

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA
TESE DEFENDIDA POR... Vinicius.....
Augusto Diniz Silva..... E APROVADA
PELA COMISSÃO JULGADORA EM 24/02/2012.


.....
ORIENTADOR

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

Vinicius Augusto Diniz Silva

Detecção de falhas em motores elétricos através das máquinas de vetores de suporte

Campinas, 2012.

12/2012

Vinícius Augusto Diniz Silva

**Detecção de falhas em motores elétricos
através das máquinas de vetores de suporte**

Dissertação apresentada ao Curso de Mestrado da Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas, como requisito para a obtenção do título de Mestre em Engenharia Mecânica.

Área de Concentração: Mecânica dos Sólidos e Projeto Mecânico

Orientador: Robson Pederiva

Campinas
2012

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

SI38d Silva, Vinícius Augusto Diniz
Detecção de falhas em motores elétricos através das
máquinas de vetores de suporte \ Vinícius Augusto Diniz
Silva. – Campinas, SP: [s.n.], 2012.

Orientador: Robson Pederiva.
Dissertação de Mestrado - Universidade Estadual de
Campinas, Faculdade de Engenharia Mecânica.

1. Motores elétricos. 2. Máquinas - Defeitos. 3.
Máquinas - Vibração. 4. Inteligência artificial. 5.
Máquinas de vetores de suporte. I. Pederiva, Robson.
II. Universidade Estadual de Campinas. Faculdade de
Engenharia Mecânica. III. Detecção de falhas em motores
elétricos através das máquinas de vetores de suporte.

Título em Inglês:	Fault detection in induction motors using support vector machines
Palavras-chave em Inglês:	Electric motors, Machines - Defects, Machinery - Vibration, Artificial intelligence, Support vector machines.
Área de concentração:	Mecânica dos Sólidos e Projeto Mecânico
Titulação:	Mestre em Engenharia Mecânica
Banca Examinadora:	Pablo Siqueira Meirelles, Luiz de Paula Nascimento
Data da defesa:	24-02-2012
Programa de Pós Graduação:	Engenharia Mecânica

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO**

DISSERTAÇÃO DE MESTRADO ACADÊMICO

**Detecção de falhas em motores elétricos
através das máquinas de vetores de suporte**

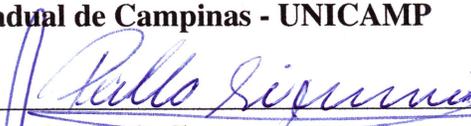
Autor: Vinícius Augusto Diniz Silva

Orientador: Robson Pederiva

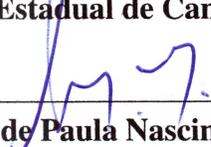
A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:



Prof. Dr. Robson Pederiva, Presidente
Universidade Estadual de Campinas - UNICAMP



Prof. Dr. Pablo Siqueira Meirelles
Universidade Estadual de Campinas - UNICAMP



Prof. Dr. Luiz de Paula Nascimento
Universidade Estadual Paulista - UNESP

Campinas, 24 de Fevereiro de 2012.

DEDICATÓRIA:

Dedico este trabalho a Deus, aos meus pais e a minha noiva.

Agradecimentos

Este trabalho não seria possível se não estivesse presente a participação de diversas pessoas, em especial:

Ao amigo e orientador Prof. Dr. Robson Pederiva pela oportunidade, confiança, orientação, apoio e pela paciência, que contribuíram para que este trabalho fosse realizado.

À Virgem Maria, por todas as vezes que intercedeu por mim e me deu forças nos momentos mais difíceis.

Aos meus pais Itamar e Iolanda pela vida, pela criação, pelos ensinamentos e por todo incentivo e esforço. A minha noiva Daniela Nattália, agradeço pelo companheirismo, carinho, compreensão e apoio. E a toda minha família, meus irmãos César, Bruna e Samuel por torcerem a favor da conclusão deste trabalho.

Aos amigos Prof. Jorge Nei, Prof. Geraldo Roberto e Prof. Paulo Lamim. Pois, se hoje esta etapa está concluída, foi graças as vocês. Obrigado por toda a força, ensinamentos e incentivo. Você fazem parte dos meus 5%...

Aos colegas do Laboratório de Vibrações e Controle e de toda a FEM, pela companhia, apoio, ajuda e amizade.

Por fim agradeço à CAPES pelo apoio financeiro, pois sem ele, a vida durante os anos de estudos não seria possível.

*Se queres conhecer a uma pessoa, não lhe
perguntas o que pensa mas sim o que ela
ama.*

Santo Agostinho

Resumo

SILVA, Vinicius Augusto Diniz. Detecção de falhas em motores elétricos através das máquinas de vetores de suporte. 2012. 122p. Dissertação de Mestrado. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas.

Motores elétricos são componentes essenciais na grande maioria dos processos industriais. As diversas falhas nas máquinas de indução podem gerar consequências drásticas para um processo industrial. Os principais problemas estão relacionados ao aumento dos custos, piora nas condições do processo e de segurança e qualidade do produto final. Muitas destas falhas mostram-se progressivas. Neste trabalho, apresenta-se uma contribuição ao estudo de métodos de detecção de falhas em motores elétricos usando Máquinas de Vetores de Suporte (SVM), treinadas a partir de sinais de vibração obtidos experimentalmente. A metodologia desenvolvida é usada para classificar a excitação devido a falhas mecânicas e elétricas, além da condição normal de funcionamento, utilizando apenas um sensor de vibração. Através da seleção de parâmetros é possível reduzir o número de entradas capazes de representar os sinais utilizados para o treinamento das SVMs. A normalização proposta permitiu melhorar as taxas de acerto, quando se quer classificar falhas em diferentes níveis de severidade das que foram utilizadas para o treinamento. Os resultados mostraram que a metodologia apresentada pode ser adaptada para ser utilizada em aplicações práticas industriais e poderá ser no futuro uma saída viável para uma manutenção industrial eficiente e eficaz.

Palavras-chave: Motores elétricos, Máquinas – Defeitos, Máquinas – Vibração, Inteligência artificial, Máquinas de vetores de suporte.

Abstract

SILVA, Vinicius Augusto Diniz. Fault detection in induction motors using support vector machines. 2012. 122p. Dissertação de Mestrado. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas.

Electric motors are essential components in most industrial processes. The several faults in induction machines can produce drastic consequences for an industrial process. The main problems are related to rising costs, decrease conditions in the process and safety and quality of the final product. Many of these failures are progressive. In this paper, we present a contribution to the study of methods for detecting faults in induction motors using Support Vector Machines (SVM) trained from vibration signals obtained experimentally. The developed methodology is used to classify the excitation due to mechanical and electrical failures, in addition to normal operating condition, using only a vibration sensor. Through the feature selection is possible to reduce the number of inputs that represent the signals used for training the SVMs. The proposed standardization has improved the accuracy rates when we want to classify failures at different levels of severity that were used for training. The results showed that this methodology can be adapted for use in industrial and practical applications and in the future may be a viable approach to an efficient and effective industrial maintenance.

Keywords: Electric motors, Machines – Defects, Machinery – Vibration, Artificial intelligence, Support vector machines.

Lista de Ilustrações

2.1	Modelo de McCulloch e Pitts.	12
2.2	Uma rede unidirecional com três camadas de conexões.	14
2.3	Exemplo de avaliação de temperatura.	25
2.4	Sistema de inferência <i>fuzzy</i> , (GOONATILAKE, 1992).	28
3.1	Hiperplano de separação ótimo (linha grossa), com os seus hiperplanos de suporte (tracejados).	36
3.2	Classificação através de hiperplanos.	37
3.3	Cálculo da distância Υ entre os hiperplanos.	38
3.4	Tipos de SVs: livres (vermelho) e limitados (preto e cinza).	44
3.5	Mudança de espaço.	45
3.6	Mapeamento não linear, (SCHOLKOPF E SMOLA, 2002).	46
3.7	Mapeamento no espaço \mathbb{R}^2	48
3.8	Mapeamento no espaço \mathbb{R}^3	49
3.9	Classificação não-linear via separação linear em espaço de maior dimensão através de um kernel.	50
3.10	Arquitetura do método DAGSVM.	53
4.1	Representação do desequilíbrio de massa.	57
4.2	Representação do desalinhamento.	58
4.3	Tipos de desalinhamento.	58
4.4	Representação da folga mecânica.	59
4.5	Elementos do rolamento.	60
4.6	Curto-circuito entre espiras.	61
4.7	Exemplo da simulação de um rotor com barras quebradas.	63
5.1	Bancada de testes.	64
5.2	Pontos de coleta.	67
5.3	Espectro da condição normal de funcionamento do motor elétrico.	68
5.4	Espectro do motor elétrico desbalanceado.	68
5.5	Espectro do motor elétrico desalinhado.	69
5.6	Espectro do motor elétrico com folga mecânica.	69

5.7	Espectro do motor elétrico com curto-circuito.	70
5.8	Espectro do motor elétrico com desequilíbrio de fase.	70
5.9	Espectro do motor elétrico com barras quebradas.	71
5.10	Média e desvio padrão para as falhas mecânicas.	72
5.11	Média e desvio padrão para as falhas elétricas.	73
5.12	Parâmetros para diagnóstico de falhas mecânicas.	74
5.13	Parâmetros para diagnóstico de falhas elétricas.	75
5.14	Distribuição espacial das falhas mecânicas.	75
5.15	Distribuição espacial das falhas elétricas.	76
5.16	Função de pertinência para a entrada para as falhas mecânicas.	85
5.17	Função de pertinência para a entrada para as falhas elétricas.	86
5.18	Função de pertinência para a condição do motor relativo à falhas mecânicas.	87
5.19	Função de pertinência para a condição do motor relativo à falhas elétricas.	88
5.20	Saída do sistema <i>fuzzy</i> para as falhas mecânicas.	92
5.21	Saída do sistema <i>fuzzy</i> para as falhas elétricas.	93
5.22	Função de pertinência de entrada para as falhas mecânicas segundo a norma VDI 2056.	102

Lista de Tabelas

1.1	Benefícios da manutenção preditiva.	2
2.1	Características da lógica <i>fuzzy</i> , (CAMARGOS, 2002).	27
3.1	Funções kernel mais comuns.	48
5.1	Classificação das falhas mecânicas com a SVM.	78
5.2	Classificação das falhas elétricas com a SVM.	79
5.3	Classificação das falhas mecânicas com a RNA.	81
5.4	Classificação das falhas elétricas com a RNA.	82
5.5	Composição da base de regras para falhas elétricas.	89
5.6	Composição da base de regras para falhas mecânicas.	90
5.7	Classificação das falhas mecânicas com a Lógica <i>Fuzzy</i>	94
5.8	Classificação das falhas elétricas com a Lógica <i>Fuzzy</i>	94
5.9	Classificação do nível 2 através da SVM treinada com o nível 1 para as falhas mecânicas.	96
5.10	Classificação do nível 2 através da RNA treinada com o nível 1 para as falhas mecânicas.	97
5.11	Classificação do nível 2 através da SVM treinada com o nível 1 para as falhas elétricas.	97
5.12	Classificação do nível 2 através da RNA treinada com o nível 1 para as falhas elétricas.	98
5.13	Classificação do nível 2 de severidade com a SVM treinada com o nível 1 para as falhas mecânicas com a normalização.	100
5.14	Classificação do nível 2 de severidade com a RNA treinada com o nível 1 para as falhas mecânicas com a normalização.	100
5.15	Classificação do nível 2 de severidade com a SVM treinada com o nível 1 para as falhas elétricas com a normalização.	101
5.16	Classificação do nível 2 de severidade com a RNA treinada com o nível 1 para as falhas elétricas com a normalização.	101
5.17	Classificação das falhas mecânicas nos níveis 1 e 2 com a Lógica <i>Fuzzy</i>	102
5.18	Classificação das falhas elétricas nos níveis 1 e 2 com a Lógica <i>Fuzzy</i>	103

Lista de Abreviaturas e Siglas

Letras Latinas

b	-	Bias na SVM	
C	-	Custo ou parâmetro de regularização	
d	-	Diâmetro do elemento rolante	[mm]
D	-	Diâmetro primitivo do rolamento	[mm]
f_l	-	Frequência de linha	[Hz]
f_r	-	Frequência de rotação	[Hz]
h	-	Dimensão VC	
H	-	Hiperplano	
H_z	-	Espaço de hipóteses	
i, j	-	Índices	
k	-	Kernel	
\mathcal{L}	-	Lagrangiano	
m	-	Número de exemplos de treinamento	
M	-	Número de classes	
n_{SV}	-	Número de SV	
N	-	Número de dimensões	
\mathbb{R}	-	Conjunto dos números reais	
s_f	-	Frequência de escorregamento do motor elétrico	[Hz]
T	-	Conjunto de treinamento	
$u(t)$	-	Potencial interno do neurônio	
w_i	-	Peso da entrada na RNA	
\mathbf{w}	-	Vetor de pesos	
x_i	-	Dados de entrada	

- \mathbf{x}_i - Vetor dos dados de entrada
 - X - Domínio de entrada
 - y_k - Saída do neurônio
 - y_i - Rótulos
-

Letras Gregas

- α_i - Multiplicadores de Lagrange
 - $\boldsymbol{\alpha}$ - Vetor dos multiplicadores de Lagrange
 - β - Ângulo de contato [graus]
 - γ, κ, Θ - Parâmetros das funções kernel
 - η - Função de transferência na RNA
 - θ - *Bias* na RNA
 - λ - Função indicadora
 - Λ - Conjunto das funções indicadoras
 - μ - Função de pertinência
 - ξ - Variável de folga
 - $\boldsymbol{\xi}$ - Vetor de todas as variáveis de folga
 - Υ - Distância entre os hiperplanos
 - Φ - Mapa de características
 - Ω - Frequência [rad/s]
-

Superescritos

- n - Dimensão do conjunto
-

Subscritos

- λ - Parâmetro da função indicadora
-

Abreviações

- Máx** - Máximo
max - Maximizar
Min - Mínimo
min - Minimizar
sgn - Função Sinal

.....

Siglas

- BPFI** - Frequência de Falha na Pista Interna (*Ball Pass Frequency Inner Racer*)
BPFO - Frequência de Falha na Pista Externa (*Ball Pass Frequency Outer Racer*)
BSF - Frequência de Falha no Elemento Rolante (*Ball Spin Frequency*)
COA - Centro de Área (*Center Of Area*)
DAG - Grafo Direcionado Acíclico (*Directed Acyclic Graph*)
ERM - Minimização do Risco Empírico (*Empirical Risk Minimization*)
FEM - Faculdade de Engenharia Mecânica
FTF - Frequência de Falha na Gaiola (*Fundamental Train Frequency*)
IA - Inteligência Artificial
KKT - Karush-Kuhn-Tucker
LIBSVM - Biblioteca para SVM (*Library for SVM*)
MAH - Motor Acoplamento Horizontal
MAV - Motor Acoplamento Vertical
MOM - Média dos máximos (*Mean Of Maximum*)
MVH - Motor Ventoinha Horizontal
MVV - Motor Ventoinha Vertical
RBF - Função de Base Radial (*Radial Basis Function*)
RNA - Redes Neurais Artificiais
SE - Sistemas Especialistas
SOM - Máximo Mínimo (*Smallest Of Maximum*)
SRM - Minimização do Risco Estrutural (*Structural Risk Minimization*)
SV - Vetores de Suporte (*Support Vector*)

- SVM** - Máquinas de Vetores de Suporte (*Support Vector Machine*)
- TF** - Transformada de Fourier
- UCT** - Um-Contra-Todos
- UCU** - Um-Contra-Um
- VC** - Vapnik-Chervonenkis
.....

SUMÁRIO

Lista de Ilustrações	ix
Lista de Tabelas	xi
Lista de Abreviaturas e Siglas	xii
SUMÁRIO	xvi
1 INTRODUÇÃO	1
1.1 Objetivos do trabalho	6
1.2 Organização do texto	7
2 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL	8
2.1 Histórico	8
2.2 Redes neurais artificiais	11
2.2.1 Modelagem de um neurônio	12
2.2.2 Modelagem da topologia	14
Rede neural unidirecional (<i>feed-forward networks</i>)	14
Modelagem da regra de aprendizado	15
2.2.3 Modelos de redes neurais artificiais	15
2.2.4 Algoritmo de <i>back-propagation</i>	16
Noções gerais	17
Treinamento	17
Deficiências do <i>back-propagation</i>	18
Paralisação da rede	19
Mínimos locais	19
<i>Momentum</i>	20
<i>Weight decay</i>	20
2.2.5 Outras fontes de problemas	20
Tamanho e topologia da rede	20

	Inicialização aleatória dos pesos	21
	Parâmetros de aprendizagem	21
	Interrupção do treinamento	22
2.3	Lógica <i>Fuzzy</i>	24
2.3.1	Sistemas de inferência <i>fuzzy</i>	28
	Fuzzificador	29
	Regras	29
	Inferência	30
	Defuzzificação	30
2.4	Considerações finais	31
3	MÁQUINAS DE VETORES DE SUPORTE	32
3.1	Teoria do aprendizado estatístico	32
3.1.1	Minimização do risco empírico	33
3.1.2	Minimização do risco estrutural	34
3.2	Fundamentação teórica das SVMs	35
3.2.1	Hiperplano de separação ótimo	37
	Margens rígidas	37
	Margens flexíveis	41
3.2.2	SVMs não lineares	44
3.2.3	SVM para multiclases	51
	Um-Contra-Todos (UCT)	51
	Um-Contra-Um (UCU)	52
	DAGSVM	53
3.3	Considerações finais	54
4	DETECÇÃO DE FALHAS ATRAVÉS DA ANÁLISE DE VIBRAÇÕES	55
4.1	Análise de vibrações	55
4.1.1	Sinais de vibrações	55
	Transformada de Fourier	56
4.2	Principais fontes de defeitos em motores elétricos	56
4.2.1	Falhas de origem mecânica	57
	Desbalanceamento	57
	Desalinhamento	58

	Folga mecânica	59
	Rolamentos	59
4.2.2	Falhas de origem elétrica	60
	Curto-circuito	61
	Desequilíbrio de fase	61
	Barras quebradas	62
4.3	Considerações finais	63
5	RESULTADOS EXPERIMENTAIS	64
5.1	Bancada e procedimentos experimentais	64
5.1.1	Condição normal de funcionamento	68
5.1.2	Desbalanceamento	68
5.1.3	Desalinhamento	69
5.1.4	Folga mecânica	69
5.1.5	Curto-circuito	70
5.1.6	Desequilíbrio de fase	70
5.1.7	Barras quebradas	71
5.1.8	Dispersão das falhas	71
5.2	SVM para classificação de falhas	73
5.2.1	Seleção dos sinais de vibração	74
5.2.2	Escolha dos parâmetros de treinamento	76
5.2.3	Diagnóstico das falhas	78
5.3	RNA para classificação de falhas	80
5.3.1	Escolha dos parâmetros de treinamento	80
5.3.2	Diagnóstico das falhas	81
5.4	Lógica <i>fuzzy</i> para classificação de falhas	84
5.4.1	Variáveis linguísticas, funções de pertinência e regras	84
5.4.2	Diagnóstico das falhas	91
5.5	Classificação de falhas em diferentes níveis de severidade	96
5.5.1	Normalização	98
5.5.2	Classificação com a normalização	99
5.6	Considerações finais	104
6	CONCLUSÃO	105

1 INTRODUÇÃO

Desde quando apareceram as primeiras máquinas, surgiu também a necessidade de conhecer seu funcionamento e monitorar seu comportamento. Porém, o papel dos homens da manutenção era apenas esperar a falha acontecer para depois corrigi-la (denomina-se hoje por manutenção corretiva não planejada). Com o aumento do nível de exigência por bens e o crescimento da indústria, não se podia mais perder tempo com o conserto de equipamentos, assim, iniciou-se trabalhos na prevenção de falhas, necessitando de profissionais na manutenção que tivessem o conhecimento da manutenção preventiva.

Com o passar dos anos, o desenvolvimento de tecnologias permitiu no período dos anos 70 e 80 que as idéias e ações mudassem de apenas prevenir para prever as falhas. Para isso foi então criada a manutenção Preditiva que utiliza técnicas que possam monitorar o equipamento, seja da forma sensitiva ou através de instrumentos de medição. Neste aspecto, os estudos foram intensificados pois os profissionais necessitavam de um nível de capacitação ainda maior, surgindo assim os primeiros especialistas.

A premissa da manutenção preditiva é que o monitoramento regular das condições reais das máquinas, e do rendimento operativo dos sistemas de processo, irão maximizar o tempo médio entre reparos. Também minimizará o número e o custo das paradas não programadas criadas por falhas da máquina, e melhorará a disponibilidade global das plantas operacionais. A inclusão da manutenção preditiva em um programa de gerência total da planta oferecerá a capacidade de otimizar a disponibilidade da maquinaria de processo e reduzirá o custo da manutenção. Na realidade, a manutenção preditiva pode ser vista como um programa de manutenção preventiva acionada por condição.

Um levantamento feito em 1988 de 500 fábricas, que implementaram com sucesso métodos de manutenção preditiva, indicou melhorias substanciais na confiabilidade, disponibilidade e custos operacionais. Realizado pela *Plant Performance Group* (uma divisão da *Technology for Energy Corporation*), este levantamento foi projetado para quantificar o impacto da inclusão de técnicas de manutenção preditiva como peça chave da filosofia da gerência de manutenção. No grupo de amostras constavam-se uma variedade de indústrias dos Estados Unidos, Canadá, Grã-Bretanha, França

e Austrália. As indústrias incluíam geração de energia elétrica, papel e celulose, processamento alimentício, têxteis, ferro e aço, alumínio, e outras indústrias de manufatura ou de processo. Cada um dos participantes tinha um programa de manutenção preditiva estabelecido com um mínimo de três anos de implementação. Os programas bem sucedidos incluídos no levantamento oferecem uma visão geral dos tipos de melhorias, que podem ser esperadas a partir de um programa de gerência de manutenção preditiva abrangente, Tabela 1.1.

Tabela 1.1: Benefícios da manutenção preditiva.

Benefícios	Percentual
Redução dos custos de manutenção	50 a 80%
Redução de quebra nas máquinas	50 a 60%
Redução de estoque de sobressalentes	20 a 30%
Redução de horas extras para manutenção	20 a 50%
Redução do tempo de parada das máquinas	50 a 80%
Aumento na vida das máquinas	20 a 40%
Aumento da produtividade	20 a 30%
Aumento dos lucros	25 a 60%

FONTE : *Plant Performance Group.*

Com as técnicas de manutenção preditiva é possível detectar falhas em nível incipiente e acompanhar seu desenvolvimento, evitando-se quebras e paradas inesperadas. Dentre as principais técnicas pode-se destacar:

- Análise de vibrações;
- Análise de fluxo magnético;
- Análise de corrente;
- Análise termográfica;
- Ferrografia, entre outras;

Os três momentos citados anteriormente (manutenção corretiva não planejada, preventiva e preditiva) foram abordados por Moubray (2000) como as gerações da manutenção. Contudo, desde os anos 80, observa-se que nenhuma nova geração de manutenção foi citada. Mas por outro lado, é visto em muitos congressos e artigos, tanto nacionais como internacionais, que o interesse na utilização de técnicas de inteligência artificial no auxílio das tradicionais técnicas de manutenção preditiva para a detecção de falhas dos mais diversos equipamentos, tem se tornado cada vez mais procurado. Pode-se assim dizer que, sem dúvida, a quarta geração da manutenção terá a presença das técnicas de Inteligência Artificial.

A análise de vibração tem sido uma das técnicas mais utilizadas para detecção e diagnóstico de falhas devido às suas potencialidades, como facilidade de utilização, custo relativamente baixo, técnica não intrusiva, dentre outros. Através da análise do espectro do sinal da vibração, pode-se detectar tanto defeitos de origem mecânica quanto de origem elétrica. Apesar de cada uma das diferentes técnicas existentes serem melhores para detectar algum tipo de falha em específico, conseguir detectar uma maior quantidade de diferentes falhas com uma mesma técnica, significa reduzir custos e otimizar o processo.

Dentre a diversidade de máquinas de uma planta industrial, pode-se destacar os motores elétricos que são largamente utilizados nos acionamentos de bombas, compressores, ventiladores, equipamentos para processamento e manuseio de cargas e em várias outras diferentes aplicações. Esta grande utilização é devido às suas características como: robustez, mecânica simples e fácil adaptação às mais variadas condições de operação. Porém, durante seu funcionamento, são submetidos a grandes cargas, expostos a ambientes agressivos e juntamente com seu envelhecimento natural esses equipamentos podem falhar.

Diferente de outros equipamentos exclusivamente mecânicos, elétricos ou hidráulicos, os motores elétricos apresentam tanto falhas de origem mecânica quanto de origem elétrica, o que pode tornar o processo de monitoramento difícil e demorado quando utiliza-se apenas uma técnica ou bastante oneroso à medida que se for utilizando um conjunto maior de técnicas para detectar e diferenciar as falhas.

Assim, nos últimos anos, vários pesquisadores tem se voltado para a aplicação de técnicas de inteligência artificial juntamente com as técnicas de manutenção preditiva para a detecção de

falhas, Chung *et al.* (1989); Frank e Köppen-Seliger (1997); Batur *et al.* (2002); Brito (2002); Jack (2002); Poyhonen *et al.* (2002); Caporalli Filho (2003); Kowalski e Orłowska-Kowalska (2003); Baccarini (2005); Yang *et al.* (2005); Aydin *et al.* (2007); Lamim Filho (2007); Rodríguez e Arkkio (2008); Kurek e Osowski (2010); Qu e Zuo (2010); Baccarini *et al.* (2011); Chen e Huang (2011). Dentre as principais técnicas de inteligência artificial pode-se destacar as Redes Neurais Artificiais, Lógica *Fuzzy*, Sistemas Especialista e Máquinas de Vetores de Suporte.

Vislumbra-se então que a aplicação de técnicas de monitoramento para motores elétricos vem se tornando um forte atrativo dentro de uma planta industrial, pois a detecção de um problema antes que o mesmo danifique a máquina irá permitir que a equipe de manutenção possa tomar as devidas precauções, evitando-se paradas inesperadas na linha de produção e com o auxílio das técnicas de inteligência artificial o processo de detecção e diagnóstico se tornaria muito mais rápido.

Seguindo esta idéia, Brito (2002) desenvolveu um sistema híbrido que usa técnicas de inteligência artificial (redes neurais e sistema especialista), para detecção e diagnóstico de falhas em motores de indução trifásicos, visando sua inclusão em programas de manutenção preditiva.

Baccarini (2005), propôs um sistema de detecção e diagnóstico de falhas elétricas (curto-circuito entre espiras do estator, quebra de barras e/ou anéis do rotor) e mecânicas (desalinhamento, desbalanceamento, folga mecânica) em motores de indução trifásicos através de modelos matemáticos e redes neurais artificiais.

Em ambos os trabalhos expressivos resultados são obtidos, porém, quando se trabalha com redes neurais artificiais, alguns inconvenientes podem ser enfrentados. Por exemplo, segundo Ghate e Dudul (2009), as redes neurais podem ser consideradas como uma "caixa preta", uma vez que, fornecem pouca explicação sobre o processo de predição e detecção de falhas.

Tem-se ainda, que as redes neurais artificiais não são dotadas de algoritmos de treinamento que maximizam a capacidade de generalização de uma forma sistemática, o que pode levar a um sobre ajuste do modelo aos dados, (HORNIK *et al.*, 1989).

Outra técnica bastante utilizada, são os sistemas baseados em lógica *fuzzy*, e que têm sido utilizados também para detecção de falhas em motores elétricos, uma vez que a interpretação da

condição do motor de indução trifásico é um conceito *fuzzy*, (ZIMMERMANN, 1996). Assim, nos últimos anos, pesquisadores têm proposto o diagnóstico de falhas baseado na lógica *fuzzy*.

Lamim Filho (2007), utilizou a lógica *fuzzy*, juntamente com sinais de fluxo magnético para o monitoramento de motores de indução trifásicos em tempo real e sem a necessidade de nenhum conhecimento prévio por parte do operador da máquina.

Em sistemas de lógica *fuzzy* é possível implementar princípios de detecção de falhas, interpretar e analisar seus resultados com um bom embasamento teórico. Porém, a lógica *fuzzy* apresenta algumas dificuldades na definição de suas regras e na forma de se processar os dados de entrada, (GHATE E DUDUL, 2009). O que implica na necessidade de um especialista para a criação das respectivas regras.

Recentemente as máquinas de vetores de suporte, vem ganhando mais aplicações na área de detecção de falhas devido a sua alta taxa de acerto e boa capacidade de generalização, (SAMANTA *et al.*, 2003).

Poyhonen *et al.* (2002), aplicaram as máquinas de vetores de suporte para classificar falhas elétricas em motores de indução trifásicos, utilizando para o treinamento um banco de dados composto por falhas simuladas.

Já em Baccarini *et al.* (2011), é apresentado as máquinas de vetores de suporte para detecção de falhas treinadas a partir de dados experimentais, mas apenas falhas mecânicas foram abordadas. Os resultados obtidos são muito bons e se observa que os procedimentos a serem adotados para a utilização da técnica são relativamente simples e a escolha dos parâmetros, utilizados no treinamento, que são escolhidos manualmente neste trabalho, podem ser automatizados.

Objetivando abranger um maior número de falhas, em Widodo e Yang (2007a) as máquinas de vetores de suporte são aplicadas em conjunto com uma extração de parâmetros não linear para detectar falhas em motores elétricos. Diferentes falhas são abordadas e significativos resultados são obtidos, porém, nada é dito sobre a capacidade de generalização das máquinas de vetores de suporte, na classificação de falhas em diferentes níveis de severidade.

Com o intuito de avaliar as pesquisas relacionadas à detecção de falhas através das máquinas de vetores de suporte, Widodo e Yang (2007b) apresentaram um levantamento bibliográfico da detecção de falhas em equipamentos usando máquinas de vetores de suporte. Suas pesquisas compreendiam artigos de 1996 a 2006. Os autores concluíram que as máquinas de vetores de suporte tendem a ser a mais promissora técnica para diagnóstico de falhas, mas ainda precisam de mais incentivo e atenção devido à falta de trabalhos voltados à pesquisa de aplicações deste conceito no monitoramento das condições dos equipamentos e diagnóstico de falhas.

As linhas de pesquisa envolvendo detecção de falhas em motores elétricos, apesar de existirem a algum tempo, nem sempre estão voltadas para o meio industrial. E as máquinas de vetores de suporte apresenta esta possibilidade e com alguns atrativos a mais, pois, comparadas com as outras técnicas, as máquinas de vetores de suporte se destacam pela sua facilidade de uso, poucas variáveis envolvendo o seu processo de treinamento, boa capacidade de generalização, entre outros.

1.1 Objetivos do trabalho

A proposta deste trabalho é a utilização das máquinas de vetores de suporte para a detecção e diagnóstico de falhas mecânicas (desalinhamento, desbalanceamento e folga mecânica) e elétricas (curto-circuito entre espiras do estator, desequilíbrio de fase e barras quebradas) em motores de indução trifásicos. Encima desta proposta objetiva-se também:

- Verificar as potencialidades das máquinas de vetores de suporte para a detecção de falhas;
- Com a utilização de apenas um sensor de aceleração para medição de vibração detectar tanto falhas mecânicas quanto falhas elétricas e caracterizá-las.
- Realizar um estudo comparativo entre as redes neurais artificiais, lógica *fuzzy* e máquinas de vetores de suporte, avaliando suas vantagens e desvantagens.
- Avaliar a capacidade das técnicas (redes neurais, lógica *fuzzy* e máquinas de vetores de suporte) em classificar corretamente falhas, a outros níveis de severidade não utilizados no treinamento.

1.2 Organização do texto

Este trabalho foi dividido em seis capítulos, onde são apresentadas as revisões bibliográficas, as metodologias abordadas, procedimentos experimentais realizados, resultados encontrados e a conclusão referente ao trabalho. A seguir, faz-se uma breve descrição do conteúdo de cada um desses capítulos:

No Capítulo II introduz-se algumas das principais técnicas de inteligência artificial, seu desenvolvimento durante os anos e suas aplicabilidades.

No Capítulo III é descrita a formulação original das máquinas de vetores de suporte, utilizada na solução de problemas com duas e múltiplas classes.

No Capítulo IV apresentam-se as principais falhas mecânicas e elétricas em motores de indução trifásicos e como a análise de vibrações é aplicada para a detecção destas falhas. Tendo como um dos objetivos a utilização dos sinais de vibração para treinamento e validação das técnicas de inteligência artificial.

No Capítulo V é mostrada a bancada experimental e os espectros de vibração, bem como o processo de inserção de cada uma das falhas em estudo. É apresentado também a parametrização das máquinas de vetores de suporte para que se consiga um bom desempenho na classificação das falhas e a proposta de uma normalização para classificar falhas que estejam em diferentes níveis de severidade em relação ao banco de dados utilizado no treinamento.

As conclusões e sugestões para trabalhos futuros são apresentadas no Capítulo VI.

2 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

2.1 Histórico

Desde os anos 30 já se encontravam correntes de pensamento em torno da inteligência artificial (IA), (PRICE, 1983), tanto que em 1943, McCulloch e Pitts propuseram um primeiro modelo matemático para um neurônio de rede neural artificial. Sendo uma rede neural artificial, um conjunto de neurônios interligados baseados no cérebro humano.

Mas apenas em 1956 oficialmente nascia a IA em uma conferência de verão em Dartmouth College, New Hampshire, USA. Na proposta desta conferência, escrita por John McCarthy, constava a intenção de realizar estudos sobre o tópico de inteligência artificial. Ao que tudo indica esta foi a primeira menção oficial da expressão Inteligência Artificial, (MIROWSKI, 2004). Mas já no início a IA gerou polêmicas, primeiramente pelo seu nome, na época considerado presunçoso, pelas suas definições, metodologias e objetivos. O desconhecimento dos princípios que fundamentam a inteligência, por um lado, e dos limites práticos da capacidade de processamento dos computadores, por outro, levou periodicamente a promessas exageradas e às correspondentes decepções.

Dada a dificuldade de uma definição formal precisa para IA, uma vez que seria necessário definir, primeiramente, a própria inteligência, foram propostas algumas definições operacionais:

- Uma máquina é inteligente se ela é capaz de solucionar uma classe de problemas que requerem inteligência para serem solucionados por seres humanos, (MCCARTHY E HAYES, 1987);
- Inteligência Artificial é a parte da ciência da computação que compreende o projeto de sistemas computacionais que exibam características associadas a inteligência humana, (PRICE, 1983); ou ainda
- Inteligência Artificial é o estudo das faculdades mentais através do uso de modelos computacionais, (CHARNIAK E MCDERMOTT, 1985);

Outros se recusam a propor uma definição para o termo e preferem estabelecer os objetivos da IA:

- Tornar os computadores mais úteis e compreender os princípios que tornam a inteligência possível, (WINSTON, 1983).

No início dos anos 60, os computadores começam a surgir. Em 1965, L. Zadeh propõe a Lógica Difusa ou Nebulosa (*Fuzzy Logic*), que para alguns autores é o marco inicial da Inteligência Computacional, pois começa a tratar também das incertezas. Por volta dos anos 70 e 80 surgem também técnicas como sistemas especialistas e mais a frente, nos anos 90, surgem as máquinas de vetores de suporte que propõem um novo princípio de aprendizagem baseado no aprendizado estatístico.

Atualmente os estudos de IA dividem-se em quatro ramos fundamentais. Uma área ligada ao estudo do aprendizado de máquinas e ao conexionismo que se relaciona também com a capacidade dos computadores aprenderem e reconhecerem padrões. Um outro ramo ligado à biologia molecular na tentativa de construir vida artificial. Um terceiro relacionado com a robótica, ligada à biologia e procurando construir máquinas que alojem vida artificial. E finalmente o ramo clássico da IA que se liga desde o início à psicologia, desde os anos 70 à epistemologia e desde os anos 80 à sociologia, e que tenta representar na máquina os mecanismos de raciocínio e de procura.

A construção de máquinas inteligentes pressupõe ou depende dos seguintes aspectos: filosofia (teorias sobre conhecimento, mente, ...), matemática (lógica, probabilidade, decisão, ...), psicologia (percepção, memória, ...), linguística (sintaxe, semântica, ...) e informática (ferramenta).

Existem vários campos de estudo dentro da IA com o propósito de dotar a máquina de capacidade de raciocínio, aprendizado e auto-aperfeiçoamento, pode-se citar alguns desses campos:

- Processamento de Linguagem Natural - Área que estuda a construção de programas capazes de compreender/interpretar a linguagem natural e gerar textos. Objetiva aperfeiçoar a comunicação entre as pessoas e os computadores.

- Reconhecimento de Padrões - É uma das áreas de pesquisa bem avançadas da IA. A capacidade de reconhecimento de padrões permite ao programa reconhecer a fala em linguagem natural, os caracteres digitados, a escrita (ex.: assinatura), entre outros. Os scanners, por exemplo, utilizam programas de reconhecimento óptico desenvolvidos pelas pesquisas em IA.
- Visão de Computador - Voltado para o desenvolvimento de maneiras de trabalhar com a visão bidimensional e tridimensional.
- Programação de Jogos - Área de estudo voltada para construção de jogos envolvendo raciocínio. Um exemplo é o jogo de xadrez, usado como uma das primeiras experiências em programação de raciocínio artificial. Neste jogo o computador deve analisar milhões de jogadas por segundo para tentar derrotar o adversário.
- Robótica - É o campo de estudo voltado para desenvolver meios de construir máquinas que possam interagir com o meio (ver, ouvir e reagir aos estímulos sensoriais). A expressão robô, que vem do tcheco *robota* e que significa trabalhador, foi criada por Karel Capek, em 1917. O primeiro robô industrial do mundo, batizado de UNIMATE, surgiu em 1962.
- Aprendizado - Existem programas de IA que conseguem aprender certos fatos por meio da experiência, desde que esse conhecimento possa ser representado de acordo com o formalismo adotado pelo programa.

A seguir será feita uma introdução às principais técnicas de inteligência artificial, reservando para as máquinas de vetores de suporte um capítulo a parte, pois são elas o foco deste trabalho.

2.2 Redes neurais artificiais

As Redes Neurais Artificiais (RNA) são uma tentativa de criar um modelo matemático capaz de apresentar algumas características de inteligência. Em outras palavras, um modelo capaz de aprender, de se adaptar mesmo em presença de sinais com distorções, de tomar decisões, etc. Nesta tentativa de criar um modelo que representasse de certa forma a estrutura e funcionalidade do sistema nervoso humano, e que pudesse ser implementado usando ferramentas computacionais disponíveis, foram desenvolvidos diversos modelos. Cada um destes modelos apresentam características diferentes, que variam de acordo com a forma como este é projetado.

Uma RNA é uma estrutura de processamento paralelo de informação consistente em elementos de processamento (neurônios) interconectados por canais de comunicação unidirecionais. Cada neurônio tem uma única saída como resultado do processamento de um conjunto de entradas, esta saída pode fazer parte do conjunto de entradas de outros neurônios ou como uma das saídas da rede, (HAYKIN, 1998). As RNAs têm normalmente um grande potencial para paralelismo, visto que seus componentes (os neurônios) são altamente independentes entre si, (HAYKIN, 1998).

Uma rede neural pode ser modelada pela especificação de um conjunto de aspectos que permitam a sua definição e caracterização. Estes aspectos englobam três níveis de modelagem que devem ser considerados no desenvolvimento das RNAs. São eles: modelagem do neurônio empregado pela rede, modelagem da topologia da rede e modelagem da regra de aprendizado.

As redes neurais têm sido desenvolvidas como generalizações de modelos matemáticos da cognição humana ou biologia neural, assumido que:

- O processamento da informação ocorre em vários elementos chamados neurônios;
- Os sinais são propagados de um elemento a outro através de conexões;
- Cada conexão possui um peso associado, que, em uma rede neural típica, pondera o sinal transmitido; e
- Cada neurônio aplica uma função de ativação à sua entrada de rede para determinar sua saída.

E podem ser caracterizadas por três aspectos principais:

1. O padrão de conexões entre as unidades (arquitetura);
2. O método de determinação dos pesos das conexões (algoritmo de treinamento ou aprendizado);
3. Sua função de ativação.

2.2.1 Modelagem de um neurônio

A modelagem dos neurônios corresponde à especificação da estrutura e funcionamento do operador neurônio, também chamado Neurônio Formal. O modelo de neurônio formal foi introduzido por McCulloch e Pitts (1988). Este modelo é composto basicamente por:

Conexões de entrada (sinapses): por ela o neurônio recebe a ativação (estímulos) de outros neurônios da rede ou do meio externo. Cada entrada x_n chega a um outro neurônio com um respectivo peso w_n . Se o estímulo for positivo ele é dito como excitatório, mas se ele for negativo recebe o nome de estímulo inibitório. Na Figura 2.1, tem-se a representação de um neurônio.

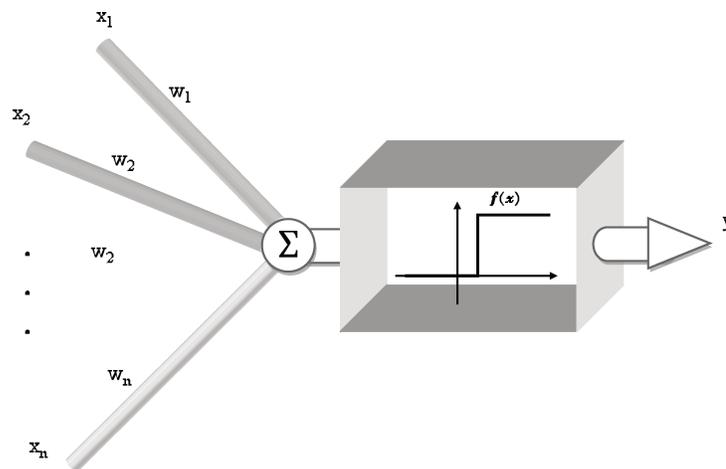


Figura 2.1: Modelo de McCulloch e Pitts.

Função somatória: combina todas as entradas em um valor único que representa a ativação global das entradas. Assim o potencial interno $u(t)$ do neurônio é dado pela ponderação das entradas com os pesos das respectivas conexões, Equação (2.1).

$$u(t) = \sum_{i=1}^n w_i x_i \quad (2.1)$$

Função de ativação: responsável por converter a soma da ativação das entradas em ativação de saída. A função de ativação é um fator básico para diferenciar os neurônios, pois é através do potencial interno do neurônio que se tem o valor que deve ser enviado aos demais neurônios.

Conexões de saída (axônios): elementos através dos quais a ativação de saída (estímulos) é levada às entradas de ativação de outros neurônios da rede.

Uma definição mais completa do neurônio inclui ainda a especificação de um deslocamento (*bias*) que é aplicado ao resultado da função somatória, antes deste resultado ser submetido à função de ativação. Este *bias* permite a obtenção de resultados mais genéricos e também permite o treinamento da rede para um número maior de possibilidades. Em alguns modelos, o *bias* aparece sob a forma de um peso complementar, cuja entrada é considerada sempre ativa, (HAYKIN, 1998). Mais formalmente, o *bias* serve para elevar o número de graus de liberdade do problema, permitindo uma melhor adaptação por parte da rede neural, ao conhecimento à ela fornecido.

Desta forma, pode-se escrever a saída do neurônio pela Equação (2.2).

$$y_k = f \left| \sum_{i=0}^n (w_i x_i + \theta_k) \right|, \quad (2.2)$$

sendo y_k a saída do neurônio k ; f é a função de ativação usada pelo neurônio; w_i é o peso da entrada i ; x_i é o valor da entrada i ; e θ_k é o *bias* do neurônio k , n é o número total de entradas do neurônio.

2.2.2 Modelagem da topologia

A modelagem da topologia de rede consiste em descrever as interconexões entre os neurônios da rede. A topologia da rede tem impacto nas características e propriedades apresentadas pelo modelo. Existem diversos tipos de topologia, entre as quais pode-se citar: totalmente conectada, linear, retro-alimentada, multi-nível, matricial ou livre. Um modelo de RNA pode também corresponder a uma combinação das topologias anteriormente citadas.

As conexões entre neurônios, que caracterizam a topologia de rede, são normalmente representadas através de uma matriz de pesos $P = p_{ij}$. Nesta matriz o elemento p_{ij} representa o peso da conexão que sai do neurônio i e entra no neurônio j . Seu valor pode ser inibitório (negativo) ou excitatório (positivo). Se não há conexão entre o neurônio i e o neurônio j então, $p_{ij} = 0$.

Rede neural unidirecional (*feed-forward networks*)

Nesta topologia o fluxo de dados é estritamente unidirecional, ou seja, os dados entram na rede pela camada de entrada, percorrem as camadas intermediárias (ocultas), nas quais são processados e saem pela camada de saída da rede, Figura 2.2. Nenhum neurônio recebe sinal de outro da mesma camada, e tão pouco de uma camada subsequente.

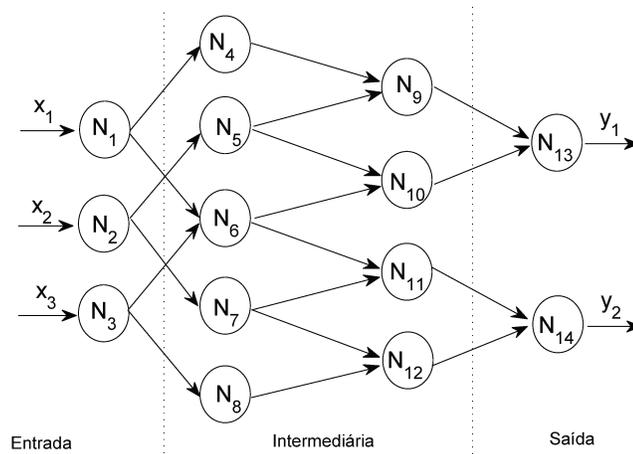


Figura 2.2: Uma rede unidirecional com três camadas de conexões.

Modelagem da regra de aprendizado

A modelagem de aprendizado ou plasticidade são as especificações do conjunto de leis que determinam como os pesos das conexões serão inicializados e alterados durante o funcionamento da rede, (LIPPMANN, 1988). Os pesos das conexões devem ser inicializados ou modificados de forma que a rede responda adequadamente aos estímulos do ambiente externo.

A principal característica desta ferramenta numérica, que é a rede neural, é a sua capacidade de aprender. Neste contexto, a aprendizagem é definida como a variação dos pesos das ligações, de modo a armazenar implicitamente uma dada informação, que poderá ser facilmente recuperada.

2.2.3 Modelos de redes neurais artificiais

Existe uma série de modelos de redes neurais artificiais, dentre os quais se destacam os modelos de aprendizado supervisionado por correção de erros. A maioria dos modelos restantes é baseada em variações deste modelo básico.

Pode-se dividir os modelos de redes em diversos grupos, de acordo com:

- Treinamento: pode ser auto-aprendizado ou associativo.
- Regra de aprendizado: algoritmo competitivo ou algoritmo adaptativo por correção de erros.
- Interconexões dos neurônios: redes com neurônios sem realimentação ou com realimentação.
- Organização da rede: um nível único ou em diversos níveis (camadas).
- Distribuição das memórias: localizada ou distribuída.
- Classificação: separador linear ou não-linear.

Dentre as divisões citadas acima a regra de aprendizado merece uma ênfase maior. A regra de aprendizado do tipo competitivo é encontrada nas redes de Hopfield e Kohonen. Este algoritmo

é caracterizado pelas conexões laterais dos neurônios com seus vizinhos, estabelecendo assim uma "competição entre os neurônios" que levará a rede a um estado estável. Já as redes com aprendizado do tipo correção de erros são baseadas no princípio da adaptação e correção dos pesos de atuação de cada neurônio, visando minimizar o erro na saída da rede até que este responda da maneira desejada. A correção de erros está diretamente ligada ao aprendizado do tipo supervisionado.

2.2.4 Algoritmo de *back-propagation*

O *back-propagation* é um algoritmo para treinamento de Redes Multi-Camadas mais difundido. Baseia-se no aprendizado supervisionado por correção de erros, (HAYKIN, 1998). É difícil dizer quem é o real criador do *back-propagation*, pois vários trabalhos foram feitos em lugares diferentes na mesma época. No entanto foi Rumelhart *et al.* (1986) quem mais difundiu este método.

Uma rede que utiliza o modelo *back-propagation* é uma rede com neurônios em três ou mais níveis: um nível de entrada (*input layer*), um ou mais níveis intermediários ou ocultos (*hidden layers*) e um nível de saída (*output layer*). A função de ativação utilizada é constituída por duas etapas:

1. Propagação: Depois de apresentado o padrão de entrada, a resposta de um neurônio é propagada como entrada para os neurônio das camadas seguintes, até a camada de saída, onde é obtida a resposta da rede e o erro é calculado;
2. Retropropagação (*backpropagation*): Desde a camada de saída até a camada de entrada, são feitas alterações nos pesos.

Durante a fase treinamento, deve-se apresentar um conjunto formado pelo par: entrada para a rede e valor desejado para resposta da entrada. A saída será comparada ao valor desejado e será computado o erro global da rede, que influenciará na correção dos pesos no passo de retropropagação. Apesar de não haver garantias que a rede forneça uma solução ótima para o problema, este processo é muito utilizado por apresentar uma boa solução para o treinamento de Multi-Camadas.

Noções gerais

Sinal funcional: um sinal funcional é um sinal (estímulo) que chega na entrada e é propagado positivamente neurônio a neurônio através da rede, e aparece na saída como um sinal de saída.

Sinal de erro: os sinais de erro originam-se nas saídas e são retropropagados neurônio a neurônio através da rede.

A camada de entrada é composta, em geral, por neurônios sensoriais, ou seja, unidades que não modificam os sinais externos, apenas fazem a distribuição para a primeira camada intermediária. As unidades de saída constituem a camada de saída da rede, e as demais unidades constituem as camadas intermediárias. As camadas intermediárias são todas aquelas que não fazem parte da entrada e nem da saída.

Cada unidade intermediária ou de saída é responsável por duas tarefas:

- Calcular o sinal na saída da unidade, que geralmente é expresso como uma função não-linear do sinal de entrada e pesos sinápticos associados;
- Calcular uma estimativa instantânea do vetor gradiente, que é necessário para a retropropagação do erro através da rede.

Treinamento

Treinar as redes multi-camadas foi por bastante tempo impraticável, até que Rumelhart *et al.* (1986), apresentou o algoritmo de *back-propagation*. O algoritmo de aprendizado proposto por Rumelhart requer a presença de pares de padrões de entrada e saída. A rede será treinada de forma que quando um dos pares de entrada for apresentado à rede, o respectivo par de saída deverá aparecer na saída.

Primeiramente, o sistema usa apenas o vetor de entrada para produzir sua própria saída (ati-

vação - *forward*). Após, então, é possível comparar a saída atual com a saída desejada e assim determinar o erro. Se não há diferenças entre a saída desejada e a saída obtida, nenhuma alteração precisa ser feita nos pesos dos neurônios. Caso contrário, os pesos têm de ser modificados para reduzir o erro (retropropagação do erro - *backward*).

A adaptação dos pesos anteriormente citada, visa garantir que a saída da rede se comporte exatamente como a saída desejada indicada juntamente com os exemplos fornecidos para a aprendizagem de uma tarefa. Desta forma, o comportamento da rede imita o comportamento apresentado pelo sistema estudado, e que é representado através de exemplos. Uma rede é capaz de aprender a:

- Reproduzir um comportamento;
- Classificar dados (informações, imagens, etc);
- Aproximar e reproduzir funções;
- Realizar previsões do "futuro" baseada nas suas experiências "passadas".

Uma vez encerrada a fase de aprendizado, ou até mesmo durante a realização desta, é realizada usualmente uma etapa de testes que permite medir o grau de generalização da rede. Existem métodos que decidem o bom momento de parar a aprendizagem (evitando o *overfitting*) baseados na evolução da curva de resposta sobre o conjunto de teste.

Deficiências do *back-propagation*

Mesmo com as vantagens do algoritmo, existem alguns casos que fazem com que este não possa ser utilizado em todo o tipo de aplicações. O maior problema é o longo processo de treinamento. Ele pode também resultar em uma taxa de acerto do aprendizado e uma taxa de generalização ruins, pois não há garantias de que os mínimos globais de erro sejam encontrados, (HAYKIN, 1998). Em outras palavras, algumas vezes, dependendo dos pesos iniciais da curva de aprendizado e dos dados de entrada, a rede neural artificial pode até não aprender.

Vários algoritmos avançados baseados no *back-propagation* otimizaram e aprimoraram o processo de treinamento, mas as falhas no aprendizado ainda ocorrem, e normalmente tem duas fontes: a paralisação da rede e os mínimos locais.

Paralisação da rede

Durante o treinamento da rede os pesos podem ser ajustados para valores muito altos. As entradas do nível oculto ou saídas do nível oculto podem alcançar valores muito altos (positivos ou negativos), e em decorrência da função de ativação a rede vai ter um valor de ativação muito próximo de zero ou de um, conseqüentemente, o processo de treinamento pode virtualmente ficar paralisado, (HAYKIN, 1998). Isto ocorre porque um neurônio que aprendeu alguma coisa (saída com valor elevado) tende a se acomodar e não mudar seus pesos.

Esta "acomodação" da rede é proposital, sua função é evitar que os neurônios fiquem mudando os seus pesos a todo o momento. A rede nunca se estabilizaria, não haveria convergência, pois, ao apresentar um novo exemplo a rede sobreporia a aprendizagem do exemplo anterior.

A implementação desta "acomodação" é feita pela função de ativação, pois na inicialização da rede os neurônios tem pesos baixos bem próximos a zero, com isto a saída da função de ativação fica em torno de 0.5 (nem zero nem 1), e a derivada da função de ativação tem seu valor mais alto, gerando assim, um valor grande que é multiplicado pelo erro estimado, e portanto resultando em uma "grande" modificação dos pesos (depende também da velocidade de aprendizado). Quando o valor dos pesos (somatório) chega a um valor alto (positivo ou negativo), a função de ativação gera saídas próximas de 1 ou de 0, e sua derivada gera valores baixos, diminuindo assim o ritmo de alteração dos pesos, com efeito, o neurônio fica menos suscetível ao aprendizado.

Mínimos locais

A superfície de erro de uma rede é cheia de picos e vales. A rede pode se concentrar em um mínimo local, deixando de se concentrar em um mínimo global, onde o erro final seria menor.

Existem alguns métodos para tentar fugir deste problema, alguns deles são o *momentum* e o *weight decay*.

Momentum

O *momentum* é uma técnica que objetiva aumentar a velocidade de convergência da RNA treinada pelo algoritmo *back-propagation* (BEALE E JACKSON, 1990; FREEMAN E SKAPURA, 1991), tentando escapar dos mínimos locais pela inércia da alteração dos pesos na direção do mínimo mais global, baseada na etapa anterior de correção dos pesos, permitindo usar velocidades de aprendizado maiores, sem no entanto correr o risco do algoritmo ignorar o mínimo global.

Weight decay

Este método tenta impedir que os pesos tenham valores muito altos pela atenuação dos valores obtidos no cálculo dos erros em função dos valores dos próprios pesos, (WERBOS, 1988). O *weight decay* soma um termo de penalidade a função de erro.

2.2.5 Outras fontes de problemas

A seguir serão apresentadas mais algumas fontes de problemas para o algoritmo de *back-propagation*.

Tamanho e topologia da rede

Este problema é de difícil solução porque ele depende diretamente da aplicação e utilização da rede, não existe uma regra para determinar uma boa topologia de rede. A topologia depende do treinamento, da quantidade de ruído e da complexidade do que a rede está tentando aprender.

Existem problemas em que são utilizados apenas um neurônio de entrada e um de saída, mas que requerem milhares de neurônios no nível intermediário, e ainda problemas com milhares de neurônios de entradas e saídas, que requerem apenas um neurônio no nível intermediário, ou até mesmo nenhum.

Atualmente este é um dos problemas mais estudados na área das redes neurais: como determinar o tamanho ideal de uma rede, assim como as conexões entre seus elementos. Alguns autores reforçam a importância da escolha adequada da topologia de rede para que se obtenha bons resultados, (TORRES *et al.*, 2004). Outros trabalhos apresentam procedimentos para a escolha da topologia de rede, como é o caso do trabalho publicado por Wang *et al.* (1994), porém são procedimentos complexos e não garantem boas taxas de acerto.

Inicialização aleatória dos pesos

A inicialização aleatória dos pesos da rede é um passo importante, pois, dependendo dos valores da inicialização, a rede pode até não conseguir aprender. Como não é possível saber quais são os pesos ideais para cada entrada dos neurônios, normalmente se opta por inicializá-los aleatoriamente. Isto não resolve o problema por completo, e muitas vezes pode ser necessário iniciar o treinamento novamente, com outros pesos.

Parâmetros de aprendizagem

Outro problema é a estimativa dos parâmetros de aprendizagem (velocidade de convergência, *momentum* e etc), estes parâmetros influenciam tanto na velocidade de aprendizado, quanto na própria capacidade de aprendizado da rede. A maior parte destes parâmetros são determinados na prática, ou seja, usam-se valores iniciais padrão, e depois vão se alterando estes parâmetros empiricamente.

Interrupção do treinamento

Descobrir o ponto em que se deve parar o treinamento, é também um problema. Como determinar que a rede já aprendeu o suficiente sobre os padrões de treinamento? Como medir a capacidade de generalização da rede?

Alguns pesquisadores usam a regra de treinar a rede um número de vezes equivalente ao número de possibilidades de casos do treinamento, mas este método é muitas das vezes inviável, ainda mais se o número de possibilidades for muito grande. Outro método é a técnica chamada de *early stopping*, que consiste no seguinte:

- Dividir os dados disponíveis para treinamento em dois grupos, o de treinamento e o de validação;
- Usar um grande número de neurônios na camada intermediária;
- Usar pesos iniciais randômicos muito pequenos;
- Usar uma baixa taxa de aprendizado;
- Calcular periodicamente a taxa de erro de validação durante o treinamento;
- Parar o treinamento quando a taxa de erro de validação começar a crescer.

As vantagens deste método são as seguintes:

- Rápido;
- Pode ser aplicado com sucesso em redes cujo o número de pesos seja bem maior que o número de exemplos;
- Requer apenas uma parametrização do usuário, que é proporcional para os dados de treinamento e de validação.

As desvantagens são:

- Definição da quantidade de dados a ser utilizada para os grupos de validação e treinamento;
- Definição da sistemática a ser utilizada para separar os dados nos dois grupos;
- Apenas se sabe que o erro de validação está aumentando após ter passado do ponto ótimo de generalização, então acaba-se parando a rede em um mínimo local ao invés de um mínimo global.

2.3 Lógica Fuzzy

A lógica *fuzzy*, também conhecida como lógica nebulosa ou difusa, foi desenvolvida por Lofti A. Zadeh, originalmente um engenheiro e cientista de sistemas, durante a década de 1960. O artigo publicado pelo autor em 1965 pela Universidade da Califórnia, em Berkeley, revolucionou o assunto com a criação de sistemas *fuzzy*, (ZADEH, 1965).

Segundo Cox (1994), o artigo de Zadeh em 1965 à respeito dos conjuntos *fuzzy* foi motivado pela idéia de que os tradicionais métodos de análise de sistemas não eram tão úteis para lidar com problemas cuja relação entre as variáveis não podiam ser representadas em termos de diferenciação ou equações diferenciais.

Explicando melhor as palavras de Zadeh; as tecnologias disponíveis não eram capazes de automatizar as atividades relacionadas a problemas que compreendessem situações ambíguas, não passíveis de processamento através da lógica booleana. Era preciso algo mais do que somente dois valores de verdade possíveis. Segundo Fernandes e Santos (2002), a lógica desenvolvida por Zadeh é uma lógica polivalente especial que é voltada para fenômenos de imprecisão e desenvolvimento de ferramentas a serem utilizadas na modelagem através de graus de verdade em uma escala ordenada.

A lógica *fuzzy* é uma extensão da lógica booleana que admite valores lógicos intermediários entre o FALSO (0) e o VERDADEIRO (1); por exemplo o valor médio "TALVEZ" (0.5). Isto significa que um valor lógico difuso é um valor qualquer no intervalo de valores entre 0 e 1. Este tipo de lógica engloba de certa forma conceitos estatísticos principalmente na área de Inferência.

As implementações da lógica *fuzzy* permitem que estados indeterminados possam ser tratados por dispositivos de controle. Desse modo, é possível avaliar conceitos não-quantificáveis, como por exemplo, avaliar a temperatura (quente, morno, frio, etc...), Figura 2.3.

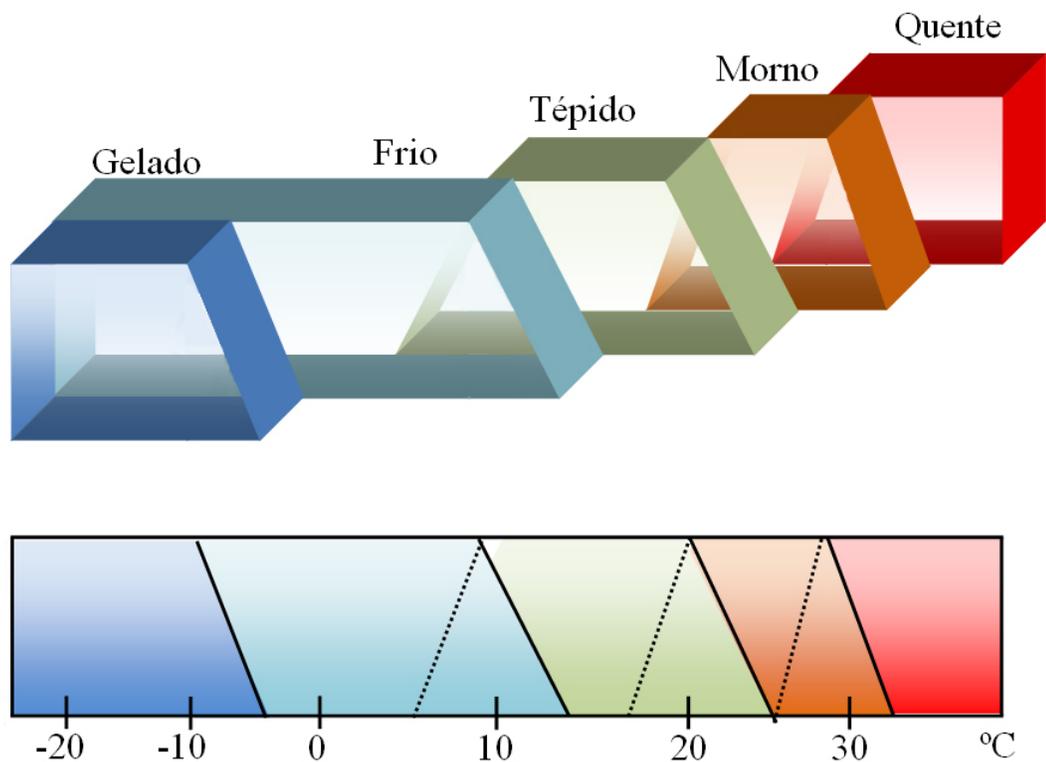


Figura 2.3: Exemplo de avaliação de temperatura.

O objetivo da lógica *fuzzy* é fazer com que as decisões tomadas pela máquina se aproximem cada vez mais das decisões humanas, principalmente ao trabalhar com uma grande variedade de informações vagas e incertas.

Os conjuntos *fuzzy* são o caminho para aproximar o raciocínio humano à forma de interpretação da máquina. Assim, eles são na verdade uma "ponte" que permite ainda o emprego de avaliações qualitativas, como por exemplo "muito quente" e "muito frio".

Isto significa que nos conjuntos convencionais tem-se limites bruscos entre os elementos pertencentes ao conjunto e os elementos não pertencentes.

Em um conjunto *fuzzy* a transição entre o membro e o não membro está numa faixa gradual, sendo associado um grau entre "0" (totalmente não membro) e "1" (totalmente membro).

Cruz (2002) sugere alguns exemplos que ajudam a ilustrar como funciona o pensamento *fuzzy*:

- Se é hora de pico, aumente a frequência dos trens;
- Se a roda deslizar, solte o freio um pouco;
- Se a terra está muito seca e a temperatura está alta, regue por muito tempo;
- Se a taxa de juros for alta e o déficit for alto, teremos uma recessão branda.

Segundo Sandri e Correa (1999), formalmente, um conjunto *fuzzy* A é definido por uma função de pertinência $\mu_A : [0 - 1]$. Essa função associa a cada elemento x do conjunto *fuzzy* A um grau $\mu_A(x)$ de pertinência. Isto representa o grau de compatibilidade entre x e o conceito expresso por A , sendo:

- $\mu_A(x) = 1$, indica que x é completamente compatível com A ;
- $\mu_A(x) = 0$, indica que x é completamente incompatível com A ;
- $0 < \mu_A(x) < 1$, indica que x é parcialmente compatível com A , com grau $\mu_A(x)$.

Desta forma, a determinação das regras de controle é feita através de um conjunto de regras formado pela união de todas as variáveis reais de forma a representarem todos os estados das grandezas utilizadas no sistema *fuzzy*.

Na Tabela 2.1, tem-se as principais características, vantagens e desvantagens da lógica *fuzzy*.

Tabela 2.1: Características da lógica *fuzzy*, (CAMARGOS, 2002).

Características	Vantagens	Desvantagens
A Lógica <i>Fuzzy</i> está baseada em palavras e não em números, ou seja, os valores verdades são expressos linguisticamente. Por exemplo: quente, muito frio, verdade, longe, perto, rápido, vagaroso, médio.	O uso de variáveis linguísticas nos deixa mais perto do pensamento humano.	Necessitam de mais simulação e testes.
Possui vários modificadores de predicado, tais como: muito, mais ou menos, pouco, bastante, médio.	Requer poucas regras, valores e decisões.	Não aprendem facilmente.
Possui também um amplo conjunto de quantificadores, como: poucos, vários, em torno de, usualmente.	Simplifica a solução de problemas e a obtenção da base do conhecimento.	Dificuldades de estabelecer regras corretamente.
Faz uso das probabilidades linguísticas (como, provável e improvável) que são interpretados como números <i>fuzzy</i> e manipulados pela sua aritmética.	Mais variáveis observáveis podem ser valoradas.	Não há uma definição matemática precisa.
Manuseia todos os valores entre 0 e 1, tomando estes, como um limite apenas.	Mais fáceis de entender, manter e testar.	
	São robustos. Operam com falta de regras ou com regras defeituosas.	
	Acumulam evidências contra e a favor.	
	Proporciona um rápido protótipo dos sistemas.	

2.3.1 Sistemas de inferência *fuzzy*

Um sistema de inferência *fuzzy* é composto de um conjunto de cinco componentes básicos principais, (JANG, 1993):

- Um fuzzificador, que transforma as informações de entrada em conjuntos *fuzzy*;
- Uma base de regras, que contém um determinado número de regras *fuzzy* do tipo "se ... então";
- Uma base de dados, onde estão definidas as funções de pertinência e os universos dos conjuntos *fuzzy* usados;
- Um método de inferência, que aplica um raciocínio *fuzzy* para obter uma saída *fuzzy*;
- Um defuzzificador, que traduz a saída para um valor numérico.

Na Figura 2.4 é ilustrado o esquema de um sistema de inferência *fuzzy*.

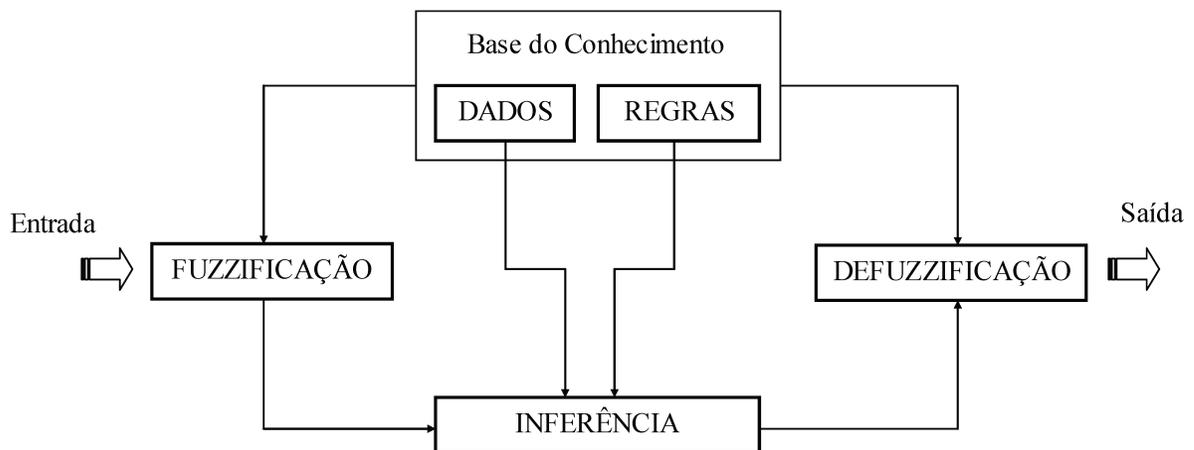


Figura 2.4: Sistema de inferência *fuzzy*, (GOONATILAKE, 1992).

A seguir são apresentados cada um dos componentes do sistema de inferência *fuzzy*.

Fuzzificador

A função do fuzzificador é receber valores numéricos do exterior e os converter em valores *fuzzy* que possam ser processado pelo mecanismo de inferência. Em outras palavras, a entrada de um sistema *fuzzy* é normalmente um valor numérico, por exemplo, a temperatura em graus Celsius ($-7^{\circ}C$) e para que o sistema *fuzzy* possa processar esta entrada é necessário convertê-la em uma "linguagem" que os mecanismos de inferência consigam processar, essa transformação é chamada de fuzzificação.

Regras

A base do pensamento *fuzzy* são as regras que estabelecem relações entre diversas variáveis *fuzzy* e uma ou mais regiões ou conjuntos. Uma regra possui duas partes, o antecedente e consequente:

$$\text{Se } \underbrace{x_1 \text{ é } A_1 \text{ E } \dots \text{ E } x_n \text{ é } A_n}_{\text{Antecedente}} \text{ Então } \underbrace{y \text{ é } B}_{\text{Consequente}}$$

Exemplo:

Se *Temperatura* é -7° Então *Clima* é *Frio*

O conjunto de regras trata de conjuntos *fuzzy* (frio, baixo, rápido, caro, etc.), contudo, tanto as entradas do sistema como suas saídas são variáveis escalares. Por esta razão deve-se inserir um módulo que fuzzifica os valores de entrada (fuzzificação) e outro que transforma as respostas *fuzzy* em respostas escalares na saída (defuzzificação).

Inferência

A inferência é que vai definir para o sistema *fuzzy* como serão processados os antecedentes, quais serão os indicadores de disparo das regras e quais os operadores utilizados sobre os conjuntos *fuzzy* existentes, para executar o processamento de conhecimento.

Tipicamente, utilizam-se modelos de inferência *fuzzy* específicos de acordo com as propriedades sintáticas definidas, ou seja, o modelo de processamento definido para o sistema de conhecimento vai depender basicamente da forma de armazenamento de informações escolhida. Em outras palavras é o modelo de inferência quem vai definir como todo o processamento do sistema se dará.

Defuzzificação

A saída gerada pelo mecanismo de inferência é uma saída *fuzzy*, e que não pode ser interpretada facilmente por um elemento externo (por exemplo, um controlador) que normalmente lidam com informações numéricas. Para fazer com que a saída do sistema *fuzzy* possa ser interpretada por elementos que processam informações numéricas, a saída é convertida. Essa conversão é feita pelo defuzzificador.

Os métodos de defuzzificação mais utilizados são:

- Menor dos máximos (SOM - *Smallest Of Maximum*): Encontra o valor de saída através do ponto em que o grau de pertinência da distribuição da ação de controle atinge o primeiro valor máximo;
- Média dos máximos (MOM - *Mean Of Maximum*): Encontra o ponto médio entre os valores que têm o maior grau de pertinência inferido pelas regras;
- Centro de área (COA - *Center Of Area*): O valor de saída é o centro de gravidade da função.

A seleção do método está relacionada diretamente com as características do processo controlado e o comportamento de controle necessário.

2.4 Considerações finais

Neste capítulo foram descritas algumas das principais técnicas de inteligência artificial, suas aplicabilidades e formas de funcionamento. Para as Máquinas de Vetores de Suporte o capítulo 3 trará uma melhor descrição da técnica que é a base do presente trabalho.

3 MÁQUINAS DE VETORES DE SUPORTE

As Máquinas de Vetores de Suporte (SVM - do inglês *Support Vector Machines*) constituem uma técnica fundamentada pela teoria do aprendizado estatístico ou teoria VC (Vapnik-Chervonenkis), desenvolvida por Vapnik (1995), a partir de estudos iniciados por Vapnik e Chervonenkis (1971). Essa teoria estabelece uma série de princípios que devem ser seguidos na obtenção de classificadores com boa generalização, sendo generalização, definida como a sua capacidade de prever corretamente a classe de novos dados do mesmo domínio em que o aprendizado ocorreu.

A formulação apresentada por Vapnik (1995), é baseada no princípio de minimização do risco estrutural (SRM, do inglês *Structural Risk Minimization*). No ano seguinte Vapnik *et al.* (1996) mostraram que este princípio é superior ao princípio de minimização do risco empírico (ERM, do inglês *Empirical Risk Minimization*), este último utilizado no projeto de redes neurais artificiais, o que atribui a SVM melhor capacidade de generalização, (LIMA, 2004).

O SRM envolve a minimização de um limite superior para o erro de generalização, enquanto que ERM envolve a minimização do erro sobre os dados de treinamento. Assim, modelos de aprendizado de máquina baseados no princípio SRM tendem a apresentar uma maior habilidade de generalização frente a dados não observados, sendo este um dos principais propósitos do aprendizado estatístico.

3.1 Teoria do aprendizado estatístico

Seja $\{\mathbf{x}_i, y_i\}_{i=1}^m$, um conjunto de treinamento com m exemplos, sendo \mathbf{x}_i um conjunto de dados de entrada ou exemplos e y_i denota-se suas classes, *labels* ou rótulos, que são uniformemente distribuídos em relação a uma função de densidade de probabilidade desconhecida $P(\mathbf{x}, y)$. Uma vez que não se conhece a função $P(\mathbf{x}, y)$, que é a probabilidade de cada dado estar contido em uma das classes, o aprendizado estatístico propõe encontrar uma função que classifique corretamente os dados das classes, sem conhecer $P(\mathbf{x}, y)$.

A Teoria de Aprendizado Estatístico (TAE) estabelece condições matemáticas que auxiliam

na escolha de um classificador a partir de um conjunto de dados de treinamento. Essas condições levam em conta o desempenho do classificador no conjunto de treinamento e a sua complexidade, com o objetivo de obter um bom desempenho também para novos dados do mesmo domínio.

Os métodos de aprendizagem estatística devem apresentar as seguintes características:

- Um conjunto flexível e suficientemente grande de funções indicadoras para representar o comportamento do conjunto de dados. As funções indicadoras são responsáveis por tentar simular o comportamento dos sistemas em que os métodos são utilizados.
- Um princípio indutivo capaz de associar o conjunto de treinamento à função que governa o sistema. São exemplos de princípios indutivos: regularização, ERM, SRM, inferência Bayesiana.
- Um algoritmo de aprendizagem, isto é, um procedimento que indica como implementar o princípio indutivo e selecionar a melhor função dentro do universo das funções indicadoras existentes.

3.1.1 Minimização do risco empírico

Na aplicação da TAE, assume-se inicialmente que os dados do domínio em que o aprendizado está ocorrendo são gerados de forma independente e identicamente distribuída de acordo com a distribuição de probabilidade desconhecida $P(\mathbf{x}, y)$, que descreve a relação entre os dados e os seus rótulos.

O erro (também denominado risco) é a probabilidade da saída obtida ser diferente da saída desejada após a escolha de uma função indicadora. O risco funcional ou erro da função, pode ser obtido pela Equação (3.1), sendo f uma função de decisão.

$$R = \int |f(\mathbf{x}) - y| P(\mathbf{x}, y) dx dy \quad (3.1)$$

Como $P(\mathbf{x}, y)$ é desconhecida, não se pode calcular diretamente o risco funcional R , de modo que deve-se recorrer a uma estimativa obtida a partir dados de treinamento, ou seja, o risco

empírico, Equação (3.2).

$$R_{\text{emp}} = \frac{1}{m} \sum_{i=1}^m |f(\mathbf{x}_i) - y_i| \quad (3.2)$$

Mostrado por Vapnik (1995), pode-se fazer essa consideração se e somente se a convergência em probabilidade de R_{emp} para R , puder ser substituída pela convergência uniforme em probabilidade. Mostrou-se também, que esta condição é suficiente para a consistência do princípio de minimização do risco empírico e também serve como limitação da dimensão VC , sendo a dimensão VC um índice escalar que mede a complexidade intrínseca das funções de decisão que serão utilizadas nas máquinas de vetores de suporte.

A dimensão VC , representada por h , mede a capacidade do conjunto de funções F . Quanto maior o seu valor, mais complexas são as funções de classificação que podem ser induzidas. Dado um problema de classificação binário, essa dimensão é definida como o número máximo de exemplos que podem ser particionados em duas classes pelas funções contidas em F , para todas as possíveis combinações binárias desses dados.

Conclui-se que a minimização do risco, que será associado ao desempenho de generalização do modelo resultante, requer que tanto o risco empírico como a razão entre a dimensão VC e o número de amostras de treinamento tem que ser pequena. Uma vez que o risco empírico é usualmente uma função decrescente de h (VAPNIK, 1995), para um dado número de amostras de treinamento, há um valor ótimo da dimensão VC . A escolha de um valor apropriado para h , que em muitos casos é controlado através do número de hiperparâmetros do modelo, é crucial para se obter bom desempenho, especialmente quando o número de amostras de treinamento é pequeno.

3.1.2 Minimização do risco estrutural

A minimização do risco empírico nem sempre é suficiente para a obtenção de resultados adequados, pois, a ERM não leva em consideração a complexidade das funções indicadoras. Quando a complexidade das funções é superior à necessidade do problema, ocorre o sobre-ajuste da função ao conjunto de treinamento. Quando ela é inferior, ocorre o sub-ajuste. Em ambos os casos, a

capacidade de generalização é reduzida.

Com o uso do conceito de dimensão VC , foi desenvolvida uma expressão, com probabilidade $(1 - \delta)$ de ocorrer, que indica que o limite superior do risco funcional será dado pela Equação (3.3), sendo R_{contorno} o erro do contorno, isto é, o erro da função em se ajustar adequadamente ao modelo, δ é um número entre 0 e 1, h é a dimensão VC e m é o número de exemplos de treinamento.

$$R \leq R_{\text{emp}} + R_{\text{contorno}}(\delta, h, m) \quad (3.3)$$

A minimização do risco estrutural é um princípio indutivo para a seleção de um modelo que será utilizado para o aprendizado a partir de conjuntos de dados finito de treinamento. A SRM descreve um modelo geral de capacidade de controle e fornece uma solução para a dificuldade de escolha entre a complexidade do espaço de hipóteses (a dimensão VC) e a qualidade de ajuste do modelo aos dados de treinamento (erro empírico).

A SRM tem como objetivo minimizar $R_{\text{contorno}}(\delta, h, m)$ da Equação (3.3). Este princípio usa a dimensão VC para controlar a complexidade das funções indicadoras, de forma a adequá-las a cada problema.

3.2 Fundamentação teórica das SVMs

O problema de classificação pode ser tratado como um problema binário, sem perda de generalidade, (GUNN, 1998). O objetivo é separar duas classes por meio de uma função que é induzida a partir das amostras de treinamento, produzindo-se um classificador que apresente um bom desempenho junto a amostras não-observadas durante o treinamento, ou seja, com boa capacidade de generalização.

A derivação do algoritmo SVM pode ser compreendida da classe mais simples das funções de decisões: funções lineares. Para ilustrar as funções de decisão, a Figura 3.1 apresenta várias funções de decisão lineares que corretamente classificam um conjunto de treinamento 2D (bidimensional).

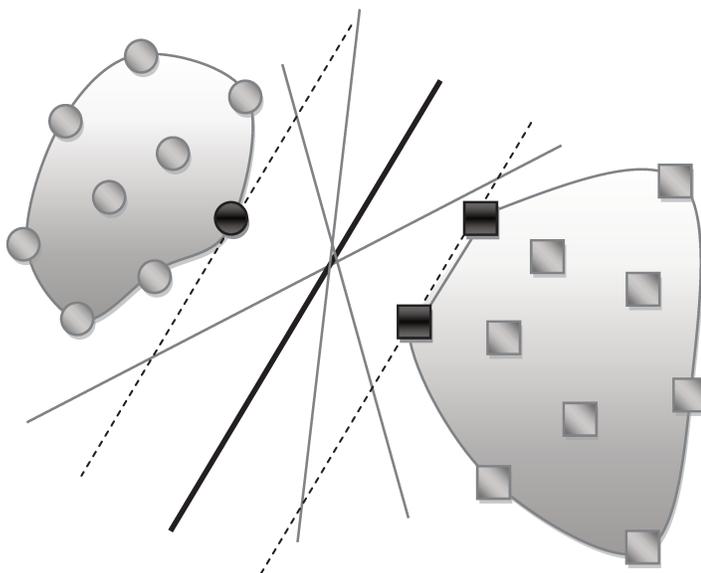


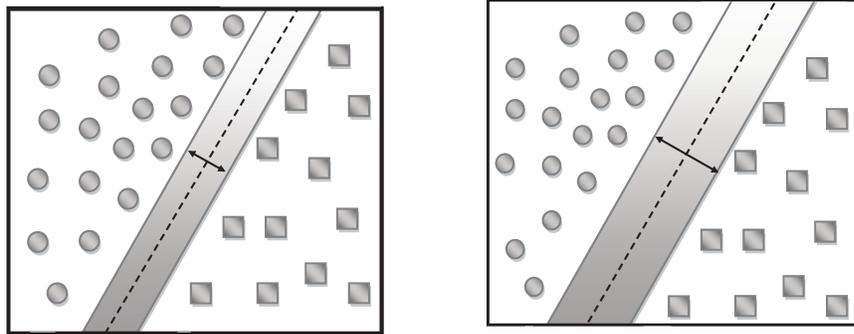
Figura 3.1: Hiperplano de separação ótimo (linha grossa), com os seus hiperplanos de suporte (tracejados).

No entanto, cada função de decisão (linhas contínuas) determinam uma área completamente diferente para cada classe. Apesar de cada função de decisão classificar corretamente o conjunto de treinamento, as previsões de novos exemplos podem diferir bastante devido às regiões de classificação determinadas por cada função de decisão. Mas há somente uma que maximiza a margem (maximiza a distância entre o classificador e a amostra mais próxima de cada classe). Este classificador linear é chamado hiperplano de separação ótimo, pois espera-se que este hiperplano generalize melhor que os demais, quando amostras não utilizadas durante a fase de treinamento forem classificadas.

Pode-se observar que o hiperplano ótimo foi construído tendo como suporte dois outros hiperplanos (linhas tracejadas), os quais por sua vez passam por alguns pontos para ambas as classes. Estes pontos são chamados de **vetores de suporte (SV)**.

3.2.1 Hiperplano de separação ótimo

Segundo Scholkopf e Smola (2002), é preferível ter limites de decisão que não somente separem corretamente duas classes, mas também estejam tão distantes quanto possível dos exemplos de treinamento. Desta forma o hiperplano de separação ótimo deve possuir margem máxima. A Figura 3.2 apresenta duas funções de decisão com diferentes margens de classificação.



(a) hiperplano com uma pequena margem de separação entre as classes.

(b) hiperplano com uma margem maior de separação entre as classes.

Figura 3.2: Classificação através de hiperplanos.

Margens rígidas

Dado um conjunto de treinamento T com m dados $\mathbf{x}_i \in X$ e seus respectivos rótulos $y_i \in Y$, sendo que X é o espaço dos dados e $Y = \{-1, +1\}$. O conjunto T será linearmente separável se for possível separar os dados das classes -1 e $+1$ por meio de um hiperplano, (SCHOLKOPF E SMOLA, 2002).

Na Equação (3.4), tem-se a equação de um hiperplano, sendo $\langle \mathbf{w}, \mathbf{x} \rangle$ o produto interno (escalar) entre os vetores \mathbf{w} e \mathbf{x} . $\mathbf{w} \in H$ é um vetor de pesos m -dimensional e $b \in \mathbb{R}$ é o termo *bias* ou a compensação (*offset*) do hiperplano em relação à origem no espaço de entrada, *bias* é um escalar e

H é um espaço do produto interno.

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \sum_{i=1}^m \mathbf{w}_i \mathbf{x}_i + b \quad (3.4)$$

Essa equação faz a divisão do espaço dos dados X em duas regiões: $\langle \mathbf{w}, \mathbf{x} \rangle + b > 0$ e $\langle \mathbf{w}, \mathbf{x} \rangle + b < 0$. Uma função sinal $g(\mathbf{x}) = \text{sgn}(f(\mathbf{x}))$ pode então ser empregada na obtenção das classificações, conforme mostrado na Equação (3.5), (SCHOLKOPF E SMOLA, 2002).

$$g(\mathbf{x}) = \text{sgn}(f(\mathbf{x})) = \begin{cases} +1 & \text{se } \langle \mathbf{w}, \mathbf{x} \rangle + b > 0 \\ -1 & \text{se } \langle \mathbf{w}, \mathbf{x} \rangle + b < 0 \end{cases} \quad (3.5)$$

Seja \mathbf{x}_1 e \mathbf{x}_2 dois pontos dentro de um hiperplano canônico em ambos os lados, Figura 3.3. Se $\langle \mathbf{w}, \mathbf{x}_1 \rangle + b = 1$ e $\langle \mathbf{w}, \mathbf{x}_2 \rangle + b = -1$, pode-se deduzir que $\langle \mathbf{w}, (\mathbf{x}_1 - \mathbf{x}_2) \rangle = 2$.

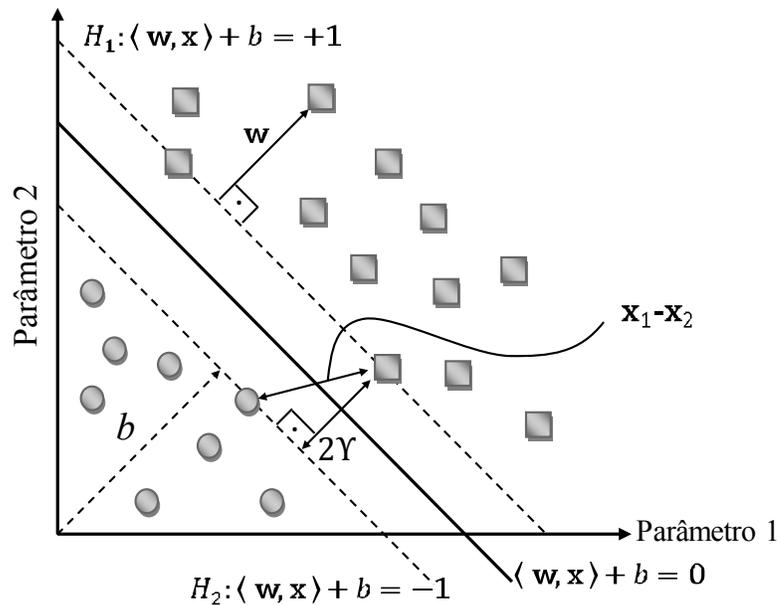


Figura 3.3: Cálculo da distância Υ entre os hiperplanos.

Para o hiperplano de separação $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$, o vetor normal é $\mathbf{w}/\|\mathbf{w}\|$, sendo $\|\mathbf{w}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. Assim, a distância entre os dois hiperplanos canônicos é igual à projeção de $\mathbf{x}_1 - \mathbf{x}_2$ para

o vetor normal $\mathbf{w}/\|\mathbf{w}\|$, que fornece $\langle(\mathbf{x}_1 - \mathbf{x}_2), \mathbf{w}\rangle/\|\mathbf{w}\| = 2/\|\mathbf{w}\|$. Como é a metade da distância entre os dois hiperplanos canônicos, a margem é dada pela Equação (3.6), (CAMPBELL E YING, 2011).

$$\Upsilon = \frac{1}{\|\mathbf{w}\|} \quad (3.6)$$

Assim, a fim de encontrar o hiperplano de separação ótimo com margem máxima, deve-se minimizar $\|\mathbf{w}\|$ sujeitas a restrições de desigualdade. Dessa forma, recorre-se ao seguinte problema de otimização, (SCHOLKOPF E SMOLA, 2002).

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (3.7)$$

$$\text{sujeito a : } y_i(\langle\mathbf{w}, \mathbf{x}_i\rangle + b) - 1 \geq 0, \forall_i = 1, \dots, m \quad (3.8)$$

Este tipo de problema é chamado de problema de otimização primal. As restrições são impostas de maneira a assegurar que não haja dados de treinamento entre as margens de separação das classes.

O problema de otimização obtido é quadrático, cuja solução possui uma ampla e estabelecida teoria matemática. Como a função objetivo que está sendo minimizada é convexa e os pontos que satisfazem as restrições formam um conjunto convexo, esse problema possui um único mínimo global. Problemas desse tipo podem ser solucionados com a introdução de uma função Lagrangiana, que engloba as restrições à função objetivo, associadas a parâmetros denominados multiplicadores de Lagrange α_i , Equação (3.7), (SCHOLKOPF E SMOLA, 2002).

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (y_i(\langle\mathbf{w}, \mathbf{x}_i\rangle + b) - 1), \quad (3.9)$$

com os multiplicadores de Lagrange $\alpha_i \geq 0$. Usar-se-á letras gregas em negrito referindo-se a vetores de variáveis, tipo, $\boldsymbol{\alpha}(\alpha_1, \dots, \alpha_m)$.

O Lagrangiano \mathcal{L} deve ser maximizado em relação a α_i , minimizado em relação a \mathbf{w} e b . Consequentemente, neste ponto de sela, a derivada de \mathcal{L} com relação às variáveis primárias devem

desaparecer.

$$\frac{\partial}{\partial b} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = 0, \quad \frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = 0, \quad (3.10)$$

o que leva a

$$\sum_{i=1}^m \alpha_i y_i = 0, \quad (3.11)$$

e

$$\mathbf{w} = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i. \quad (3.12)$$

O vetor solução, portanto, tem uma expansão em termos de exemplos de treinamento. Observa-se que, embora a solução \mathbf{w} é única (devido à convexidade estrita da Equação (3.7), e da convexidade da Equação (3.8)), os coeficientes α_i não precisam ser únicos, (SCHOLKOPF E SMOLA, 2002).

De acordo com o teorema KKT, apenas os multiplicadores de Lagrange α_i que não são zero no ponto de sela, correspondem às restrições da Equação (3.8) que são precisamente conhecidas, (SCHOLKOPF E SMOLA, 2002). Assim, para todo $i = 1, \dots, m$ tem-se:

$$\alpha_i [y_i (\langle \mathbf{x}_i, \mathbf{w} \rangle + b) - 1] = 0. \quad (3.13)$$

Observa-se na Equação (3.13) que α_i pode ser diferente de 0 somente para os dados que se encontram sobre os hiperplanos H_1 e H_2 . Formalmente **vetores de suporte (SV)** são os padrões \mathbf{x}_i para o qual $\alpha_i > 0$. Esta terminologia está relacionada com termos correspondentes na teoria de conjunto convexo, relevantes para a otimização convexa. De acordo com a Equação (3.13), os vetores de suporte estão exatamente sobre a margem. Todos os exemplos de treinamento restantes são irrelevantes.

Voltando ao problema de otimização, pode-se chegar ao fim substituindo as Equações (3.11)

e (3.12) na Equação Lagrangiana (3.9), chega-se a forma dual do problema de otimização:

$$\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \quad (3.14)$$

$$\text{sujeito a } \alpha_i \geq 0, i = 1, \dots, m, \quad (3.15)$$

$$\text{e } \sum_{i=1}^m \alpha_i y_i = 0 \quad (3.16)$$

O valor de b é calculado a partir dos SVs e das condições da Equação (3.13), (SCHOLKOPF E SMOLA, 2002), denota-se n_{SV} o número de SVs e SV representa o conjunto dos SVs.

$$b = \frac{1}{n_{SV}} \sum_{\mathbf{x}_j \in SV} \frac{1}{y_j} - \langle \mathbf{w}, \mathbf{x}_j \rangle = \frac{1}{n_{SV}} \sum_{\mathbf{x}_j \in SV} \left(\frac{1}{y_j} - \sum_{\mathbf{x}_i \in SV} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x}_j \rangle \right) \quad (3.17)$$

Por fim da substituição da expansão, Equação (3.12), na função de decisão, Equação (3.5), obtém-se uma expressão que pode ser calculada em termos dos produtos internos, tomada entre o padrão a ser classificado e os vetores de suporte.

$$g(\mathbf{x}) = \text{sgn}(f(\mathbf{x})) = \text{sgn} \left(\sum_{i=1}^m \alpha_i y_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b \right) \quad (3.18)$$

Esta função linear representa o hiperplano que separa os dados com maior margem, considerado aquele com melhor capacidade de generalização.

Margens flexíveis

Em situações reais, devido a ruídos e/ou condições do problema, é difícil encontrar aplicações em que os dados sejam linearmente separáveis. Assim as margens rígidas são estendidas para lidar com conjuntos de treinamento mais gerais.

Cortes e Vapnik (1995) escolheram uma diferente aproximação para as SVM, permitindo que as restrições da Equação (3.8) sejam violadas, isto é possível introduzindo, as, então chamadas,

variáveis de folga ξ_i .

$$\xi_i \geq 0, \text{ com } i = 1, \dots, m \quad (3.19)$$

Essas variáveis relaxam as restrições impostas ao problema de otimização primal, que se tornam, (SCHOLKOPF E SMOLA, 2002):

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \forall i = 1, \dots, m \quad (3.20)$$

Este tipo de procedimento deixa as margens do classificador mais suaves, permitindo que alguns pontos fiquem entre os hiperplanos H_1 e H_2 e também a ocorrência de alguns erros de classificação. Por esse motivo, as SVMs obtidas neste caso também podem ser referenciadas como SVMs com margens flexíveis ou suaves.

Claramente, fazendo ξ_i grande o suficiente, a restrição em (\mathbf{x}_i, y_i) pode sempre ser atendida. A fim de não obter a solução trivial onde todos ξ_i assumem valores grandes, há a necessidade de penalizá-los na função objetivo. Para este fim, um termo $C \sum_i \xi_i$ é incluído na Equação (3.7), (SCHOLKOPF E SMOLA, 2002).

$$\min_{\mathbf{w}, \xi} \left(\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i \right), \quad (3.21)$$

sujeito as restrições das Equações (3.19) e (3.20). A constante C , chamada de custo ou em inglês *cost*, é um termo de regularização que impõe um peso à minimização dos erros no conjunto de treinamento em relação à minimização da complexidade do modelo.

A solução da Equação (3.21) envolve passos matemáticos semelhantes aos apresentados anteriormente, com a introdução de uma função Lagrangiana e tornando suas derivadas parciais nulas. Tem-se como resultado o seguinte problema dual:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \quad (3.22)$$

$$\text{sujeito a } 0 \leq \alpha_i \leq C, \forall i = 1, \dots, m, \quad (3.23)$$

$$e \sum_{i=1}^m \alpha_i y_i = 0 \quad (3.24)$$

Pode-se observar que essa formulação é igual à apresentada para as SVMs de margens rígidas, a não ser pela restrição nos α_i , que agora são limitados pelo valor de C .

Seja α a solução do problema dual, enquanto \mathbf{w} , b e ξ denotam as soluções da forma primal. O vetor \mathbf{w} continua sendo determinado pela Equação (3.12). As variáveis ξ_i podem ser calculadas pela Equação (3.25), (CRISTIANINI E SHAWE-TAYLOR, 2000).

$$\xi_i = \max \left\{ 0.1 - y_i \sum_{j=1}^m y_j \alpha_j \langle \mathbf{x}_j, \mathbf{x}_i \rangle + b \right\} \quad (3.25)$$

A variável b provém novamente de α e de condições de KKT, que neste caso são:

$$\alpha_i (y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 + \xi_i) = 0 \quad (3.26)$$

$$(C - \alpha_i) \xi_i = 0 \quad (3.27)$$

Como nas SVMs de margens rígidas, os pontos \mathbf{x}_i para os quais $\alpha_i > 0$ são denominados vetores de suporte, sendo os dados que participam da formação do hiperplano separador. Porém, neste caso, pode-se distinguir tipos distintos de SVs (LORENA, 2006).

Se $\alpha_i < C$, pela Equação (3.27), e $\xi_i = 0$ então, da Equação (3.26), estes SVs encontram-se sobre as margens e também são denominados livres. Os SVs para os quais $\alpha_i = C$ podem representar três casos, (LORENA, 2006):

- Erros, se $\xi_i > 1$;
- Pontos corretamente classificados, porém entre as margens, se $0 < \xi_i \leq 1$, ou pontos sobre as margens, se $\xi_i = 0$;
- O último caso ocorre raramente e os SVs anteriores são denominados limitados. Na Fi-

gura 3.4 são ilustrados os possíveis tipos de SVs. Pontos na cor vermelha representam SVs livres. SVs limitados são ilustrados em preto e cinza. Pontos pretos correspondem a SVs limitados que são erros de treinamento. Todos os outros dados, verdes e azuis, são corretamente classificados e encontram-se fora das margens, possuindo $\xi_i = 0$ e $\alpha_i = 0$.

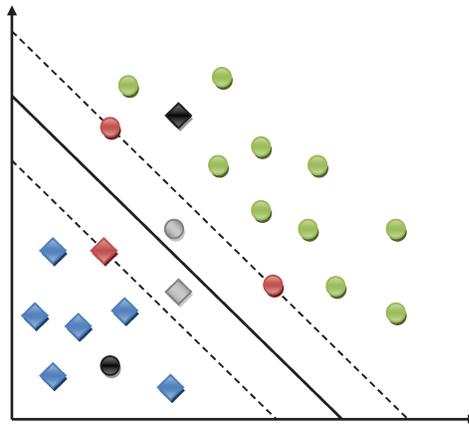


Figura 3.4: Tipos de SVs: livres (vermelho) e limitados (preto e cinza).

Para calcular b , computa-se a média da Equação (3.17) sobre todos SVs \mathbf{x}_j entre as margens, ou seja, com $\alpha_i < C$, (SCHOLKOPF E SMOLA, 2002).

Tem-se como resultado final a mesma função de classificação representada na Equação (3.18), porém neste caso as variáveis α_i são determinadas pela solução da Expressão (3.22).

3.2.2 SVMs não lineares

Até agora, foi mostrado que um hiperplano de margem larga é bom do ponto de vista estatístico, e como calculá-lo. Embora essa abordagem já resolva alguns problemas, ainda há uma grande desvantagem em trabalhar apenas com SVM lineares, pois, há muitos casos em que não é possível dividir satisfatoriamente os dados de treinamento por um hiperplano.

Para permitir superfícies de decisão muito mais gerais, as SVMs usam funções kernel, que

têm a seguinte forma:

$$k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle, \quad (3.28)$$

permitindo computar o valor do produto interno em H sem ter que, explicitamente, computar o mapa Φ .

As funções kernel tem a função de transformar não linearmente os dados no espaço de entrada $x_1, \dots, x_m \in X$ em um espaço de maior dimensão chamado de espaço de características (*feature space*), usando o mapeamento $\Phi : x_i \mapsto \Phi(x_i)$, onde é possível fazer uma separação linear, Figura 3.5.

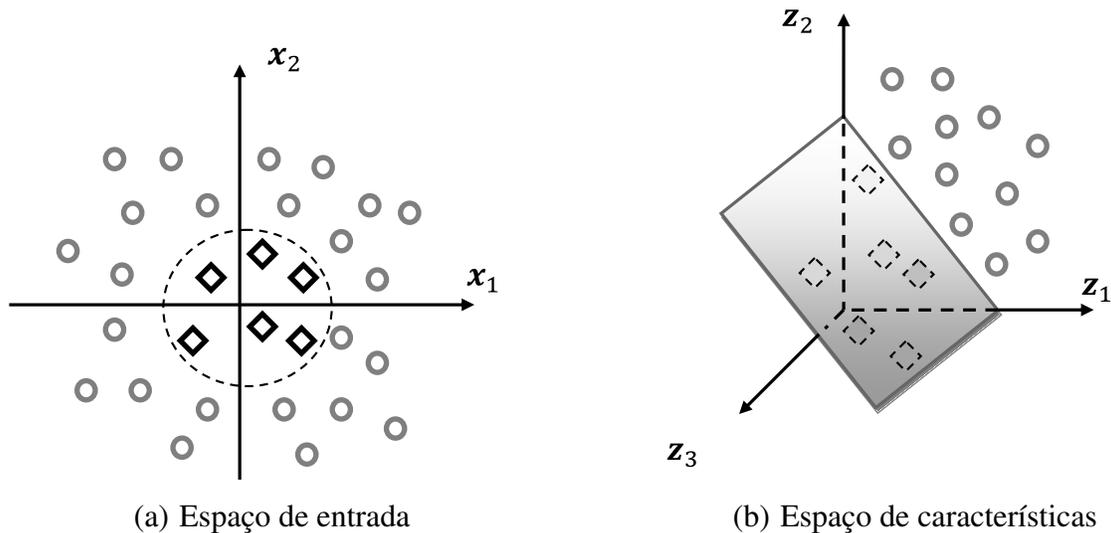


Figura 3.5: Mudança de espaço.

Cover (1965) propôs um teorema que motiva o procedimento descrito anteriormente. O teorema de Cover caracteriza o número de possíveis separações lineares de m pontos em posições diversas em um espaço N -dimensional.

Quanto mais se aumentar N , mais termos existirão na soma, e, portanto, maior é o número resultante. Este teorema formaliza a intuição de que o número de separações aumenta com a di-

mensionalidade.

Para ilustrar estes conceitos, considere dois vetores de duas dimensões: $\mathbf{x}_i = (x_{i1}, x_{i2})$ e $\mathbf{x}_j = (x_{j1}, x_{j2})$ e um kernel polinomial de grau 2. Assim tem-se:

$$\begin{aligned}
 & \langle \mathbf{x}_i, \mathbf{x}_j \rangle^2 \\
 & \langle (x_{i1}, x_{j1}) + (x_{i2}, x_{j2}) \rangle^2 \\
 & x_{i1}^2 x_{j1}^2 + 2x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 \\
 & x_{i1}^2 x_{j1}^2 + \sqrt{2} x_{i1} x_{i2} \cdot \sqrt{2} x_{j1} x_{j2} + x_{i2}^2 x_{j2}^2 \\
 & \langle (x_{i1}^2, x_{i2}^2, \sqrt{2} x_{i1} x_{i2}), (x_{j1}^2, x_{j2}^2, \sqrt{2} x_{j1} x_{j2}) \rangle
 \end{aligned} \tag{3.29}$$

Os cálculos da Equação (3.29) transformam os dados de \mathbb{R}^2 para \mathbb{R}^3 , assim o conjunto de dados não linear em \mathbb{R}^2 torna-se linearmente separável em \mathbb{R}^3 . É possível então encontrar um hiperplano capaz de separar esses dados. Pode-se verificar que a função ($f(x)$) da Figura 3.6, embora linear em \mathbb{R}^3 , corresponde a uma fronteira não linear em \mathbb{R}^2 . Na Figura 3.6, tem-se a representação do mapeamento de \mathbb{R}^2 para \mathbb{R}^3 , tomando com exemplo o vetor \mathbf{x}_i , o mesmo acontece para \mathbf{x}_j .

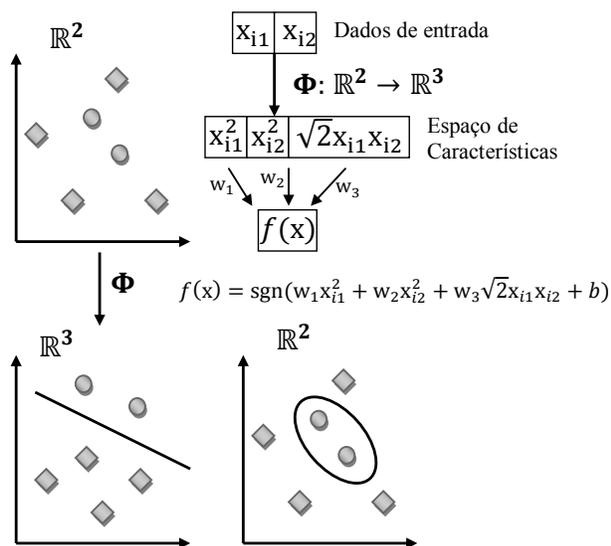


Figura 3.6: Mapeamento não linear, (SCHOLKOPF E SMOLA, 2002).

Então o processo consiste em mapear os dados para um espaço de maior dimensão utilizando o mapa Φ e aplica-se a SVM linear sobre este espaço. Essa encontra o hiperplano com maior margem de separação, garantindo assim uma boa capacidade de generalização. Para realizar o mapeamento, basta aplicar Φ aos exemplos presentes no problema de otimização da Equação(3.22):

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j k(x_i, x_j), \quad (3.30)$$

sujeito as restrições das Equações (3.23) e (3.24).

De forma semelhante, o classificador extraído se torna:

$$g(\mathbf{x}) = \text{sgn}(f(\mathbf{x})) = \text{sgn} \left(\sum_{i=1}^m \alpha_i y_i k(x_i, x_j) + b \right) \quad (3.31)$$

Em que b é adaptado da Equação (3.17) também aplicando o mapeamento aos dados:

$$b = \frac{1}{n_{SV}} \sum_{\mathbf{x}_j \in SV} \left(\frac{1}{y_j} - \sum_{\mathbf{x}_i \in SV} \alpha_i y_i k(x_i, x_j) \right) \quad (3.32)$$

É comum empregar a função kernel sem conhecer o mapeamento Φ , que é gerado implicitamente. A utilidade dos kernels está, portanto, na simplicidade de seu cálculo e em sua capacidade de representar espaços abstratos, (LORENA, 2006).

Para garantir a convexidade do problema de otimização formulado na Equação (3.30) e também que o kernel represente mapeamentos nos quais seja possível o cálculo de produtos internos, utiliza-se funções kernel que seguem as condições estabelecidas pelo teorema de Mercer, (POYHONEN *et al.*, 2002; GUYON *et al.*, 2006; WANG *et al.*, 2008; QU E ZUO, 2010). De forma simplificada, um kernel que satisfaz as condições de Mercer é caracterizado por dar origem a matrizes positivas semi-definidas k , em que cada elemento k_{ij} é definido por $k_{ij} = k(x_i, x_j)$, para todo $i, j = 1, \dots, n$, (HERBRICH, 2001).

Por fim não há maneiras de entender intuitivamente por que o mapeamento do kernel aumenta as chances de uma separação, em termos de conceitos da teoria de aprendizagem estatís-

tica, (SCHOLKOPF E SMOLA, 2002). Mas pode-se dizer que usar um kernel normalmente significa usar uma função de classe maior, aumentando assim a capacidade de aprendizado da máquina e a capacidade em separar linearmente problemas que no início não eram possíveis.

Alguns dos kernels mais utilizados na prática estão listados na Tabela 3.1. Cada um deles apresenta hiperparâmetros que devem ser determinados pelo usuário, que também estão indicados na tabela. Lembrando que quando se utiliza a SVM de margens flexíveis, deve-se considerar também o parâmetro C .

Tabela 3.1: Funções kernel mais comuns.

Tipo de kernel	Função $k(x_i, x_j)$	Hiperparâmetros
Linear	$\langle x_i, x_j \rangle$	
Polinomial	$(\kappa \langle x_i, x_j \rangle + \Theta)^d$	κ, Θ, d
Gaussiano (RBF)	$\exp(-\gamma \ x_i - x_j\ ^2)$	γ
Sigmoidal	$\tan(\kappa \langle x_i, x_j \rangle + \Theta)$	κ, Θ

Na Figura 3.7, tem-se o exemplo de um mapeamento bidimensional e Na Figura 3.8, tem-se o exemplo de um mapeamento tridimensional, utilizando o kernel RBF, sendo Parâmetro 1, Parâmetro 2 e Parâmetro 3, quaisquer duas ou três entradas que o usuário queira classificar.

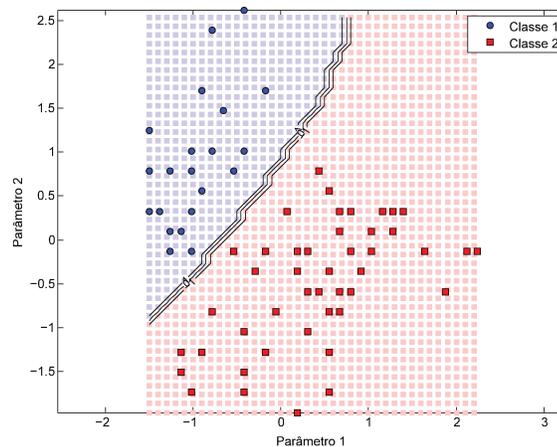


Figura 3.7: Mapeamento no espaço \mathbb{R}^2 .

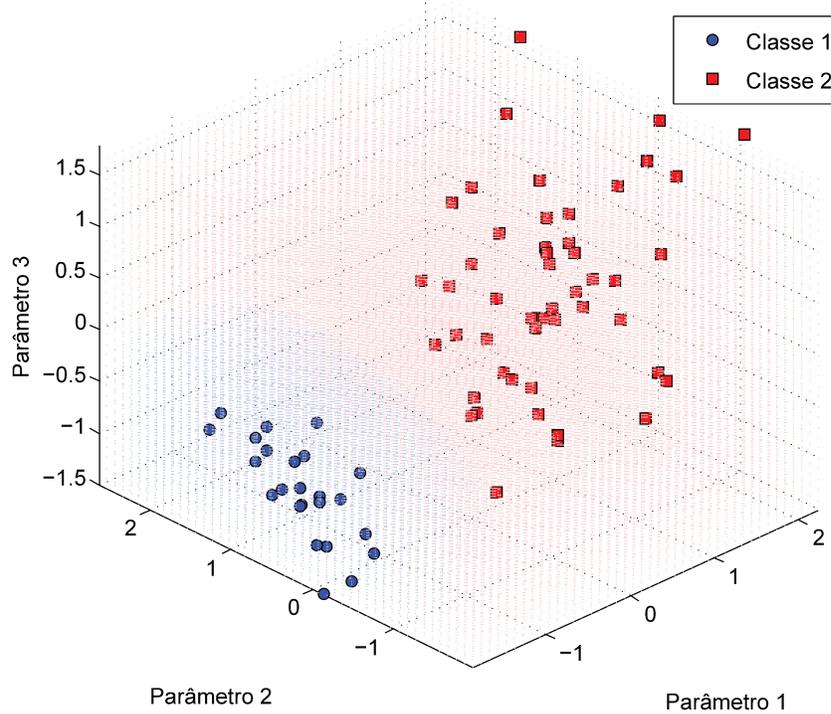


Figura 3.8: Mapeamento no espaço \mathbb{R}^3 .

As divisões com os kernel são sempre divisões lineares em um espaço de maior dimensão (espaço de características), o que resulta em uma divisão não linear no espaço original (espaço de entrada). Na Figura 3.9, tem-se a representação de uma classificação não linear utilizando um kernel.

Figura 3.9(a): Padrões a serem classificados. Figura 3.9(b): Padrões no espaço de entrada. Figura 3.9(c): Início do mapeamento pelo kernel. Figura 3.9(d): Padrões no espaço de características. Figura 3.9(f): Classificação no espaço de maior dimensão. Figura 3.9(g): Retorno ao espaço de entrada. Figura 3.9(h): Padrões classificados.

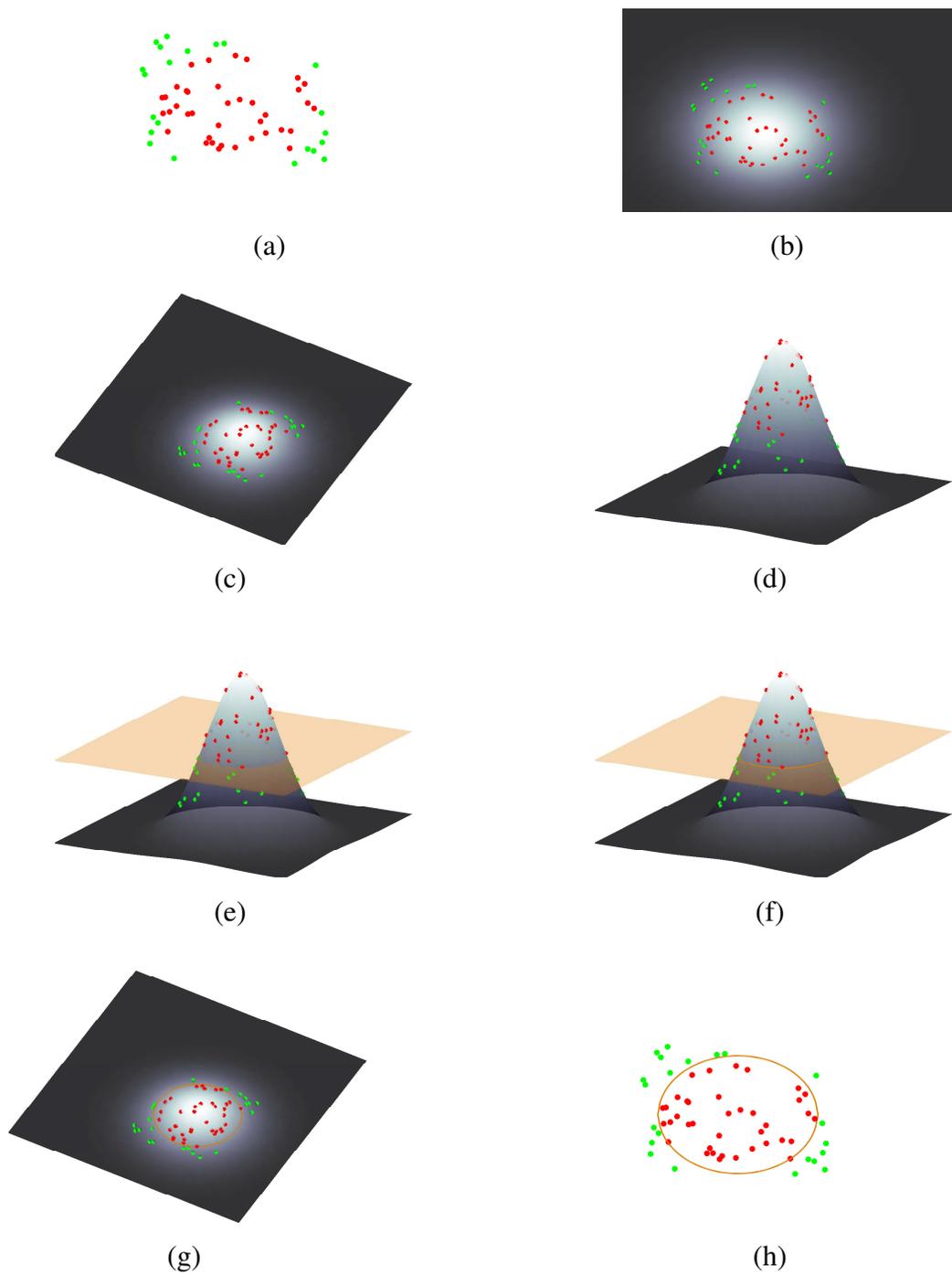


Figura 3.9: Classificação não-linear via separação linear em espaço de maior dimensão através de um kernel.

3.2.3 SVM para multiclass

Diversas técnicas de aprendizado de máquinas são originalmente formuladas para a solução de problemas de classificação binários, porém, muitos destes problemas apresentam mais que duas classes.

A forma mais direta de geração de classificadores multiclass a partir de técnicas de classificação binária, consiste em decompor o problema multiclass em subproblemas binários. As saídas dos classificadores binários gerados na solução de cada um desses subproblemas são então combinadas na obtenção do classificador final.

Formalmente, em um sistema multiclass o conjunto de treinamento é composto por pares (\mathbf{x}_i, y_i) , tal que $y_i \in \{1, \dots, K\}$, com $K > 2$, sendo K o número de classes. Dentre as principais abordagens que utilizam como base essa decomposição, podem-se destacar os métodos: "Um-Contra-Todos", "Um-Contra-Um" e "DAGSVM" (do inglês *Directed Acyclic Graph Support Vector Machines*) que quer dizer SVM com Grafo Direcionado Acíclico.

Um-Contra-Todos (UCT)

A primeira implementação usando SVM para classificação com multiclass foi provavelmente o método Um-Contra-Todos (do inglês, *One-Against-All* ou *One-Against-the-Rest*), (BOTTOU *et al.*, 1994).

O UCT é um método simples e efetivo para problemas com multiclass classes. Este constrói K modelos SVM, sendo K o número de classes. O K -ésimo SVM é treinado com todos os exemplos da K -ésima classe com rótulos positivos, e todos os outros exemplos com rótulos negativos.

Para se obter um classificador para K classes, é comum construir um conjunto de classificadores binários f^1, \dots, f^K , cada um treinado para separar uma classe do resto, e combiná-las, fazendo a classificação multiclass de acordo com a saída de maior número de votos.

Os valores $f^K(x)$ também podem ser usados como rejeitores de decisões. Para um melhor entendimento, considere a diferença entre os dois maiores valores de $f^K(x)$ como uma medida de confiança na classificação de x . Se essa medida está aquém de um limiar θ , o classificador rejeita o padrão e não o atribui a nenhuma classe. Isto tem como consequência que os padrões restantes podem alcançar uma menor taxa de erro.

Um-Contra-Um (UCU)

O método Um-contra-Um, (do inglês, *One-Against-One*) ou também chamado na literatura de método *pairwise*, foi introduzido por Knerr *et al.* (1990). Na classificação UCU, treina-se um classificador para cada par de classes possíveis, (FRIEDMAN, 1996; SCHMIDT E GISH, 1996; HASTIE E TIBSHIRANI, 1998).

Este método constrói $K(K - 1)/2$ classificadores, onde cada classificador é treinado sobre os dados de duas classes. Para dados de treinamento envolvendo as i -ésimas e j -ésimas classes. Há diferentes procedimentos que permitem combinar estes classificadores visando atribuir uma única classe a um novo exemplo \mathbf{x} não pertencente ao conjunto de treinamento, depois que todos os $K(K - 1)/2$ classificadores são construídos.

Hsu e Lin (2002) sugeriram usar a seguinte estratégia de votação: se $\text{sgn}(\langle \mathbf{w}^{ij}, \Phi(x_t) \rangle + b^{ij})$ diz que \mathbf{x} está na i -ésima classe, então a i -ésima classe recebe um voto. Caso contrário, a j -ésima classe recebe um voto. Então, prediz-se que \mathbf{x} está na classe com maior número de votos. A abordagem de votação descrita é também chamada de estratégia Max Wins.

No método UCU o número de classificadores é geralmente maior do que o número de classificadores do método UCT, por exemplo, se $K = 10$, precisa-se treinar 45 classificadores binários ao invés de 10 como no método UCT. Embora isto sugere um maior tempo de treinamento, os problemas individuais que a SVM treina são significativamente menores.

Considerações semelhantes se aplicam à velocidade de execução. Quando for classificar um padrão de teste, todos os 45 classificadores binários serão avaliados, e o padrão será classificado de acordo com a classe que recebeu o maior número de votos. A classe recebe um voto uma vez

que o classificador coloca um padrão dentro dela. Os classificadores individuais, no entanto, são geralmente menores em tamanho (eles têm menos SVs) do que teriam na abordagem UCT. Isto por duas razões: Primeiro, os conjuntos de treinamento são menores e, segundo, os problemas a serem aprendidos são geralmente mais fáceis, já que as classes têm menos sobreposições.

No caso de duas classes receberem o mesmo número de votos, uma estratégia de desempate deve ser adotada. Hsu e Lin (2002) simplesmente selecionam aquela com menor índice. No entanto, é claro que tal estratégia não é a mais adequada, uma vez que não existe nenhuma justificativa teórica, apenas uma abordagem empírica e depende da implementação das classes. Na literatura, existem algumas estratégias para tratar deste problema.

DAGSVM

O DAGSVM ou simplesmente DAG, foi proposto por Platt *et al.* (2000). A fase de treinamento é a mesma do método UCU, (HSU E LIN, 2002). O método requer também a solução de $K(K - 1)/2$ problemas de classificação binária. Entretanto, na fase de teste, é empregado um grafo direcionado acíclico com raiz binária que tem $K(K - 1)/2$ nós internos e K folhas (Figura 3.10). Cada nó é um SVM binário envolvendo as i -ésima e j -ésima classes. Dado um exemplo de teste \mathbf{x} , partindo do nó raiz, a função decisão binária é avaliada. Então, move-se para a esquerda ou para a direita, dependendo do valor da saída. Desta forma, caminha-se através de uma das alternativas, até encontrar um nó terminal que indica a classe.

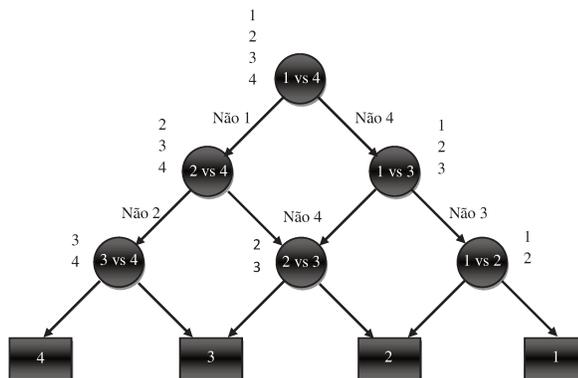


Figura 3.10: Arquitetura do método DAGSVM.

Uma vantagem do DAG é que algumas análises de generalização podem ser estabelecidas, (PLATT *et al.*, 2000).

3.3 Considerações finais

Neste capítulo foram descritos os conceitos básicos a respeito das SVMs. Embasada no princípio do aprendizado estatístico, essa técnica de aprendizado de máquinas caracteriza-se, principalmente, pela boa capacidade de generalização. Apresentam também bom desempenho diante de dados de grande dimensão, sobre os quais outras técnicas provavelmente levariam a um sub ou sobre ajuste do modelo.

Um outro ponto que torna a técnica interessante, é devido à convexidade da formulação do problema de otimização, remetendo a um único mínimo global, o que é mais uma vantagem das SVMs sobre outras técnicas de inteligência artificial. Por fim, tem-se que o uso das funções kernel torna o algoritmo mais eficiente e computacionalmente estável, principalmente quando se trata de classificação de dados não lineares.

Entre as principais limitações das SVMs, pode-se destacar a sua sensibilidade a escolha de parâmetros e a dificuldade de interpretação do modelo gerado por essa técnica.

4 DETECÇÃO DE FALHAS ATRAVÉS DA ANÁLISE DE VIBRAÇÕES

4.1 Análise de vibrações

Chama-se de vibração ou oscilação qualquer movimento que se repita após um intervalo de tempo. O balançar de um pêndulo e o movimento de uma corda dedilhada são exemplos típicos de vibração.

De forma análoga ao que sucede com a medicina, em que a partir de alguns indicadores (parâmetros) do estado de saúde do indivíduo se traça o quadro clínico que o caracteriza e se faz o respectivo diagnóstico, também nos equipamentos mecânicos é possível ter idéia da sua condição de funcionamento e conservação a partir do conhecimento da evolução dos seus parâmetros significativos, e a partir desta evolução consegue-se diagnosticar seu estado e fazer um prognóstico do desempenho do equipamento no futuro.

Nas máquinas, uma alteração nos sinais de vibração pode indicar problemas em seu componente ou em sua montagem. O fato de os sinais de vibração de uma máquina trazerem informações relacionadas com o seu funcionamento, indicam a saúde da máquina e orienta sobre a decisão de execução ou não da manutenção da mesma.

Assim, a análise de vibrações é o processo pelo qual as falhas em componentes móveis de um equipamento são descobertas através da taxa de variação das forças dinâmicas geradas. Tais forças afetam o nível de vibração, que por sua vez, pode ser avaliado em pontos acessíveis das máquinas, sem interromper o funcionamento das mesmas.

4.1.1 Sinais de vibrações

Os sinais de vibrações na prática consistem usualmente de muitas frequências que ocorrem simultaneamente. Esses sinais podem ser analisados pela amostragem amplitude vs tempo. Mas normalmente precisa-se descobrir quantos componentes, por exemplo da frequência de falha, exis-

tem no sinal, e em que frequência eles ocorrem. Desta forma esses componentes são melhores vistos no espectro amplitude de vibração vs frequência, e para efetuar essa mudança de domínio utiliza-se a Transformada de Fourier.

Transformada de Fourier

A transformada de Fourier (TF) é uma ferramenta matemática usada para converter uma função de um domínio para outro. A transformada de Fourier e a transformada inversa de Fourier de um sinal contínuo $x(t)$ são dadas respectivamente por:

$$X(\Omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\Omega t} dt \quad (4.1)$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\Omega)e^{j\Omega t} d\Omega \quad (4.2)$$

Sendo Ω a frequência em rad/s.

Neste trabalho utilizou-se o algoritmo FFT para o cálculo da transformada de Fourier.

4.2 Principais fontes de defeitos em motores elétricos

Motores elétricos são dispositivos que transformam energia elétrica em energia mecânica. Devido a sua grande utilização, estão expostos a uma ampla variedade de ambientes e condições que os tornam sujeitos a diversos defeitos.

Os fatores que afetam o comportamento dos motores podem ser agrupados em problemas de origem mecânica e problemas de origem elétrica ou magnética.

4.2.1 Falhas de origem mecânica

A grande maioria das ocorrências de defeitos mecânicos se manifestam nas primeiras harmônicas de rotação do motor. A seguir, serão analisadas algumas das fontes de vibrações de origem mecânica.

Desbalanceamento

O desbalanceamento de rotores é, sem dúvida, a mais frequente causa de vibrações em máquinas rotativas. Este fenômeno é caracterizado pela existência de desequilíbrios de massa em relação aos eixos de rotação. Tais desequilíbrios são originados por assimetrias, tolerâncias e desvios de forma, além das imperfeições da matéria-prima e da montagem. Na Figura 4.1 tem-se a representação básica do desequilíbrio de massa.

A vibração resultante é predominantemente radial, com forte componente em uma vez a frequência de rotação ($1 \times f_r$), (GIRDHAR E SCHEFFER, 2004).

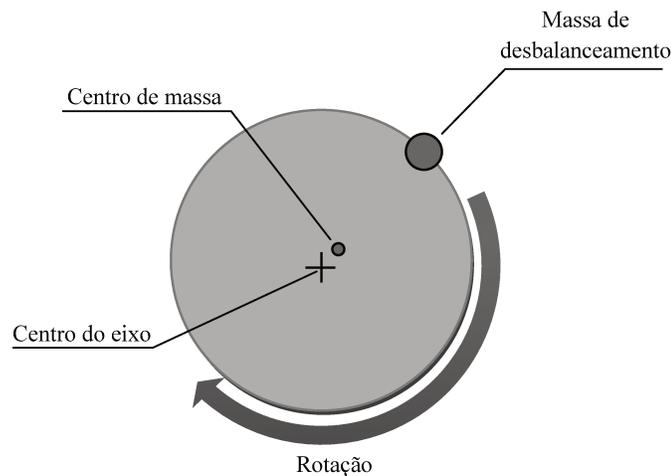
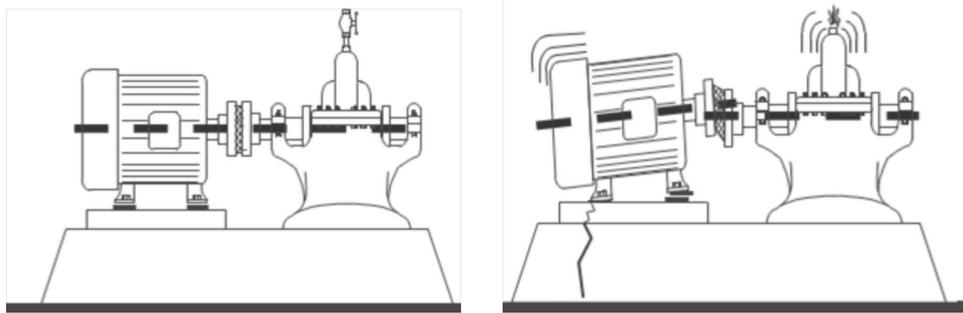


Figura 4.1: Representação do desequilíbrio de massa.

Desalinhamento

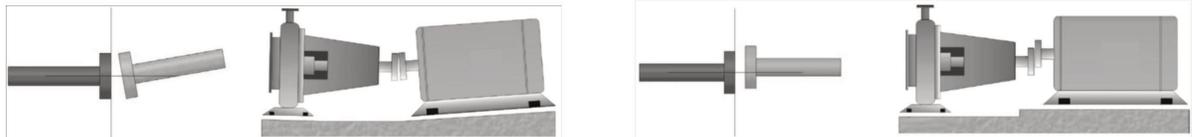
O desalinhamento é um problema tão comum quanto o desbalanceamento. Nas montagens mecânicas, geralmente, tem-se vários eixos, mancais e acoplamentos com características dinâmicas diferentes. Quando o conjunto opera, aparecem forças de interação, que acabam provocando vibrações. O desalinhamento é o desvio em relação à linha de centro durante a operação, Figura 4.2. O desalinhamento pode ser paralelo, angular ou uma combinação dos dois, Figura 4.3.



(a) Alinhado.

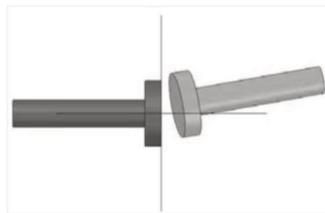
(b) Desalinhado.

Figura 4.2: Representação do desalinhamento.



(a) Angular.

(b) Paralelo.



(c) Misto.

Figura 4.3: Tipos de desalinhamento.

No desalinhamento angular, a vibração é mais forte na direção axial com componentes em $1xf_r$ e $2xf_r$. No desalinhamento paralelo, a vibração é maior na direção radial, com fortes componentes nas harmônicas da frequência de rotação ($1xf_r$, $2xf_r$, $3xf_r$, $4xf_r$), (GIRDHAR E SCHEFFER, 2004).

Folga mecânica

A folga mecânica se caracteriza pela presença de múltiplos harmônicos da frequência de rotação ($1xf_r$) e gera vibrações em máquinas rotativas devido a parafusos frouxos, folgas excessivas nos mancais, entre outros, Figura 4.4. O plano dominante é o radial com harmônicas mais altas e também sub harmônicas de $1xf_r$ ($0.5xf_r$, $1.5xf_r$, $2.5xf_r$, etc.), (MOBLEY E KEITH, 1999).

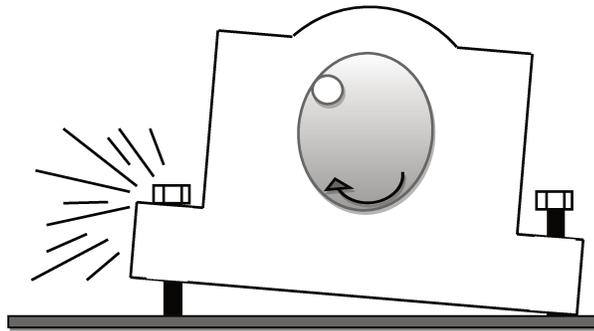


Figura 4.4: Representação da folga mecânica.

Rolamentos

Nos mancais de rolamento, os defeitos locais produzem uma série de impactos que podem excitar ressonâncias na estrutura da caixa e na carcaça do motor, entre 1kHz e 20kHz.

As frequências características em (Hz) da pista externa (BPFO, do inglês *Ball Pass Frequency Outer Race*), da pista interna (BPFI, do inglês *Ball Pass Frequency Inner Race*), do elemento rolante (BSF, do inglês *Ball Spin Frequency*) e da gaiola (FTF, do inglês *Fundamental Train Frequency*), são dadas, respectivamente, pelas Equações (4.3) a (4.6), sendo f_r a frequência de rotação (Hz),

d o diâmetro do elemento rolante (mm), D o diâmetro primitivo (mm), n o número de elementos rolantes e β o ângulo de contato do rolamento, conforme mostrado na Figura 4.5, (ALMEIDA E GÓZ, 1994).

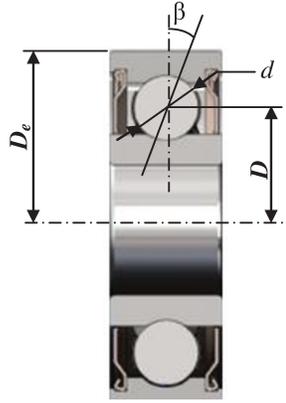


Figura 4.5: Elementos do rolamento.

$$BPFO = \frac{n}{2} f_r \left(1 - \frac{d}{D} \right) \cos \beta \quad (4.3)$$

$$BPFI = \frac{n}{2} f_r \left(1 + \frac{d}{D} \right) \cos \beta \quad (4.4)$$

$$BSF = \frac{d}{2D} f_r \left[1 - \left(\frac{d}{D} \right)^2 \cos^2 \beta \right] \quad (4.5)$$

$$FTF = \frac{1}{2} f_r \left(1 - \frac{d}{D} \right) \cos \beta \quad (4.6)$$

4.2.2 Falhas de origem elétrica

As vibrações causadas por falhas de origem elétrica são normalmente o resultado de forças magnéticas desequilibradas. Estas forças agem sobre o rotor e estator e podem ser devidos a: mancais do eixo do rotor excêntricos, barras do rotor trincadas ou quebradas, rotor não centrado dentro do estator, furo do estator elíptico e ventoinhas quebradas ou defeituosas.

Curto-circuito

Este tipo de falha no isolamento é tipicamente causado por contaminação do enrolamento, abrasão, vibração ou surtos de tensão, podendo ser agravada por falhas ou ineficiência do processo de impregnação, incluindo-se aí a utilização de condutores e verniz ou resina de má qualidade, mal preservados, ou incompatíveis com a classe térmica e tensão do equipamento, além do processo inadequado de cura. Na Figura 4.6, tem-se o exemplo de um curto-circuito entre espiras.



Figura 4.6: Curto-circuito entre espiras.

Segundo Nandi e Toliyat (2000), o 21° harmônico da frequência de linha (1260Hz) está sempre presente quando há uma falha no estator. Após a comparação de mais de 1000 espectros de fluxo magnético, pôde-se verificar, para um motor de 4 pólos, que os harmônicos 19° e 21° da frequência de linha (f_l) foram os mais excitados pela inserção do curto circuito, (LAMIM FILHO, 2007).

Desequilíbrio de fase

O desequilíbrio de fase ou desbalanceamento de tensão é caracterizado pela existência de níveis de tensão diferentes entre duas fases. Ele é geralmente identificado através da análise da corrente do motor elétrico. No entanto, como a técnica de análise de vibração é normalmente aplicada em plantas industriais, a detecção do desequilíbrio de fase através dessa técnica torna-se uma

importante contribuição para os programas de manutenção preditiva.

Lamim Filho (2007), mostrou que no espectro de vibração o 2º harmônico da rede (120Hz) está relacionado com o desequilíbrio de fase.

Barras quebradas

Uma das falhas frequentes no rotor das máquinas de indução é a quebra de barras e soldas frias na gaiola. Os sintomas característicos são vibração e ruído anormal. Segundo Thomson e Fenger (2001), as razões para quebras e trincas de barras ou de anéis podem ser provenientes de esforços:

- Térmico: causado por sobrecarga;
- Magnético: devido a vibrações, excentricidade;
- Residual: problemas de fabricação;
- Dinâmico: forças centrífugas, conjugado da carga;
- Ambiental: contaminação química ou devido à umidade;
- Mecânico: fadiga.

Brito e Pederiva (2002) mostraram que no espectro de vibração a detecção de barras quebradas é feita levando em conta $1x.f_r \pm 2x.s_f$ sendo s_f a frequência de escorregamento do motor elétrico.

Na Figura 4.7, tem-se um exemplo da simulação de um rotor com barras quebradas. Neste caso os pequenos furos representam um estágio inicial da falha de barras quebradas.



Figura 4.7: Exemplo da simulação de um rotor com barras quebradas.

4.3 Considerações finais

Neste capítulo foram descritas as principais falhas mecânicas e elétricas em motores de indução trifásicos, e as características de cada falha no espectro de vibrações. A análise de vibrações foi escolhida por ser uma técnica não intrusiva e consagrada no meio industrial. Já os motores elétricos foram escolhidos devido à sua grande utilização nas plantas fabris, pela facilidade de manuseio, diversidade de falhas e facilidade de inserção das mesmas. No Capítulo 5 será descrito como os sinais de vibração serão utilizados para a detecção das falhas por meio das SVMs, RNAs e lógica *fuzzy*.

5 RESULTADOS EXPERIMENTAIS

Neste capítulo apresentam-se a bancada experimental e os espectros coletados através da análise de vibrações para o treinamento e validação das técnicas de inteligência artificial utilizadas (RNA, Lógica *Fuzzy* e SVM). Os experimentos foram realizados para a condição normal de funcionamento, para as excitações mecânicas (desbalanceamento, desalinhamento e folga mecânica) e excitações elétricas (curto-circuito entre espiras, desequilíbrio de fase e barras quebradas), essas falhas foram escolhidas por se tratarem de alguns dos principais defeitos que os motores elétricos podem ter em situações reais, (BRITO, 2002).

5.1 Bancada e procedimentos experimentais

A bancada de teste é composta por um motor de indução trifásico {1}, uma máquina de corrente contínua {4}, um banco de resistência variável, um varivolt trifásico, um sistema de medição e um microcomputador. A máquina de corrente contínua funciona como gerador, alimentando um banco de resistências, atuando como carga para o motor de indução. Variando-se a corrente de excitação do campo do gerador CC ou alterando o banco de resistências, consegue-se, conseqüentemente, variação da carga do motor. Na Figura 5.1 tem-se uma vista geral da bancada de testes.

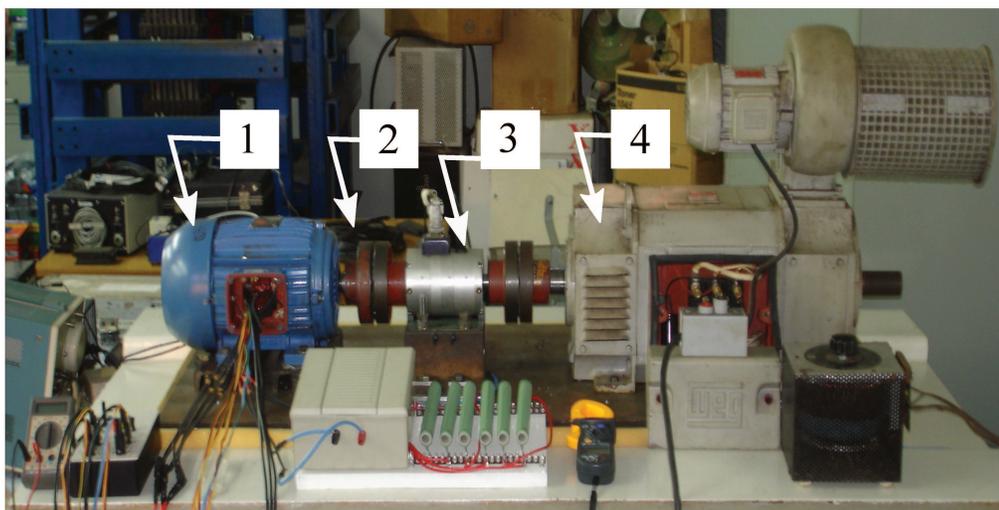


Figura 5.1: Bancada de testes.

Os defeitos foram introduzidos no motor elétrico, WEG (FH 88747), rotor gaiola, 5 CV, 1730 rpm, 220 V, 60 Hz, 4 pólos, categoria N, 44 barras, 36 ranhuras, rolamento SKF 6205-2Z, ID-1, carcaça 100L, classe de isolamento B, FS 1.15, Ip/In 7.5, IP 55, 13.8 A.

O gerador está acoplado ao motor elétrico através de acoplamentos flexíveis {2} e um torquímetro {3} da S. Himmelstein and Company, modelo MCRT 9-02T(1-3), 0-7500 rpm, bidirecional e torque máximo de 1000 lb-in que serviu para garantir a mesma condição de funcionamento em todos os testes realizados.

O motor de indução foi modificado de modo a possibilitar testes de falhas elétricas, tais como barras quebradas ou trincadas do rotor e baixa isolação (curto-circuito) entre espiras do enrolamento do estator.

Para simular uma baixa isolação, entre espiras de uma mesma fase, foram extraídas quatro derivações em uma bobina. Essas derivações foram dispostas externamente e ligadas em série (duas de cada vez) com um banco de resistência, de 1Ω , 100 Watts (cada uma) conectadas em paralelo e que são adicionadas ao circuito por meio de um grupo de disjuntores para que se controle a intensidade da corrente elétrica mantendo-se sempre a carga nominal do motor.

Cada bobina é constituída por 26 espiras com bitola do fio igual a 16. Como cada fase é formada por 6 bobinas, tem-se o total de 156 espiras por fase. Assim, a configuração permite analisar baixa isolação entre, no mínimo, duas espiras e, no máximo, dez espiras para uma mesma fase, correspondendo aos percentuais de 1.2% (2/156) e 6.4% (10/156) de baixa isolação.

A corrente de curto-circuito foi limitada em 10A, inserindo um resistor entre os terminais das espiras em curto-circuito. Dessa forma, pôde-se garantir que o motor iria funcionar por um tempo indeterminado e sem o comprometimento da isolação do mesmo. Caracterizando-se um início de uma real situação de baixa isolação (curto-circuito) entre espiras.

A excitação por desequilíbrio de fase foi obtida inserindo-se uma resistência variável em série com uma das fases de alimentação do motor elétrico. Para a simulação de barras quebradas efetuou-se o rompimento/trinca das barras utilizando, para isso, uma broca com diâmetro de $5mm$.

Já para as falhas mecânicas, o desbalanceamento foi inserido através da colocação de massas com diferentes pesos e alternadamente em diferentes posições em um disco fixado no eixo traseiro do motor elétrico. O desalinhamento foi causado movimentando-se o motor tanto na direção vertical como na horizontal, permitindo desalinhamentos do tipo paralelo, angular e misto. A folga mecânica foi inserida afrouxando-se os parafusos de fixação do motor elétrico.

Para cada falha inserida, o espectro foi coletado. Então retornava-se para a condição normal de funcionamento. Só então outra falha em uma escolha totalmente aleatória era inserida. Este procedimento garantiu que a falha, após retirada, não modificasse o funcionamento do motor.

Foram coletados aleatoriamente 680 sinais de vibração, sendo estes distribuídos entre as falhas em estudo: 110 sinais coletados para cada uma das falhas mecânicas e para a condição sem falha mecânica e 60 sinais coletados para cada uma das falhas elétricas e para a condição sem falha elétrica. Este banco de dados foi extraído dos trabalhos de Brito (2002) e Lamim Filho (2007), porém em contextos diferentes.

Em Brito (2002), as redes neurais artificiais foram utilizadas para diagnosticar falhas de desbalanceamento, desalinhamento, folga mecânica e barras quebradas.

Em Lamim Filho (2007), foi criado em sistema *fuzzy* para diagnosticar baixa isolação (curto-circuito), desequilíbrio de fase e barras quebradas.

Para todas as falhas os acelerômetros foram posicionados em quatro localizações diferentes, Figura 5.2, sendo elas denominadas: Motor Ventoinha Vertical (MVV), Motor Ventoinha Horizontal (MVH), Motor Acoplamento Vertical (MAV) e Motor Acoplamento Horizontal (MAH).

Porém verificou-se que o ponto de medição cujas falhas, no geral, tinham uma melhor caracterização foi o ponto MAV, (BRITO, 2002). Sendo, assim, MAV foi o ponto de medição escolhido para extrair dos espectros os parâmetros para treinamento.

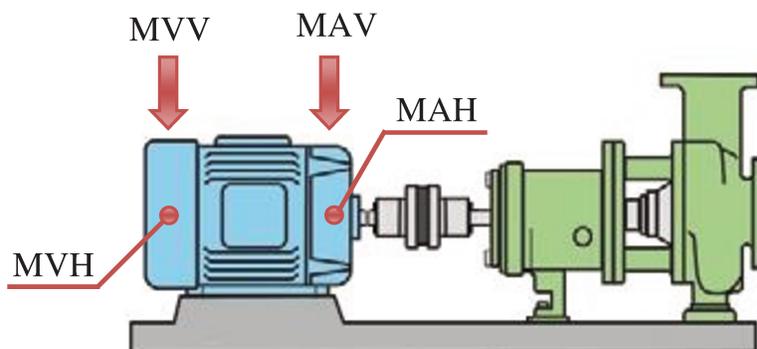


Figura 5.2: Pontos de coleta.

Na obtenção dos dados, utilizou-se a placa NI-6251 da National Instruments. Essa placa possui 16 canais analógicos de entrada que podem amostrar em até 200kHz e 2 contadores digitais de 24 bits cada. As entradas analógicas possuem resolução de 16 bits. Os sinais de vibração foram submetidos a um filtro *anti-aliasing* com frequência de corte de 2kHz. Para a implementação do algoritmo de aquisição de dados foi usado o software MATLAB R2010b.

Os sinais foram obtidos no plano radial (posição vertical e horizontal) do motor elétrico, tanto no lado do acoplamento quanto no lado da ventoinha (lado oposto ao acoplamento). Os sinais foram coletados a uma frequência de amostragem de 5 kHz e 20480 pontos, de modo a analisar toda a faixa de frequência na qual os defeitos em estudo são identificados.

Utilizou-se uma janela tipo Hanning e 4 médias foram feitas no sinal no domínio da frequência. Os sinais foram transformados para o domínio da frequência utilizando o algoritmo da FFT.

Antes dos testes, a bancada foi balanceada e alinhada objetivando eliminar as fontes de vibrações indesejáveis. Dessa forma foi possível determinar a condição normal de funcionamento do conjunto motor-gerador (assinatura da bancada), a qual se estipulou como sendo no máximo 0.5 mm/s de amplitude de vibração segundo a norma VDI 2056, (VDI, 1964).

5.1.1 Condição normal de funcionamento

Na Figura 5.3, tem-se o espectro da condição normal de funcionamento do motor elétrico.

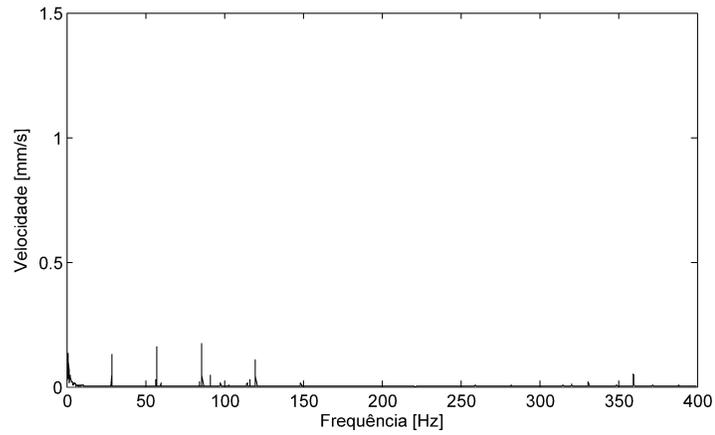


Figura 5.3: Espectro da condição normal de funcionamento do motor elétrico.

5.1.2 Desbalanceamento

Na Figura 5.4, tem-se o espectro da falha de desbalanceamento do motor elétrico.

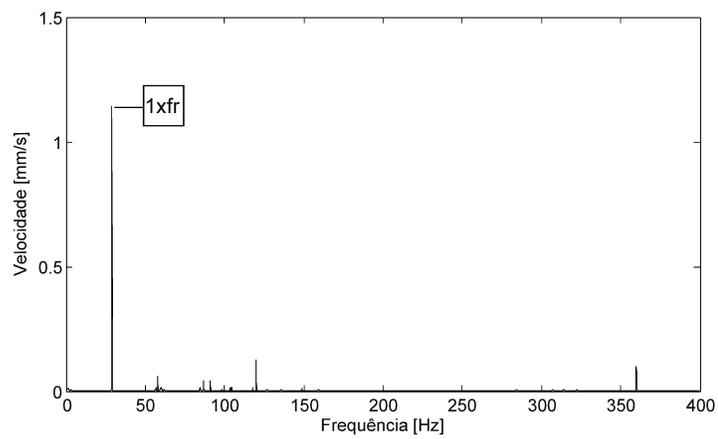


Figura 5.4: Espectro do motor elétrico desbalanceado.

5.1.3 Desalinhamento

Na Figura 5.5, tem-se o espectro da falha de desalinhamento do motor elétrico.

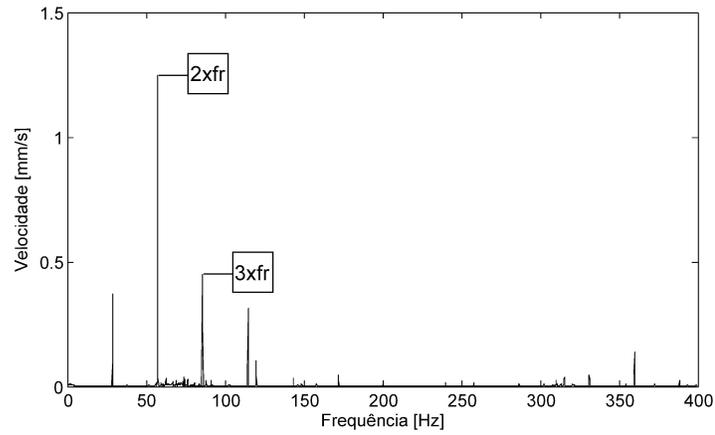


Figura 5.5: Espectro do motor elétrico desalinhado.

5.1.4 Folga mecânica

Na Figura 5.6, tem-se o espectro da falha de folga mecânica do motor elétrico.

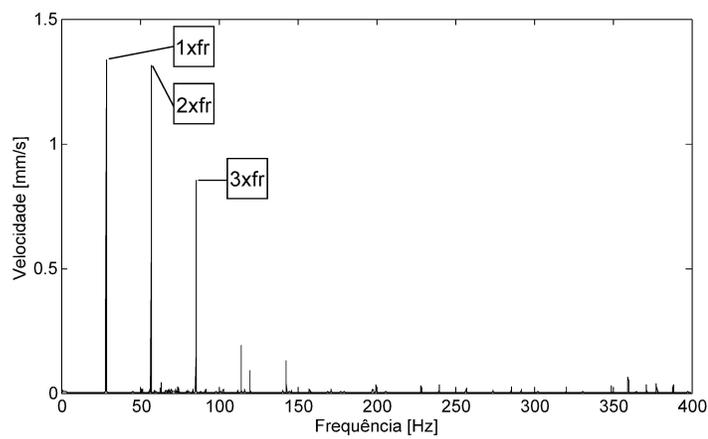


Figura 5.6: Espectro do motor elétrico com folga mecânica.

5.1.5 Curto-circuito

Na Figura 5.7, tem-se o espectro da falha de curto-circuito do motor elétrico.

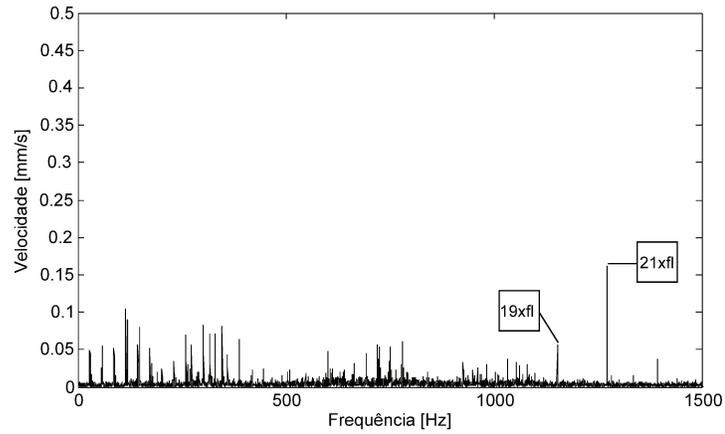


Figura 5.7: Espectro do motor elétrico com curto-circuito.

5.1.6 Desequilíbrio de fase

Na Figura 5.8, tem-se o espectro da falha de desequilíbrio de fase do motor elétrico.

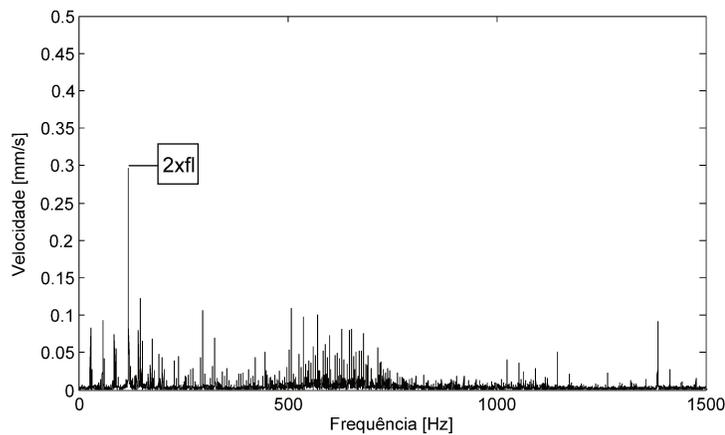


Figura 5.8: Espectro do motor elétrico com desequilíbrio de fase.

5.1.7 Barras quebradas

Na Figura 5.9, tem-se o espectro da falha de barras quebradas do motor elétrico.

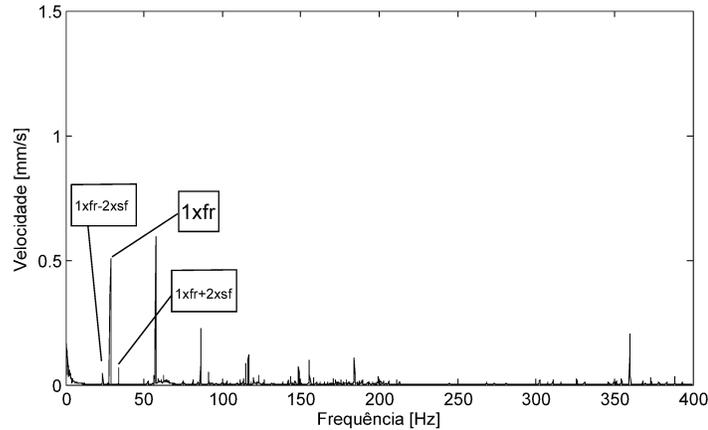
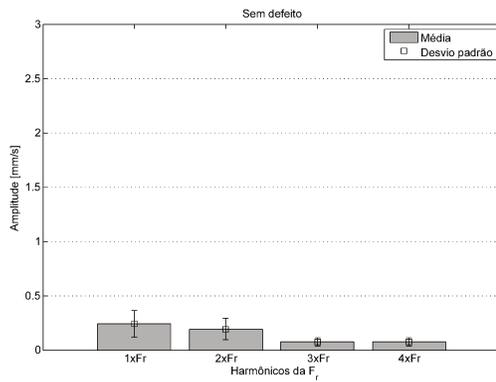


Figura 5.9: Espectro do motor elétrico com barras quebradas.

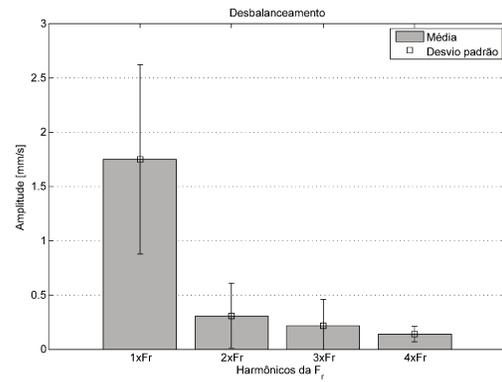
5.1.8 Dispersão das falhas

Segundo a norma VDI 2056, máquinas com até 20 CV de potência são consideradas sem falha, quando possuem níveis de vibração de até 0.71 mm/s, são consideradas com nível 1 (Permissível) quando os níveis de vibração estão entre 0.71 e 1.80 mm/s, são consideradas com nível 2 (Tolerável) quando os níveis de vibração estão entre 1.80 e 4.50 mm/s e nível 3 (Não Permissível) quando apresentam níveis de vibração superiores à 4.50 mm/s.

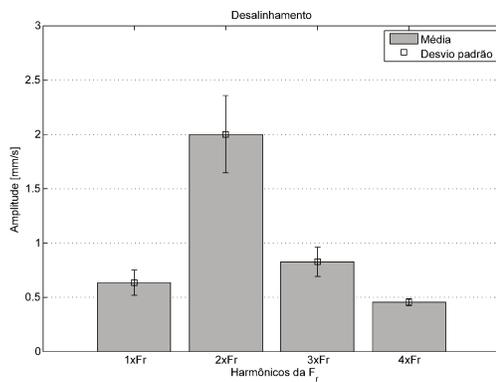
Para este estudo serão considerados apenas 2 níveis de severidade, nível 1 com amplitudes de vibração entre 0.71 e 1.80 mm/s e nível 2 com amplitudes de vibração acima de 1.80 mm/s. Na Figura 5.10 têm-se, para as falhas mecânicas, a média e o desvio padrão das frequências características para os testes realizados, nos dois níveis de severidade.



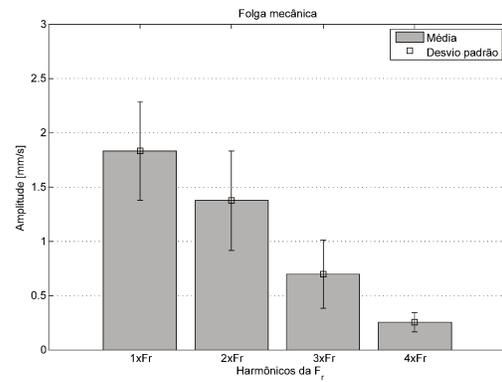
(a) Sem defeito.



(b) Desbalanceamento.



(c) Desalinhamento.



(d) Folga mecânica.

Figura 5.10: Média e desvio padrão para as falhas mecânicas.

Para defeitos de origem elétrica os níveis de vibração são relativamente baixos, não sendo possível utilizar a norma VDI 2056. Para estas falhas os níveis foram definidos de acordo com a intensidade das falhas inseridas. Para a falha de curto-circuito, quando inseria-se 2 espiras em curto considerou-se como sendo nível 1 e quando inseria-se 10 espiras em curto considerou-se como nível 2.

Para a falha de desequilíbrio de fase o nível 1 foi considerado quando duas fases estavam à uma voltagem de 220V e a terceira com 210V e o nível 2 foi considerado quando duas fases estavam à uma voltagem de 220V e a terceira com 200V. Por último foi considerado como nível 1 quando haviam três barras quebradas e nível 2 quando haviam sete barras quebradas.

Na Figura 5.11 têm-se, para as falhas elétricas, a média e o desvio padrão das frequências características para os testes realizados, nos dois níveis de severidade.

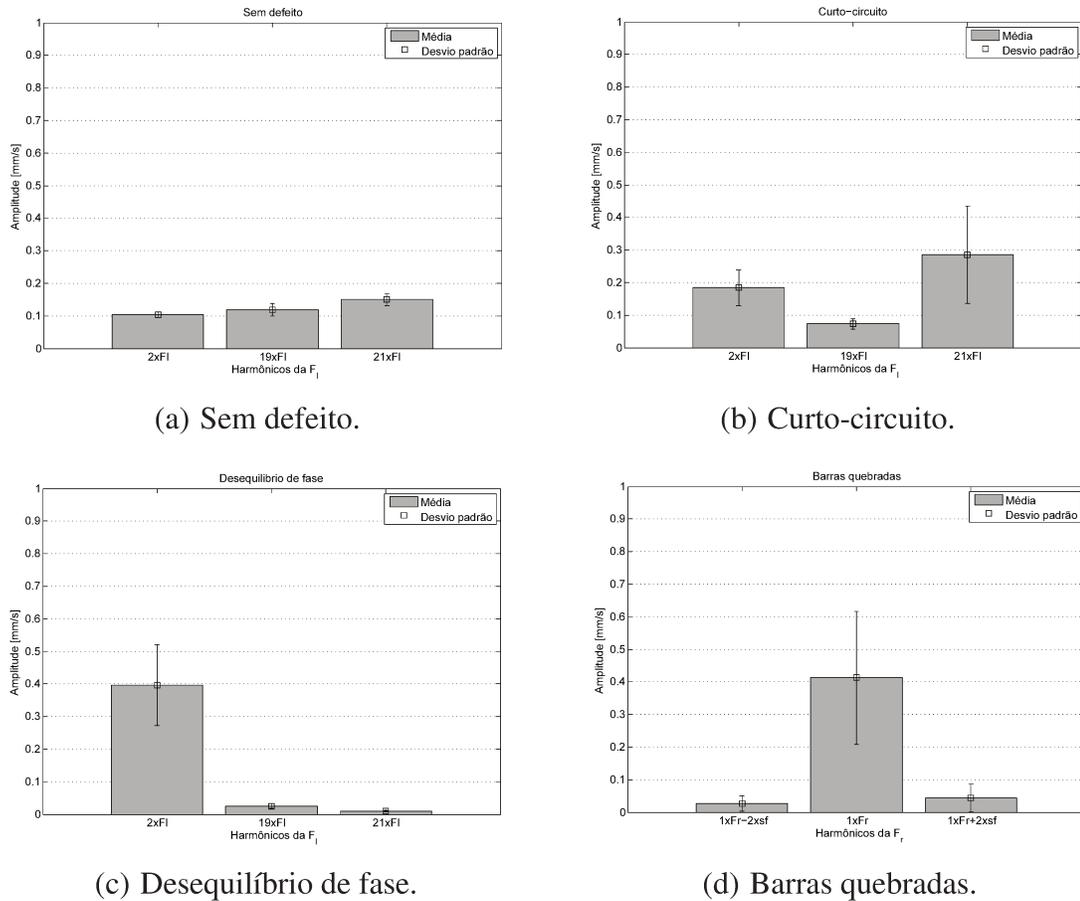


Figura 5.11: Média e desvio padrão para as falhas elétricas.

5.2 SVM para classificação de falhas

Como visto a SVM é uma técnica útil para classificação de dados com boa capacidade de generalização. Mas para que se alcance uma boa taxa de acerto, parâmetros e métodos devem ser adotados.

5.2.1 Seleção dos sinais de vibração

Os dados obtidos a partir da análise do espectro de vibração contém não só informações sobre as falhas através das frequências determinísticas, mas também diversos parâmetros que podem e devem ser negligenciados, como o ruído.

Para a seleção dos parâmetros mais relevantes para a predição das falhas, utilizou-se o algoritmo de *forward feature selection* disponível no *toolbox* do MATLAB. E, como esperado, os parâmetros que tiveram maior ocorrência dentre as falhas, foram também os encontrados na teoria.

Na Figura 5.12, tem-se a ocorrência dos harmônicos da frequência de rotação durante a seleção de parâmetros para a detecção de falhas mecânicas (desbalanceamento, desalinhamento e folga mecânica). E na Figura 5.13 apresenta a ocorrência dos harmônicos da frequência de linha durante a seleção de parâmetros para a detecção de falhas elétricas (curto-circuito, desequilíbrio de fase).

A Figura 5.14 e 5.15 mostra os valores de amplitude de vibração no espaço \mathbb{R}^3 para as falhas mecânicas e elétricas respectivamente. Pode-se notar que os dados amostrados estão agrupados e sobrepostos e, portanto, não podem ser linearmente separados.

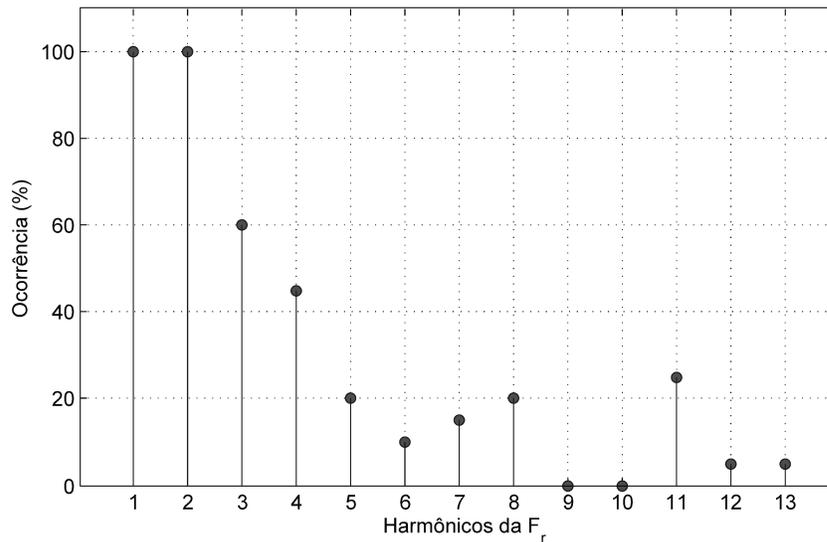


Figura 5.12: Parâmetros para diagnóstico de falhas mecânicas.

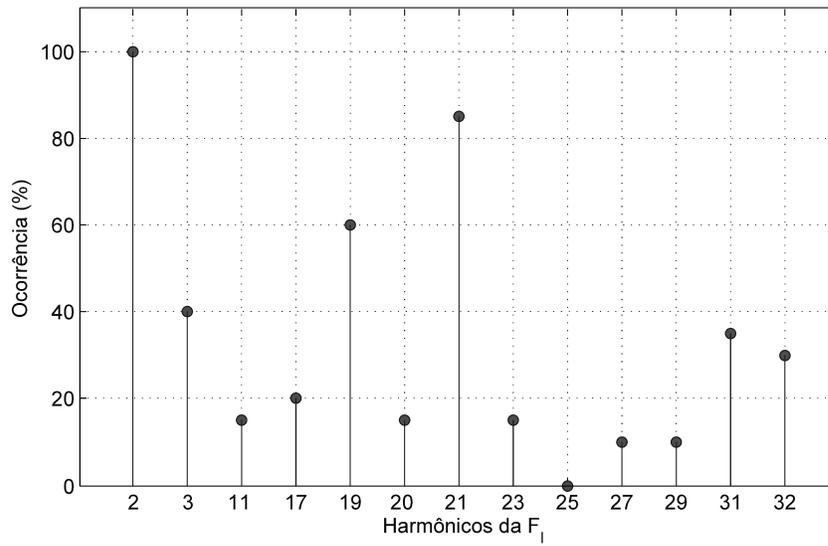


Figura 5.13: Parâmetros para diagnóstico de falhas elétricas.

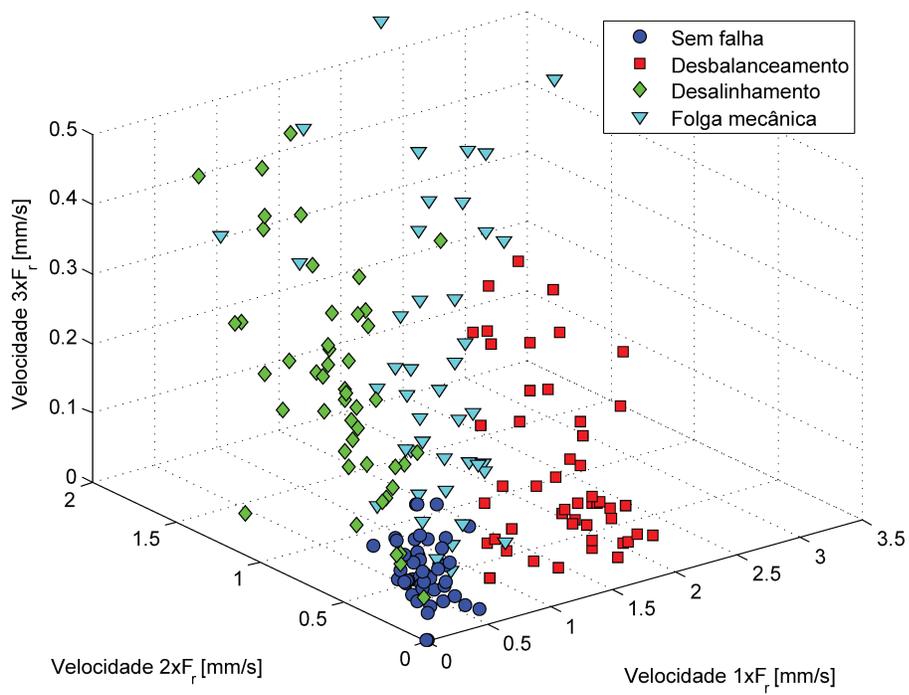


Figura 5.14: Distribuição espacial das falhas mecânicas.

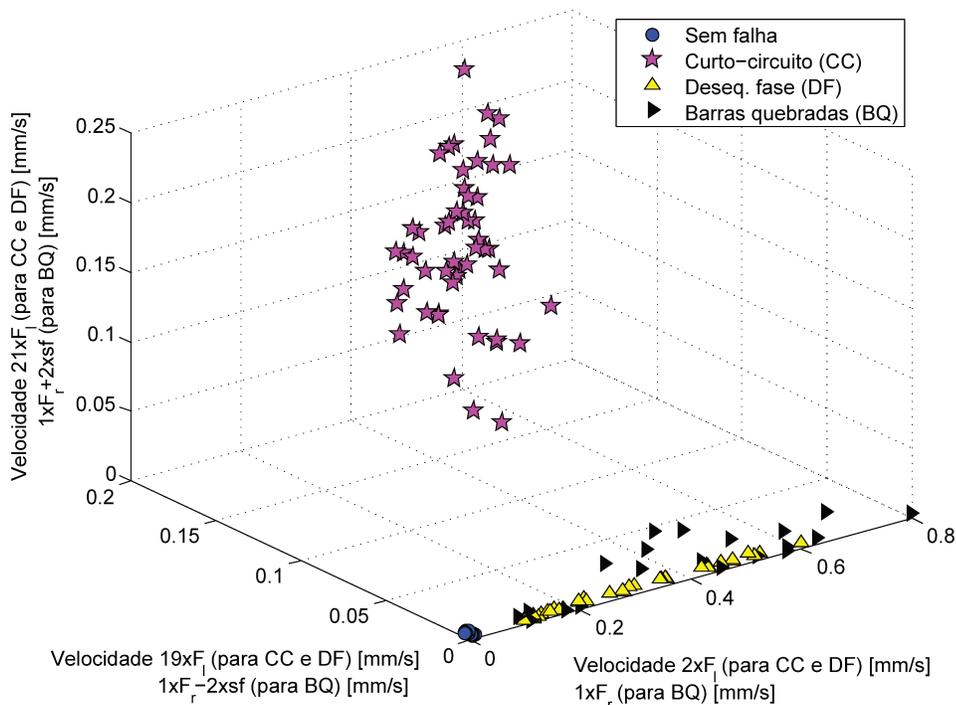


Figura 5.15: Distribuição espacial das falhas elétricas.

5.2.2 Escolha dos parâmetros de treinamento

A partir daí foram definidos os parâmetros de entrada: $1xf_r$, $2xf_r$, $3xf_r$ e $4xf_r$ para detecção de falhas mecânicas, $2xf_l$, $19xf_l$ e $21xf_l$ para curto-circuito e desequilíbrio de fase e $1xf_r$ e $1xf_r \pm 2xsf$ para as barras quebradas. Efetuou-se o treinamento da SVM¹ com o kernel RBF (*Radial Basis Function*). Esse tipo de kernel foi escolhido, pois ao contrário do kernel linear o RBF, mapeia as amostras em um espaço dimensional superior ao do problema inicial, o que o torna capaz de trabalhar em situações onde há não linearidade.

Outra razão para a escolha do kernel RBF é o número de hiperparâmetros que influenciam na complexidade da escolha do modelo, já que o kernel polinomial tem mais hiperparâmetros do que o kernel RBF. Por último, deve-se notar que o kernel sigmóide não é válido sob alguns parâmetros.

¹Para a classificação dos defeitos mecânicos e elétricos através das SVMs, utilizou-se o software LIBSVM desenvolvido por Chang e Lin (2011), versão 3.1.

O kernel RBF possui dois parâmetros (C e γ) que devem ser escolhidos pelo usuário (ver Tabela 3.1). Uma escolha errada desses parâmetros pode levar a má generalização do classificador. Para evitar este problema efetuou-se uma validação cruzada v -vezes (do inglês *v-fold cross-validation*), que consiste em dividir o conjunto de treinamento em v subconjuntos de tamanhos iguais. Sequencialmente, um subconjunto é testado usando o classificador treinado sobre os restantes $v - 1$ subconjuntos. Assim, cada dado de todo o conjunto treinado, é predito com o maior acerto da validação cruzada. Esta etapa é importante uma vez que a validação cruzada pode prevenir o problema de sobre ajuste, (HSU *et al.*, 2003).

Juntamente com a validação cruzada é feita uma pesquisa em grade (do inglês *search-grid*), dos melhores valores de C e γ . Vários pares de (C, γ) são julgados e aquele com o melhor acerto na validação cruzada é escolhido. Tentar um crescimento exponencial de sequências de C e γ é um método prático para identificar os melhores parâmetros, (HSU *et al.*, 2003). Para ambas as falhas em estudo buscou-se entre valores de $C = 2^{-5}, 2^{-4}, \dots, 2^{15}$ e para $\gamma = 2^{-15}, 2^{-14}, \dots, 2^3$.

Para a decomposição multiclases foi utilizado o método "um-contra-um". Pois, apesar de o método "um-contra-todos" ser um bom método cujo desempenho é comparável ao "um-contra-um", o método "um-contra-um" tem um tempo de treinamento menor, (HSU E LIN, 2002; WIDODO E YANG, 2007A; WANG E CHIANG, 2009).

Fazendo um resumo dos passos para a utilização da SVM tem-se:

- Transformar os dados para o formato de entrada da SVM;
- Caso os dados estejam em escalas muito diferente, colocá-los na mesma escala;
- Considerar em primeiro lugar a utilização do kernel RBF, devido às vantagens apresentadas;
- Utilizar uma validação cruzada e a pesquisa em grade para buscar pelos melhores valores dos hiperparâmetros;
- Utilizar os melhores valores dos hiperparâmetros encontrados para o treinamento;
- Testar a SVM.

5.2.3 Diagnóstico das falhas

Foi escolhido como parâmetros de entrada, as amplitudes das frequências anteriormente descritas. O conjunto de dados foi misturado e particionado na metade, efetuou-se este processo três vezes, obtendo assim, três conjuntos de treinamento e três conjuntos de validação, com dados variados. Os conjuntos constituídos por falhas mecânicas possuíam 220 dados de treinamento e 220 dados de validação cada um, e os conjuntos de falhas elétricas possuíam 120 dados de treinamento e 120 dados de validação cada um. Efetuou-se o treinamento da SVM com o conjunto de treinamento contendo falhas em nível 1 e em seguida validou-a com os conjunto de teste contendo também falhas no nível 1. Efetuou-se este processo por três vezes, utilizando cada um dos diferentes conjuntos criados, sendo eles denominados conjunto 1 (c1), conjunto 2 (c2) e conjunto 3 (c3). Essa divisão em sub-conjuntos com dados variados teve o intuito de verificar a significância dos dados experimentais.

Na Tabela 5.1, tem-se os resultados da classificação da SVM para as falhas mecânicas além da condição normal de funcionamento.

Tabela 5.1: Classificação das falhas mecânicas com a SVM.

Falha	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Sem defeito	96.00	96.00	98.00
Desbalanceamento	96.00	96.00	92.00
Desalinhamento	88.00	88.00	88.00
Folga mecânica	80.00	80.00	84.00
Parâmetros (C, γ)	(128,0.250)	(64,0.500)	(2048,3.125 e^{-2})

Na Tabela 5.2, tem-se os resultados da classificação da SVM para as falhas elétricas além da condição normal de funcionamento.

Tabela 5.2: Classificação das falhas elétricas com a SVM.

Falha	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Sem defeito	100	100	100
Curto-circuito	100	100	100
Desequilíbrio de fase	77,78	100	100
Barras quebradas	100	100	100
Parâmetros (C, γ)	(32,8)	(512,8)	(64,8)

Observando os resultados das Tabelas 5.1 e 5.2, tem-se que a metodologia utilizada obteve resultados satisfatórios. Pode-se perceber que a SVM é uma técnica que apresenta poucas diferenças nos resultados mesmo alterando-se os conjuntos de treinamento e validação. Apesar da diferença entre os parâmetros (C, γ) dos diferentes treinamentos, o usuário não precisa saber qual o melhor par que deve ser utilizado para cada conjunto de treinamento, pois a escolha é feita automaticamente, evitando-se escolhas erradas, o que comprometeria o desempenho da SVM.

Nas falhas elétricas observa-se porcentagens de acerto de 100% na maioria os casos, isso se deve ao fato de que estas falhas estão distantes entre si na sua distribuição espacial, o que facilita o trabalho da SVM. O caso observado diferente de 100% se deve ao conjunto de falha da desequilíbrio de fase possuir alguns dados que estão um pouco dispersos do restante do conjunto, o que resultou em erros de classificação. Como o processo de divisão dos sub-conjuntos (c1, c2 e c3) foi aleatório, aconteceu que no conjunto de treinamento c1 estes dados dispersos não fizeram parte do conjunto de treinamento, implicando que o modelo não foi capaz de reconhecer estes dados durante a classificação. Este problema pode ser resolvido utilizando uma amostra maior de treinamento, uma vez que os dados se mostraram significativos para caracterizar as falhas, não necessitando desta divisão em conjuntos menores.

Para as falhas mecânicas foi obtida porcentagens de acerto boas porém inferiores à 100%, isso se deve ao fato de que as falhas mecânicas estão sobrepostas a agrupadas, o que dificulta o processo de classificação.

5.3 RNA para classificação de falhas

Os mesmos três conjuntos de dados utilizados para o treinamento da SVM também foram utilizados para o treinamento da RNA, com o objetivo de realizar um estudo comparativo. A rede neural foi desenvolvida utilizando o *Neural Network Toolbox* (NNTOOL) do software MATLAB.

5.3.1 Escolha dos parâmetros de treinamento

As entradas da RNA foram as mesmas da SVM. Dessa forma, foram treinadas e comparadas três arquiteturas de RNAs, 4x3x1, 4x5x1 e 4x10x1 para as falhas mecânicas e 3x3x1, 3x5x1 e 3x10x1 para as falhas elétricas, que representa, por exemplo, a arquitetura 4x3x1, 4 neurônios de entrada, 3 neurônios na camada intermediária e 1 neurônio na camada de saída. O erro desejado foi de 0.5%, taxa de aprendizagem de 0.01, constante de *momentum* 0.9, inicialização de pesos aleatórios e número máximo de épocas igual a 1000.

Esses parâmetros de treinamento da RNA foram escolhidos, pois, para garantir o sucesso do aprendizado deve-se usar uma taxa de aprendizado baixa. Mas isso faz com que o algoritmo se torne lento demais, então, para compensar este problema, tenta-se acelerar a velocidade da rede através do *momentum*, (HAYKIN, 1998).

O *momentum* garante uma aceleração e desaceleração gradual da velocidade de alteração dos pesos. O valor do *momentum* é um valor real entre 0 e 1, mas normalmente utiliza-se um valor de *momentum* alto, por exemplo 0.9, com o objetivo de acelerar a escolha dos pesos ideais, o que reduz o erro na saída do neurônio mais rapidamente, sem, no entanto, provocar perturbações exageradas no processo de convergência ao mínimo global do erro. Todos os cálculos feitos pela RNA são realizados sobre cada um dos exemplos da base de aprendizado. Chama-se de época, a apresentação de todo o conjunto de exemplos disponíveis, de modo que a rede possa corrigir os pesos, (HAYKIN, 1998).

As topologias foram desenvolvidas utilizando o método de treinamento Retropropagação do erro (*backpropagation*) através do aprendizado supervisionado e sendo a RNA, otimizada pelo

método de Levenberg-Marquardt. Utilizou-se a função sigmoïdal, como função de ativação dos neurônios de entrada e neurônios intermediários e para o neurônio de saída utilizou-se a função linear. A velocidade de ativação utilizada foi de 0.001.

5.3.2 Diagnóstico das falhas

Foram utilizados os mesmos três conjuntos, usados no treinamento e na validação da SVM. Cada um dos conjuntos foi usado para treinar cada uma das arquiteturas propostas (4x3x1, 4x5x1 e 4x10x1 falhas mecânicas e 3x3x1, 3x5x1 e 3x10x1 falhas elétricas), houve essa diferenciação do número de neurônios de entrada para as falhas mecânicas e elétricas devido ao fato de haver quatro frequências características de falhas para os defeitos mecânicos e três frequências características de falhas para os defeitos elétricos. Na Tabela 5.3, tem-se os resultados da classificação da RNA para as falhas mecânicas além da condição normal de funcionamento.

Tabela 5.3: Classificação das falhas mecânicas com a RNA.

Falha	Topologia	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Sem falha	4x3x1	81.24	80.57	77.46
	4x5x1	86.24	84.49	91.29
	4x10x1	83.80	90.96	91.56
Desbalanceamento	4x3x1	76.16	74.71	73.20
	4x5x1	90.80	86.97	83.35
	4x10x1	78.64	86.25	75.16
Desalinhamento	4x3x1	85.82	80.69	77.30
	4x5x1	94.08	84.77	81.18
	4x10x1	91.87	91.06	87.49
Folga mecânica	4x3x1	61.29	59.68	60.36
	4x5x1	74.65	70.72	60.43
	4x10x1	62.04	64.97	71.29

Na Tabela 5.4, tem-se os resultados da classificação da RNA para as falhas elétricas além da condição normal de funcionamento.

Tabela 5.4: Classificação das falhas elétricas com a RNA.

Falha	Topologia	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Sem falha	3x3x1	95.11	94.85	95.03
	3x5x1	94.73	90.93	92.93
	3x10x1	95.30	94.45	92.14
Curto-circuito	3x3x1	91.22	89.73	83.61
	3x5x1	86.50	85.35	81.33
	3x10x1	84.40	81.18	77.76
Deseq. de fase	3x3x1	94.90	95.47	94.94
	3x5x1	95.05	92.63	93.35
	3x10x1	94.59	94.49	92.49
Barras quebradas	3x3x1	92.60	93.23	91.48
	3x5x1	88.44	86.11	85.99
	3x10x1	84.29	85.91	88.42

Analisando os resultados das Tabelas 5.3 e 5.4, tem-se que a metodologia utilizada para as RNAs obtiveram também bons resultados. Observa-se que os resultados obtidos com a utilização da RNA para os mesmos dados de treinamento, foram inferiores aos acertos obtidos com a SVM. Quando se utiliza as RNAs, há diversos parâmetros de aprendizagem que influenciam no seu desempenho. As topologias de rede são fatores muito importantes no processo de aprendizagem de uma RNA, como visto nos resultados anteriormente mostrados, diferentes topologias resultam em diferente taxas de acerto, por exemplo, para o caso das falhas mecânicas a topologia que apresentou melhores resultados foi a 4x5x1 e para as falhas elétricas a melhor topologia foi a 3x3x1.

Essa dificuldade de escolha da topologia adequada ao problema em estudo, faz com que se busque métodos que auxiliem nesta escolha. Estes métodos muitas das vezes são complexos e não garantem bons resultados. Outra saída seria treinar varias redes com topologias diferentes, como

está sendo feito, porém diversos teste serão necessários até que se obtenha a melhor topologia, e o que não garantirá que essa topologia será boa caso o conjunto de dados mude. Tem-se ainda, que à medida que se deseja analisar bancos de dados muito grandes, essa busca da melhor topologia vai se tornando cada vez mais demorada.

Além da topologia de rede, existem também outros parâmetros que influenciam no resultado final como: o erro desejado, a taxa de aprendizagem, *momentum*, número de épocas, algoritmo de treinamento, método de otimização, função de ativação de entrada e de saída, velocidade de ativação, entre outros, o que leva a entender que deve haver um compromisso grande por parte do usuário nesta escolha, buscando os melhores valores para cada caso, lembrando que esta busca é empírica. Tem-se ainda, que além de todos este fatores no processo de utilização da RNA, no fim, a rede pode não aprender.

Assim, os resultados obtidos pelas redes neurais artificiais mostraram-se bons, pois, sem utilizar nenhuma técnica de IA, um especialista teria muito trabalho para analisar todos os espectros. Porém, utilizar as RNAs, requer um pouco mais de conhecimento por parte do usuário. E finalmente fazendo-se uma busca mais refinada dos parâmetros que influenciam o processo, melhores resultados podem, talvez, serem obtidos.

5.4 Lógica *fuzzy* para classificação de falhas

Na lógica *fuzzy* também foram utilizado os três mesmos conjuntos de dados que serviram para o treinamento da SVM e da RNA. Para a detecção dos defeitos, as amplitudes dos espectros de vibração foram consideradas como variáveis de entrada para o sistema *fuzzy*. Variáveis de entrada *fuzzy*: $1xf_r$, $2xf_r$ e $3xf_r$ para detecção de falhas mecânicas, $2xf_l$, $19xf_l$ e $21xf_l$ para curto-circuito e desequilíbrio de fase e $1xf_r$ e $1xf_r \pm 2xs_f$ para as barras quebradas. Para as variáveis de saída, têm-se as possíveis falhas de desbalanceamento (DB), desalinhamento (DS), folga mecânica (FM), curto-circuito (CC), desequilíbrio de fase (DF) e barras quebradas (BQ) e a condição normal de funcionamento (SF).

5.4.1 Variáveis linguísticas, funções de pertinência e regras

Uma variável linguística u no universo U é definida como sendo parte de um conjunto de termos $T(u)$. Por exemplo, se u for velocidade, então seu conjunto de termos $T(u)$ poderia ser, Equação (5.1):

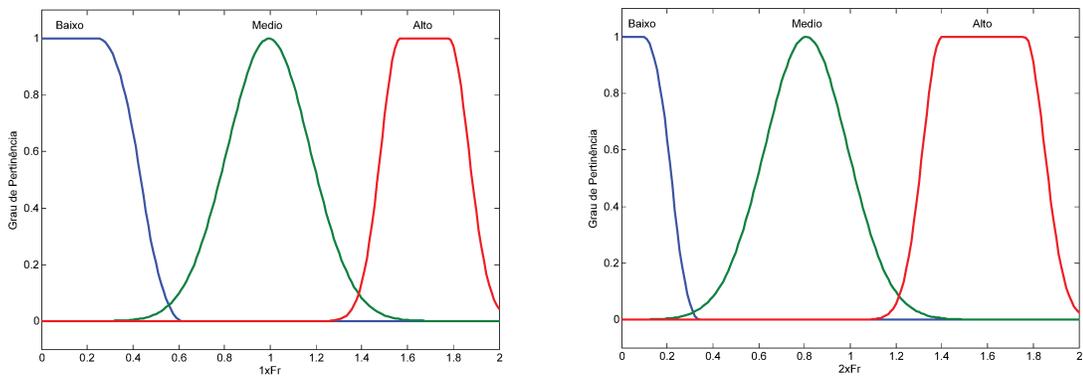
$$T(\text{velocidade}) = \{\text{baixa}, \text{media}, \text{alta}\} \quad (5.1)$$

As variáveis de entrada do sistema *fuzzy* foram definidas como sendo: $E = 1xf_r, 2xf_r$ e $3xf_r$ para falhas mecânicas e $E = 2xf_l, 19xf_l, 21xf_l, 1xf_r$ e $1xf_r \pm 2xs_f$ para as falhas elétricas. Essas entradas são as amplitudes de cada uma destas frequências, ou seja, um valor numérico. Para que o sistema *fuzzy* interprete estes valores são necessários convertê-los para valores *fuzzy*. Estes valores *fuzzy* são termos que aproximam os valores numéricos (amplitudes), para algo mais próximo do entendimento humano. Foi definido que as entradas *fuzzy* para falhas mecânicas serão: Baixa (B), Média (M) ou Alta (A), Equação (5.2), e para as falhas elétricas serão Baixa (B) ou Alta (A), Equação (5.3).

$$T(E) = \{\text{Baixo (B)}, \text{Medio (M)}, \text{Alto (A)}\} \quad (5.2)$$

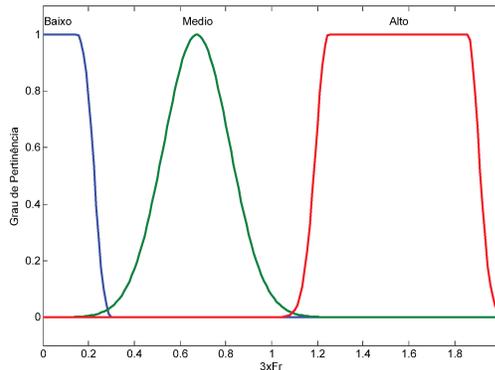
$$T(E) = \{\text{Baixo (B)}, \text{Alto (A)}\} \quad (5.3)$$

Porém é necessário informar ao sistema *fuzzy* qual a faixa de amplitude que a frequência determinística deve estar para se encaixar em cada uma das entradas do sistema, ai entram as funções de pertinência de entrada. As funções de pertinência foram geradas para os valores Baixo (B), Médio (M) e Alto (A) para falhas mecânicas e Baixo (B) e Alto (A) para as falhas elétricas, de acordo com uma análise prévia das amplitudes das frequências determinísticas de cada falha. São as funções de pertinência de entrada que avaliam se as amplitudes das frequências são B, M ou A para falhas mecânicas ou B ou A para falhas elétricas, para posteriormente serem classificadas de acordo com as regras. Na Figura 5.16, tem-se as funções de pertinência para cada uma das entradas das falhas mecânicas ($1x f_r$, $2x f_r$, $3x f_r$).



(a) Função de pertinência para $1x f_r$.

(b) Função de pertinência para $2x f_r$.

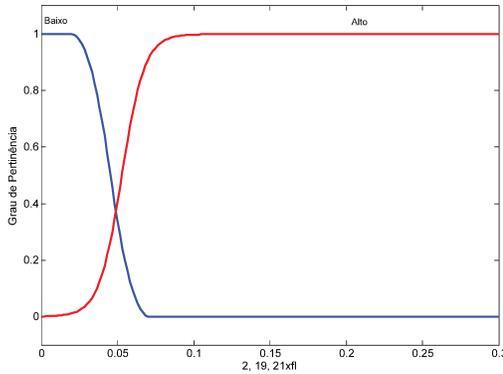


(c) Função de pertinência para $3x f_r$.

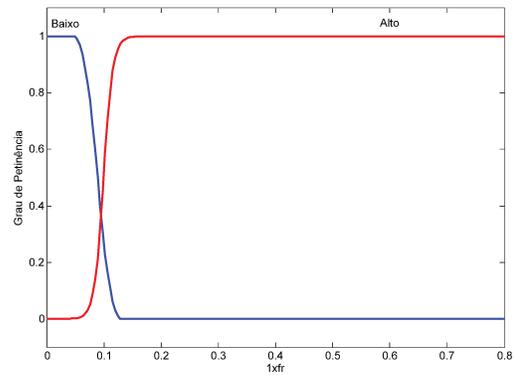
Figura 5.16: Função de pertinência para a entrada para as falhas mecânicas.

Na Figura 5.17, têm-se as funções de pertinência para cada uma das entradas

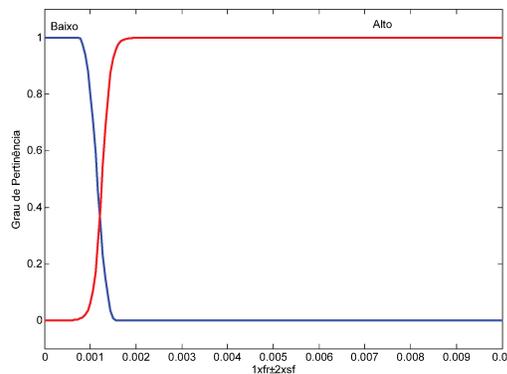
($2x_{f_l}$, $19x_{f_l}$ e $21x_{f_l}$ para curto-circuito e desequilíbrio de fase e $1xf_r \pm 2xs_f$ para as barras quebradas.) das falhas elétricas.



(a) Função de pertinência para $2x_{f_l}$, $19x_{f_l}$ e $21x_{f_l}$.



(b) Função de pertinência para $1xf_r$.



(c) Função de pertinência para $1xf_r \pm 2xs_f$.

Figura 5.17: Função de pertinência para a entrada para as falhas elétricas.

Deve-se agora definir como o sistema *fuzzy* irá responder à estas entradas, ou seja, como o sistema classificará a condição final do motor. Para a interpretação da condição do motor, têm-se as variáveis de saída: para falhas mecânicas DB, DS e FM e para falhas elétricas CC, DF e BQ interpretadas pelas variáveis linguísticas da Equação (5.4) para falhas mecânicas e Equação (5.5) para falhas elétricas. Sendo $S = DB, DS$ ou FM para falhas mecânicas e $S = CC, DF$ ou BQ , para falhas elétricas. Em outras palavras, em resposta às entradas, foi definido que o sistema *fuzzy* poderá

diagnosticar o motor como: para falhas mecânicas, sem defeito mecânico, nível intermediário de uma falha ou nível crítico de uma falha e para falhas elétricas, sem defeito elétrico ou com defeito devido à uma das falhas elétricas.

$$T(S) = \{Sem\ Defeito\ (SD),\ Intermediario\ (NI),\ Critico\ (NC)\} \quad (5.4)$$

$$T(S) = \{Sem\ Defeito\ (SD),\ Com\ Defeito\ (CD)\} \quad (5.5)$$

Visto que a saída do sistema são termos linguísticos (Equação (5.4) e (5.5)), a interpretação destes termos por parte do computador ficaria difícil. Então, entra a defuzzificação, que transforma a saída para valores numéricos. Estes valores ficariam vagos e sem sentido, uma vez que o usuário não saberia o que cada um deles representaria, então utiliza-se as funções de pertinência de saída para se definir qual a faixa de valores representam as condições do motor na saída do sistema.

Na Figura 5.18, tem-se a função de pertinência de saída para as falhas mecânicas e na Figura 5.19, tem-se a função de pertinência de saída para as falhas elétricas. Estas funções de saída representam como o sistema *fuzzy* dará o diagnóstico do motor elétrico.

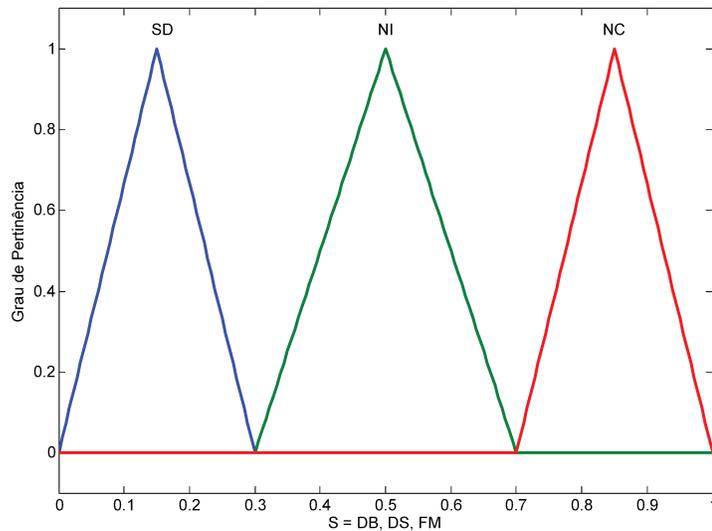


Figura 5.18: Função de pertinência para a condição do motor relativo à falhas mecânicas.

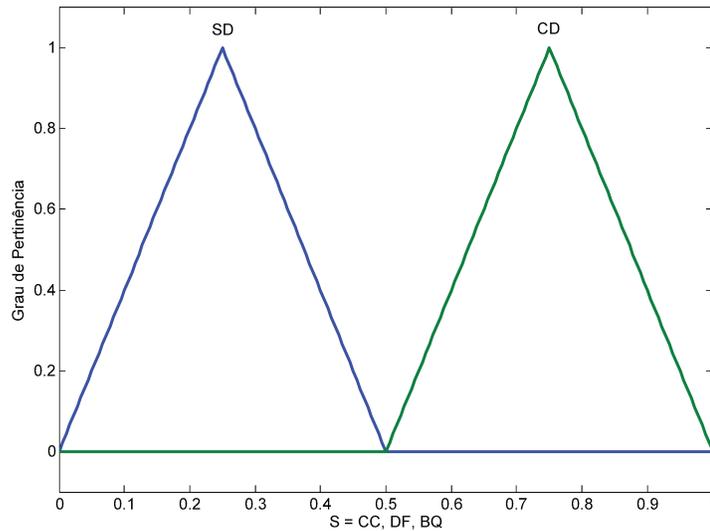


Figura 5.19: Função de pertinência para a condição do motor relativo à falhas elétricas.

Assim a faixa de valores que o sistema irá retornar como resposta é escolhida pelo usuário. Em relação às falhas mecânicas, se o sistema tem como saída SD, interpreta-se de que o motor encontra-se sem defeito mecânico, sendo que a faixa de valores estipulados para representar a saída SD foi de 0 a 0.3. NI representa o nível intermediário da falha mecânica, podendo variar de 0.30 a 0.70 e NC representa o nível crítico da falha mecânica, podendo variar de 0.70 a 1.0, Figura 5.18.

Já nas falhas elétricas, para SD, tem-se a interpretação de que o motor encontra-se sem defeito elétrico, sendo que a faixa de saída numérica que representa a situação SD foi definida de 0 a 0.50 e CD representa a condição com defeito elétrico, podendo variar de 0.50 a 1.0, Figura 5.19. Valores também escolhidos pelo usuário.

É necessário estabelecer esta faixas para cada saída, pois, como dito, a saída da *fuzzy* é convertida num valor numérico para que o computador possa interpretar. Assim, por exemplo, se o sistema *fuzzy* retorna qualquer valor entre 0.30 e 0.70, durante a classificação de uma falha mecânica, isso que dizer que aquela falha que está sendo classificada, apresenta nível intermediário.

Por fim é necessário informar ao sistema *fuzzy* como cada uma das amplitudes das frequências características de falha, agora já transformadas pelas funções de pertinência em B, M ou A

para falhas mecânicas ou em B ou A para falhas elétricas, sejam classificadas como cada uma das diferentes falhas em estudo, para isso utiliza-se um banco de regras.

Na Tabela 5.5 (falhas elétricas) e Tabela 5.6 (falhas mecânicas), têm-se as bases de regras composta por uma coleção de proposições *fuzzy*, apresentadas na forma *if-then*.

Estas bases de regras foram construídas a partir de informações das amplitudes de suas respectivas frequências características observadas para a condição da máquina e testadas para as condições consideradas. Deve-se ressaltar a dependência do especialista para a confecção das regras.

Tabela 5.5: Composição da base de regras para falhas elétricas.

Regra	Entradas						Saídas		
	$2x f_l$	$19x f_l$	$21x f_l$	$1x f_r - 2x s_f$	$1x f_r$	$1x f_r + 2x s_f$	CC	DF	BQ
01	B	B	B	-	-	-	SD	SD	-
02	B	B	A	-	-	-	CD	SD	-
03	B	A	B	-	-	-	SD	SD	-
04	B	A	A	-	-	-	CD	SD	-
05	A	B	B	-	-	-	SD	CD	-
06	A	B	A	-	-	-	CD	CD	-
07	A	A	B	-	-	-	SD	CD	-
08	A	A	A	-	-	-	CD	CD	-
09	-	-	-	B	B	B	-	-	SD
10	-	-	-	B	B	A	-	-	SD
11	-	-	-	B	A	B	-	-	SD
12	-	-	-	B	A	A	-	-	CD
13	-	-	-	A	B	B	-	-	SD
14	-	-	-	A	B	A	-	-	SD
15	-	-	-	A	A	B	-	-	CD
16	-	-	-	A	A	A	-	-	CD

SD - Sem Defeito CD - Com Defeito

Tabela 5.6: Composição da base de regras para falhas mecânicas.

Regra	Entradas			Saídas		
	$1x.f_r$	$2x.f_r$	$3x.f_r$	Desbalanceamento	Desalinhamento	Folga Mecânica
01	B	B	B	SD	SD	SD
02	B	B	M	SD	SD	SD
03	B	B	A	SD	NC	SD
04	B	M	B	SD	NI	SD
05	B	M	M	SD	NI	SD
06	B	M	A	SD	NC	SD
07	B	A	B	SD	NC	SD
08	B	A	M	SD	NC	SD
09	B	A	A	SD	NC	SD
10	M	B	B	NI	SD	SD
11	M	B	M	NI	NI	SD
12	M	B	A	NI	NC	SD
13	M	M	B	SD	SD	NC
14	M	M	M	SD	SD	NC
15	M	M	A	SD	NC	NC
16	M	A	B	SD	NC	SD
17	M	A	M	SD	NC	SD
18	M	A	A	SD	NC	SD
19	A	B	B	NC	SD	SD
20	A	B	M	NC	NI	SD
21	A	B	A	NC	NC	SD
22	A	M	B	NC	NI	SD
23	A	M	M	NC	NI	NI
24	A	M	A	NC	NC	SD
25	A	A	B	NC	NC	SD
26	A	A	M	NC	NC	NI
27	A	A	A	NC	NC	NC

SD - Sem Defeito NI - Nível Intermediário NC - Nível Crítico

5.4.2 Diagnóstico das falhas

Para a implementação do sistema, utilizou-se o *Toolbox* de Lógica *Fuzzy* do Software MATLAB. Na avaliação dos dados de entrada e diagnóstico da condição do motor, utilizou-se o Método de Inferência Mamdani.

O Método de Inferência Mamdani combina os graus de pertinência referentes a cada um dos valores de entrada através do operador mínimo, e agrega as regras através do operador máximo. Para a etapa de defuzzificação, utilizou-se o método do centro de gravidade (centroid), (MAMDANI, 1974).

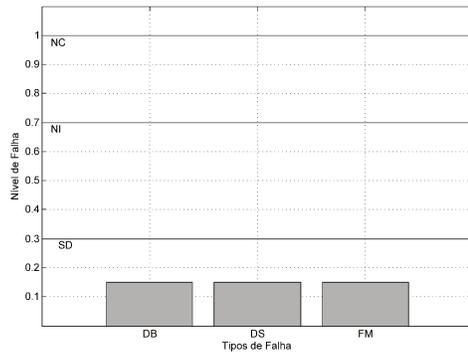
Por exemplo:

Entrando com os dados ($1xf_r = 0.2810$ mm/s, $2xf_r = 0.2925$ mm/s, $3xf_r = 0.0420$ mm/s) no sistema *fuzzy*, o que indica B B B de acordo com as funções e pertinência de entrada para as falhas mecânicas, recai-se que a regra número 1 da Tabela 5.6 foi solicitada e que a saída correspondente para esta situação é considerada sem defeito de origem mecânica, Figura 5.20(a).

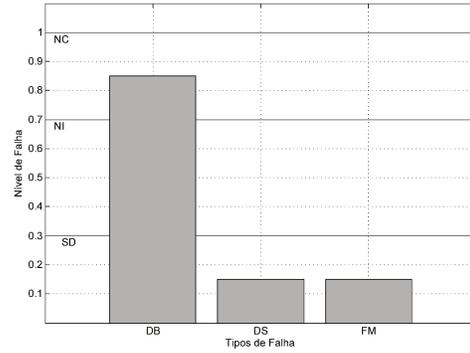
Na Figura 5.20(b), tem-se o resultado do sistema *fuzzy* para uma entrada de dados ($1xf_r = 1.7946$ mm/s, $2xf_r = 0.3254$ mm/s, $3xf_r = 0.1832$ mm/s), o que indica A B B, recai-se que a regra 20 da Tabela 5.6 foi solicitada e que a saída correspondente para esta situação é um nível crítico de desbalanceamento.

Na Figura 5.20(c), tem-se o resultado do sistema *fuzzy* para uma entrada de dados ($1xf_r = 0.4642$ mm/s, $2xf_r = 1.6848$ mm/s, $3xf_r = 0.8204$ mm/s), o que indica M A M, recai-se que a regra número 17 da Tabela 5.6 foi solicitada e que a saída correspondente para esta situação é um nível crítico de desalinhamento.

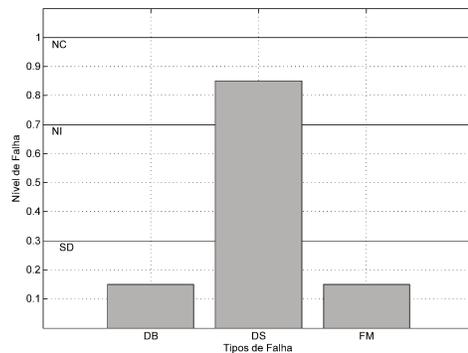
Na Figura 5.20(d), tem-se o resultado do sistema *fuzzy* para uma entrada de dados ($1xf_r = 1.1586$ mm/s, $2xf_r = 0.8842$ mm/s, $3xf_r = 0.7386$ mm/s), o que indica M M M, recai-se que a regra 14 da Tabela 5.6 foi solicitada e que a saída correspondente para esta situação é um nível crítico de folga mecânica.



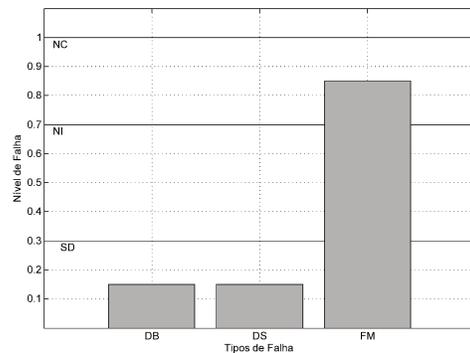
(a) Saída para condição sem defeito.



(b) Saída para o desbalanceamento.



(c) Saída para o desalinhamento.



(d) Saída para a folga mecânica.

Figura 5.20: Saída do sistema *fuzzy* para as falhas mecânicas.

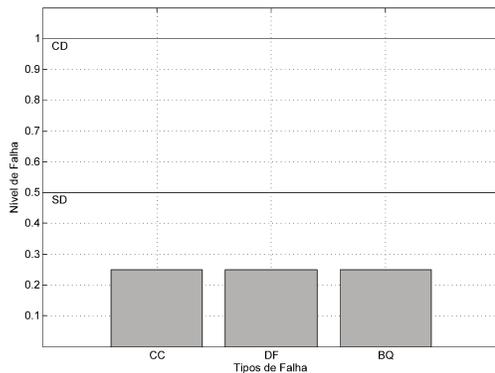
Na Figura 5.21(a), tem-se o resultado do sistema *fuzzy* para uma entrada de dados ($2x_{f_l} = 0.1187$ mm/s, $19x_{f_l} = 0.0386$ mm/s, $21x_{f_l} = 0.0135$ mm/s), o que indica B B B, recai-se que a regra 1 da Tabela 5.5 foi solicitada e que a saída correspondente para esta situação é considerada sem defeito de origem elétrica.

Na Figura 5.21(b), tem-se o resultado do sistema *fuzzy* para uma entrada de dados ($2x_{f_l} = 0.0335$ mm/s, $19x_{f_l} = 0.1388$ mm/s, $21x_{f_l} = 0.2473$ mm/s), o que indica B B A, recai-se que a regra 2 da Tabela 5.5 foi solicitada e que a saída correspondente para esta situação é de falha de curto-circuito.

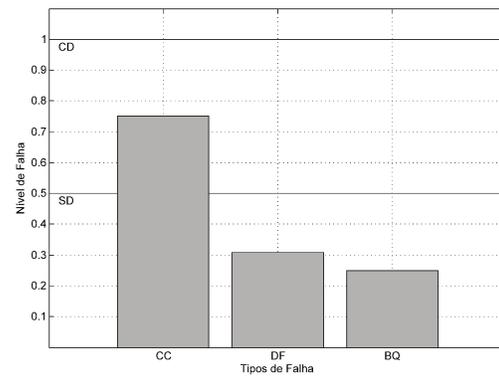
Na Figura 5.21(c), tem-se o resultado do sistema *fuzzy* para uma entrada de dados ($2x_{f_l} =$

0.1707 mm/s, $19x_{f_l} = 0.0303$ mm/s, $21x_{f_l} = 0.0398$ mm/s), o que indica A B B, recai-se que a regra 5 da Tabela 5.5 foi solicitada e que a saída correspondente para esta situação é de falha de desequilíbrio de fase.

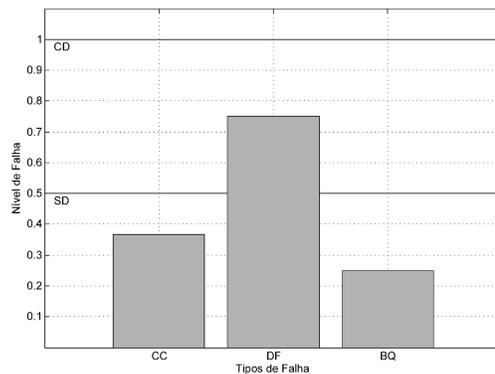
Na Figura 5.21(d), tem-se o resultado do sistema *fuzzy* para uma entrada de dados ($1x_{f_r} - 2x_{s_f} = 0.0453$ mm/s, $1x_{f_r} = 0.3617$ mm/s, $1x_{f_r} + 2x_{s_f} = 0.0625$ mm/s), o que indica A A B, recai-se que a regra 15 da Tabela 5.5 foi solicitada e que a saída correspondente para esta situação é de falha de barras quebradas.



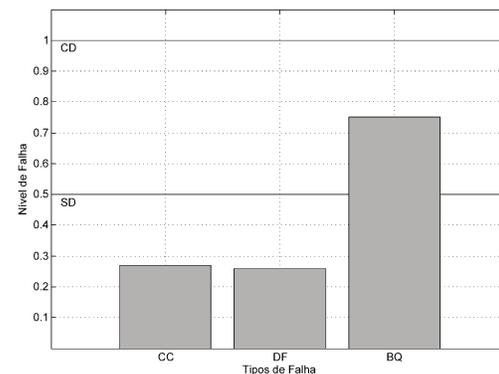
(a) Saída para condição sem defeito.



(b) Saída para o curto-circuito.



(c) Saída para o deseq. de fase.



(d) Saída para as b. quebradas.

Figura 5.21: Saída do sistema *fuzzy* para as falhas elétricas.

Na Tabela 5.7 (falhas mecânicas) e Tabela 5.8 (falhas elétricas), tem-se o percentual de acertos das saídas do sistema *fuzzy* para os conjuntos de dados.

Tabela 5.7: Classificação das falhas mecânicas com a Lógica *Fuzzy*.

Falha	Saída do sistema	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Sem falha	SD	100	100	100
	NI	0	0	0
	NC	0	0	0
Desbalanceamento	SD	4	4	12
	NI	40	36	40
	NC	56	60	48
Desalinhamento	SD	16	16	12
	NI	80	80	84
	NC	4	4	4
Folga mecânica	SD	28	16	20
	NI	20	28	36
	NC	52	56	44

Tabela 5.8: Classificação das falhas elétricas com a Lógica *Fuzzy*.

Falha	Topologia	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Sem falha	SD	100	100	100
	CD	0	0	0
Curto-circuito	SD	0	0	0
	CD	100	100	100
Deseq. de fase	SD	0	0	0
	CD	100	100	100
Barras quebradas	SD	11.11	16.67	22.22
	CD	88.89	83.33	77.78

Observando os resultados das Tabelas 5.7 e 5.8, tem-se que a metodologia utilizada para a lógica *fuzzy* obteve bons resultados. Nas falhas mecânicas, para as falhas de desbalanceamento, desalinhamento e folga mecânica, os valores classificados como sem defeito (SD), representam erros de classificação. Por exemplo, para a falha de desbalanceamento tem-se a classificação SD = 4, NI = 40 e NC = 56 na coluna Acerto c1 da Tabela 5.7. Estes valores representam que de todos os dados entregues na entrada do sistema *fuzzy*, ele classificou 4% dos desbalanceamentos como SD, ou seja, dados classificados erroneamente, pois a falha inicialmente existia mas ficou fora do limite de falha da função de pertinência de entrada. Os outros 40% e 56% são dados classificados corretamente somando um total de 96% de acerto, porém divididos em nível intermediário (NI) e nível crítico (NC), pois foi definido que a saída seria assim. Na condição sem defeito todos os dados foram classificados como SD, o que corresponde a um acerto de 100%. Para as falhas elétricas, as mesmas análises podem ser realizadas, porém com apenas duas saídas, ou existia a falha (CD) ou não (SD).

Mas para que chegasse a estes resultados foi realizada uma análise de todo o banco de dados para que fosse possível definir os limites das funções de pertinência de entrada. Para elaboração do banco de regras de falhas mecânicas, foi necessário estudar o comportamento das frequências determinísticas de cada falha e assim definir como cada amplitude influenciaria o aparecimento ou não de cada falha. Já para as falhas elétricas, há poucos estudos sobre o comportamento das mesmas no espectro de vibrações, pois, normalmente seu diagnóstico é feito usando a técnica de análise de corrente e/ou fluxo magnético. Realizou-se então um estudo sobre os espectros para que chegasse a uma conclusão de como cada frequência influenciaria no aparecimento ou não de cada falha elétrica, elaborando assim a Tabela 5.5. As funções de pertinência foram criadas através do comparativo dos espectros de falha com os espectros da condição sem defeito. Pelos resultados pode-se concluir que é possível detectar tanto falhas mecânicas quanto falhas elétricas com a lógica *fuzzy* utilizando a análise de vibrações.

Contudo, ressalta-se a dificuldade de se definir as regras e a modelagem da funções de pertinência. Uma modelagem das funções um pouco fora dos limites de falha e regras não tão precisas fazem com que alguns dados sejam classificados de maneira errada, o que mostra a dependência da lógica *fuzzy* com um especialista que possua conhecimentos específicos de cada falha, para que se consiga regras e funções de pertinência cada vez mais precisas.

5.5 Classificação de falhas em diferentes níveis de severidade

Foi mostrado até agora a utilização da SVM, RNA e Lógica *Fuzzy*, para a classificação de falhas, porém, os níveis de severidade de falha dos parâmetros de entrada, durante a fase de treinamento e verificação, sempre foram os mesmos (nível 1). Mas em situações reais os equipamentos apresentam diversos níveis de falha e o ponto de coleta pode variar devido às restrições de acesso ao equipamento. Segundo Baccarini *et al.* (2011), uma SVM treinada com o sinal obtido de um acelerômetro em determinada posição não terá o mesmo acerto caso a posição dos mesmos mude em uma nova verificação.

Tem-se ainda que, se a SVM ou RNA for treinada com o nível 1 de severidade de uma falha, ela não será capaz de classificar corretamente a mesma falha caso ela esteja em um nível 2 de severidade.

Com os modelos de SVM e RNA já treinados para classificar as falhas no nível 1, decidiu-se classificar o nível 2 com os mesmos modelos. A intenção desta etapa é: o usuário treinou sua SVM ou RNA com os dados que tinha disponível, e agora vai utilizá-la para classificar um novo dado recebido, só que este dado está em outro nível. Na Tabela 5.9 e Tabela 5.10, têm-se a os resultados da classificação da SVM e RNA respectivamente para as falhas mecânicas, em diferentes níveis.

Tabela 5.9: Classificação do nível 2 através da SVM treinada com o nível 1 para as falhas mecânicas.

Falha	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Desbalanceamento	44.00	80.00	48.00
Desalinhamento	72.00	32.00	28.00
Folga mecânica	60.00	16.00	74.00
Parâmetros (C, γ)	(128,0.250)	(64,0.500)	(2048,3.125 e^{-2})

Tabela 5.10: Classificação do nível 2 através da RNA treinada com o nível 1 para as falhas mecânicas.

Falha	Topologia	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Desbalanceamento	4x3x1	68.40	68.37	73.59
	4x5x1	87.20	81.52	69.59
	4x10x1	79.68	79.27	64.67
Desalinhamento	4x3x1	22.96	51.24	36.18
	4x5x1	49.91	56.94	43.49
	4x10x1	56.97	39.27	35.21
Folga mecânica	4x3x1	64.06	42.41	68.88
	4x5x1	79.47	72.60	83.12
	4x10x1	64.97	69.46	66.32

Na Tabela 5.11 e Tabela 5.12, têm-se a os resultados da classificação da SVM e RNA para as falhas elétricas respectivamente, em diferentes níveis.

Tabela 5.11: Classificação do nível 2 através da SVM treinada com o nível 1 para as falhas elétricas.

Falha	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Curto-circuito	66.67	77.78	88.89
Deseq. de fase	0	0	22.22
Barras quebradas	0	11.11	44.44
Parâmetros (C, γ)	(32,8)	(512,8)	(64,8)

Tabela 5.12: Classificação do nível 2 através da RNA treinada com o nível 1 para as falhas elétricas.

Falha	Topologia	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Curto-circuito	3x3x1	91.80	83.01	89.31
	3x5x1	76.32	85.42	86.54
	3x10x1	69.67	61.03	89.84
Deseq. de fase	3x3x1	97.48	95.26	97.66
	3x5x1	97.07	95.18	80.73
	3x10x1	97.17	97.99	98.00
Barras quebradas	3x3x1	83.91	97.33	97.81
	3x5x1	78.95	97.07	78.87
	3x10x1	80.67	41.047	98.90

Pode-se observar pelas Tabelas 5.9 e 5.11, que quando se tem uma SVM treinada para falhas em um nível 1 de severidade e classifica-se as mesmas falhas treinadas, porém um outro nível de severidade as taxas de acerto caem consideravelmente, tanto para as falhas mecânicas quanto para as falhas elétricas. O mesmo pode-se dizer para a classificação de falhas mecânicas utilizando a RNA (Tabela 5.10), já para as falhas elétricas a rede neural conseguiu lidar com essa diferença de níveis.

Quando se utilizam falhas no nível 2 na lógica *fuzzy*, a classificação não é possível, pois, as funções de pertinência foram criadas de acordo com as amplitudes das falhas do nível 1. Então para lidar com o problema de classificação com mais de um nível de severidade com a lógica *fuzzy*, deve-se recriar as funções de pertinência. Este procedimento será comentado no item 5.5.2.

5.5.1 Normalização

Devido às baixas taxas de acerto quando se treina a SVM ou RNA com o nível 1 de uma falha e classifica-se o nível 2, o presente trabalho propõe uma normalização para lidar com este problema.

A normalização consiste em:

- Falhas mecânicas:
 - Normalizar as amplitudes das frequências 2 , 3 e $4x f_r$ em relação a amplitude de $1x f_r$.
- Falhas elétricas:
 - Curto-circuito: Normalizar as amplitudes das frequências 2 e $19x f_l$ em relação a amplitude de $21x f_l$.
 - Desequilíbrio de fase: Normalizar as amplitudes das frequências 19 e $21x f_l$ em relação a amplitude de $2x f_l$.
 - Barras quebradas: Normalizar as amplitudes das frequências $\pm 2x s_f$ em relação a amplitude de $1x f_r$.

Com este procedimento, é possível, por exemplo, treinar a SVM com uma falha em nível 1 de severidade e classificar uma falha em um nível 2 de severidade, com melhores taxas de acerto, sendo esses níveis os definidos no item 5.1.8.

5.5.2 Classificação com a normalização

Nesta etapa foi utilizada a normalização proposta, para tentar classificar falhas no nível 2 de severidade, assumindo-se que só tenha disponível dados no nível 1 para o treinamento. O processo consiste em normalizar os dados de entrada no nível 1 e então treinar a SVM ou a RNA, em seguida normaliza-se os dados no nível 2 e classifica-os.

Na Tabela 5.13 e Tabela 5.14, têm-se a os resultados da classificação da SVM e RNA para as falhas mecânicas respectivamente, ambos com a normalização.

Tabela 5.13: Classificação do nível 2 de severidade com a SVM treinada com o nível 1 para as falhas mecânicas com a normalização.

Falha	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Desbalanceamento	100	100	100
Desalinhamento	100	100	100
Folga mecânica	72.00	76.00	92.00
Parâmetros (C, γ)	(4,8)	(16,8)	(2,8)

Tabela 5.14: Classificação do nível 2 de severidade com a RNA treinada com o nível 1 para as falhas mecânicas com a normalização.

Falha	Topologia	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Desbalanceamento	4x3x1	88.61	86.39	87.59
	4x5x1	77.53	85.40	85.60
	4x10x1	85.31	81.87	91.19
Desalinhamento	4x3x1	75.69	81.90	66.15
	4x5x1	81.97	85.32	72.40
	4x10x1	84.02	83.14	77.26
Folga mecânica	4x3x1	33.09	57.93	49.25
	4x5x1	47.97	73.13	33.62
	4x10x1	46.86	82.29	31.25

Na Tabela 5.15 e Tabela 5.16, têm-se a os resultados da classificação da SVM e RNA para as falhas elétricas respectivamente, ambos com a normalização.

Tabela 5.15: Classificação do nível 2 de severidade com a SVM treinada com o nível 1 para as falhas elétricas com a normalização.

Falha	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Curto-circuito	100	100	100
Deseq. de fase	88.89	88.89	100
Barras quebradas	100	100	100
Parâmetros (C, γ)	(4,8)	$(3.125e^{-2}, 3.052e^{-5})$	(8,8)

Tabela 5.16: Classificação do nível 2 de severidade com a RNA treinada com o nível 1 para as falhas elétricas com a normalização.

Falha	Topologia	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Curto-circuito	3x3x1	91.92	98.83	99.85
	3x5x1	99.10	83.73	96.26
	3x10x1	84.89	75.67	91.30
Deseq. de fase	3x3x1	90.72	98.08	89.26
	3x5x1	94.52	82.80	99.60
	3x10x1	99.05	99.80	93.56
Barras quebradas	3x3x1	90.59	99.63	99.83
	3x5x1	92.76	81.91	99.98
	3x10x1	99.30	99.94	91.30

Na classificação das falhas com o nível 2 de severidade com a lógica *fuzzy*, as funções de pertinência tiveram de serem recriadas. Para que as funções de pertinência ficassem mais genéricas na classificação de falhas mecânicas utilizou-se a faixa de severidade da norma VDI 2056 (ver ítem 5.1.8). Desta forma a lógica *fuzzy* poderá lidar com ambos os níveis.

Na Figura 5.22, tem-se as funções de pertinência para as entradas ($1x f_r$, $2x f_r$ e $3x f_r$) das

falhas mecânicas segundo a norma VDI 2056.

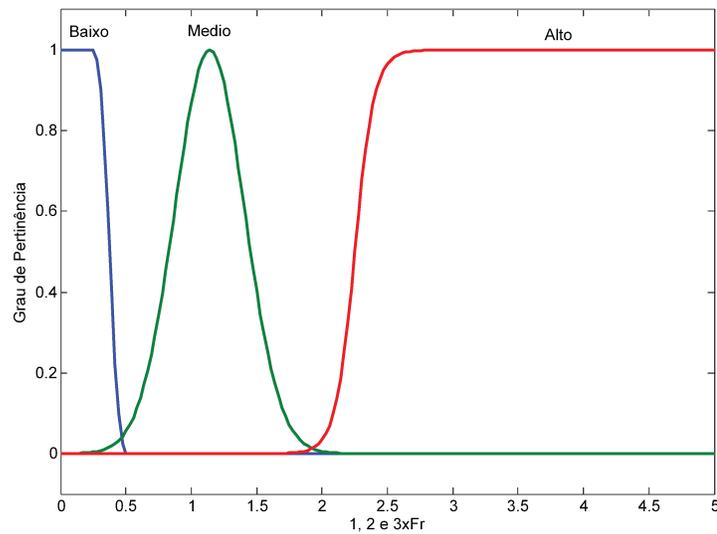


Figura 5.22: Função de pertinência de entrada para as falhas mecânicas segundo a norma VDI 2056.

Na Tabela 5.17, tem-se o percentual de acertos das saídas do sistema *fuzzy* para os conjuntos de dados mecânicos nos níveis 1 e 2.

Tabela 5.17: Classificação das falhas mecânicas nos níveis 1 e 2 com a Lógica *Fuzzy*.

Falha	Saída do sistema	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Desbalanceamento	SD	2	2	4
	NI	50	48	54
	NC	48	50	42
Desalinhamento	SD	16	12	14
	NI	70	68	68
	NC	14	20	18
Folga mecânica	SD	28	24	24
	NI	22	20	20
	NC	50	56	52

Na classificação das falhas elétricas, as mesmas funções de pertinência foram adotadas, Figura 5.17, pois, não havendo uma norma que tratasse de falhas com níveis de vibração tão baixos, as funções de pertinência já foram criadas com base na análise do todo o conjunto de falhas elétricas.

Na Tabela 5.18, tem-se o percentual de acertos das saídas do sistema *fuzzy* para os conjuntos de dados elétricos nos níveis 1 e 2.

Tabela 5.18: Classificação das falhas elétricas nos níveis 1 e 2 com a Lógica *Fuzzy*.

Falha	Topologia	Acerto c1 (%)	Acerto c2 (%)	Acerto c3 (%)
Curto-circuito	SD	0	0	0
	CD	100	100	100
Deseq. de fase	SD	0	0	0
	CD	100	100	100
Barras quebradas	SD	5.56	5.56	11.11
	CD	94.44	94.44	88.89

Em relação às SVMs, a normalização conseguiu contornar o problema da classificação de outros níveis de severidade melhorando as taxas de acerto, tanto para falhas mecânicas quanto para as falhas elétricas.

As RNAs não sofreram tanta influência quanto a SVM devido aos diferentes níveis para treinamento e classificação. Porém, com a normalização foi possível obter melhores taxas de acerto, mas inferiores aos obtidos com a SVM com dados normalizados. É importante salientar que no item 5.3.2, foi visto que a topologia de rede que apresentou melhores resultados para a classificação de falhas mecânicas com a RNA foi a 4x5x1, já neste processo de normalizar para melhorar as taxas de acerto a topologia que melhor apresentou resultados foi a 4x10x1, reforçando então, a dificuldade de se escolher a topologia adequada. Para as falhas elétricas a topologia que apresentou melhores resultados se manteve a mesma, que foi a 3x3x1.

Por fim, para classificar diferentes níveis com a lógica *fuzzy* novas funções de pertinência tiveram de ser criadas, uma vez que a normalização não se aplica, pois, a normalização mudaria as amplitudes das frequências de entrada, podendo, então, deixar os dados fora da faixa das funções

de pertinência. O que mostra mais uma vez a necessidade de um especialista para a criação das funções de pertinência, pois, o acerto final do sistema *fuzzy* dependerá delas e do banco de regras.

Como entrada do sistema *fuzzy*, utilizou-se apenas três frequências características de falhas mecânicas, ao contrário da SVM e da RNA que foram utilizadas quatro. Optou-se por utilizar apenas três, pois, a quantidade de entradas influenciaria no tamanho do banco de regras, que ao invés de 27 regras para cada um dos defeitos, seriam necessárias 81 regras para cada defeito. Assim, apesar dos bons resultados da lógica *fuzzy*, deve-se ressaltar que um sistema *fuzzy* mais completo ficaria muito mais complexo de se elaborar.

À respeito da normalização proposta, diz-se que, apesar dos bons resultados que ela propiciou, esta é ainda uma proposta inicial, envolvendo uma análise linear, e que precisa de mais estudo, principalmente quando se trabalhar com diferentes máquinas e diferentes falhas.

5.6 Considerações finais

Neste capítulo foi apresentada a bancada e descritos os procedimentos experimentais para obtenção dos sinais de vibração. Foi mostrado também a metodologia para utilização das máquinas de vetores de suporte, redes neurais artificiais e lógica *fuzzy* para classificação de falhas mecânicas e elétricas. Para lidar com problemas de classificação de falhas em outros níveis de severidade foi proposta uma normalização, e a sua utilização permitiu que a SVM e a RNA conseguissem classificar as falhas com melhores taxas de acerto, porém, mais uma vez, a SVM apresentou melhor capacidade de generalização do que a RNA.

6 CONCLUSÃO

Esta dissertação apresentou um estudo da detecção e diagnósticos de falhas mecânicas e elétricas em motores de indução trifásicos utilizando técnicas de inteligência artificial. Foram estudadas as principais técnicas de IA, suas aplicações, vantagens e desvantagens.

A revisão bibliográfica à respeito de detecção de falhas através da análise de vibrações, serviu para o conhecimento das principais falhas e de como detectá-las. Sendo a análise de vibrações uma das principais técnicas utilizadas no meio industrial, devido ao seu baixo custo, facilidade de utilização e por não ser intrusiva.

Os dados experimentais utilizados neste trabalho foram importantes, pois, tornou possível o treinamento das técnicas de inteligência artificial com dados de falhas reais, aproximando mais a utilização da IA para o meio industrial. A utilidade destas técnicas está em que algumas das falhas apresentadas pelo motor elétrico podem, muitas vezes, serem de difícil detecção ou necessitem de rápido diagnóstico, como é o caso da falha de curto-circuito entre espiras que é considerada uma falha severa, pois sua evolução é rápida, o que sugere que o sistema de diagnóstico deve detectar falhas em um estágio inicial.

Uma seleção de parâmetros foi realizada com o intuito de se obter os melhores parâmetros de treinamento, e como esperado, os melhores parâmetros encontrados pelo seletor foram também os apresentados na teoria. Para diagnóstico de falhas mecânicas adotou-se componentes da frequência de rotação do motor e para falha de origem elétrica adotou-se componentes da frequência de linha (60Hz).

Metodologias utilizadas para se obter um bom desempenho das técnicas de inteligência artificial foram apresentadas no Capítulo 5. Os procedimentos foram aplicados e classificou-se as falhas em estudo. Os resultados mostraram a viabilidade das técnicas de inteligência artificial.

A lógica *fuzzy* é uma técnica que possibilita modelar e manipular matematicamente informações vagas e imprecisas, porém, um conhecimento específico do que se deseja modelar é indispensável. Como mostrado, o banco de regras e as funções de pertinência são fatores de suma

importância para a obtenção de bons resultados. A tarefa de montar o banco de regras passa a ficar cada vez mais complicada à medida que se for adicionando mais parâmetros de entrada ou, por exemplo, cruzando-se informações de diferentes sensores. Neste trabalho utilizou-se apenas um sensor de vibração, mas em outras pesquisas poderia também querer utilizar, por exemplo, a análise de corrente do motor, pois, talvez teria um diagnóstico mais preciso, neste caso um conhecimento mais profundo ainda seria necessário para a confecção do banco de regras.

Já as máquinas de vetores de suporte e as redes neurais artificiais, não exigem este conhecimento prévio do problema. A rede neural apresentou bons resultados, mas como visto, a RNA apresenta vários parâmetros de treinamento que o usuário deve definir, afim de que a rede consiga "aprender". A SVM treinada utilizando o kernel RBF, apresenta apenas dois parâmetros de treinamento (C, γ) , e que podem ser obtidos automaticamente através de uma validação cruzada juntamente com uma pesquisa em grade, encontrando assim o melhor par de (C, γ) , evitando-se uma possível má generalização.

A SVM e RNA diminuem sua capacidade de generalização, quando são apresentadas, para a classificação, as mesmas falhas treinadas mas em diferentes níveis de severidade. Tentando resolver este problema, este trabalho propôs uma normalização, e que após sua utilização bons resultados podem ser obtidos. Porém, a SVM apresentou melhor capacidade de generalização. Para a lógica *fuzzy* lidar com este problema de diferentes níveis de severidade, novas funções de pertinência tiveram de ser criadas.

Para detecção de falhas elétricas com as tradicionais técnicas de manutenção preditiva é necessário retirar o motor de operação para que se possam realizar os testes de curto-circuito e barras quebradas, sendo que para a detecção do desequilíbrio de fase é necessário o acesso a cabos energizados e que na maioria das vezes não estão disponíveis para medição, sendo um serviço altamente perigoso para o operador.

Pensando nisso, a metodologia apresentada possibilita utilizar as SVMs em aplicações práticas como uma forma de monitoramento *on-line* por um operador sem muitos conhecimentos de análise e detecção de falhas. Os procedimentos adotados fizeram a SVM treinada capaz de caracterizar falhas mecânicas de falhas elétricas com a utilização de apenas um sensor. E a normalização proposta permite que a SVM treinada classifique, com uma melhor taxa de acerto, falhas em dife-

rentes níveis de severidade.

Em suma, destaca-se que o desenvolvimento experimental de técnicas de inteligência artificial, combinadas com as tradicionais técnicas de manutenção preditiva, proporciona detecção e diagnóstico de falhas quase que imediato para o equipamento monitorado, além de consistir de uma ferramenta de baixo custo, e com a sua constante implementação e desenvolvimento, poderá ser, no futuro a saída mais viável para uma manutenção industrial eficiente e eficaz.

Como contribuição desta dissertação, destacam-se os seguintes pontos principais:

- Utilização de apenas um sensor para detectar e caracterizar falhas elétricas e falhas mecânicas.
- Proposta de normalização para lidar com o problema da classificação de falhas em diferentes níveis de severidade.
- Métodos para a aplicação da SVM como forma de monitoramento *on-line*.
- Estudo comparativo entre as principais técnicas de inteligência artificial.

Para trabalhos futuros propõe-se:

- Detectar falhas através da fusão de sinais de diferentes sensores.
- Introduzir novas excitações na bancada experimental, como o problema de rolamento.
- Estender o estudo para diferentes grupos de máquinas.
- Utilização de um banco de falhas teórico, com a possibilidade de inserir novos dados de falhas medidas, e assim ter um sistema adaptativo que contorne a dificuldade da vivência da falha para o treinamento das técnicas de inteligência artificial.

REFERÊNCIAS

ALMEIDA, M.T. e GÓZ, R.D.S. Curso de vibrações em motores. Relatório técnico, FUPAI, Itajubá, MG, 1994.

AYDIN, I.; KARAKOSE, M. e AKIN, E. Artificial immune based support vector machine algorithm for fault diagnosis of induction motors. In **Electrical Machines and Power Electronics, 2007. ACEMP '07. International Aegean Conference on**, pp. 217 –221. sept. 2007.

BACCARINI, Lane Maria Rabelo. **Deteção e diagnóstico de falhas em motores de indução**. 2005. Tese de Doutorado. Faculdade de Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte.

BACCARINI, L.M.R.; SILVA, V.V.R.; MENEZES, B.R. e CAMINHAS, W.M. SVM practical industrial application for mechanical faults diagnostic. **Expert Systems with Applications**, Vol. 38, nº 6, 6980 – 6984, 2011.

BATUR, C.; ZHOU, L. e CHAN, C.C. Support vector machines for fault detection. In **Decision and Control, 2002, Proceedings of the 41st IEEE Conference on**, Vol. 2, pp. 1355 – 1356 vol.2. dec. 2002.

BEALE, R. e JACKSON, T. **Neural computing: An introduction**. IOP Publishing Ltd., Bristol, UK, 1990. ISBN 0-85274-262-2.

BOTTOU, L.; CORTES, C.; DENKER, J.S.; DRUCKER, H.; GUYON, I.; JACKEL, L.D.; LECUN, Y.; MULLER, U.A.; SACKINGER, E.; SIMARD, P. e VAPNIK, V. Comparison of classifier methods: a case study in handwritten digit recognition. **Pattern Recognition, 1994. Vol. 2 - Confe-**

rence B: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on, Vol. 2, 77–82, outubro 1994.

BRITO, Jorge Nei. **Desenvolvimento de um sistema inteligente híbrido para diagnóstico de falhas em motores de indução trifásicos**. 2002. Tese de Doutorado. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas.

BRITO, J.N. e PEDERIVA, R. Using artificial intelligence tools to detect problems in induction motors. In **Proceedings of the 1st International Conference on Soft Computing and Intelligent Systems (International Session of 8th SOFT Fuzzy Systems Symposium) and 3rd International Symposium on Advanced Intelligent Systems (SCIS & ISIS 2002), Suzuka, Japão**, Vol. 1. 2002.

CAMARGOS, F.L. **Lógica nebulosa: Uma abordagem filosófica e aplicada**. Relatório técnico, Universidade Federal de Santa Catarina, 2002.

CAMPBELL, C. e YING, Y. **Learning with support vector machines**. Morgan & Claypool Publishers series, 2011.

CAPORALLI FILHO, Angelo. **Sistema especialista para o forjamento a quente de precisão**. 2003. Tese de Doutorado. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas.

CHANG, C.C. e LIN, C.J. LIBSVM: A library for support vector machines. **ACM Transactions on Intelligent Systems and Technology**, Vol. 2, 27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

CHARNIAK, E. e MCDERMOTT, D. **Introduction to artificial intelligence**. Addison-Wesley Publishing Company, 1985.

CHEN, X. e HUANG, H. Immune feedforward neural network for fault detection. **Tsinghua Science & Technology**, Vol. 16, nº 3, 272 – 277, 2011.

CHUNG, D.T.; MODARRES, M. e HUNT, R.M. Gotres: An expert system for fault detection and analysis. **Reliability Engineering & System Safety**, Vol. 24, nº 2, 113 – 137, 1989.

CORTES, C. e VAPNIK, V. Support-vector networks. **Machine Learning**, Vol. 20, 273–297, September 1995.

COVER, T.M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. **Electronic Computers, IEEE Transactions on**, Vol. EC-14, nº 3, 326 –334, june 1965.

COX, E. **The fuzzy systems handbook: a practitioner’s guide to building, using, and maintaining fuzzy systems**. Academic Press Professional, Inc., San Diego, CA, USA, 1994. ISBN 0-12-194270-8.

CRISTIANINI, N. e SHAWE-TAYLOR, J. **An introduction to support vector machines and other kernel-based learning methods**. Cambridge University Press, United Kingdon, 2000.

CRUZ, J.A.O. Lógica nebulosa. Relatório técnico, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2002.

FERNANDES, M.C. e SANTOS, R.H. Lógica nebulosa x lógica paraconsistente. Relatório técnico, Universidade Federal de São Carlos, 2002.

FRANK, P.M. e KÖPPEN-SELIGER, B. Fuzzy logic and neural network applications to fault diagnosis. **International Journal of Approximate Reasoning**, Vol. 16, nº 1, 67 – 88, 1997.

FREEMAN, J.A. e SKAPURA, D.M. **Neural networks: Algorithms, applications, and pro-**

gramming techniques. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1991. ISBN 0-201-51376-5.

FRIEDMAN, J.H. Another approach to polychotomous classification. Relatório técnico, Department of Statistics and Stanford Linear Accelerator Center, Stanford University, 1996.

GHATE, V.N. e DUDUL, S.V. Induction machine fault detection using support vector machine based classifier. **WTOS**, Vol. 8, 591–603, May 2009.

GIRDHAR, P. e SCHEFFER, C. **Practical machinery vibration analysis and predictive maintenance.** Elsevier, 2004.

GOONATILAKE, S.; KHEBBAL, S. Intelligent hybrid systems. **Proceedings of First International Conference on Intelligent Systems**, 1992.

GUNN, S.R. Support vector machines for classification and regression. Relatório técnico, Dept. of Electronics and Computer Science, University of Southampton, 1998.

GUYON, I.; GUNN, S.; NIKRAVESH, M. e ZADEH, L.A. **Feature extraction: Foundations and applications (Studies in fuzziness and soft computing).** Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 3540354875.

HASTIE, T. e TIBSHIRANI, R. Classification by pairwise coupling. In **Proceedings of the 1997 conference on Advances in neural information processing systems 10**, NIPS '97, pp. 507–513. MIT Press, Cambridge, MA, USA, 1998. ISBN 0-262-10076-2.

HAYKIN, S. **Neural networks: A comprehensive foundation.** Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd ed., 1998. ISBN 0132733501.

HERBRICH, R. **Learning kernel classifiers: Theory and algorithms (Adaptive computation**

and machine learning). The MIT Press, dezembro 2001. ISBN 026208306X.

HORNIK, K.; STINCHCOMBE, M. e WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Netw.**, Vol. 2, 359–366, July 1989.

HSU, C.W.; CHANG, C.C. e LIN, C.J. **A practical guide to support vector classification**. Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2003.

HSU, C.W. e LIN, C.J. A comparison of methods for multiclass support vector machines. **IEEE Transactions on Neural Networks**, Vol. 13, nº 2002, 415–425, 2002.

JACK, L. Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms. **Mechanical Systems and Signal Processing**, Vol. 16, nº 2-3, 373–390, março 2002.

JANG, J.R. Adaptive-network based fuzzy inference systems. **IEEE Transactions on Systems, Man and Cybernetics**, Vol. 23, nº 3, 665–685, 1993.

KNERR, S.; PERSONNAZ, L. e DREYFUS, G. Single-layer learning revisited: A stepwise procedure for building and training a neural network. **NATO ASI Series Springer**, 1990.

KOWALSKI, C.T. e ORLOWSKA-KOWALSKA, T. Neural networks application for induction motor faults diagnosis. **Mathematics and Computers in Simulation**, Vol. 63, nº 3-5, 435 – 448, 2003.

KUREK, J. e OSOWSKI, S. Support vector machine for fault diagnosis of the broken rotor bars of squirrel-cage induction motor. **Neural Comput. Appl.**, Vol. 19, 557–564, June 2010.

LAMIM FILHO, Paulo César Monteiro. **Monitoramento permanente de motores de indução**

trifásicos. 2007. Tese de Doutorado. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas.

LIMA, Clodoaldo Aparecido Moraes. **Comitê de máquinas: Uma abordagem unificada empregando máquinas de vetores-suporte**. 2004. Tese de Doutorado. Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas.

LIPPMANN, R.P. An introduction to computing with neural nets. **SIGARCH Comput. Archit. News**, Vol. 16, 7–25, March 1988.

LORENA, Ana Carolina. **Investigação de estratégias para a geração de máquinas de vetores de suporte multiclases**. 2006. Tese de Doutorado. Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo.

MAMDANI, E. Application of fuzzy algorithms for control of simple dynamic plant. **Electrical Engineers, Proceedings of the Institution of**, Vol. 121, nº 12, 1585–1588, december 1974.

MCCARTHY, J. e HAYES, P.J. **Some philosophical problems from the standpoint of artificial intelligence**, pp. 26–45. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987. ISBN 0-934613-45-1.

MCCULLOCH, W.S. e PITTS, W. **A logical calculus of the ideas immanent in nervous activity**, pp. 15–27. MIT Press, Cambridge, MA, USA, 1988. ISBN 0-262-01097-6.

MIROWSKI, P. Mccorduck's machines who think after twenty-five years: revisiting the origins of ai. **AI Mag.**, Vol. 24, 135–138, January 2004.

MOBLEY, R. e KEITH, R.K. **Vibration Fundamentals**. Butterworth-Heinemann, 1999.

MOUBRAY, J. **Reliability-centered maintenance**. Industrial Press Inc., 2 ed., 2000.

NANDI, S. e TOLIYAT, H. Novel frequency domain based technique to detect incipient stator inter-turn faults in induction machines. In **Industry Applications Conference, 2000. Conference Record of the 2000 IEEE**, Vol. 1, pp. 367–374. 2000.

PLATT, J.C.; CRISTIANINI, N. e SHAWE-TAYLOR, J. Large margin dags for multiclass classification. In **Advances in Neural Information Processing Systems**, pp. 547–553. MIT Press, 2000.

POYHONEN, S.; NEGREA, M.; ARKKIO, A.; HYOTYNIEMI, H. e KOIVO, H. Support vector classification for fault diagnostics of an electrical machine. In **Signal Processing, 2002 6th International Conference on**, Vol. 2, pp. 1719 – 1722 vol.2. aug. 2002.

PRICE, K. Review of "the handbook of artificial intelligence, volume 2 by a. barr & e. a. feigenbaum, eds., volume 3 by p. r. cohen & e. a. feigenbaum, eds.". **SIGART Bull**, pp. 13–14, January 1983.

QU, J. e ZUO, M.J. Support vector machine based data processing algorithm for wear degree classification of slurry pump systems. **Measurement**, Vol. 43, nº 6, 781 – 791, 2010.

RODRÍGUEZ, P.V.J. e ARKKIO, A. Detection of stator winding fault in induction motor using fuzzy logic. **Applied Soft Computing**, Vol. 8, nº 2, 1112 – 1120, 2008.

RUMELHART, D.E.; HINTON, G.E. e WILLIAMS, R.J. **Learning internal representations by error propagation**, pp. 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 0-262-68053-X.

SAMANTA, B.; AL-BALUSHI, K. e AL-ARAIMI, S. Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection. **Engineering Applications of Artificial Intelligence**, Vol. 16, nº 7-8, 657 – 665, 2003.

SANDRI, S. e CORREA, C. Lógica nebulosa. Relatório técnico, V Escola de Redes Neurais,

Conselho Nacional de Redes Neurais, ITA, 1999.

SCHMIDT, M. e GISH, H. Speaker identification via support vector classifiers. In **Acoustics, Speech, and Signal Processing, ICASSP-96. Conference Proceedings, IEEE International Conference on**, Vol. 1, pp. 105–108. may 1996.

SCHOLKOPF, B. e SMOLA, A.J. **Learning with kernels: Support vector machines, regularization, optimization, and beyond**. The MIT Press, Cambridge, Massachusetts, London, England, 2002.

THOMSON, W. e FENGER, M. Current signature analysis to detect induction motor faults. **Industry Applications Magazine, IEEE**, Vol. 7, n^o 4, 26–34, jul/aug 2001.

TORRES, J.; MUÑOZ, M.; MARRO, J. e GARRIDO, P. Influence of topology on the performance of a neural network. **Neurocomputing**, Vol. 58–60, n^o 0, 229–234, 2004.

VAPNIK, V.; GOLOWICH, S.E. e SMOLA, A. Support vector method for function approximation, regression estimation, and signal processing. In **Advances in Neural Information Processing Systems 9**, pp. 281–287. MIT Press, 1996.

VAPNIK, V.N. **The nature of statistical learning theory**. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8.

VAPNIK, V.N. e CHERVONENKIS, A.Y. On the uniform convergence of relative frequencies of events to their probabilities. **Theory of Probability and its Applications**, Vol. 16, n^o 2, 264–280, 1971.

VDI, VEREIN DEUTSCHER INGENIEURE. **VDI 2056: Criteria for Assessing Mechanical Vibrations of Machines**. 1964. p. 12.

WANG, T.Y. e CHIANG, H.M. One-against-one fuzzy support vector machine classifier: An approach to text categorization. **Expert Systems with Applications**, Vol. 36, nº 6, 10030 – 10034, 2009.

WANG, W.; MEN, C. e LU, W. Online prediction model based on support vector machine. **Neurocomput.**, Vol. 71, 550–558, January 2008.

WANG, Z.; MASSIMO, C.D.; THAM, M.T. e MORRIS, A.J. A procedure for determining the topology of multilayer feedforward neural networks. **Neural Networks**, Vol. 7, nº 2, 291–300, 1994.

WERBOS, P. Backpropagation: past and future. In **Neural Networks, IEEE International Conference on**, pp. 343 –353 vol.1. jul 1988.

WIDODO, A. e YANG, B.S. Application of nonlinear feature extraction and support vector machines for fault diagnosis of induction motors. **Expert Syst. Appl.**, Vol. 33, 241–250, July 2007a.

WIDODO, A. e YANG, B.S. Support vector machine in machine condition monitoring and fault diagnosis. **Mechanical Systems and Signal Processing**, Vol. 21, nº 6, 2560 – 2574, 2007b.

WINSTON, P.H. Artificial intelligence research at the artificial intelligence laboratory, Massachusetts Institute of Technology. **AI Magazine**, pp. 44–48, 1983.

YANG, B.S.; LIM, D.S. e TAN, A.C.C. Vibex: an expert system for vibration fault diagnosis of rotating machinery using decision tree and decision table. **Expert Systems with Applications**, Vol. 28, nº 4, 735 – 742, 2005.

ZADEH, L.A. Fuzzy sets. **Information and Control**, Vol. 8, 338–353, 1965.

ZIMMERMANN, H.J. **Fuzzy set theory - and its applications**. Kluwer Academic Publishers,

Norwell, MA, USA, 3rd ed., 1996. ISBN 0-7923-9624-3.