: no caso de desenvolvimento de um sistema mais robusto, com maior variedade de ensaios, essa variável torna-se ainda mais relevante.

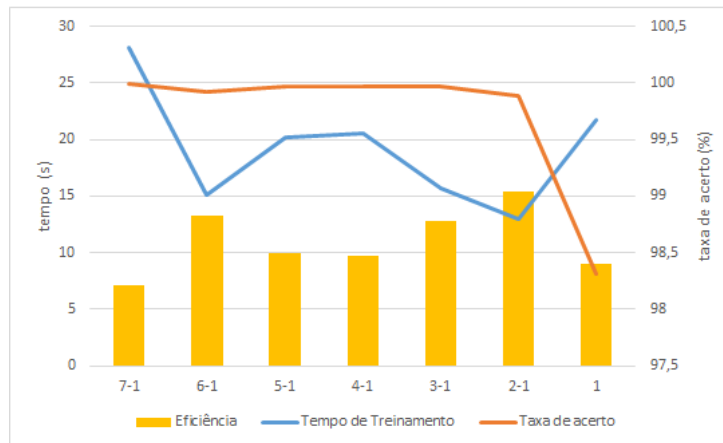


Figura 22 – Gráfico de eficiência (tempo de treinamento vs. taxa de acerto) para as topologias testadas.

Fonte: Autoria própria.

Assumindo então a topologia 15-2-1 como a que melhor se adequou às condições de contorno do problema, estudou-se ainda a possibilidade de outros algoritmos de treina-

¹Número de Falsos Positivos²Número de Falsos Negativos³Conjunto de Treinamento⁴Conjunto de Validação

mento atingirem resultados superiores (em tempo ou taxa de acerto) aos do `trainlm`. Testou-se portanto de forma análoga ao feito para as topologias, os algoritmos `traingd`, `traingdm` e `trainrp`, conforme visto no Capítulo 4. Os resultados são apresentados na Tabela 8.

Tabela 8 – Resultados de treinamento da rede com diferentes algoritmos.

	Algoritmo de Treinamento			
	GD	GDM	LM	RPROP
Núm. épocas	>50000	>50000	33	550
Tempo (s)	>430	>554	12,954	4,830
FP [tr]	N/A ⁵	N/A	12	248
FN [tr]	N/A	N/A	13	477
FP [val]	N/A	N/A	4	59
FN [val]	N/A	N/A	6	100
FP total	N/A	N/A	16	307
FN total	N/A	N/A	19	577
Erros totais	N/A	N/A	35	884
Erro (%)	N/A	N/A	0,119	3,011

A primeira informação trazida pelo teste foi acerca dos algoritmos mais simples, o GD e o GDM: seu tempo de convergência é bem maior quando comparado ao LM e ao RPROP, e o processo supera as 50.000 épocas de iteração. De fato, observa-se que o LM apresenta o menor erro de regime, e um tempo de processamento bastante justo frente ao RPROP, que apesar de ser o mais rápido em quase três vezes, apresenta um erro na saída quase 30 vezes superior, o que é também considerável. Isto posto, baseou-se na topologia 15-2-1 do PMC com algoritmo de treinamento `trainlm` para as etapas seguintes.

6.2 Supressão de dados de curto-circuito

Nesta etapa, fez-se o primeiro teste de generalização do classificador. As amostras correspondentes aos ensaios de motores com curto-circuitos de 3% e 5% entre bobinas de mesma fase foram removidas do conjunto original, restando apenas os dados de motores saudáveis e com curtos-circuitos de 1% e 10%. Esses valores foram escolhidos a fim de informar valores extremos do domínio durante o processo de treinamento, e verificar se ao final deste, a rede era capaz de inferir corretamente os status das amostras de teste.

Nesta nova abordagem, o gráfico de performance da Figura 23, que aponta os erros quadráticos para os conjuntos de treinamento e validação aleatórios criados pelo algoritmo a partir do conjunto inicial, aponta que o processo atingiu resultado ótimo em 46 épocas, que tiveram duração total de 14,098 segundos. Os valores são bastante semelhantes aos

⁵N/A: não se aplica. O valor indicado não foi obtido devido à interrupção precoce do processo de treinamento, limitado pelo número de épocas.

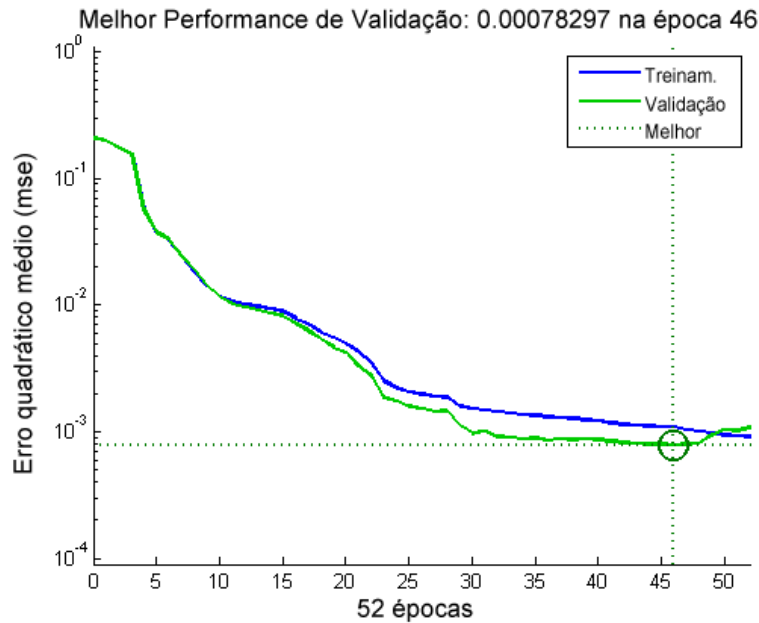


Figura 23 – Gráfico de performance para testes de generalização para curtos de 3% e 5%.

Fonte: Autoria própria.

encontrados utilizando-se a base da dados completa, o que leva a crer que os resultados serão também semelhantes.

De fato, na Figura 24, é possível observar as matrizes de confusão retornadas pelo MATLAB. No conjunto de treinamento, verificam-se 12 falsos positivos e 12 falsos negativos. No conjunto de validação, eles são respectivamente iguais a 2 e 3. No conjunto de teste, com os motores em curto, todas as amostras foram corretamente classificadas. No total, foram 29 erros em 29.361 amostras, o que representa um erro de aproximadamente 0,01%.

É importante ressaltar que, como foram utilizadas apenas amostras provenientes de motores com falhas para este teste, o cálculo das taxas de acerto para a linha e coluna correspondentes à classe de motores saudáveis torna-se obsoleto. Em termos matemáticos, porém, trata-se de uma indeterminação do tipo 0/0, e o MATLAB aponta os resultados como NaN, do inglês *Not a Number* para indicar que não se trata de um resultado válido.

Os resultados encontrados confirmam portanto o poder de generalização do classificador neural para as situações de curto-circuito intermediárias. Essa é uma característica de grande interesse para testes futuros: conhecendo as condições mínimas de curto capazes de sacrificar a vida útil do equipamento e as mais extremas condições que a máquina pode suportar antes de vir efetivamente a deixar de funcionar, é possível criar um sistema universal de detecção, o que reduz a quantidade de testes necessários para treinamento e comprova a robustez do sistema criado.



Figura 24 – Matriz de Confusão para testes de generalização para curtos de 3% e 5%.

Fonte: Autoria própria.

6.3 Acionamento Direto *versus* Inversor

O segundo teste com supressão de dados na base de dados de treinamento foi feito no sentido de testar a generalização do algoritmo em relação à frequência de acionamento, e à influência da resposta da indutância de enrolamento do motor à tensão PWM aplicada pelo inversor.

De maneira análoga ao que foi feito na Seção 6.2, dividiu-se o *dataset* original em dois subconjuntos: no primeiro, utilizado para o treinamento da rede neural, continha apenas os dados das amostras em que os motores foram acionados diretamente da rede elétrica; o segundo, apenas para aquelas em que se utilizou o inversor de frequência.

O gráfico de performance da Figura 25 mostra uma situação diferente daquela ocorrida no teste anterior, com a base de dados completa: aqui, o processo de treinamento foi

interrompido por ter sido atingido o gradiente mínimo definido, isto é, a rede atingiu uma configuração de erro quadrático mínimo (*mse*) antes que o *early-stopping* ocorresse. Além disso, devido ao reduzido número de amostras, o algoritmo foi encerrado em apenas 16 épocas, ou 1,808 segundos.

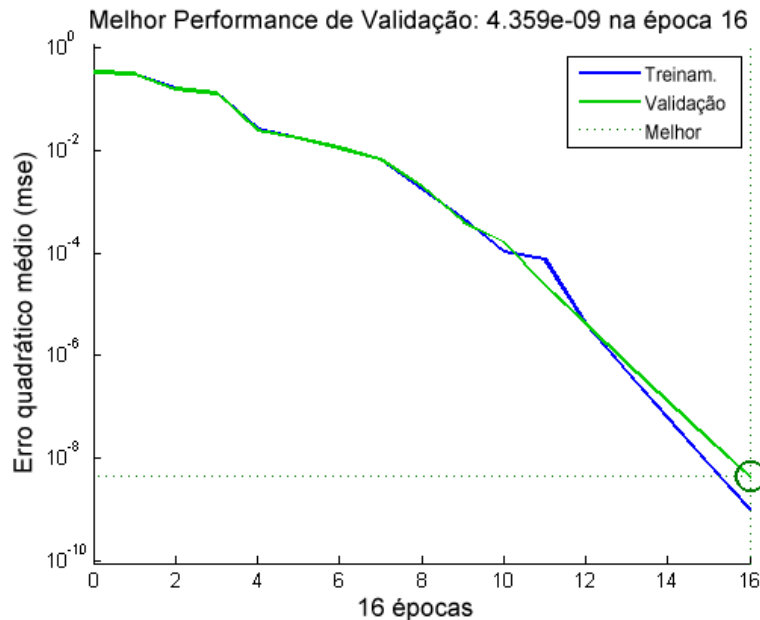


Figura 25 – Gráfico de performance para o teste de acionamento direto vs. inversor de frequência.
Fonte: Autoria própria.

Apesar dessa restrição, bastante considerável — em que pese o fato de que a quantidade de amostras de acionamento direto ser inferior a um quarto do total para acionamento por inversor (5.652 contra 23.709) —, o sistema ainda foi capaz de identificar corretamente o status do motor sob condições para as quais ele não havia sido treinado, como mostra a Figura 26.

Nota-se, todavia, que o número de motores defeituosos incorretamente classificados é bastante significativo, passando dos 50%. Isto é, o algoritmo mostrou-se mais eficiente em identificar aqueles que estavam em condições normais (cerca de 90%), o que pode sugerir que a falta de informações para os ensaios em ambos os extremos dificultou o trabalho de inferência da rede para os valores desconhecidos.

6.4 Amostras ruidosas

Conforme visto no Capítulo 4, um sistema baseado em Rede Neural Artificial deve ser resistente aos ruídos. Nesta etapa, testaremos o classificador desenvolvido em relação a essa característica. Novamente, a rede foi treinada com o *dataset* integral, dividido em porções de 80% para treinamento, 10% para validação e 10% para teste; neste último conjunto, adicionou-se um “ruído branco” às amostras, em diferentes magnitudes.

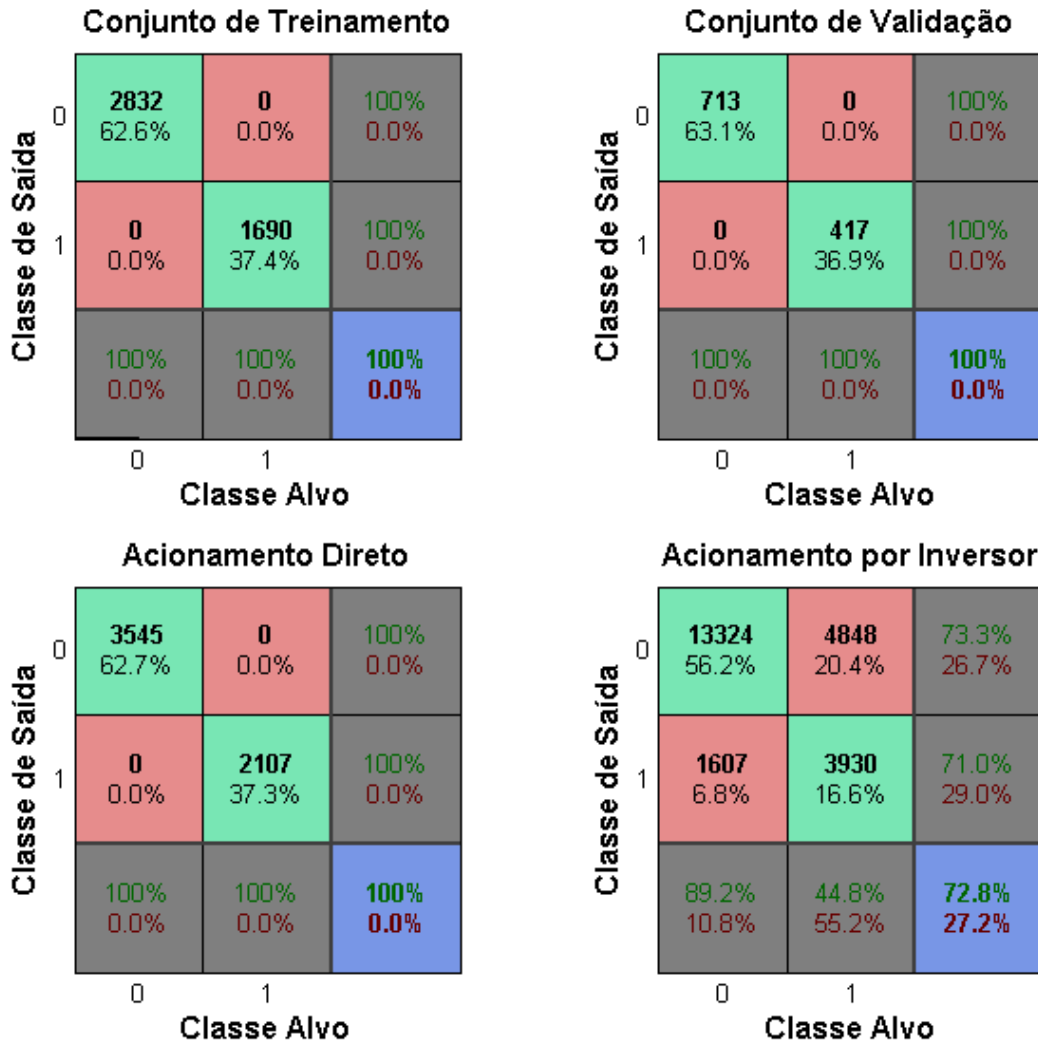


Figura 26 – Matriz de confusão para o teste de acionamento direto vs. inversor de frequência.

Fonte: Autoria própria.

Para inserção do ruído, utilizou-se a função `randn` do MATLAB, que cria números pseudoaleatórios com distribuição normal, onde a variância (m_2 ou σ) vale sempre 1 e a média (m_1), zero.

Observou-se que, conforme esperado, após a aplicação do ruído, a taxa de acerto do classificador sofreu uma redução: a capacidade da rede de absorver esses erros, porém, fica bem ilustrada na Figura 27.

Quando a magnitude do ruído inserido é de 10% sobre os valores originais, os erros no conjunto de testes sobem de 0,2% (2 falsos negativos e 3 falsos positivos, em números absolutos) para 1,7%, com o mesmo número de falsos negativos e 49 falsos positivos, um crescimento expressivo. Isso se deve ao fato de que alterações nos sinais de entrada advindos de motores saudáveis retiram-lhes das condições que o algoritmo de classificação compreende como a normalidade, e os enquadram como defeituosos.

Um comportamento bastante semelhante pode ser observado quando o ruído cresce,

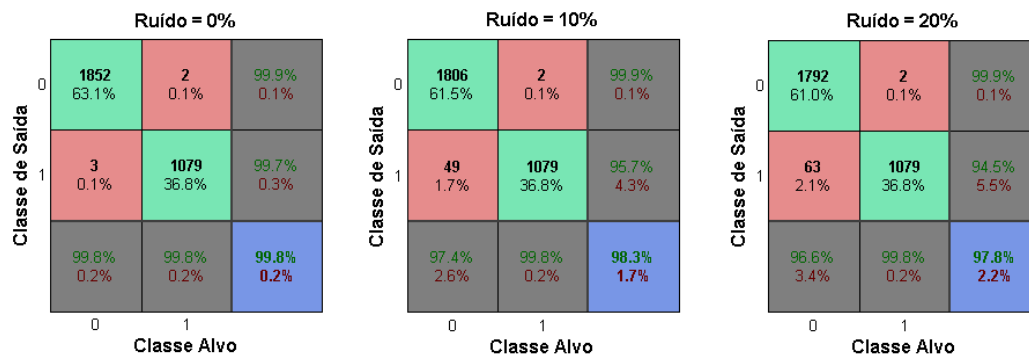


Figura 27 – Matriz de confusão do classificador para diferentes níveis de ruído.

Fonte: Autoria própria.

atingindo até 20% em magnitude. Nesse caso, há novamente um crescimento considerável para os falsos positivos, que chegam a 63 ocorrências, ou 2,1% do total. Nota-se ainda que, apesar do crescimento do número de amostras incorretamente classificadas ser grande em termos relativos, em termos absolutos eles ainda representam um montante muito pequeno em relação ao todo.

Uma possível solução para o problema do ruído em implementação de sistemas semelhantes ao aqui proposto seria a realização de mais de uma medida independente para cada ensaio: assim, a chance de reincidência de um erro de classificação seria diminuída, já que o ruído branco é um processo estocástico, e dificilmente ocasionaria seguidos erros de classificação em uma rede bem treinada.

Conclusões e Diretivas Futuras

Neste trabalho, foi feita uma investigação acerca da topologia e algoritmos de treinamento para classificadores de falhas de curto-circuito estatístico baseados em RNA. Também realizou-se um estudo sobre a relevância de determinados atributos do sinal de corrente do motor comumente utilizados em sistemas desse tipo, sempre visando à otimização do sistema classificador. Ademais, foram feitas aquisições de amostras reais em bancada de experimentos e uma série de testes da rede para garantir sua eficácia, sua capacidade de generalização e sua resiliência frente a amostras ruidosas.

A primeira etapa do projeto consistiu na obtenção em bancada de laboratório das amostras de sinais elétricos dos motores. Aqui, contou-se com uma estrutura previamente desenvolvida na UTFPR-CP, parte do escopo do Projeto Casadinho/Procad realizado em parceria com a EESC/USP, e que foi crucial para que a grande variedade de ensaios pudesse ser realizada com praticidade e em tempo hábil para o desenvolvimento deste trabalho.

A bancada de experimentos do Laboratório de Sistemas Inteligentes (LSI/UTFPR-CP) permitiu a obtenção de dados de equipamentos em condições originais de fábrica — para conjunto de controle — e de um motor modificado, que permitia a simulação das falhas de curto-circuito estatístico, que se desejava estudar, em uma variedade de magnitudes. Essas máquinas foram ensaiadas sob acionamento direto da rede elétrica e a partir de um inversor de frequência PWM, em diferentes níveis de torque, que varreram desde o quase-vazio até a sobrecarga do MIT.

Após a criação de uma ampla base dados através dos ensaios supracitados, partiu-se para o estudo e avaliação dos atributos que poderiam ser aplicados sobre os sinais aquisitados, a fim de extrair-lhes o conteúdo de informação que pudesse ser correlacionado ao status real da máquina.

Para isso, utilizou-se de técnicas estatísticas já encontradas na literatura relacionada, que no entanto foram ranqueadas através de algoritmos convenientes de seleção de atributos. Nesta etapa ficou claro, por exemplo, o quão relevantes são o fator de pico, da variância e do momento de 4ª ordem do sinal na detecção da falha; ademais, eliminaram-se

aqueles atributos de menor importância, cujo cálculo e inserção no método de classificação representariam um ganho muito pequeno em resultado, quando comparado com o custo computacional acrescido.

A seguir, e de posse dos dados necessários para tal, anteriormente obtidos, foi possível proceder à investigação da topologia ótima para solução do problema. A melhor solução encontrada, na configuração 15-2-1 — 15 entradas, 2 neurônios na camada oculta e 1 na de saída — e treinamento via algoritmo de Levenberg-Maquardt (LM), sendo quase quatro vezes menor em número de nós neuronais, 4,25 vezes menor em número de sinapses e levando cerca de 10 vezes menos tempo para treinamento do que o modelo inicial confirma o poder de extração da informação proporcionado pelos atributos selecionados.

A eficácia do classificador também se mostrou notável, haja vista a taxa de acerto superior aos 99,88%. De fato, a topologia ótima encontrada tem um desempenho semelhante ao encontrado na literatura, sendo menor em tamanho do que as apontadas em outros trabalhos em análise no domínio do tempo e guardando similaridade com as soluções encontradas em análises no domínio da frequência.

Por fim, foram feitos testes para comprovar a capacidade de generalização da rede frente a situações por esta desconhecidas. No primeiro, suprimiu-se os dados de motores com curto-circuitos de valores intermediários (3% e 5%) do processo de treinamento, deixando apenas os valores extremos (1% e 10%) disponíveis nesta etapa de projeto. Na etapa de validação, todavia, o classificador mostrou-se capaz de identificar corretamente todas as amostras desconhecidas a ele apresentadas.

O resultado aqui encontrado é relevante por mostrar que não há necessidade de se fazer uma miríade de testes para que o algoritmo possa identificar o status do motor com a devida acurácia: o sistema realmente foi capaz de “aprender” e identificar a tendência dos valores dos atributos que permitem fazer a classificação.

O teste seguinte, porém, exibiu uma limitação do classificador desenvolvido: quando treinado apenas com amostras de acionamento direto da rede, ele encontrou dificuldades em apontar o correto status do motor ensaiado quando este era acionado via inversor de frequência, apresentando uma taxa de acerto inferior a 50%. Todavia, esse pode ser dito um teste extremo para o algoritmo, ainda mais levando-se em consideração a variedade de frequências em que os ensaios com o inversor foram realizados. Esta pode ser considerada uma desvantagem da análise no domínio do tempo quando confrontada com a feita no da frequência.

Por fim, no último teste realizado comprovou-se a resiliência da rede neural quando confrontada com valores ruidosos: mesmo quando o ruído branco adicionado ao sinal alcançou a magnitude considerável de 20%, o classificador ainda apresentou um resultado bastante satisfatório, com uma taxa de erro de aproximadamente 2,2% e sem apresentar aumento de falsos negativos, exibindo queda de desempenho apenas no que se refere aos falsos positivos.

O panorama geral encontrado sugere, portanto, que atingiu-se um excelente resultado para o problema inicialmente proposto. Outras análises, no entanto, podem conduzir a um aumento ainda maior no desempenho desse tipo de algoritmo, como segue:

- ❑ **Comparação do PMC com outros SCI:** outras soluções podem ser comparadas com a aqui apresentada, incluindo as árvores de decisão, sistemas de inferência *fuzzy*, mapas auto-organizáveis, algoritmos genéticos, e ainda outras topologias de RNA, como a RBFNN;
- ❑ **Outros métodos de seleção de atributos:** para reduzir a dimensionalidade das entradas do algoritmo, e assim aumentar sua eficiência, uma boa aposta seria a Análise de Componentes Principais (PCA), dentre outros;
- ❑ **Comparação com a MCSA no domínio da frequência:** pode-se confrontar o sistema aqui desenvolvido com um similar que faça a análise da assinatura de corrente no domínio da frequência, e ainda verificar a possibilidade de montar um sistema misto.

Além disso, outros testes podem ser realizados a fim de aumentar a robustez e o escopo do classificador, levando-o no sentido de um protótipo em hardware, que pudesse ser efetivamente utilizado em problemas práticos:

- ❑ **Quantificação da falha identificada:** investigação de técnicas que permitam ao sistema retornar o status do equipamento não apenas em termos qualitativos, mas também quantitativos, informação relevante para os sistemas de manutenção;
- ❑ **Ampliação da base de dados:** o ensaio de motores de diferentes potências e classes de rendimento, e o desenvolvimento de metodologias que compreendam contemplar toda essa variedade de máquinas no mesmo algoritmo;
- ❑ **Ampliação do escopo da técnica:** investigar técnicas que permitam a incorporação da detecção de outros tipos de falha comuns em MIRGE, como os defeitos de rolamento, barras quebradas e excentricidade, tanto isoladamente quanto no caso de ocorrência simultânea de duas ou mais dessas falhas.

Referências

- ASFANI, D. A. et al. Temporary short circuit detection in induction motor winding using combination of wavelet transform and neural network. **Expert Systems with Applications**, 2012. v. 39, n. 5, p. 5367–5375, 2012.
- BOGLIETTI, A. et al. International standards for the induction motor efficiency evaluation: A critical analysis of the stray-load loss determination. **IEEE Transactions on Industry Applications**, 2004. v. 40, n. 5, p. 1294–1301, 2004.
- BRONIERA JR., P.; GOEDEL, A.; GRACIOLA, C. L. Uma estratégia neural para diagnóstico de curto-circuito entre espiras no enrolamento de estator em máquinas de indução trifásicas. **Proceedings of XI Simpósio Brasileiro de Automação Inteligente**, 2013. XI Simpósio Brasileiro de Automação Inteligente (SBAI 2013), p. 1–6, 2013.
- CASELLA, G.; BERGER, R. L. **Inferência estatística**. [S.l.]: São Paulo: Cengage Learning, 2010.
- CHENG, J. .; AI, L.; XU, S. . **The research on ant colony algorithm-neural networks model in fault diagnosis of asynchronous motor**. [S.l.: s.n.], 2013. 1625-1628 p. (Advanced Materials Research, v. 614-615).
- COELHO, D.; MEDEIROS, C. **Short Circuit Incipient Fault Detection and Supervision in a Three-Phase Induction Motor with a SOM-Based Algorithm**. [S.l.]: Springer-Verlag, 2013. 315-323 p.
- DRIF, M.; ESTIMA, J. O.; CARDOSO, A. J. M. Discriminating rotor cage faults and mechanical load oscillations in three-phase induction motors by the stator instantaneous complex apparent impedance. In: **Proceedings of IEEE Energy Conversion Congress and Exposition, ECCE 2012**. [S.l.: s.n.], 2012. p. 3024–3031.
- EMPRESA DE PESQUISA ENERGÉTICA, EPE. **Anuário Estatístico de Energia Elétrica**. Rio de Janeiro, 2013. 253 p.
- FITZGERALD, A.; KINGSLEY JR., C.; UMANS, S. **Máquinas Elétricas**. [S.l.]: Bookman, Porto Alegre, RS, Brasil., 2006.
- GARDEL, P. et al. Neural network broken bar detection using time domain and current spectrum data. In: **Proceedings - 2012 20th International Conference on Electrical Machines, ICEM 2012**. [S.l.: s.n.], 2012. p. 2492–2497.

- GHATE, V. N.; DUDUL, S. V. Optimal mlp neural network classifier for fault detection of three phase induction motor. **Expert Systems with Applications**, 2010. v. 37, n. 4, p. 3468–3481, 2010.
- GOEDTEL, A. **Estimador de velocidade em motores de indução trifásicos usando redes neurais artificiais**. Tese (Doutorado) — Escola de Engenharia de São Carlos da Universidade de São Paulo — EESC-USP, 2007.
- GONGORA, W. S. **Uma abordagem neural no diagnóstico de falhas em rolamentos de motores de indução trifásicos**. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná - UTFPR, 2013.
- GYFTAKIS, K. N. et al. A novel approach for broken bar fault diagnosis in induction motors through torque monitoring. **IEEE Transactions on Energy Conversion**, 2013. 2013.
- HAGAN, M. T.; MENHAJ, M. B. Training feedforward networks with the marquardt algorithm. **Neural Networks, IEEE Transactions on**, 1994. IEEE, v. 5, n. 6, p. 989–993, 1994.
- HALL, M. et al. The WEKA data mining software: An update. **ACM SIGKDD Explorations Newsletter**, 2009. ACM, v. 11, n. 1, p. 10–18, 2009.
- HAYKIN, S. **Neural Networks - A comprehensive foundation**. [S.l.]: Upper Saddle River, NJ: Prentice Hall, 1999.
- HONG, J. et al. Monitoring of airgap eccentricity for inverter-fed induction motors based on the differential inductance. In: **2012 IEEE Energy Conversion Congress and Exposition, ECCE 2012**. [S.l.: s.n.], 2012. p. 3000–3007.
- HUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, 1986. v. 323, n. 6088, p. 533–536, 1986.
- MAHAMAD, A. K.; HIYAMA, T. Fault classification based artificial intelligent methods of induction motor bearing. **International Journal of Innovative Computing, Information and Control**, 2011. v. 7, n. 9, p. 5477–5494, 2011.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, 1943. Springer, v. 5, n. 4, p. 115–133, 1943.
- MCPHERSON, G.; LARAMORE, R. D. **An introduction to electrical machines and transformers**. [S.l.]: New York: John Wiley and Sons, 1990.
- MONARO, R. M. **Lógica fuzzy aplicada na melhoria da proteção digital de geradores síncronos**. Tese (Doutorado) — Escola de Engenharia de São Carlos da Universidade de São Paulo — EESC-USP, 2013.
- MOOSAVI, S. S. et al. Impedance angle changes analysis applied to short circuit fault detection. In: **20th International Conference on Electrical Machines, ICEM 2012**. [S.l.: s.n.], 2012. p. 1724–1729.
- MORGAN, N.; BOURLARD, H. Generalization and parameter estimation in feedforward nets: Some experiments. In: **NIPS**. [S.l.: s.n.], 1989. p. 630–637.

- MURUGANATHAM, B. et al. Roller element bearing fault diagnosis using singular spectrum analysis. **Mechanical Systems and Signal Processing**, 2013. v. 35, n. 1-2, p. 150–166, 2013.
- NANDI, S.; TOLIYAT, H. A.; LI, X. Condition monitoring and fault diagnosis of electrical motors - a review. **IEEE Transactions on Energy Conversion**, 2005. v. 20, n. 4, p. 719–729, 2005.
- PICAZO-RÓDENAS, M. J. et al. Use of the infrared data for heating curve computation in induction motors: Application to fault diagnosis. **Engineering Failure Analysis**, 2013. 2013.
- PIRES, V. F. et al. Motor square current signature analysis for induction motor rotor diagnosis. **Measurement: Journal of the International Measurement Confederation**, 2013. v. 46, n. 2, p. 942–948, 2013.
- RIEDMILLER, M.; BRAUN, H. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: IEEE. **Neural Networks, 1993., IEEE International Conference on**. [S.l.], 1993. p. 586–591.
- SANTOS, T. H. dos. **Estimador neural de velocidade aplicado a um driver de controle escalar do motor de indução trifásico**. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná - UTFPR, 2012.
- SARKHANLOO, M. S. et al. A new method for stator winding turn-fault diagnosis of induction motor by space vector model based on neural network. **International Review on Modelling and Simulations**, 2011. v. 4, n. 5, p. 2182–2189, 2011.
- SEERA, M. et al. Application of the fuzzy min-max neural network to fault detection and diagnosis of induction motors. **Neural Computing and Applications**, 2012. p. 1–10, 2012.
- SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. **Redes Neurais Artificiais para engenharia e ciências aplicadas**. [S.l.]: São Paulo: Artliber Editora, 2010.
- SUETAKE, M. **Sistemas Inteligentes para Monitoramento e Diagnósticos de Falhas em Motores de Indução Trifásicos**. Tese (Doutorado) — Escola de Engenharia de São Carlos da Universidade de São Paulo — EESC-USP, 2012.
- TRZYNADLOWSKI, A. M. **Control of induction motors**. [S.l.]: Academic Pr, 2001.
- WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining: Practical machine learning tools and techniques**. [S.l.]: Morgan Kaufmann, 2005.
- YADAV, M.; WADHWANI, S. Automatic fault classification of rolling element bearing using wavelet packet decomposition and artificial neural network. **International Journal of Engineering and Technology**, 2011. v. 3, n. 4, p. 270–276, 2011.
- YANG, C.; ZHU, C.; XIE, H. Fault diagnosis based on rprop neural network for asynchronous motor. **Dianli Zidonghua Shebei/Electric Power Automation Equipment**, 2012. v. 32, n. 1, p. 80–83, 2012.

YEH, C. et al. A reconfigurable motor for experimental emulation of stator winding inter-turn and broken bar faults in polyphase induction machines. **Electric Machines and Drives Conference, 2007. IEMDC '07. IEEE International**, 2007. v. 2, p. 1413–1419, 2007.