



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

David Nascimento Coelho

**SOBRE O DESEMPENHO DE ALGORITMOS DE
APRENDIZADO DE MÁQUINAS NA DETECÇÃO DE
FALHAS EM MOTORES DE INDUÇÃO TRIFÁSICOS: UM
ESTUDO COMPARATIVO**

Fortaleza

2015

David Nascimento Coelho

**SOBRE O DESEMPENHO DE ALGORITMOS DE
APRENDIZADO DE MÁQUINAS NA DETECÇÃO DE
FALHAS EM MOTORES DE INDUÇÃO TRIFÁSICOS: UM
ESTUDO COMPARATIVO**

Dissertação apresentada à coordenação do Programa de Pós-Graduação em Engenharia de Teleinformática, da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Engenharia de Teleinformática.

Área de concentração: Sinais e Sistemas.

Orientador: Prof. Dr. Guilherme de Alencar Barreto

Fortaleza

2015

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca de Pós-Graduação em Engenharia - BPGE

-
- C615s Coelho, David Nascimento.
Sobre o desempenho de algoritmos de aprendizado de máquinas na detecção de falhas em motores de indução trifásicos: um estudo comparativo / David Nascimento Coelho. – 2015.
97 f. : il., enc. ; 30 cm.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Departamento de Engenharia de Teleinformática, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2015.
Área de concentração: Sinais e Sistemas.
Orientação: Prof. Dr. Guilherme Alencar Barreto.
Coorientação: Prof. Dr. Cláudio Marques de Sá Medeiros.
1. Teleinformática. 2. Curto circuito. 3. Inteligência computacional. I. Título.



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA
CAMPUS DO PICI, CAIXA POSTAL 6007 CEP 60.738-640
FORTALEZA – CEARÁ - BRASIL
FONE (+55) 85 3366-9467 – FAX (+55) 85 3366-9468

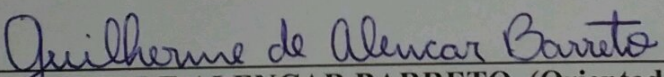
DAVID NASCIMENTO COELHO

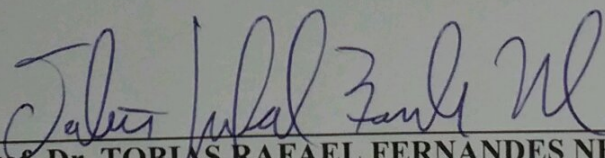
SOBRE O DESEMPENHO DE ALGORITMOS DE APRENDIZADO DE MÁQUINAS NA DETEÇÃO DE FALHAS EM MOTORES DE INDUÇÃO TRIFÁSICOS: UM ESTUDO COMPARATIVO

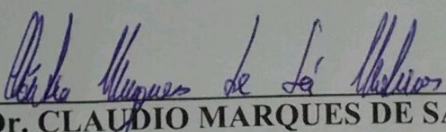
Dissertação submetida à Coordenação do Programa de Pós-Graduação em Engenharia de Teleinformática, da Universidade Federal do Ceará, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Teleinformática.
Área de concentração: Sinais e Sistemas.

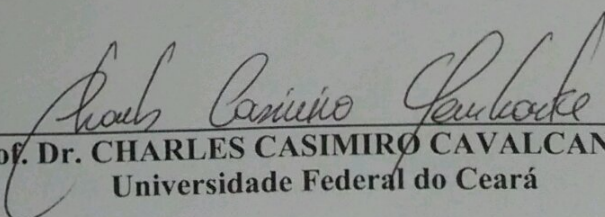
Aprovada em: 29/09/2015.

BANCA EXAMINADORA


Prof. Dr. GUILHERME DE ALENCAR BARRETO (Orientador)
Universidade Federal do Ceará


Prof. Dr. TOBIAS RAFAEL FERNANDES NETO
Universidade Federal do Ceará


Prof. Dr. CLAUDIO MARQUES DE SA MEDEIROS
Instituto Federal de Educação Ciência e Tecnologia do Ceará


Prof. Dr. CHARLES CASIMIRO CAVALCANTE
Universidade Federal do Ceará

Agradecimentos

A Deus, por sempre se mostrar presente em minha vida.

A meus pais, que sempre me apoiaram em todos os momentos e são exemplos em tudo para mim.

Aos meus irmãos, que mesmo já não dividindo o mesmo teto são extremamente presentes e importantes.

À Nice, por ter enfrentado cada fase dessa dissertação, e por sempre me aconselhar e apoiar nas decisões mais importantes.

Ao Prof. Guilherme de Alencar Barreto e ao Prof. Cláudio Marques de Sá Medeiros, por toda a paciência e ensinamentos durante o mestrado.

Aos professores Tobias Fernandes e Charles Casimiro pelo tempo, pelas valiosas colaborações e sugestões.

Aos amigos do mestrado (Ananda, César, Daniel, Amauri, Davyd, Rodrigo, Rafael) pelas reflexões, críticas, sugestões recebidas e ajudas com o LaTeX.

Aos amigos e família, os quais contribuíram para a finalização deste trabalho direta ou indiretamente.

À CAPES, pelo apoio financeiro com a manutenção da bolsa de auxílio.

Resumo

Esta dissertação visa a detecção de falhas incipientes por curto-circuito entre espiras de um motor de indução trifásico do tipo gaiola de esquilo acionado por conversor de frequência com modulação por largura de pulso do tipo senoidal. Para detectar este tipo de falha, uma bancada de testes é utilizada para impor diferentes condições de operação ao motor, e cada amostra do conjunto de dados foi extraída das correntes de linha do conversor de frequência supracitado. Para extração de características, a análise da assinatura de corrente do motor foi utilizada. Para solucionar este problema, a detecção desta falha é tratada como um problema de classificação, por isso, diferentes algoritmos supervisionados de aprendizado de máquina são utilizados: Mínimos Quadrados Ordinários, Redes Perceptron Simples, Redes Perceptron Multicamadas, Máquina de Aprendizado Extremo, Máquina de Vetor de Suporte, Máquina de Vetor de Suporte por Mínimos Quadrados, Máquina de Aprendizado Mínimo, e Classificadores Gaussianos. Juntamente com a técnica de opção de rejeição, estes classificadores são testados e os resultados destes são comparados entre si e com outros trabalhos que fizeram uso mesmo banco de dados. Taxas de acerto máximo de 100% com os classificadores Máquina de Vetor de Suporte e Máquina de Vetor de Suporte por Mínimos Quadrados sugerem que, em um futuro próximo, um sistema embarcado pode ser desenvolvido com estes algoritmos.

Palavras-Chave: Inteligência Computacional. Aprendizado de Máquina. Motores de Indução. Detecção de Falhas. Opção de Rejeição.

Abstract

This dissertation aims at the detection of short-circuit incipient fault condition in a three-phase squirrel-cage induction motor fed by a sinusoidal PWM inverter. In order to detect this fault, a test bench is used to impose different operation conditions to an induction motor, and each sample of the data set is taken from the line currents of the PWM inverter aforementioned. For feature extraction, the Motor Current Signature Analysis is used. The detection of this fault is treated as a classification problem, therefore different supervised algorithms of machine learning are used so as to solve it: Ordinary Least Squares, Single-layer Perceptron, Multi-layer Perceptron, Extreme Learning Machine, Support-Vector Machine, Least-Squares Support-Vector Machine, the Minimal Learning Machine, and Gaussian Classifiers. Together with Reject Option technique, these classifiers are tested and the results are compared with other works that use the same data set. Maximum accuracy rates of 100% with Support-Vector Machine and Least-Squares Support-Vector Machine classifiers suggest that, in near future, an embedded system can be developed with these algorithms.

Key-words: Computational Intelligence. Machine Learning. Induction Motors. Fault Detection. Reject Option.

Lista de ilustrações

Figura 1 – Formação do Banco de Dados Reais. Adaptado de OLIVEIRA, 2014.	23
Figura 2 – Bancada de testes utilizada nesta pesquisa.	24
Figura 3 – Características das Amostras. (OLIVEIRA; MEDEIROS, 2013)	26
Figura 4 – Mapeamento linear dos dados. Adaptado (MONTEIRO, 2009).	35
Figura 5 – Modelo matemático de um neurônio da rede PS	38
Figura 6 – Modelo dos neurônios da rede MLP (a) camada oculta (b) camada de saída	40
Figura 7 – Problema da Multilateração. (Souza Jr. et al., 2015).	52
Figura 8 – Metodologia Dos Experimentos	55
Figura 9 – Distribuição dos dados não-normalizados.	65
Figura 10 – Distribuição dos dados não-normalizados.	66
Figura 11 – Distribuição dos dados normais, normalizados.	67
Figura 12 – Distribuição dos dados de falhas, normalizados.	67
Figura 13 – Gráfico de Dispersão 1.	68
Figura 14 – Gráfico de Dispersão 3.	68
Figura 15 – Metodologia 01.	69
Figura 16 – Metodologia 01.	70
Figura 17 – Metodologia 02.	72
Figura 18 – Metodologia 03.	73
Figura 19 – Metodologia 03.	75
Figura 20 – Metodologia 04.	76
Figura 21 – Problema com 7 classes. Banco Desequilibrado de Dados.	79
Figura 22 – Problema com 7 classes. Banco equilibrado de Dados.	79
Figura 23 – Problema com 3 classes. Banco Desequilibrado de Dados.	80
Figura 24 – Problema com 3 classes. Banco equilibrado de Dados.	80
Figura 25 – Problema com 4 classes. Banco Desequilibrado de Dados.	81
Figura 26 – Problema com 4 classes. Banco equilibrado de Dados.	82

Figura 27 – Problema Binário. Com opção de Rejeição. Banco Desequilibrado de Dados.	83
Figura 28 – Problema Binário. Com opção de Rejeição. Banco equilibrado de Dados.	83
Figura 29 – Problema com 7 classes. Com opção de Rejeição. Banco Desequilibrado de Dados.	84
Figura 30 – Problema com 7 classes. Com opção de Rejeição. Banco equilibrado de Dados.	85
Figura 31 – Problema com 3 classes. Com opção de Rejeição. Banco Desequilibrado de Dados.	86
Figura 32 – Problema com 3 classes. Com opção de Rejeição. Banco equilibrado de Dados.	86
Figura 33 – Problema com 4 classes. Com opção de Rejeição. Banco Desequilibrado de Dados.	87
Figura 34 – Problema com 4 classes. Com opção de Rejeição. Banco equilibrado de Dados.	87

Lista de tabelas

Tabela 1 – Características do MIT utilizado.	24
Tabela 2 – Características das amostras (COELHO; MEDEIROS, 2013).	25
Tabela 3 – Hiperparâmetros dos classificadores avaliados nesta dissertação	57
Tabela 4 – Parâmetros dos Classificadores	59
Tabela 5 – Matriz de Confusão MLP - Método 1.	71
Tabela 6 – Matriz de Confusão OLS - Método 1.	71
Tabela 7 – Matriz de Confusão MLP - Método 2.	72
Tabela 8 – Matriz de Confusão OLS - Método 2.	72
Tabela 9 – Matriz de Confusão MLP - Método 3.	74
Tabela 10 – Matriz de Confusão OLS - Método 3.	74

Lista de abreviaturas e siglas

BP	Back Propagation
ELM	Extreme Learning Machine
FFT	Fast Fourier Transform
LSSVM	Least-squares Support Vector Machine
MAP	Maximum a Posteriori
MCSA	Motor Current Signature Analysis
MIT	Motor de Indução Trifásico
MLM	Minimal Learning Machine
MLP	Multi-layer Perceptron
SOM	Self-organizing Map
SVD	Singular Value Decomposition
SVM	Support Vector Machine
OLS	Ordinary Least Squares
PS	Perceptron Simples
RBF	Radial Basis Function

Lista de símbolos

w_r	Velocidade angular rotórica do motor (em RPM)
f_s	Frequência de alimentação do motor
P	Pares de polos do motor
s	Escorregamento do motor
f_h	Componentes de frequência
k_1	Ordem dos harmônicos temporais
k_2	Ordem dos harmônicos espaciais
N	Número total de amostras
n	Índice da amostra atual
\mathbf{x}_n	Vetor de amostras
\mathbf{X}	Matriz de amostras
C_n	Classe da amostra
p	Número de atributos do problema
K	Número de classes do problema
\mathbf{d}_n	Rótulo da amostra
\mathbf{D}	Matriz de rótulos
$p()$	Probabilidade
$\boldsymbol{\mu}_i$	Vetor médio da i -ésima classe
$\boldsymbol{\Sigma}_i$	Matriz de Covariância da i -ésima classe

$ \cdot $	Determinante de uma matriz
$g_i()$	função discriminante da i -ésima classe
$\ln()$	logaritmo natural
$Q_i()$	Distância de Mahalanobis da i -ésima classe
b	Bias
\mathbf{y}_n	Saída do classificador
\mathbf{Y}	Matriz de saídas do classificador
$N1$	Número de amostras para treinamento e validação
$N2$	Número de amostras para teste
q	Número de neurônios ocultos
$\ \cdot\ ^2$	Norma Euclidiana quadrática
T	Transposição de uma matriz
l	Constante da regularização de Tikhonov
L	Quantidade de parâmetros de um classificador
\mathbf{W}	Matriz de pesos
\mathbf{w}_i	Vetor de pesos do i -ésimo neurônio da camada oculta
\mathbf{m}_k	Vetor de pesos do k -ésimo neurônio da camada de saída
t	Iteração
u_i	Ativação do i -ésimo neurônio
$\phi()$	Função de ativação
$\phi'()$	Derivada da função de ativação
δ_k	Gradiente local do k -ésimo neurônio
e_k	Erro do k -ésimo neurônio
λ	Fator de momento
η	Taxa de aprendizagem
$J()$	Função custo

$L()$	Função lagrangeana
α_i	Multiplicador de Lagrange
ξ_i	Variável de folga
C	Constante de regularização do SVM
$\kappa(.,.)$	Função de Kernel
σ^2	Hiperparâmetro da função de Kernel gaussiano
γ	Constante de regularização do LSSVM
R	Pontos de referência do MLM
\mathbf{m}_x	Entradas de referência do MLM
\mathbf{D}_x	Distância entre as entradas
\mathbf{t}_y	Saída de referência do MLM
$\mathbf{\Delta}_y$	Distância entre as saídas
\mathbf{B}	Modelo de regressão do MLM
r	Número de divisões da validação cruzada
c_r	Custo de rejeição

Sumário

1	INTRODUÇÃO	16
1.1	Motores de Indução	16
1.2	O Problema da Detecção de Falhas em MIT	17
1.3	Objetivos	18
1.3.1	Objetivo Geral	18
1.3.2	Objetivos Específicos	18
1.4	Produção Científica	19
1.5	Organização Geral da Dissertação	19
2	DETECÇÃO DE FALHAS EM MOTORES DE INDUÇÃO	21
2.1	Descrição do problema	21
2.2	Descrição do Banco de Dados	23
2.3	Conclusão	27
3	CLASSIFICADORES	28
3.1	O problema da classificação	29
3.2	Classificadores Estatísticos e Critério MAP	29
3.2.1	Maximum a Posteriori	29
3.2.2	Classificadores Gaussianos	31
3.3	Classificador Linear de Mínimos quadrados	34
3.3.1	Estimação da Matriz W via Mínimos Quadrados Ordinários	35
3.3.2	Implementação da Inversão de Matrizes	36
3.4	Classificadores Neurais	37
3.4.1	Perceptron Multicamadas	38
3.4.2	Máquinas de Aprendizado Extremo	42
3.5	Classificadores Via Vetores de Suporte	43
3.5.1	Máquinas de Vetor de Suporte	44
3.5.2	Máquinas de Vetor de Suporte por Mínimos Quadrados	49
3.6	Classificação Via Regressão Baseada em Distâncias	50
3.7	Conclusão	52
4	METODOLOGIA DOS EXPERIMENTOS	53

4.1	Análise dos Dados	53
4.2	Banco de Dados	53
4.3	Metodologia de Comparação entre Classificadores	54
4.3.1	Holdout	56
4.3.2	Seleção do modelo	56
4.3.3	Treinamento dos Classificadores	58
4.3.4	Teste do Classificador	59
4.4	Problema Multiclasses	60
4.5	Opção de Rejeição	61
4.6	Comparação entre Classificadores	63
4.7	Conclusão	64
5	ANÁLISE DOS RESULTADOS	65
5.1	Análise dos Dados	65
5.2	Comparação entre Classificadores	69
5.3	Seleção dos Hiperparâmetros	76
5.4	Resultados Com Problema Multiclasses	78
5.5	Resultados com Opção de Rejeição	81
6	CONCLUSÃO	88
6.1	Objetivo Geral	88
6.2	Objetivos Específicos	89
6.3	Trabalhos Futuros	89
A	IMPLEMENTAÇÕES	90
A.1	Implementação do Classificador ELM	90
	Referências	92

Introdução

Neste capítulo, é apresentada uma breve introdução sobre o motor de indução trifásico, a detecção de falhas neste, e a abordagem deste problema. Além disso, este capítulo contém os objetivos, a organização, e as contribuições desta dissertação.

1.1 Motores de Indução

Os motores de indução trifásicos (MIT), devido a sua robustez, eficiência e simplicidade, são a principal força motora da indústria atual (BACHA et al., 2008), (SESHADRINATH; SINGH; PANIGRAHI, 2014). De acordo com Thomson e Fenger (2001), estes podem consumir tipicamente de 40 a 50% de toda energia elétrica gerada em uma nação industrializada. Dentre as aplicações destas máquina, pode-se citar o acionamento de ventiladores, bombas, bobinadeiras, esteiras transportadoras e elevadores (JUNIOR, 2013).

Com a finalidade de adequar este motor às diversas aplicações, vários estudos sobre o controle de velocidade e torque desta máquina já foram realizados. Dentre os meios para controlar a velocidade desta, o controle vetorial e o escalar podem ser citados (VENKADESAN; HIMAVATHI; MUTHURAMALINGAM, 2013), (SAWA; KUME, 2004), (NIRALI; SHAH, 2011). Neste contexto, quando a variação de velocidade do motor de indução se mostra necessária, os conversores de frequência são muito utilizados. Estes dispositivos variam a velocidade do motor de indução modificando a frequência da tensão de alimentação deste.

A equação que relaciona a velocidade do motor à frequência da tensão de alimentação deste, pode ser descrita como:

$$w_r = \frac{120f_s}{2P}(1 - s), \quad (1.1)$$

onde w_r é a velocidade angular rotórica (em RPM), f_s é a frequência de alimentação, em Hertz (Hz), do motor, P é o número de pares de polos, e s é o escorregamento deste.

Mesmo com todos os estudos sobre esta máquina e de toda a sua robustez, devido ao envelhecimento da máquina, condições adversas do ambiente, aplicações inadequadas, ou falta de um programa de manutenção, o motor de indução é sujeito a várias falhas (GHATE; DUDUL, 2010), (NANDI; TOLIYAT; LI, 2005). De acordo com Vico e Hunt (2010), a taxa de falhas nestes motores é conservadoramente estimada de 3 a 5% por ano.

Os tipos mais comuns destas são falhas em rolamento, falhas de isolamento no estator ou rotor, abertura de barras ou quebra de anéis, e excentricidade rotórica (NANDI; TOLIYAT; LI, 2005), (BONNETT, 2010). Dentre todas estas falhas, a perda de isolamento nos enrolamentos do estator corresponde a aproximadamente 40% de todas as falhas de motores (NANDI; TOLIYAT; LI, 2005), (MARTINS; PIRES; PIRES, 2007).

Devido a todas estas falhas, se não houver um monitoramento periódico destes motores, podem ocorrer paradas não programadas em linhas de produção, e o custo destas é muito alto (OLIVEIRA; MEDEIROS, 2013). Segundo Avelar, Baccarini e Amaral (2011), custos de manutenção podem representar de 15% a 40% de muitos produtos.

Por isso, a detecção de falhas e o monitoramento periódico das condições de máquinas rotativas trazem um diferencial competitivo para as indústrias, visto que estes procedimentos podem garantir eficiência, segurança e bom funcionamento ao diminuir as quebras destas máquinas, conduzindo, assim, a uma maior produtividade e redução na perda de capital (YADAV; WADHWANI, 2011).

Especificamente, o monitoramento em tempo real do curto circuito entre espiras no enrolamento do estator é uma ferramenta importante para reduzir custos e proteger a máquina, visto que esta falha demora apenas alguns minutos para evoluir (THOMSON; FENGER, 2001). Também, se a falha de curto circuito ocorrer em apenas uma fase, a detecção prematura desta irá minimizar os danos em outras fases.

1.2 O Problema da Detecção de Falhas em MIT

Diversos estudos sobre detecção de falhas por curto circuito entre espiras do estator do motor de indução trifásico foram feitos recentemente. Asfani et al. (2012) utilizaram densidade espectral, transformada Wavelet Haar, e redes neurais para detecção de curto circuito. Por outro lado, Avelar, Baccarini e Amaral (2011) utilizaram máquinas de vetores de suporte, além das leituras da componente fundamental de tensão e das componentes fundamental e 3ª harmônica de corrente para este mesmo tipo de falha. Por fim, Seshadrinath, Singh e Panigrahi (2014) utilizaram uma rede neural probabilística baseada em Wavelets para detectar falhas entre espiras.

Nos trabalhos supracitados e em outros trabalhos que visam a detecção de falhas em motores de indução, tenta-se dividir as amostras coletadas dos motores entre funcionamento normal ou com falha. Assim a detecção de falhas pode ser tratada como um problema de reconhecimento de padrões, onde os dados devem ser divididos em duas ou mais classes.

Vários algoritmos de aprendizado de máquina, tais como máquinas de vetores de suporte (AMEL et al., 2013) (SAWITRI et al., 2013) e redes neurais (GHATE; DUDUL, 2011) (ZAREI, 2012) podem ser utilizados para resolver este tipo de problema. Estes

algoritmos não necessitam de um modelo matemático do motor (MARTINS; PIRES; PIRES, 2007), porém é necessário uma quantidade consistente e significativa de dados que represente adequadamente o problema específico.

De modo a gerar estes dados, existem diversas técnicas para extrair informações dos motores. Análise de sinais de vibração, análise da assinatura de corrente do motor, análise de óleo, fusão de sensores, análise de temperatura, análise de ruído audível e ultrassom são alguns exemplos destas técnicas (LEE et al., 2014).

Dentre estas técnicas, a análise da assinatura de corrente do motor (do inglês, *Motor Current Signature Analysis* - MCSA) é muito utilizada devido ao seu baixo custo e por ser um método não invasivo (EFTEKHARI et al., 2013). Particularmente, na detecção de curto circuito entre espiras do estator, é comum utilizar algumas componentes do espectro de frequências de corrente para compor a entrada para o sistema de detecção.

Porém, este problema não é simples, visto que as mesmas componentes de frequência utilizadas para detecção de falhas podem existir previamente no sistema elétrico ou serem afetadas por mais de uma falha, e o ruído proveniente do ambiente pode estar fortemente incorporado aos sinais de corrente (GHATE; DUDUL, 2010). Essas condições podem criar muitas dificuldades na detecção de falhas, e tornar o problema não linear.

Por isso, de modo a abordar este problema, almeja-se avaliar diversos paradigmas de aprendizagem de máquina no problema da detecção de falhas em MIT acionado por conversor de frequência.

1.3 Objetivos

Uma vez exposto o problema de interesse na seção anterior, os seguintes objetivos foram traçados para esta dissertação.

1.3.1 Objetivo Geral

O objetivo geral desta pesquisa é, através da comparação de diversas técnicas de aprendizagem de máquina, detectar falhas incipientes por curto circuito em motores de indução trifásicos gaiola de esquilo, acionados por um conversor de frequência.

1.3.2 Objetivos Específicos

Os objetivos específicos desta dissertação estão listados a seguir:

Analisar estatisticamente o banco de dados utilizado.

Implementar diferentes paradigmas de classificadores de padrões, tais como MLP, SVM e MLM.

Estudar a eficácia destes classificadores na detecção de falhas de motores de indução trifásicos.

Utilizar estratégias de classificação com opção de rejeição, visando aumentar a confiabilidade do sistema de classificação.

Comparar, através de testes estatísticos, classificadores de padrões quanto a sua capacidade de detecção de falhas.

1.4 Produção Científica

Os resultados parciais deste trabalho foram reunidos no artigo a seguir.

COELHO, D. N. ; SANTOS, J. D. A. ; MEDEIROS, C. M. S. ; BARRETO, G. A. Performance Comparison of Classifiers in The Detection of Short Circuit Incipient Fault in a Three-phase Induction Motor. In: IEEE Symposium Series on Computational Intelligence, 2014, Orlando, FL, USA. IEEE-SSCI, 2014.

Além deste artigo, durante o período no qual esta dissertação foi feita, o autor desta também foi co-autor nos seguintes artigos:

SILVA, R. D. C. ; COELHO, D. N. ; THE, G. A. P. . Comparison Between K-Nearest Neighbors, Self-organizing Maps and Optimum-path Forest in the recognition of Packages using Image Analysis by Zernike Moments. In: International Conference on Industry Applications, 2014, Juiz de Fora. INDUSCON, 2014.

SILVA, R. D. ; COELHO, D. N. ; THE, G. A. P. Performance Analysis of Classifiers to Recognition of Objects from Low-resolution Images Industrial Sensor. In: Simpósio Brasileiro de Automação Inteligente, 2013, Fortaleza. SBAI/DINCON 2013, 2013.

OLIVEIRA, A. G. ; COELHO, D. N. ; BESSA, R. ; MEDEIROS, C. M. S. ; PONTES, R. T. . Técnicas Computacionais Para a Detecção de Falhas por Curto-circuito Entre Espiras de um Motor de Indução Acionado por Conversor de Frequência. In: Simpósio Brasileiro de Automação Inteligente, 2013, Fortaleza. SBAI/DINCON 2013, 2013.

1.5 Organização Geral da Dissertação

Por fim, o restante desta dissertação está organizada na seguinte formatação:

No capítulo 2, é mostrado um estudo sobre a falha por curto circuito entre espiras do estator do MIT, e sobre a detecção de falhas neste tipo de motor. Além disso, descreve-se o conjunto de dados utilizado neste trabalho.

No capítulo 3, apresenta-se o problema de classificação. É ainda realizada uma breve explicação dos algoritmos utilizados e das especificidades do problema da detecção

de falhas.

Já no capítulo 4, os detalhes da implementação dos algoritmos, a metodologia geral de comparação destes, e as técnicas computacionais (utilizadas juntamente com os classificadores) são discriminadas.

No capítulo 5, os resultados do trabalho desenvolvido são discutidos e analisados.

Por fim, no capítulo 6, são feitas as considerações finais e as perspectivas para trabalhos futuros.

Detecção de Falhas em Motores de Indução

Neste capítulo, as características do problema investigado nesta dissertação são abordadas. Primeiramente a falha por curto circuito e os meios para a sua detecção são evidenciados. Por fim, é mostrado o procedimento para formação do banco de dados.

2.1 Descrição do problema

Como já mencionado anteriormente, as falhas por curto circuito entre espiras do estator dos motores de indução trifásicos estão entre as de maior ocorrência dentre todas as falhas. Os maiores contribuidores para este tipo de falha podem ser agrupados como estresses térmicos, elétricos, mecânicos e do ambiente (SESHADRINATH; SINGH; PANIGRAHI, 2014). Na presença de um ou mais destes estresses, o motor pode vir a falhar.

Especificamente, a falha por curto circuito no estator do motor de indução trifásico inicia como uma falha de alta impedância (NATARAJAN, 1989). Em seguida, a corrente de falha pode causar um aquecimento local, fazendo com que a falha se espalhe rapidamente no bobinamento (TALLAM et al., 2003), configurando uma falha de baixa impedância.

No geral, existem duas formas de detectar falhas: através do diagnóstico ou de um prognóstico. O diagnóstico é feito após ocorrer a falha, e serve tanto para determinar as causas da falha como para nortear os procedimentos de reparos. É uma abordagem reativa ao problema, ou seja, não previne as falhas. Já o prognóstico é a detecção do início da falha, podendo estimar (predizer) a vida útil remanescente da máquina e programar a sua manutenção. Esta é uma abordagem que previne o problema (proativa) (LEE et al., 2014).

No caso das falhas por curto circuito no estator do motor de indução trifásico acionado por um conversor de frequência, é muito importante fazer o prognóstico da falha, visto que, se esta falha for detectada quando se está iniciando sua ocorrência, a equipe de manutenção pode atuar e evitar acréscimos aos custos de produção, proteger o conversor de frequência, e a máquina de indução pode ser reutilizada após o rebobinamento (THOMSON; FENGER, 2001).

Para detectar este tipo de falha, primeiramente, é necessário algum método para extrair características do motor. Dentre os diversos métodos, a análise da assinatura de corrente do motor (do inglês, *Motor Current Signature Analysis* - MCSA) tem sido vastamente utilizado, visto que não é um método invasivo, não precisa ser adaptado para áreas com risco de explosão e pode ser aplicado, sem restrição de potência, a qualquer máquina (THORSEN; DALVA, 1997).

Este método consiste em utilizar algumas componentes do espectro de frequência da corrente do motor como características da falha. Para definir quais são estas frequências, Penman et al. (1994) perceberam que as harmônicas geradas pelas falhas por curto circuito obedeciam à equação

$$f_h = \left[\frac{k_1 \pm k_2(1-s)}{P} \right] f_s, \quad (2.1)$$

em que f_h é a componente de frequência em função do curto-circuito entre espiras; $k_1 = 1, 3, 5, \dots$ é a ordem das harmônicas temporais¹; $k_2 = 1, 2, 3, \dots$ é a ordem das harmônicas espaciais²; s é o escorregamento; P é o número de pares de pólos; e f_s é a frequência fundamental de alimentação do motor.

Assim, no caso da utilização de um conversor de frequência, sabendo-se o valor da frequência fundamental com a qual este alimenta o motor e medindo-se ou estimando-se o escorregamento, pode-se extrair as demais componentes para a detecção de falhas.

É importante salientar que algumas das frequências, obtidas da Equação 2.1, também podem estar relacionadas a outros tipos de falhas no motor ou na própria fonte de alimentação. De acordo com Das, Purkait e Chakravorti (2012), a tensão de alimentação desbalanceada pode produzir uma assinatura de corrente que é, aparentemente, idêntica à assinatura causada pelo curto circuito entre espiras por exemplo.

A partir da Equação (2.1), alguns autores tentaram encontrar componentes de frequência para a detecção de falhas por curto circuito entre espiras. Thomson e Fenger (2001) realizaram experimentos com motores de baixa potência submetidos a curto-circuito entre espiras. As componentes encontradas apenas para curto-circuito, ocorreram para $k_1 = 1$ e $k_2 = 3$, ou $k_1 = 1$ e $k_2 = 5$. Em um motor com dois pares de pólos e com escorregamento próximo de zero, estas componentes equivalem aproximadamente a $2,5f_s$ e $3,5f_s$.

Já Coelho e Medeiros (2013), após aplicarem a transformada rápida de Fourier (do inglês, *fast Fourier transform* - FFT) a diversas amostras de sinais de correntes do motor de indução trifásico sujeitos a vários níveis de curto circuito, verificaram que as componentes do espectro de frequências que mais variavam eram as seguintes: $0, 5f_s, 1f_s, 1,5f_s, 2f_s, 3f_s, 5f_s$ e $7f_s$.

¹ decorrentes de outros componentes da rede elétrica, tais como conversores eletrônicos e pontes retificadoras.

² decorrentes da construção física do MIT e do posicionamento de seus enrolamentos.

Por fim, Oliveira e Medeiros (2013), após coletarem, do espectro de frequência de um motor sujeito a vários níveis de curto-circuito, 16 componentes entre $0,5f_s$ e $8f_s$ (a cada $0,5f_s$), aplicaram uma rede MLP a diversas combinações destas componentes para tentar detectar este tipo de falha. Os melhores resultados de classificação foram obtidos através da seguinte combinação de componentes: $0,5f_s$, $1,5f_s$, $2,5f_s$, $3f_1$, $5f_s$ e $7f_s$.

Na presente dissertação, os mesmo sinais usados para gerar as componentes em Coelho e Medeiros (2013) e Oliveira e Medeiros (2013) foram utilizados. Os sinais e a bancada de testes que serviu para gerá-los são explicados no tópico a seguir.

2.2 Descrição do Banco de Dados

Nesta seção, é apresentado o fluxograma que evidencia a abordagem do problema e os passos para a formação do banco de dados. A partir disto, são descritas a sequência de passos para obter a quantidade total dos dados, a distribuição dos exemplos entre as classes, as características de cada amostra, e as técnica para seleção e extração de atributos.

Na Figura 1, tem-se uma visão geral dos módulos do sistema para detecção de falhas e da sequência de passos para obter cada amostra do banco de dados gerado.

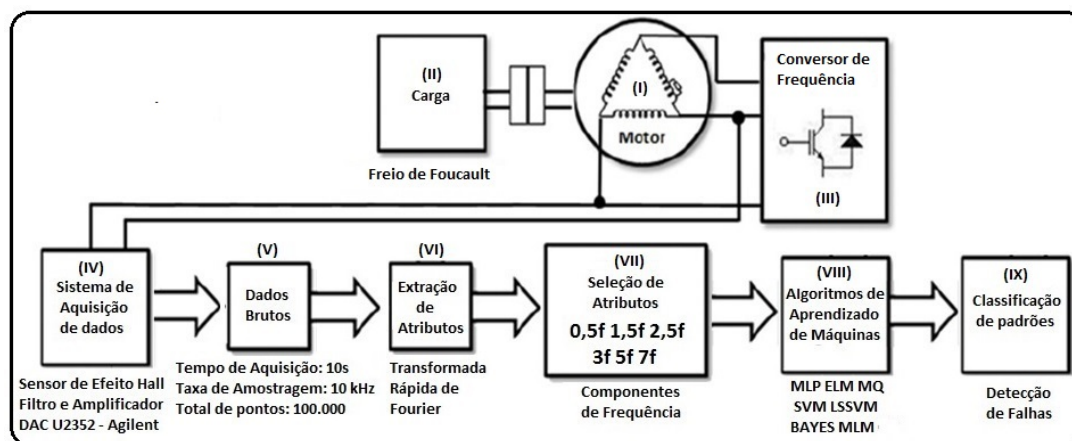


Figura 1 – Formação do Banco de Dados Reais. Adaptado de OLIVEIRA, 2014.

Primeiramente, sabe-se que o sucesso da aplicação de reconhecimento de padrões depende da construção de um modelo confiável para os dados. Por isso, para gerar o conjunto de dados representantes de diversas condições de operação do MIT, uma bancada de testes foi montada. Os componentes principais desta bancada, representados pelos módulos I e II da Figura 1, são evidenciados na Figura 2, onde podem ser vistos um motor de indução trifásico, e um eixo conectando o motor a um Freio de Foucault.

Um motor de indução trifásico do tipo gaiola de esquilo (*standard*) (módulo I da Figura 1) foi utilizado como base para este banco de dados. Suas principais características

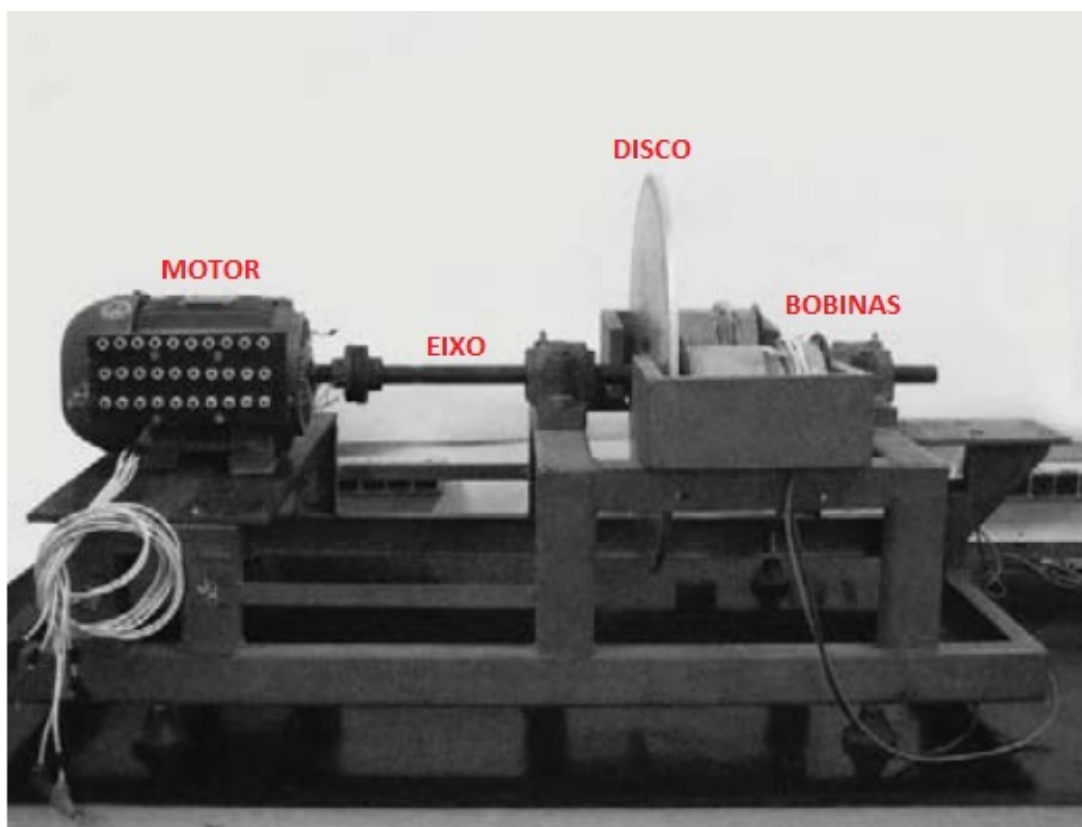


Figura 2 – Bancada de testes utilizada nesta pesquisa.

estão listadas na Tabela 1.

Tabela 1 – Características do MIT utilizado.

Grandeza	Valor	Unidade
Potência Nominal	0,75	kW
Tensão Nominal	220/380	V
Corrente Nominal (In)	3,02/1,75	A
Eficiência	79,5%	-
Velocidade Nominal	1720	RPM
Corrente de Pico	7,2In	A
Fator de Potência	0,82	-

Para aplicar três níveis de carga a este motor (0% - sem carga, 50%, e 100% da carga nominal), um Freio de Foucault foi utilizado (módulo II da Figura 1).

Também, para variar a velocidade do motor, um conversor de frequência de fabricação WEG (CFW-09 - módulo III da Figura 1) foi utilizado para aplicação de tensão com diversas condições de componente fundamental de frequência: 30 Hz, 35 Hz, 40 Hz, 45 Hz, 50 Hz, 55 Hz e 60 Hz. Este conversor operou no modo escalar V/F para a geração da modulação por largura de pulsos do tipo senoidal.

Além disso, três sensores baseados em efeito Hall (módulo IV da Figura 1) foram utilizados para medir as correntes de linha do motor.

Normalmente, este motor possui apenas dois terminais disponíveis por fase (para que este seja alimentado). Porém, para gerar este conjunto de dados, o motor foi rebobinado, de modo que mais terminais estão disponíveis, expondo derivações dos enrolamentos de cada fase. Deste modo, pode-se emular diferentes níveis de curto circuito entre espiras do mesmo enrolamento. Neste trabalho, falhas por curto circuito entre fases não são consideradas.

Três níveis de curto circuito foram utilizados. No Nível 1, 5 espiras de um enrolamento são curto-circuitadas, totalizando 1,41% das espiras de uma fase. No Nível 2, 17 espiras são curto circuitadas (correspondendo a 4,8% do enrolamento). Por fim, no Nível 3, 32 espiras (9,26% do enrolamento) são deixadas em curto circuito.

Finalmente, um sistema auxiliar de comando foi construído para executar dois tipos de curto circuito: alta impedância (A - emulando o início do processo de curto circuito), e baixa impedância (B - emulando a propagação da falha). Com estes dois tipos de curto circuito, e com os três níveis estabelecidos, existem seis condições diferentes de falha: A1, A2, A3, B1, B2 e B3.

É importante mencionar que, em ambos os tipos de de curto circuito, as correntes de falha foram limitadas a corrente nominal do motor. Esta limitação foi necessária para que o motor não fosse danificado e pudesse ser reutilizado após várias emulações de falhas.

Todas estas condições de operação estão representadas na Tabela 2. A combinação das três fases do conversor, dos três níveis de carga, das sete frequências do conversor e das sete extensões de falha do motor formam um banco de dados composto por 441 amostras ($3 \times 3 \times 7 \times 7$), onde cada amostra representa uma condição específica de operação.

Tabela 2 – Características das amostras (COELHO; MEDEIROS, 2013).

	Características das Amostras
Fase do Conversor	Fase 1; Fase 2; Fase 3
Nível de Carga	0%; 50%; 100%
Frequência do Conversor	30 Hz; 35 Hz; 40 Hz; 45 Hz; 50 Hz; 55 Hz; 60 Hz
Extensão da Falha	Normal; A1; A2; A3; B1; B2; B3

É importante mencionar que, como representado na Figura 1, o motor foi conectado em delta. Nesta configuração, duas correntes de linha do conversor de frequência foram diretamente conectadas à fase do motor que continha as falhas. Como um dos objetivos deste projeto é construir um sistema que possa detectar falhas utilizando apenas uma fase do conversor (independentemente do sensor estar conectado diretamente ou não à fase com falha), apenas uma das duas fases citadas acima é utilizada para que não haja informação redundante.

Assim, no conjunto de dados final, são utilizadas 294 amostras ($2 \times 3 \times 7 \times 7$): 147 da fase 1 (diretamente conectada à corrente de falha) e 147 da fase 3 (indiretamente

conectada à corrente de falha). Todas estas amostras são representadas na Figura 3. Como pode ser visto nesta figura, o problema pode ser tratado a partir de 7 classes, se cada nível de falha for considerado uma classe (normal, AI1, AI2, AI3, BI1, BI2, BI3). Com esta configuração, cada classe possui 42 amostras.

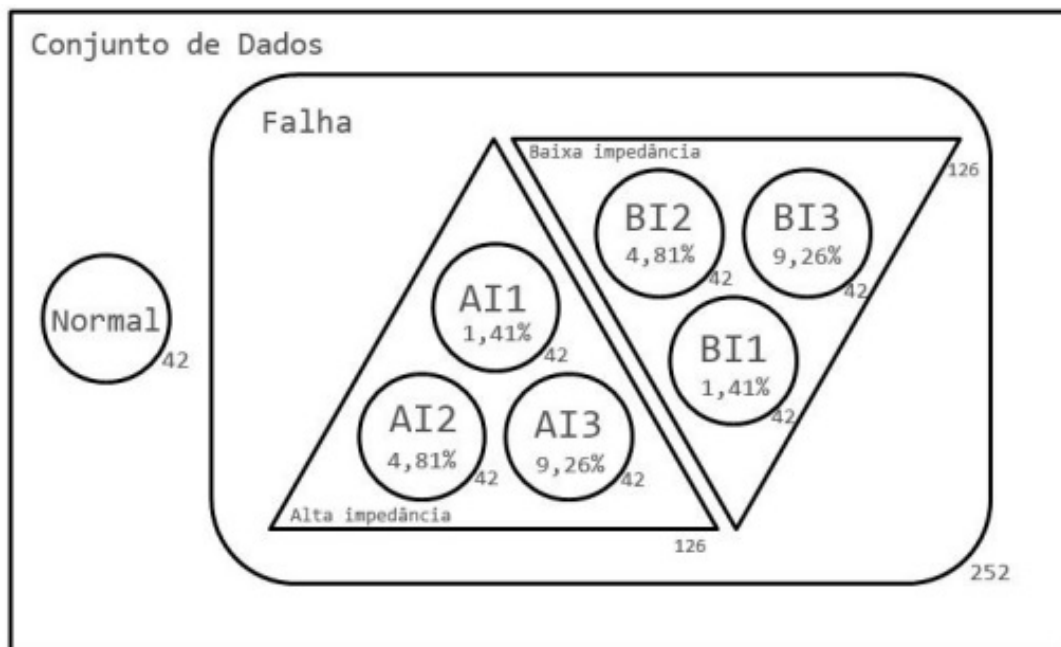


Figura 3 – Características das Amostras. (OLIVEIRA; MEDEIROS, 2013)

Por outro lado, quando se quer classificar o estado do motor entre condição normal de operação, falha de alta impedância ou falha de baixa impedância, o problema mantém-se multi-classes, porém com apenas 3 classes: uma com 42 amostras (normal), outra com 126 amostras (AI1, AI2, AI3) e a última também com 126 amostras (BI1, BI2, BI3).

Porém, é importante mencionar que para uma aplicação industrial não é um problema considerável confundir os níveis ou tipos de falhas. Assim, o problema aqui estudado pode ser tratado como um de classificação binária: motor em funcionamento normal ou com falha. Neste último caso, existem 42 amostras de motor em funcionamento normal e 252 de motor em curto circuito (AI1, AI2, AI3, BI1, BI2, BI3).

Vale salientar que esta distribuição de dados entre classes não é comum em um meio industrial, visto que amostras de motores em funcionamento normal são mais frequentes que as amostras de falhas.

Todos estes dados foram adquiridos através de um sistema de aquisição Agilent U2352, de 16 bits de resolução, passando por um filtro analógico passa-baixa de 1 kHz e um amplificador de sinal para adequação ao sistema de aquisição de dados.

Para coletar um determinada amostra, após escolhidas as condições de operação do motor (fase do conversor, nível de carga, frequência do conversor e extensão da falha), foi

feita uma aquisição de 10 s à uma taxa de amostragem de 10 kHz. Assim, cada amostra é um sinal estacionário de 100.000 pontos (módulo V, Figura 1).

Como apenas se está interessado na situação estacionária dos sinais de corrente, após esta aquisição, foi aplicada a FFT (módulo VI, Figura 1) para obter o espectro de frequência de cada uma das amostras. Como existe um filtro analógico de 1 kHz, pode-se escolher componentes de frequência de até 500 Hz³ como um atributo do problema.

Além disso, como a máxima frequência de alimentação do conversor é de 60 Hz, a maior componente múltipla desta frequência, considerando um máximo de 500 Hz, é a oitava harmônica ($8 \times 60 = 480$).

Por fim, como em [Oliveira e Medeiros \(2013\)](#) foi feito um estudo minucioso com diversas combinações de componentes de frequência na detecção de falhas através de uma rede MLP, as mesmas componentes utilizadas pelo trabalho anteriormente citado foram usadas nesta dissertação, a saber: $0,5f_s$, $1,5f_s$, $2,5f_s$, $3f_s$, $5f_s$ e $7f_s$, onde f_s é a frequência fundamental do conversor (módulo VII, Figura 1).

Em suma, o conjunto final de dados é composto por 294 amostras, onde cada um dos 6 atributos de cada amostra corresponde a uma harmônica da frequência fundamental de alimentação do conversor. Assim, cada amostra pode ser representada por

$$\mathbf{x}_n = \begin{bmatrix} 0,5f_s \\ 1,5f_s \\ 2,5f_s \\ 3f_s \\ 5f_s \\ 7f_s \end{bmatrix}, \quad (2.2)$$

em que \mathbf{x}_n é o vetor coluna (vetor de atributos) contendo valores de harmônicas da frequência fundamental de alimentação do conversor.

2.3 Conclusão

Neste capítulo, apresentou-se o problema da detecção de falhas por curto-circuito entre espiras do MIT, e as técnicas para montar um banco de dados a ser utilizado para a solução deste.

A seguir, nos Capítulos 3 e 4, serão apresentadas as ferramentas e metodologias utilizadas para detecção de falhas em motores.

³ teorema de Nyquist.

Classificadores

Neste capítulo, o problema da classificação é abordado e, em seguida, os algoritmos dos classificadores utilizados nessa dissertação são apresentados.

Inicialmente, é importante definir alguns termos que serão utilizados durante esta dissertação:

- **Atributo:** uma característica do sistema a ser tratado, que serve de entrada para os classificadores.
- **Amostra:** se refere a um vetor de atributos, descrevendo um objeto a ser classificado.
- **Hiperparâmetro:** uma variável do classificador, que deve ser definida, antes do treinamento, para que este classificador se adapte ao problema ao qual ele se propõe a solucionar.
- **Parâmetro:** uma variável do classificador que é ajustada durante o treinamento.
- **Treinamento:** etapa na qual os parâmetros de um classificador são ajustados.
- **Iteração de treinamento:** por iteração, entende-se o instante de apresentação de uma amostra de treinamento para ajustar os parâmetros de um classificador.
- **Época de treinamento:** ocorre após todas as amostras de treinamento terem sido apresentadas (uma única vez cada) a um classificador, de modo a atualizar os parâmetros deste.
- **Validação:** etapa na qual o algoritmo é aplicado a dados não utilizados durante o treinamento, e as estatísticas de desempenho do classificador são geradas.
- **Realização:** Quando finaliza-se as etapas de treinamento e teste, e valida-se o classificador com os dados restantes, ocorre uma realização do classificador.
- **Generalização:** é a característica mais desejada de um classificador, visto que, quanto maior a capacidade de generalização deste, maior será sua taxa de acerto para novos dados (dados de validação).

A partir dos termos anteriores, o problema da classificação será definido.

3.1 O problema da classificação

Muitos problemas no mundo dos negócios, ciência, indústria e medicina podem ser tratados como problemas de classificação. Exemplos incluem previsão de falências, reconhecimento de caracteres manuscritos, diagnóstico médico, controle de qualidade, detecção de falhas e reconhecimento de fala (ZHANG, 2000).

Formalmente, neste tipo de problema, assume-se que se está de posse de um conjunto de N pares $\{\mathbf{x}_n, C_n\}_{n=1}^N$, em que o vetor coluna $\mathbf{x}_n \in \mathbb{R}^p$ representa a n -ésima amostra de entrada¹ e C_n é a classe à qual pertence \mathbf{x}_n . Assume-se ainda que se tem um número finito e pré-definido de K classes ($K \ll N$), i.e. $C_n \in \{C_1, C_2, \dots, C_K\}$. Por fim, seja n_i o número de exemplos da i -ésima classe (i.e. C_i), temos que $N = n_1 + n_2 + \dots + n_K = \sum_{i=1}^K n_i$.

Dependendo do problema, há várias formas de codificar os rótulos das classes. Nesta dissertação, escolheu-se um vetor coluna $\mathbf{d}_n \in \mathbb{R}^K$, no qual a posição referente a classe à qual o dado de entrada pertence tem o valor +1 e as demais posições tem o valor -1.

Como exemplo, se um problema tem 3 classes, e a amostra atual pertence à classe 2, seu rótulo será: $\mathbf{d}_n = [-1 \ 1 \ -1]^T$. Assim, o problema aqui estudado possui N pares $\{\mathbf{x}_n, \mathbf{d}_n\}_{n=1}^N$.

No caso do problema de detecção de falhas por curto circuito, este pode ser tratado como binário ou multi classes. Se os dados forem divididos entre motores em funcionamento normal ou com falha, o problema em questão se torna binário. Porém, se determinados níveis de curto circuito forem tratados como uma classe, o problema se torna multi classes. Esta divisão dos dados entre as classes será melhor explicada no capítulo 4.

No restante do presente capítulo, tanto os algoritmos utilizados para o desenvolvimento deste trabalho, como alguns conceitos importantes são explicados.

3.2 Classificadores Estatísticos e Critério MAP

Nesta seção introduz-se o critério de decisão ótima, conhecido como critério *Maximum a Posteriori* (MAP), além de quatro classificadores gaussianos obtidos a partir da suposição de que os exemplos de uma dada classe seguem uma lei de distribuição de probabilidade normal.

3.2.1 Maximum a Posteriori

Inicialmente, define-se $p(C_i)$ como a probabilidade *a priori* da i -ésima classe, ou seja, a probabilidade da classe C_i ser selecionada antes do experimento ser realizado, sendo

¹ p é o número de atributos do problema em questão.

o experimento o ato de observar e classificar um certo padrão. Este é um experimento aleatório, visto que não se sabe de antemão a que classe o padrão será atribuído. Logo, uma modelagem probabilística se torna justificável.

Em seguida, levando em conta apenas os dados da classe C_i , ou seja, ao subconjunto de padrões \mathbf{x}_n cujos rótulos são iguais a C_i , um modelo probabilístico comum para estes dados é a densidade normal multivariada, denotada por $p(\mathbf{x}_n|C_i)$, de vetor médio $\boldsymbol{\mu}_i$ e matriz de covariância $\boldsymbol{\Sigma}_i$. Matematicamente, este modelo é dado pela seguinte expressão:

$$p(\mathbf{x}_n|C_i) = \frac{1}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_i) \right\}, \quad (3.1)$$

em que $|\boldsymbol{\Sigma}_i|$ denota o determinante da matriz de covariância $\boldsymbol{\Sigma}_i$, e $\boldsymbol{\Sigma}_i^{-1}$ denota a inversa desta matriz. Alguns detalhes sobre a implementação da matriz inversa estão dispostos na seção 3.3.

A densidade $p(\mathbf{x}_n|C_i)$, no contexto de classificação de padrões, também é chamada de *função de verossimilhança*² da classe C_i . A função de verossimilhança da classe C_i pode ser entendida como o modelo probabilístico que "explica" como os dados estão distribuídos nesta classe.

Além disso, dado que um novo padrão \mathbf{x}_n é observado, qual seria a probabilidade de que este padrão pertença à classe C_i ? Esta informação pode ser modelada através da função densidade *a posteriori* da classe, $p(C_i|\mathbf{x}_n)$.

Através do Teorema da Probabilidade de Bayes, a densidade a posteriori $p(C_i|\mathbf{x}_n)$ pode ser relacionada com a densidade a priori $p(C_i)$ e a função de verossimilhança $p(\mathbf{x}_n|C_i)$ por meio da seguinte expressão:

$$p(C_i|\mathbf{x}_n) = \frac{p(C_i)p(\mathbf{x}_n|C_i)}{p(\mathbf{x}_n)}. \quad (3.2)$$

Um critério comumente usado para tomada de decisão em classificação de padrões é o critério *máximo a posteriori* (MAP). Ou seja, um determinado padrão \mathbf{x}_n é atribuído à classe C_j se a moda da densidade a posteriori $p(C_j|\mathbf{x}_n)$ for a maior dentre todas. Em outras palavras, tem-se a seguinte regra de decisão:

$$\text{Atribuir } \mathbf{x}_n \text{ à classe } C_j, \text{ se } p(C_j|\mathbf{x}_n) > p(C_i|\mathbf{x}_n), \forall i \neq j.$$

Este critério MAP também pode ser definido como

$$C_j = \arg \max_{i=1, \dots, K} \{p(C_i|\mathbf{x}_n)\}, \quad (3.3)$$

em que o operador "arg max" retorna o "argumento do máximo", ou seja, o conjunto de pontos para os quais a função de interesse atinge seu valor máximo.

² Do inglês, *likelihood function*.

Ao substituir a Eq. (3.2) na regra de decisão do critério MAP, obtém-se uma nova regra de decisão, dada por

$$\text{Atribuir } \mathbf{x}_n \text{ à classe } C_j, \text{ se } p(C_j)p(\mathbf{x}_n|C_j) > p(C_i)p(\mathbf{x}_n|C_i), \forall i \neq j,$$

em que o termo $p(\mathbf{x}_n)$ é eliminado por estar presente em ambos os lados da inequação. Em outras palavras, o termo $p(\mathbf{x}_n)$ não influencia na tomada de decisão feita por meio do critério MAP.

Na verdade, o critério MAP pode ser generalizado para usar qualquer *função discriminante* $g_i(\mathbf{x}_n)$, passando a ser escrito como

$$\text{Atribuir } \mathbf{x}_n \text{ à classe } C_j, \text{ se } g_j(\mathbf{x}_n) > g_i(\mathbf{x}_n), \forall i \neq j.$$

É importante ressaltar que, em um sentido amplo, uma função discriminante $g_i(\mathbf{x}_n)$ é qualquer função matemática que fornece um valor numérico que permita quantificar a pertinência do padrão \mathbf{x}_n à classe C_i . Assim, as classes podem ser ranqueadas (ordenadas) em função dos valores de suas respectivas funções discriminantes.

Vários tipos de funções discriminantes para classificadores gaussianos podem ser obtidos a partir da especificação da matriz de covariância, conforme será apresentado a seguir.

3.2.2 Classificadores Gaussianos

No contexto dos classificadores bayesianos gaussianos, uma das funções discriminantes mais utilizadas é dada por

$$\begin{aligned} g_i(\mathbf{x}_n) &= \ln p(C_i|\mathbf{x}_n), \\ &= \ln[p(C_i)p(\mathbf{x}_n|C_i)], \\ &= \ln p(C_i) + \ln p(\mathbf{x}_n|C_i), \end{aligned} \tag{3.4}$$

$$= g_i^{(1)}(\mathbf{x}_n) + g_i^{(2)}(\mathbf{x}_n), \tag{3.5}$$

em que $\ln(u)$ é a função logaritmo natural de u e a função $g_i^{(2)}(\mathbf{x}_n) = \ln p(\mathbf{x}_n|C_i)$ é chamada de função log-verossilhança da classe C_i .

Substituindo a função de verossilhança mostrada na Eq. (3.1) em $g_i^{(2)}(\mathbf{x}_n)$, chega-se à seguinte expressão:

$$g_i^{(2)}(\mathbf{x}_n) = \ln \left[\frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} Q_i(\mathbf{x}_n) \right\} \right], \tag{3.6}$$

$$= -\frac{1}{2} Q_i(\mathbf{x}_n) - \frac{p}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i|, \tag{3.7}$$

em que $Q_i(\mathbf{x}_n) = (\mathbf{x}_n - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_i)$.

Dado que o termo $-\frac{p}{2} \ln 2\pi$ é constante e aparece nas funções discriminantes de todas as classes ($i = 1, \dots, c$). Logo, este termo não influencia na tomada de decisão, podendo ser eliminado.

A partir disto, podem-se definir algumas configurações de discriminantes para o classificador bayesiano gaussiano:

- Caso 1: A função discriminante geral para o classificador gaussiano é dada por

$$g_i(\mathbf{x}_n) = -\frac{1}{2}Q_i(\mathbf{x}_n) - \frac{1}{2} \ln |\boldsymbol{\Sigma}_i| + \ln p(C_i). \quad (3.8)$$

- Caso 2: A partir da função discriminante geral do caso 1, uma suposição comumente feita na prática é a de que as densidades a priori das classes são iguais, ou seja

$$p(C_1) = P(C_2) = \dots = P(C_K), \quad (3.9)$$

o que equivale a supor que as classes são equiprováveis³. Com isto, é possível simplificar ainda mais a função discriminante mostrada na Eq. (3.8):

$$g_i(\mathbf{x}_n) = -\frac{1}{2}Q_i(\mathbf{x}_n) - \frac{1}{2} \ln |\boldsymbol{\Sigma}_i|, \quad (3.10)$$

uma vez que o termo $\ln p(C_i)$ é igual para todas as c funções discriminantes. Vale ressaltar que usar esta função discriminante equivale a reescrever o critério MAP como

$$\text{Atribuir } \mathbf{x}_n \text{ à classe } C_j, \text{ se } \ln p(\mathbf{x}_n|C_j) > \ln p(\mathbf{x}_n|C_i), \forall i \neq j,$$

de tal forma que a regra de decisão passa a depender somente das funções de log-verossimilhança das classes. Neste caso, o critério MAP passa a ser chamado de critério da máxima verossimilhança (*maximum likelihood criterion*, ML).

Além desta aproximação, existem algumas considerações que levam a outras configurações de classificadores gaussianos. São elas:

- Caso 3: As estruturas de covariâncias das K classes são iguais, ou seja, suas matrizes de covariância são iguais. Em outras palavras,

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \dots = \boldsymbol{\Sigma}_K = \boldsymbol{\Sigma}. \quad (3.11)$$

Neste caso, a função discriminante da classe C_i passa a ser escrita simplesmente como

$$g_i(\mathbf{x}_n) = -\frac{1}{2}Q_i(\mathbf{x}_n) = -\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_i), \quad (3.12)$$

³ Esta suposição pode ser encontrada na prática em situações nas quais o número de exemplos (padrões) por classe é aproximadamente igual.

em que o termo $-\frac{1}{2} \ln |\Sigma_i|$ foi eliminado por não influenciar mais na tomada de decisão. É possível notar que a função discriminante $g_i(\mathbf{x}_n)$ é proporcional a $Q_i(\mathbf{x}_n)$, que é a distância de Mahalanobis quadrática. Assim, pode-se fazer $g_i(\mathbf{x}_n) = Q_i(\mathbf{x}_n)$, de tal forma que o critério de decisão passa a ser escrito como

$$\text{Atribuir } \mathbf{x}_n \text{ à classe } C_j, \text{ se } Q_j(\mathbf{x}_n) < Q_i(\mathbf{x}_n), \forall i \neq j,$$

o que, em palavras, significa classificar \mathbf{x}_n como sendo da classe C_j se a distância (de Mahalanobis) de \mathbf{x}_n ao centróide da classe C_j (i.e. $\boldsymbol{\mu}_j$) for *menor* que as distâncias de \mathbf{x}_n aos centróides restantes.

Matriz de Covariância Agregada - Σ_{pool} : Uma forma muito comum de se implementar o classificador gaussiano cuja função discriminante é mostrada na Eq. (3.12) envolve o cálculo da matriz de covariância agregada, definida como

$$\begin{aligned} \Sigma_{pool} &= \left(\frac{n_1}{N}\right) \Sigma_1 + \left(\frac{n_2}{N}\right) \Sigma_2 + \cdots + \left(\frac{n_K}{N}\right) \Sigma_K, \\ &= p(C_1)\Sigma_1 + p(C_2)\Sigma_2 + \cdots + p(C_K)\Sigma_K, \\ &= \sum_{i=1}^K p(C_i)\Sigma_i, \end{aligned} \quad (3.13)$$

em que $p(C_i)$ é a probabilidade a priori da classe i . Percebe-se assim que a matriz Σ_{pool} é a média ponderada das matrizes de covariância das c classes, com os coeficientes de ponderação sendo dados pelas respectivas probabilidades a priori.

A matriz Σ_{pool} costuma ser melhor condicionada que as matrizes de covariância individuais e, por isso, sua inversa tende a causar menos problemas de instabilidade numérica.

- **Caso 4:** Os atributos de \mathbf{x}_n são descorrelacionados entre si e possuem mesma variância (que pode ser feita igual a 1). Neste caso, tem-se que a matriz de covariância de todas as classes é dada por

$$\Sigma = \mathbf{I}_p, \quad (3.14)$$

em que \mathbf{I}_p é a matriz identidade de ordem p . Logo, tem-se que $\Sigma^{-1} = \mathbf{I}_p$. Neste caso, a função discriminante da classe C_i passa a ser escrita como

$$g_i(\mathbf{x}_n) = -\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_i)^T \mathbf{I}_p (\mathbf{x}_n - \boldsymbol{\mu}_i), \quad (3.15)$$

$$= -\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_i)^T (\mathbf{x}_n - \boldsymbol{\mu}_i), \quad (3.16)$$

$$= -\frac{1}{2}\|\mathbf{x}_n - \boldsymbol{\mu}_i\|^2, \quad (3.17)$$

em que $\|\mathbf{u}\|^2$ denota a norma euclidiana quadrática de \mathbf{u} . Assim, pode-se fazer $g_i(\mathbf{x}_n) = \|\mathbf{x}_n - \boldsymbol{\mu}_i\|^2$, de tal forma que o critério de decisão passa a ser escrito como

$$\text{Atribuir } \mathbf{x}_n \text{ à classe } C_j, \text{ se } \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 < \|\mathbf{x}_n - \boldsymbol{\mu}_i\|^2, \forall i \neq j,$$

o que significa dizer que \mathbf{x}_n deve ser classificado como pertencente à classe C_j se a distância euclidiana de \mathbf{x}_n ao centróide $\boldsymbol{\mu}_j$ for *menor* que as distâncias de \mathbf{x}_n aos centróides restantes.

Este classificador também é chamado de classificador de máxima verossimilhança, pois o rótulo atribuído ao padrão de entrada é definido pela máxima proximidade deste ao centróide de determinada classe.

Nesta dissertação, os classificadores gaussianos cujas funções discriminantes estão descritas nos casos 2 (função não linear) e 4 (equação 3.10 e 3.17 respectivamente) foram utilizados.

No caso 4, chegou-se a um classificador linear a partir da suposição inicial de que os dados são distribuídos de forma Gaussiana. Na próxima seção, será mostrada a teoria sobre classificadores lineares, bem como um classificador linear específico: o baseado em mínimos quadrados ordinários

3.3 Classificador Linear de Mínimos quadrados

Nos classificadores lineares, busca-se obter um mapeamento linear entre os dados de entrada (no caso, vetores de atributos $\mathbf{x}_n \in \mathbb{R}^{p+1}$), e os dados de saída correspondentes (no caso, vetores $\mathbf{d}_n \in \mathbb{R}^K$ que indicam a qual classe pertence determinada amostra).

É importante mencionar que a dimensão do vetor \mathbf{x}_n é $p + 1$ pela adição de um viés b (também conhecido como *bias* ou *threshold*) aos atributos do problema.

Dado que tanto os vetores de atributos como os rótulos das classes são vetores coluna, este mapeamento pode ser realizado matematicamente pela equação

$$\mathbf{d}_n = \mathbf{W}\mathbf{x}_n, \quad (3.18)$$

onde $\mathbf{W} \in \mathbb{R}^{K \times (p+1)}$ é a matriz responsável pela transformação e cada elemento d_{ni} do vetor \mathbf{d}_n corresponde ao resultado da função discriminante da i -ésima classe. A representação gráfica deste classificador está ilustrada na Figura 4, onde \mathbf{X} e \mathbf{D} são, respectivamente, as matrizes que contêm os vetores de entrada e os vetores de saída de cada amostra.

Assim, a função discriminante destes classificadores consiste na multiplicação do vetor de entrada da amostra atual pela matriz \mathbf{W} , obtendo um vetor de saída. Então, esta amostra é atribuída à classe C_j , onde C_j corresponde a posição do vetor de saída no qual é encontrado o componente de máximo valor. Isto pode ser descrito como

$$\text{Atribuir } \mathbf{x}_n \text{ à classe } C_j, \text{ se } d_{nj} > d_{ni}, \forall i \neq j.$$

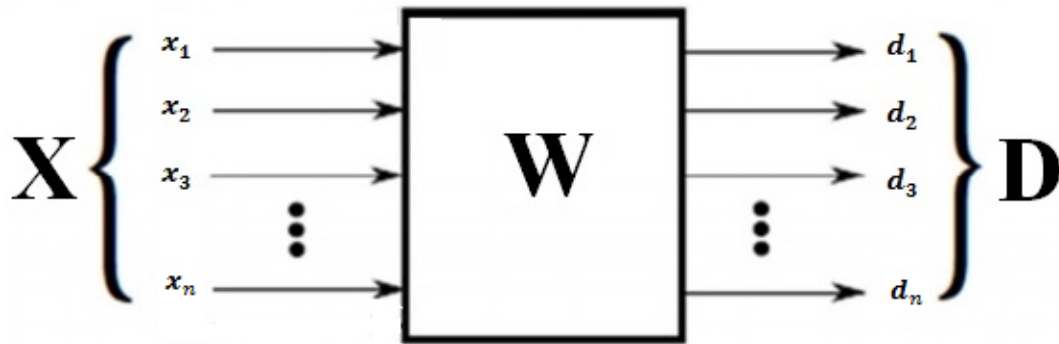


Figura 4 – Mapeamento linear dos dados. Adaptado (MONTEIRO, 2009).

A priori, não se sabe o valor dos elementos da matriz \mathbf{W} , mas existem várias técnicas, durante o treinamento dos classificadores, para estimar os pesos desta matriz a fim de minimizar os erros de atribuições das entradas às classes.

Uma destas técnicas é através do método dos mínimos quadrados ordinários, que é especificado a seguir.

3.3.1 Estimação da Matriz \mathbf{W} via Mínimos Quadrados Ordinários

Inicialmente, assume-se que se está de posse de N pares $\{\mathbf{x}_n, d_n\}_{n=1}^N$, onde \mathbf{x}_n e \mathbf{d}_n são, respectivamente, os vetores colunas de atributos e os de rótulos de determinada classe.

Agrupando-se estes vetores em matrizes, tem-se que $\mathbf{X} = [\mathbf{x}_1 | \mathbf{x}_2 | \dots | \mathbf{x}_n]$ é a matriz das amostras de entrada, e $\mathbf{D} = [\mathbf{d}_1 | \mathbf{d}_2 | \dots | \mathbf{d}_n]$ é a matriz dos rótulos de saída. Assim, a partir destas matrizes, a equação 3.18 pode ser representada matricialmente por

$$\mathbf{D} = \mathbf{W}\mathbf{X}, \quad (3.19)$$

na qual todos os vetores \mathbf{d}_n são calculados simultaneamente.

Se a matriz \mathbf{X} for quadrada, ou seja, se o número n de amostras for igual a $p + 1$, a matriz \mathbf{W} pode ser obtida através da seguinte equação:

$$\mathbf{W} = \mathbf{D}\mathbf{X}^{-1}, \quad (3.20)$$

Porém, normalmente, nos problemas de classificação, tem-se uma quantidade bem superior de amostras comparada ao número de atributos. Assim, \mathbf{X} não pode ser invertida diretamente.

Por isso, para obter \mathbf{W} , uma técnica bastante difundida é a matriz pseudoinversa (*Moore-Penrose Pseudoinverse*) (MOORE, 1920), (PENROSE, 1955), também conhecida como técnica dos mínimos quadrados ordinários (*Ordinary Least Squares - OLS*). Através desta, visa-se minimizar a soma quadrática dos erros na aproximação representada na equação 3.19.

Para isto, isola-se \mathbf{W} através da seguinte equação:

$$\mathbf{W} = \mathbf{D}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}, \quad (3.21)$$

onde a matriz $\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$ é chamada de matriz pseudoinversa de \mathbf{X} . A implementação desta será detalhada na seção a seguir.

Finalmente, após o cálculo da matriz \mathbf{W} , pode-se testar o desempenho do classificador OLS para as amostras que não foram utilizadas para treinamento, comparando a saída estimada pelo classificador, dada por

$$\vec{y} = \tilde{w}\vec{x}_i, \quad (3.22)$$

com os rótulos destas amostras, para validar se esta matriz representa (modela) bem o mapeamento entrada-saída dos dados.

3.3.2 Implementação da Inversão de Matrizes

Todos os algoritmos aplicados nesta dissertação foram implementados no software MATLAB. Assim, os meios para inverter matrizes partem do pressuposto que se está utilizando este software.

Inicialmente, considera-se o mesmo problema da equação 3.18, onde existe uma transformação linear de $\mathbf{X} \in \mathbb{R}^{(p+1) \times N}$ em $\mathbf{D} \in \mathbb{R}^{K \times N}$, através de $\mathbf{W} \in \mathbb{R}^{K \times (p+1)}$. O objetivo é estimar a matriz \mathbf{W} , usando as matrizes \mathbf{X} e \mathbf{D} , sabendo-se que $\mathbf{D} = \mathbf{W}\mathbf{X}$. Assim, dependendo dos valores de $p + 1$ e c , pode-se formular este problema do seguinte modo:

- $\mathbf{W} = \mathbf{D}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$ caso $c \neq p + 1$

onde a matriz $\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$ é chamada de matriz pseudoinversa de \mathbf{X} .

A matriz $\mathbf{X}\mathbf{X}^T$ a ser invertida pode estar mal condicionada ou até ser singular, e isto pode levar a resultados numericamente instáveis, o que não é desejável. Recomenda-se sempre avaliar o posto da matriz \tilde{X} , que é definido como

$$p(\tilde{x}) \leq \min(p + 1, N), \quad (3.23)$$

ou seja, o posto de uma matriz é, no máximo, igual ao número de linhas ou colunas (o menor deles). O posto é uma medida do número de linhas ou colunas linearmente independentes de uma matriz. Se $p(\tilde{x}) = \min(p + 1, N)$, então a matriz é dita de posto completo; caso contrário, ela é dita ser de posto incompleto.

Para evitar este problema, pode-se utilizar o método de regressão de cumeeira (*ridge regression*), definido para lidar com problemas mal-condicionados. O uso desse método, também conhecido como regularização de Tikhonov (TIKHONOV; ARSENIN, 1977) (HOERL; KENNARD, 1970), leva à seguinte estimativa da matriz \mathbf{W} :

- $\mathbf{W} = \mathbf{D}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + l\mathbf{I})^{-1}$ caso $c \neq p + 1$

em que $l > 0$ é a constante de regularização e \mathbf{I} é a matriz identidade de dimensões $(p + 1) \times (p + 1)$.

Existem as seguintes possibilidades para a estimação da matriz \mathbf{W} usando Matlab:

- Função `inv()`:
 - $W = DX' * inv(X * X')$; ou
 - $W = DX' * inv(X * X' + lI)$;
- Operadores Backlash ("`/`"): este método pode ser usado para solução da pseudoinversa, e utiliza decomposição QR como algoritmo. Dada a formulação acima, a pseudo inversa pode ser implementada como:
 - $W = D/X$
- `pinv()`: Assim como os operadores Backlash, este método pode ser utilizado para estimar a pseudoinversa de \tilde{X} , porém este utiliza decomposição em valores singulares (*Singular Value Decomposition* - SVD), e já trata melhor a questão do posto da matriz X . Assim, a pseudoinversa pode ser implementada simplesmente por:
 - $W = D * pinv(X)$;

No caso desta dissertação, o método `pinv()` foi utilizado toda vez que necessitou-se calcular a pseudoinversa de \tilde{X} . O método do operador barra invertida ("`/`") só pode ser usado se matriz \tilde{X} for de posto completo, limitação que não existe para a função `pinv()`.

Após entender os classificadores lineares e suas implementações, nas sessões a seguir, diferentes classificadores não-lineares serão evidenciados.

3.4 Classificadores Neurais

As redes neurais artificiais (HAYKIN, 2000) possuem, como base para sua estrutura, o modelo matemático de McCulloch e Pitts (1943). A partir deste, foram desenvolvidos vários algoritmos que são utilizados para a resolução de diversos problemas, incluindo a classificação de padrões.

Nesta dissertação, dois classificadores foram utilizados, baseados nas seguintes redes neurais: perceptron multicamadas (*multilayer perceptron* - MLP) e a máquina de aprendizado extremo (*extreme learning machine* - ELM). As redes MLP são conhecidas por serem aproximadoras universais de funções (HORNIK; STINCHCOMBE; WHITE, 1989), porém, o treinamento das redes ELM é bem mais rápido, mesmo possuindo arquitetura semelhante às redes MLP. O algoritmo de ambas é descrito a seguir.

É importante salientar que, durante este trabalho, ao se mencionar rede MLP, o autor se refere às redes MLP, com apenas uma camada escondida, treinadas pelo algoritmo de retropropagação do erro (*error backpropagation* - BP).

3.4.1 Perceptron Multicamadas

As redes MLP tem como base as redes Perceptron Simples (PS) (ROSENBLATT, 1958). Nestas, assim como no classificador OLS, busca-se um mapeamento linear entre os dados de entrada e saída.

Porém, as redes PS se diferenciam do classificador OLS pela sua regra de aprendizagem e pelo fato de que cada linha da matriz \mathbf{W} representa um neurônio artificial. Na figura 5, a arquitetura de um neurônio da rede neural perceptron simples é representada. Nesta, cada x_i representa um atributo (uma variável de entrada), cada w_i representa um parâmetro do neurônio (peso sináptico), w_0 é o limiar (*bias ou threshold*) associado a este neurônio, e ao bloco que relaciona $u(t)$ com $y(t)$ é definido como função de ativação do neurônio.

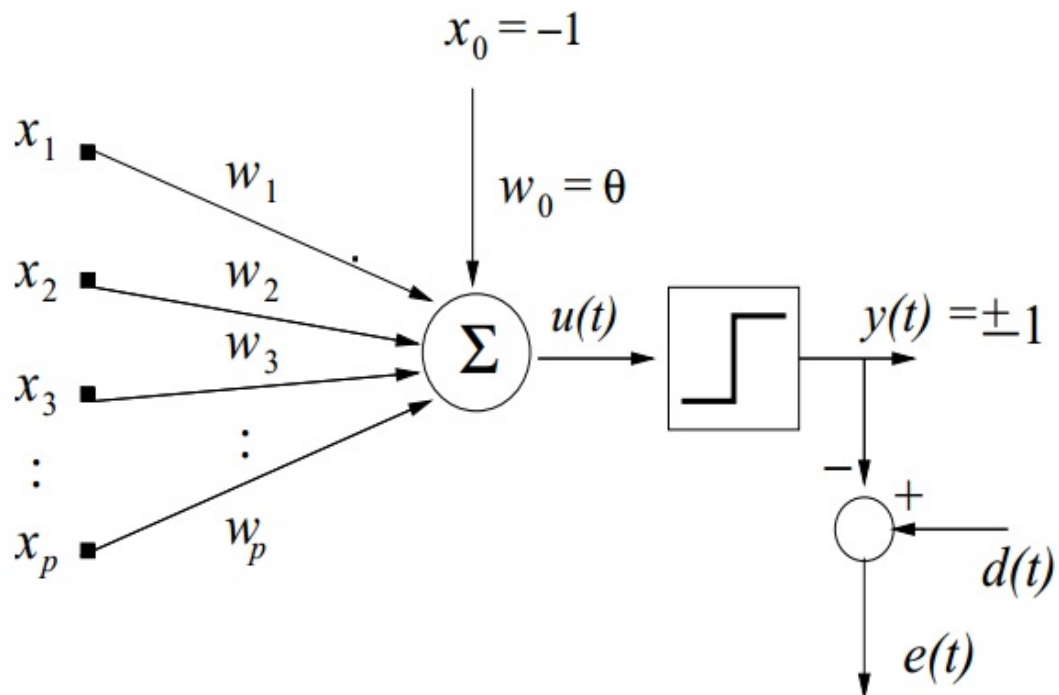


Figura 5 – Modelo matemático de um neurônio da rede PS

A função de ativação representada na figura 5 é a função sinal, onde a saída desta é -1 se sua entrada for negativa, e +1 se for positiva.

Assim, matematicamente, a relação entre entrada e saída de um neurônio é dada por

$$y(t) = \text{sign}(u(t)) = \text{sign}(\mathbf{w}_n \mathbf{x}_n), \quad (3.24)$$

Diferentemente das redes PS, as redes MLP, além da camada de neurônios que tem contato direto com os dados de saída (camada de saída), possuem uma ou mais camadas de neurônios ocultos (camada oculta ou camada escondida), que são responsáveis pelo processamento não linear da informação de entrada, de modo a facilitar a resolução do problema para os neurônios da camada de saída.

Uma rede MLP com uma camada oculta pode ser representada por: $MLP(p, q, c)$, onde p é o número de entradas, q é o número de neurônios ocultos e K o número de neurônios de saída. Assim, o número de parâmetros deste classificador é dado por: $L_{mlp} = (p + 1)q + (q + 1)K$.

As especificações de p e K são ditadas pela forma como o problema em questão é tratado. Num caso específico de classificação de padrões por exemplo, p é definido pelo número de atributos, e K pelo número de classes do problema.

Por outro lado, não há uma heurística definida para calcular a quantidade de neurônios ocultos (q) de uma rede MLP. Esta depende da complexidade do problema, e devem ser realizados vários testes até que o valor mais adequado seja encontrado. Assim, este número é um hiperparâmetro da rede MLP.

Além das diferenças previamente citadas, nos problemas de classificação, a função de ativação utilizada pela rede MLP possui uma não linearidade suave, podendo obter em sua saída, diversos valores no intervalo $[0, 1]$ (caso a função utilizada seja do tipo sigmoideal), ou valores no intervalo $[-1, +1]$ (caso a função utilizada seja do tipo tangente hiperbólica). Por outro lado, em casos de aproximação de funções, a saída pode ser linear.

Nesta dissertação, a tangente hiperbólica foi escolhida para ser utilizada como função de ativação. Esta é definida como

$$y_i(t) = \frac{1 - \exp(-u_i(t))}{1 + \exp(-u_i(t))}, \quad (3.25)$$

onde $y_i(t)$ é a saída do i -ésimo neurônio de uma camada na iteração t , e $u_i(t)$ é o produto escalar entre o vetor de pesos do neurônio e a sua atual entrada.

O algoritmo da retropropagação do erro (*Error Backpropagation* - BP) (RUMELHART; HINTON; WILLIAMS, 1986) é utilizado para atualizar os parâmetros da rede MLP. Este será descrito a seguir.

Inicialmente, define-se que o vetor associado a cada neurônio i da camada oculta, é representado por $\mathbf{w}_i = [w_{i0} \dots w_{ip}]^T$, em que w_{i0} é o limiar associado ao neurônio i .

Por outro lado, a cada neurônio k da camada de saída, é associado um vetor de pesos representado por $\mathbf{m}_k = [m_{k0} \dots m_{kq}]^T$, em que m_{k0} é o limiar associado ao neurônio k , e q é o número de neurônios da camada oculta. Ambas representações estão ilustradas na figura 6, em que $\phi(u_i)$ representa a função de ativação (tangente hiperbólica).

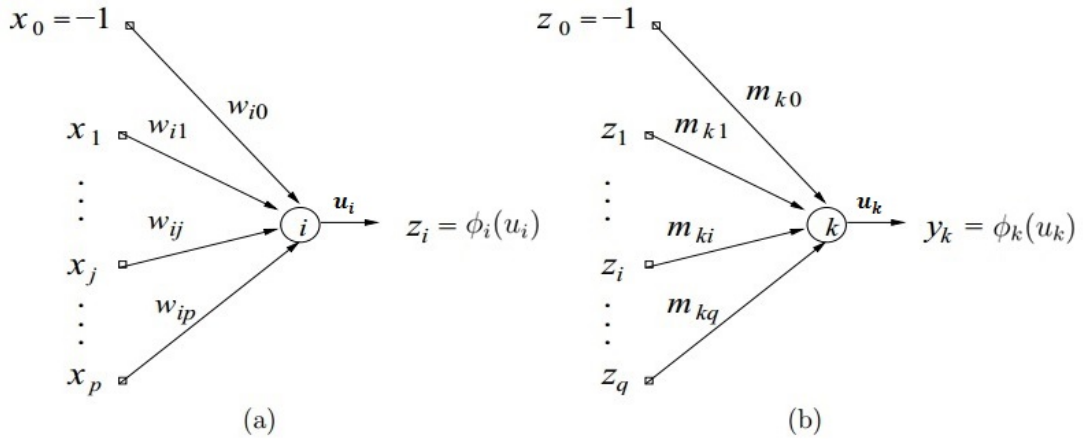


Figura 6 – Modelo dos neurônios da rede MLP (a) camada oculta (b) camada de saída

A partir dessas definições, o algoritmo de treinamento BP se divide em duas fases: sentido direto e sentido inverso.

No sentido direito (*forward*), o fluxo de informações se dá dos neurônios de entrada para os neurônios de saída, passando pelos da camada escondida. Essa fase se divide em dois passos:

Primeiramente, após a apresentação de um vetor de entrada \mathbf{x}_n , na iteração t , à rede MLP, a saída de cada neurônio da camada escondida é calculada por

$$z_i(t) = \phi_i(u_i(t)) = \phi_i\left(\sum_{j=0}^p w_{ij}(t)x_j(t)\right) = \phi_i(\mathbf{w}_i^T(t)\mathbf{x}(t)). \quad (3.26)$$

Em seguida, a saída de cada neurônio da camada de saída é dada por

$$y_k(t) = \phi_k(u_k(t)) = \phi_k\left(\sum_{i=0}^q m_{ki}(t)z_i(t)\right) = \phi_k(\mathbf{m}_k^T(t)\mathbf{z}(t)). \quad (3.27)$$

Após estes dois passos, inicia-se o sentido inverso (*backward*), que é a fase onde os parâmetros (pesos sinápticos) dos neurônios são calculados. Nesta fase, o fluxo de informações se dá dos neurônios de saída, para os neurônios da camada oculta.

Após calcular, no sentido direto, as ativações e saídas de cada neurônio, o primeiro passo da segunda fase de treinamento é calcular os gradientes locais dos neurônios de saída

$$\delta_k(t) = e_k(t)\phi'(u_k(t)), \quad (3.28)$$

em que $e_k(t)$ é o erro entre a saída desejada $d_k(t)$ para o neurônio k e a saída gerada por ele, $y_k(t)$:

$$e_k(t) = d_k(t) - y_k(t), \quad (3.29)$$

Como a tangente hiperbólica foi escolhida como função de ativação, a derivada $\phi'(u_k(t))$ é calculada por

$$\phi'_k(u_k(t)) = \frac{d\phi_k(u_k(t))}{du_k(t)} = \frac{1}{2} [1 - y_k^2(t)]. \quad (3.30)$$

O segundo passo do sentido inverso consiste em calcular os gradientes locais dos neurônios da camada escondida. Assim, para cada neurônio i desta camada, utiliza-se a seguinte expressão:

$$\delta_i(t) = \phi'(u_i(t)) \sum_{k=1}^n m_{ki} \delta_k(t), \quad (3.31)$$

em que, assim como para os neurônios de saída, a derivada $\phi'(u_i(t))$ é dada por

$$\phi'_i(u_i(t)) = \frac{d\phi_i(u_i(t))}{du_i(t)} = \frac{1}{2} [1 - y_i^2(t)]. \quad (3.32)$$

Por fim, o último passo da fase 2, corresponde à atualização ou ajuste de parâmetros para a camada de saída e para as camadas ocultas. No caso de uma MLP com apenas uma camada oculta, a regra de atualização dos pesos desta é dada por

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) = w_{ij}(t) + \eta \delta_i(t) x_j(t), \quad (3.33)$$

em que η é a taxa de aprendizagem da rede (um hiperparâmetro desta). Por fim, para a camada de saída, a atualização dos pesos se dá por

$$m_{ki}(t+1) = m_{ki}(t) + \Delta m_{ki}(t) = m_{ki}(t) + \eta \delta_k(t) z_i(t). \quad (3.34)$$

A estas duas funções de adaptação, também pode ser adicionado um termo adicional, chamado "termo de momento", cujo objetivo é tornar o processo de aprendizagem mais estável. Com estes termos, as equações 3.33 e 3.34 passam a ser dadas por

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_i(t) x_j(t) + \lambda \Delta w_{ij}(t-1), \quad (3.35)$$

e

$$m_{ki}(t+1) = m_{ki}(t) + \eta \delta_k(t) z_i(t) + \lambda \Delta m_{ki}(t-1), \quad (3.36)$$

em que λ é um hiperparâmetro da rede MLP, e é chamado "fator de momento". Este fator, por questões de estabilidade de aprendizagem, deve ser mantido, em geral, na faixa de valores entre 0,5 e 1.

Após apresentado o algoritmo de treinamento para a rede MLP, também será evidenciada uma classe de redes neurais supervisionadas, com uma única camada oculta, chamada *Máquina de Aprendizado Extremo* ou simplesmente rede ELM, proposta por Huang, Zhu e Ziew (2006).

3.4.2 Máquinas de Aprendizado Extremo

Diferentemente da rede MLP, na rede ELM, os pesos entre as camadas de entrada e oculta são escolhidos aleatoriamente, enquanto os pesos entre as camadas oculta e de saída são determinados analiticamente.

A rede ELM é uma rede que apresenta rápida velocidade de aprendizado e facilidade de implementação (HUANG; WANG; LAN, 2011) e, devido principalmente a isso, vários autores têm aplicado a rede ELM (e as mais sofisticadas variantes dela) a uma ampla gama de problemas complexos de classificação de padrões e regressão (NEUMANN; STEIL, 2013; HORATA; CHIEWCHANWATTANA; SUNAT, 2012; MOHAMMED et al., 2011; ZONG; HUANG, 2011; MICHE et al., 2011; MICHE et al., 2010; LIU; WANG, 2010; DENG; ZHENG; CHEN, 2009).

Mais especificamente, a rede ELM é uma rede *feedforward* com uma única camada oculta, que oferece menor necessidade de intervenção humana, no que se refere ao ajuste de seus parâmetros (pesos e limiares), quando comparada a redes *feedforward* mais tradicionais, tais como as redes MLP (*Perceptron multicamadas*) e RBF (Funções de Base Radial).

Assume-se que N pares de dados $\{(\mathbf{x}_n, \mathbf{d}_n)\}_{n=1}^N$ estejam disponíveis para construir e avaliar o modelo, em que $\mathbf{x}_n \in \mathbb{R}^{p+1}$ é o n -ésimo padrão de entrada⁴ e $\mathbf{d}_n \in \mathbb{R}^K$ é o rótulo da classe alvo correspondente, com K denotando o número de classes.

Seleciona-se aleatoriamente, então, N_1 ($N_1 < N$) pares de dados a partir do conjunto de dados disponível e os organizando em colunas das matrizes \mathbf{D} e \mathbf{X} , tem-se que:

$$\mathbf{X} = [\mathbf{x}_1 \mid \mathbf{x}_2 \mid \cdots \mid \mathbf{x}_{N_1}] \quad \text{e} \quad \mathbf{D} = [\mathbf{d}_1 \mid \mathbf{d}_2 \mid \cdots \mid \mathbf{d}_{N_1}]. \quad (3.37)$$

em que $\dim(\mathbf{X}) = (p + 1) \times N_1$ e $\dim(\mathbf{D}) = K \times N_1$.

Para uma rede com $p + 1$ unidades de entrada, q neurônios ocultos e K saídas, a i -ésima saída, para o n -ésimo padrão de entrada \mathbf{x}_n , é dada por

$$y_{in} = \mathbf{w}_i^T \mathbf{z}_n, \quad (3.38)$$

em que $\mathbf{w}_i \in \mathbb{R}^q$, $i = 1, \dots, K$, é o vetor peso conectando os neurônios ocultos ao i -ésimo neurônio de saída, e $\mathbf{z}_n \in \mathbb{R}^q$ é o vetor de saídas dos neurônios ocultos para um dado padrão de entrada $\mathbf{x}(t) \in \mathbb{R}^p$. O vetor $\mathbf{z}(t)$ propriamente dito é definido como

$$\mathbf{z}_n = [f(\mathbf{m}_1^T \mathbf{x}_n + b_1) \quad f(\mathbf{m}_2^T \mathbf{x}_n + b_2) \quad \cdots \quad f(\mathbf{m}_q^T \mathbf{x}_n + b_q)]^T, \quad (3.39)$$

em que b_l , $l = 1, \dots, q$, é o limiar (*bias*) do l -ésimo neurônio oculto, $\mathbf{m}_l \in \mathbb{R}^{p+1}$ é o vetor de pesos do l -ésimo neurônio oculto e $f(\cdot)$ é uma função de ativação sigmoideal ou de base radial. Usualmente, os vetores de peso \mathbf{m}_l são aleatoriamente amostrados a partir de uma distribuição uniforme ou normal.

⁴ Primeira componente de \mathbf{x}_n é igual a 1 para possibilitar a inclusão do bias.

Define-se $\mathbf{Z} = [\mathbf{z}(1) \ \mathbf{z}(2) \ \cdots \ \mathbf{z}(N_1)]$ como uma matriz $q \times N_1$ cujas N_1 colunas são os vetores de saída da camada oculta $\mathbf{z}_n \in \mathbb{R}^q$, $n = 1, \dots, N_1$, em que N_1 é o número de padrões de entrada disponíveis para treinamento. Similarmente, seja $\mathbf{D} = [\mathbf{d}(1) \ \mathbf{d}(2) \ \cdots \ \mathbf{d}(N_1)]$ uma matriz $K \times N_1$ cuja n -ésima coluna é o vetor alvo (desejado) $\mathbf{d}_n \in \mathbb{R}^K$ associado com o padrão de entrada \mathbf{x}_n , $n = 1, \dots, N_1$. Finalmente, seja \mathbf{W} uma matriz $K \times q$, cuja i -ésima linha é o vetor de pesos $w_i^T \in \mathbb{R}^q$, associado ao i -ésimo neurônio de saída, $i = 1, \dots, K$.

Assim, essas três matrizes estão relacionadas pelo seguinte mapeamento linear:

$$\mathbf{D} = \mathbf{WZ}, \quad (3.40)$$

em que as matrizes \mathbf{D} e \mathbf{Z} são conhecidas, enquanto a matriz de pesos \mathbf{W} não. A solução baseada no critério OLS do sistema linear da Eq. (3.40) é dada pela inversa generalizada de Moore-Penrose como

$$\mathbf{W} = \mathbf{DZ}^T (\mathbf{ZZ}^T)^{-1}. \quad (3.41)$$

A expressão mostrada na Eq. (3.41) pode ser dividida em K equações de estimação individuais, uma para cada neurônio de saída i , ou seja

$$\mathbf{w}_i = (\mathbf{ZZ}^T)^{-1} \mathbf{ZD}_i^T, \quad i = 1, \dots, c, \quad (3.42)$$

em que \mathbf{D}_i representa a i -ésima linha da matriz \mathbf{D} .

No Apêndice A.1 se encontra a implementação deste classificador no software Matlab.

Após finalizado o estudo sobre os classificadores neurais, será iniciado o estudo dos baseados em vetores de suporte.

3.5 Classificadores Via Vetores de Suporte

A base dos classificadores via vetores de suporte está fundamentada na teoria do aprendizado estatístico (VAPNIK, 1998), (VAPNIK, 2000), onde tenta-se obter a minimização do risco empírico e do risco estrutural.

Entende-se por risco empírico, como o erro obtido pelo classificador, ao buscar separar as amostras de duas classes durante o treinamento. Já o risco estrutural, está relacionado à complexidade da função que o classificador gerou para separar as classes.

Assim, visando obter o melhor resultado de separabilidade dos padrões de treinamento, através de uma função menos complexa possível, o processo de aprendizagem supervisionado busca obter uma boa capacidade de generalização (capacidade de separar os dados de testes entre as classes do problema).

Inicialmente, as máquinas de vetores de suporte (*support vector machine* - SVM), foram introduzidas para solucionar problemas binários de reconhecimento de padrões (BUR-

GES, 1998). Mesmo existindo formas de, com estes classificadores, solucionar problemas multiclases, nesta dissertação, utilizou-se apenas o classificador binário.

O algoritmo para a classificação binária de padrões, através do SVM, é explicado a seguir. Em seguida, também será detalhado o classificador baseado nas máquinas de vetores de suporte por mínimos quadrados (*least squares SVM - LSSVM*).

3.5.1 Máquinas de Vetor de Suporte

Para obter o algoritmo não-linear do classificador SVM, faz-se necessário apresentar os classificadores SVM lineares de margem rígida e de margem flexível.

De maneira geral, assim como outros algoritmos de aprendizado de máquina, no problema de classificação, os algoritmos baseados em vetores de suporte buscam um hiperplano (problema linear) ou uma hiper superfície (problema não linear) que separa os dados de duas classes, a partir de alguns exemplos destas.

Inicialmente, considerando um problema linearmente separável, as soluções deste podem ser representadas, matematicamente, pela equação de um hiperplano, dada por

$$\mathbf{w}^T \mathbf{x}_n + b = 0 \quad (3.43)$$

em que \mathbf{w} é um vetor de pesos, b é o viés (também conhecido como *bias*, *threshold* ou *limiar*), e \mathbf{x}_n é uma amostra de entrada.

Caso o hiperplano consiga colocar todos as amostras de uma determinada classe em posição oposta ao da outra, este representará uma solução para o problema em estudo. Neste caso, este hiperplano deve obedecer as seguintes restrições:

$$\mathbf{w}^T \mathbf{x}_i + b > a \rightarrow d_i = +1$$

$$\mathbf{w}^T \mathbf{x}_i + b < -a \rightarrow d_i = -1$$

em que $a > 0$, e \mathbf{x}_i representa a i -ésima amostra de treinamento.

Neste tipo de problema, há vários hiperplanos que podem dividir as amostras de entrada em dois espaços multidimensionais distintos.

Dentre estes, deve-se escolher como solução o hiperplano que possua a máxima distância em relação aos padrões mais próximos de treinamento, ou seja, possua a maior margem de separação⁵. Este hiperplano é definido como ótimo e representado pela equação

$$\mathbf{w}_o^T \mathbf{x}_n + b_o = 0. \quad (3.44)$$

Assim, para problemas de classificação binários, a função discriminante do classificador SVM pode ser definida como:

$$g(\mathbf{x}) = \text{sign}(\mathbf{w}_o^T \mathbf{x}_n + b_o) \quad (3.45)$$

⁵ distância entre hiperplano e amostra de treinamento mais próxima

de modo que

$$\begin{aligned}\mathbf{w}^T \mathbf{x}_i + b < 0 &\rightarrow f(\mathbf{x}) = -1, \\ \mathbf{w}^T \mathbf{x}_i + b \geq 0 &\rightarrow f(\mathbf{x}) = +1.\end{aligned}$$

Para obter o hiperplano ótimo, ou seja, para encontrar os valores de w_o e b_o a partir dos dados de treinamento, pode-se, inicialmente, impor ao problema da equação 3.44, as seguintes restrições

$$\begin{aligned}\mathbf{w}^T \mathbf{x}_i + b \geq +1 &\rightarrow d_i = +1 \\ \mathbf{w}^T \mathbf{x}_i + b \leq -1 &\rightarrow d_i = -1\end{aligned}$$

onde estas equações podem ser resumidas por

$$d_i(\mathbf{w}^T \mathbf{x}_i + b) \geq +1. \quad (3.46)$$

As amostras de treinamento que satisfazem a igualdade na equação anterior são denominadas vetores de suporte e são as que possuem a menor distância ao hiperplano ótimo.

Durante o treinamento dos classificadores SVM, busca-se maximizar a margem de separação. Isto é conseguido através da minimização da norma do vetor de pesos (VAPNIK, 2000) (HAYKIN, 2000), representada pela função

$$J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} = \frac{1}{2} \|\vec{w}\|^2 \quad (3.47)$$

onde $J(\mathbf{w})$ é a função a ser minimizada.

Ao maximizar a margem de separação, infere-se que a dimensão VC (Vapnik-Chervonenkis)⁶ é minimizada.

Assim, dada as equações 3.46 3.47, tem-se que o treinamento de um classificador SVM pode ser convertido em um problema de otimização, onde busca-se minimizar $J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}$ com a restrição $d_i(\mathbf{w}^T \mathbf{x}_i + b) \geq +1$.

O problema de otimização com restrição anteriormente citado é conhecido como problema primal, e pode ser solucionado pelo método de Lagrange, utilizando a seguinte função lagrangeana:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^N \alpha_i (d_i (\mathbf{x}_i^T \mathbf{w} + b) - 1) \quad (3.48)$$

onde α_i são os multiplicadores de Lagrange, e estes são não-negativos.

⁶ dimensão associada à complexidade da função discriminante, ou seja, ao risco estrutural.

A solução deste problema de otimização é determinado pelo ponto de sela da função lagrangeana (minimizada em relação a \mathbf{w} e b e maximizada em relação ao vetor de coeficientes de Lagrange α). Essa minimização é feita ao substituir os resultados das derivadas parciais da Equação 3.48 por \mathbf{w} e b nesta própria equação.

Nestas condições, desenvolvendo-se a equação 3.48, chega-se ao problema de otimização dual, dado por

$$\max(L(\alpha)) = \max \left\{ \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j \right\}, \quad (3.49)$$

$$\begin{aligned} \sum_{i=1}^N \alpha_i d_i &= 0 \\ \alpha_i &\geq 0 \quad \forall i \end{aligned}$$

Solucionando-se este problema, encontram-se os multiplicadores de Lagrange ótimos α_i^o . Por fim, através dos multiplicadores de lagrange ótimos, computam-se o hiperplano ótimo \mathbf{w}_o e o viés ótimo pelas equações

$$\mathbf{w}_o = \sum_{i=1}^N \alpha_i^o d_i \mathbf{x}_i \quad (3.50)$$

e

$$b_o = 1 - \mathbf{w}_o^T \mathbf{x}_{(s)} \quad (3.51)$$

quando $d_{(s)} = 1$, em que $(x_{(s)}, d_{(s)})$ representam um vetor de suporte.

Assim, reescrevendo-se a função discriminante definida pela equação 3.45, utilizando os resultados das Equações (3.50) e (3.51), tem-se que

$$g(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^N \alpha_i^o d_i \mathbf{x}_i^T \mathbf{x} + b_o \right) \quad (3.52)$$

No problema de otimização primal desenvolvido até o momento (também conhecido como SVM de margem rígida), considerou-se que existe um hiperplano que separa totalmente as duas classes, sem que ocorra nenhum erro. Porém, muitas vezes, esta condição não é atingida por haver sobreposição entre alguns padrões pertencentes a classes distintas, ou pela presença de *outliers*.

A flexibilização da margem de separação entre as classes é um meio para levar em conta também estas situações. A partir desta, alguns erros, durante o treinamento do algoritmo, são permitidos de modo a reduzir a complexidade da função discriminante e melhorar o desempenho desta para o conjunto de teste.

Assim, esta flexibilização se dá a partir de um relaxamento das restrições do SVM de margens rígidas, e pode ser formulada por

$$d_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad (3.53)$$

onde os limiares $\xi_i \geq 0$ são chamados de variáveis de folga.

A partir disto, o problema de otimização primal dos classificadores SVM de margem flexível pode ser definido como

$$J(\mathbf{w}, \xi) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \xi_i \quad (3.54)$$

$$\begin{aligned} d_i(\mathbf{w}^T \mathbf{x}_i + b) &\geq 1 - \xi_i \quad \forall i \\ \xi_i &\geq 0 \quad \forall i \end{aligned}$$

onde C é a constante responsável pela regularização entre o primeiro e o segundo termo da função a ser otimizada. Esta constante é um hiperparâmetro dos classificadores SVM de margem flexível.

A partir do problema formulado anteriormente, pode-se perceber que tanto tenta-se maximizar a margem de separação entre as classes (através da minimização de $\frac{1}{2} \mathbf{w}^T \mathbf{w}$), como tenta-se minimizar os valores das variáveis de folga (ξ_i).

Do mesmo modo que os classificadores de margem rígida, os de margem flexível também podem ser resolvidos através de uma função lagrangeana, formulada por:

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i (d_i(\mathbf{x}_i^T \mathbf{w} + b) - 1 + \xi_i) - \sum_{i=1}^N \beta_i \xi_i, \quad (3.55)$$

em que α e β são multiplicadores de lagrange, ξ são as variáveis de folga e todos os elementos destes conjuntos são não-negativos.

A solução deste problema é determinada ao se minimizar a função lagrangeana em relação à \mathbf{w} , b e ξ_i . Nestas condições, e desenvolvendo-se a Equação 3.55, chega-se ao problema de otimização dual, dado por

$$\begin{aligned} \max(L(\alpha)) &= \max \left\{ \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j \right\} \quad (3.56) \\ &\sum_{i=1}^N \alpha_i d_i = 0 \\ &0 \leq \alpha_i \leq C \quad \forall i \end{aligned}$$

Como pode-se perceber, a única diferença entre este problema dual e o da Equação (3.49), é a restrição aplicada aos multiplicadores de lagrange α_i . Assim, tanto o classificador

de margem rígida, como o de margem flexível são formatados para solucionar problemas linearmente separáveis. Porém, existem diversos problemas reais onde uma separação linear dos dados entre classes não é possível.

Por isso, para tratar este tipo de problemas, funções de kernel $\kappa(\mathbf{x}, \mathbf{x}_i)$ são aplicadas aos classificadores SVM, de modo que os dados possam ser mapeados em um espaço de características de ordem superior, fazendo com que, neste novo espaço, o problema se torne linear. Ao utilizar estas funções, evita-se o uso explícito de um espaço de características de elevada dimensão, trabalhando-se indiretamente neste.

Dentre os tipos de kernels, o linear, o polinomial, e o gaussiano são muito utilizados (HAYKIN, 2000). Nesta dissertação, optou-se pelo kernel gaussiano, formulado por

$$\kappa(\mathbf{x}, \mathbf{x}_i) = \exp \left\{ -\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma^2} \right\} \quad (3.57)$$

onde σ^2 é um hiperparâmetro a ser definido e $\|\mathbf{x} - \mathbf{x}_i\|^2$ é a distância euclidiana entre as amostras \mathbf{x} e \mathbf{x}_i

Com a aplicação de um kernel ao classificador SVM, o problema de otimização dual, para o caso de margem flexível, torna-se

$$\max(L(\alpha)) = \max \left\{ \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \right\}, \quad (3.58)$$

$$\begin{aligned} \sum_{i=1}^N \alpha_i d_i &= 0 \\ 0 &\leq \alpha_i \leq C \quad \forall i \end{aligned}$$

e, por fim, sua função discriminante pode ser reescrita como

$$g(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^N \alpha_i d_i \kappa(\mathbf{x}_i, \mathbf{x}) + b_o \right) \quad (3.59)$$

Por fim, para solucionar o problema de otimização dual, no software Matlab, utilizou-se a função `quadprog()`. Esta tem por finalidade solucionar um conjunto de equações quadráticas, podendo utilizar diversos algoritmos. Nesta dissertação, optou-se pelo algoritmo *Interior Point Convex*⁷.

Um variante do classificador SVM é o SVM por mínimos quadrados (*least squares SVM - LSSVM*), que reduz o problema de programação quadrático dos SVM a um sistema linear. O algoritmo do classificador LSSVM é descrito a seguir.

⁷ <http://www.mathworks.com/help/optim/ug/quadprog.html>

3.5.2 Máquinas de Vetor de Suporte por Mínimos Quadrados

Assim como o classificador SVM, o classificador LSSVM (SUYKENS; VANDEWALLE, 1999a) também é capaz de solucionar problemas binários de classificação, porém sua formulação faz com que seu treinamento seja simplificado em relação ao do classificador SVM.

Levando em consideração o classificador de margens flexíveis, o classificador LSSVM apresenta duas diferenças em relação ao SVM. Primeiramente, a restrição de desigualdade presente no SVM torna-se uma de igualdade. Além disso, na função custo, leva-se em conta a soma dos quadrados das variáveis de folga.

Deste modo, o problema primal de otimização do classificador LSSVM torna-se

$$J(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \frac{1}{2} \sum_{i=1}^n \xi_i^2 \quad (3.60)$$

$$d_i ((\mathbf{w}^T \mathbf{x}) + b) = 1 - \xi_i \quad \forall i$$

onde γ , assim como o parâmetro C nos classificadores SVM, tem por finalidade regularizar a função de otimização. Vale ressaltar que, no problema acima, as variáveis de folga ξ_i podem possuir valores negativos. Este problema também pode ser solucionado por uma função lagrangeana, dada por

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \frac{1}{2} \sum_{i=1}^n \xi_i^2 - \sum_{i=1}^n \alpha_i (d_i ((\mathbf{w}^T \mathbf{x}) + b) - 1 + \xi_i) \quad (3.61)$$

em que as condições de otimização (minimização de \mathbf{w} , b e ξ_i através de gradientes, e maximização de α_i) geram, respectivamente, as seguintes funções:

$$\begin{aligned} \mathbf{w} &= \sum_{i=1}^n \alpha_i x_i d_i \\ \sum_{i=1}^n \alpha_i d_i &= 0 \\ \alpha_i &= \gamma \xi_i \\ d_i (\mathbf{x}_i^T \mathbf{w} + b) - 1 + \xi_i &= 0 \end{aligned} \quad (3.62)$$

Pode-se rearranjar as igualdades acima, de modo a representar o problema do classificador por um sistema de equações lineares, dado por:

$$\begin{bmatrix} 0 & \mathbf{d}^T \\ \mathbf{d} & \boldsymbol{\Omega} + \gamma^{-1} \mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix} \quad (3.63)$$

em $\Omega_{i,j} = d_i d_j \mathbf{x}_i^T \mathbf{x}_j$, $\mathbf{d} = [d_1 \ d_2 \ \dots \ d_n]^T$, $\boldsymbol{\alpha} = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_n]^T$ e $\mathbf{1} = [1 \ 1 \ \dots \ 1]^T$.

Para solucionar este sistema de equações lineares, utilizou-se, no software Matlab, a função `linsolve()`. Esta funciona problemas do tipo $\mathbf{Ax} = \mathbf{b}$, e utiliza dois algoritmos principais para isto: fatoração LU com pivoteamento parcial (se a matriz \mathbf{A} for quadrada), ou fatoração QR com pivoteamento por coluna (se as dimensões da matriz \mathbf{A} forem diferentes).

Para aplicar este classificador à problemas não-lineares, também pode-se utilizar o mesmo conceito de kernel usado nos classificadores SVM. Com isso, a matriz $\mathbf{\Omega}$ pode ser redefinida como $\Omega_{i,j} = d_i d_j \kappa(\mathbf{x}_i, \mathbf{x}_j)$ e a saída do classificador (função discriminante), fica definida como:

$$g(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^I \alpha_i d_i \kappa(\mathbf{x}, \mathbf{x}_i) + b \right). \quad (3.64)$$

Como citado anteriormente, uma grande vantagem do LSSVM em relação ao SVM é a diminuição da complexidade para resolução do problema de otimização. Porém, o classificador LSSVM também possui limitações: a principal se refere aos valores não-nulos dos multiplicadores de Lagrange, o que significa perda na esparsidade da solução do problema, pois todos os elementos de treinamento deverão ser armazenados. Isto pode trazer problemas para um banco de dados com muitas amostras.

3.6 Classificação Via Regressão Baseada em Distâncias

Inicialmente projetado para problemas de regressão, a máquina de aprendizado mínimo (*minimal learning machine* - MLM) é um algoritmo recente de aprendizado supervisionado que pressupõe a existência de um mapeamento entre as configurações geométricas de pontos no espaço de entrada e as configurações geométricas dos respectivos pontos no espaço de saídas (Souza Jr. et al., 2015).

Este mapeamento é representado por matrizes de distância de entrada e saída que são calculadas com respeito aos padrões de entrada e saídas desejadas e dois conjuntos de pontos de referência tomados nos espaço de entrada e saída respectivamente. Assim, a capacidade de generalização destes modelos vem do mapeamento linear entre estas matrizes de distância (Souza Jr. et al., 2015).

Primeiramente, o banco de dados deve ser dividido em duas partes: conjunto de treinamento e conjunto de teste. Feito isto, R pontos de referência devem ser aleatoriamente escolhidos. Estes pontos de referência correspondem a algumas amostras do conjunto de treinamento. Cada amostra possui suas entradas (atributos) e saídas (rótulos). O número de pontos de referência é o único hiperparâmetro deste algoritmo. O modo como este número é definido será especificado no capítulo 4.

A etapa de treinamento é iniciada calculando-se $D_x \in \mathbb{R}^{N_1 \times K}$, onde cada elemento d_{nk} representa a distância da n -ésima amostra de treinamento até a entrada do k -ésimo

ponto de referência (representada por \mathbf{m}_x). Neste trabalho, a distância Euclidiana é utilizada para calcular os elementos d_{nk} .

O segundo passo desta etapa, consiste em calcular $\Delta_y \in \mathbb{R}^{N_1 \times K}$, onde cada elemento desta matriz representa a distância de cada saída das amostras do conjunto de treinamento a cada saída dos pontos de referência (representadas por \mathbf{t}_y).

Por fim, deve-se estimar um modelo de regressão linear relacionando as matrizes D_x e Δ_y . Este modelo linear é representado na equação 3.65, onde E corresponde aos resíduos, e $B \in \mathbb{R}^{K \times K}$ é uma matriz $R \times R$ que deve ser estimada a partir dos dados. Esta estimação pode ser feita pelo método da pseudo-inversa, representado na equação 3.66.

$$\Delta_y = D_x \cdot B + E \quad (3.65)$$

$$\hat{B} = (D_x' D_x)^{-1} D_x' \Delta_y \quad (3.66)$$

Já a etapa de teste consiste de, para cada amostra de teste \mathbf{x}_i , calcular um vetor $d \in \mathbb{R}^{1 \times R}$, onde cada elemento representa a distância entre as entradas dos pontos de referências e da amostra de teste \mathbf{x}_i .

Em seguida, as distâncias entre as saídas da amostra \mathbf{x}_i e dos pontos de referência são estimadas por

$$\hat{\delta}(\mathbf{y}, \mathbf{t}_k) = d \hat{B}, \quad (3.67)$$

onde cada elemento de $\hat{\delta}(\mathbf{y}, \mathbf{t}_k) \in \mathbb{R}^{1 \times R}$ representa a distância entre a saída dos pontos de referência e a da amostra \mathbf{x}_i .

Neste ponto, chega-se a um problema de multilateração, que consiste em estimar a saída \mathbf{y} a partir de pontos fixos e das distâncias entre estes pontos e a saída. A figura a seguir ilustra esta situação para uma saída de dimensão 2.

Assim, a saída \mathbf{y} é estimada pela solução de um problema de otimização descrito na equação 3.68.

$$\hat{y} = \underset{y}{\operatorname{argmin}} \left(\sum_{k=1}^K \left((\mathbf{y} - \mathbf{t}_k)' (\mathbf{y} - \mathbf{t}_k) - \hat{\delta}^2(\mathbf{y}, \mathbf{t}_k) \right) \right) \quad (3.68)$$

Neste trabalho, a função `fsolve()` do Matlab, juntamente com o algoritmo Levenberg-Marquardt, é utilizada para solucionar o problema supracitado. Esta função é capaz de solucionar um sistema de equações não lineares, o que a tornou uma boa candidata para ser utilizada.

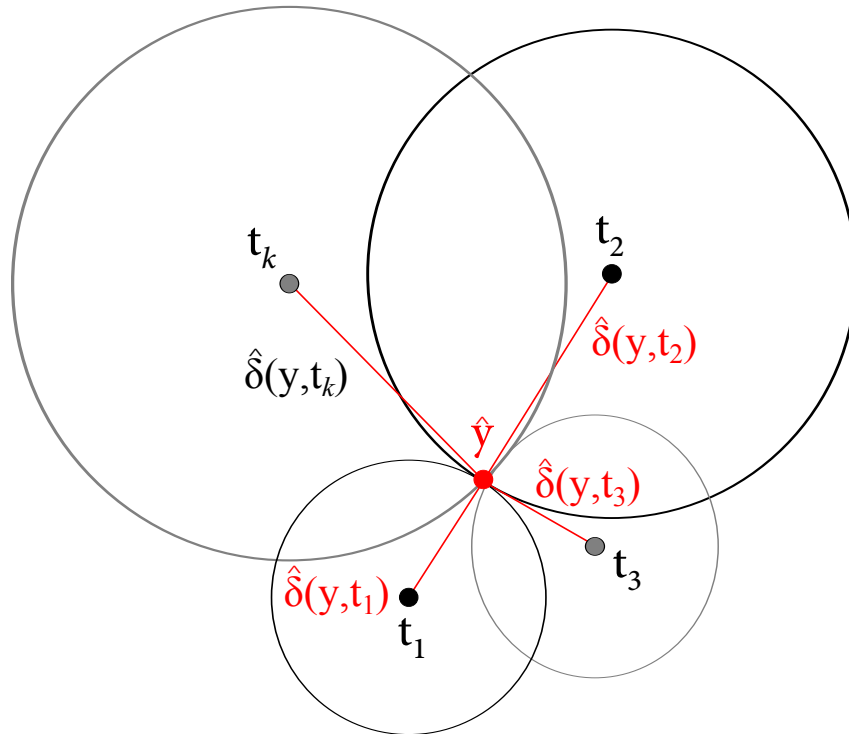


Figura 7 – Problema da Multilateração. (Souza Jr. et al., 2015).

3.7 Conclusão

Neste capítulo, as definições e a formalização do problema de classificação foram apresentadas. Além disso, abordou-se o algoritmo de cada classificador utilizado para solucionar o problema de detecção de falhas: OLS, Gaussianos, MLP, ELM, SVM, LSSVM e MLM.

Assim, após a descrição de cada algoritmo de aprendizado de máquina, as técnicas utilizadas, juntamente com estes, para a detecção de falhas em MIT, serão detalhadas no capítulo a seguir.

Metodologia dos Experimentos

Neste capítulo, são apresentadas as técnicas utilizadas para analisar os dados, comparar os classificadores e detectar as falhas nos motores de indução trifásicos. Além disso, alguns testes realizados com base nas características específicas de cada classificador são evidenciados.

4.1 Análise dos Dados

Como mostrado no Capítulo 2, já haviam sido realizados alguns estudos teóricos sobre as harmônicas que indicam falhas por curto circuito, e alguns estudos empíricos sobre as estatísticas destes atributos. Porém, a única métrica estatística utilizada para analisar os dados, foi a variância destes. Assim, nesta dissertação, utilizaram-se mais duas ferramentas para analisar os dados.

Primeiramente, diagramas de caixa foram utilizados para verificar a faixa de valores nas quais os dados de cada classe variam. Isto foi feito para verificar se havia um limiar de separação, em uma dimensão, entre os dados de motores em funcionamento normal ou com falhas. Além dos diagramas de caixas, foram feitos gráficos de dispersão com as harmônicas que seriam utilizadas para a avaliação dos classificadores. A partir destes gráficos, buscava-se um limiar de separação, em duas dimensões, entre as classes de motores em funcionamento normal ou com falhas. Se comprovada a existência de um destes limiares, não haveria a necessidade de aplicação de algoritmos complexos para solução do problema.

4.2 Banco de Dados

Como explicado no Capítulo 2, o banco de dados reais utilizado nesta dissertação possui 294 amostras representantes de motores. Se o problema for tratado como binário, 42 destas amostras representam a classe de motores em funcionamento normal. O restante (252) são representantes de motores com falhas por curto circuito entre espiras. Este banco de dados é definido como Banco de dados 1. Pode-se perceber que o banco de dados 1 é bem desequilibrado, visto que a quantidade de amostras da classe de falhas (86% do total) equivale a seis vezes a quantidade de amostras da classe de motores em funcionamento normal.

Além disso, este conjunto não representa uma situação real na indústria, visto que, nesta, a quantidade de amostras de motores em funcionamento normal é bem maior (apenas uma pequena parcela de motores falha).

Para tentar sanar este problema, de modo a equilibrar o número de dados nas duas classes, algumas amostras de motores em funcionamento normal foram geradas, artificialmente, de acordo com o método a seguir (*noise injection*) (MATSUOKA, 1992):

- Passo 1: Seleciona-se, aleatoriamente, uma amostra da classe de funcionamento normal.
- Passo 2: Adiciona-se, a cada atributo desta amostra, um valor aleatório, de distribuição uniforme, com valores entre $-0,01 * x_{nj}$ e $+0,01 * x_{nj}$, onde x_{nj} corresponde ao valor de determinado atributo.

Estes passos foram feitos até que fossem gerados 210 amostras de motores em funcionamento normal. Assim, o novo banco de dados é composto de 504 dados, contendo 252 dados de cada classe. Este banco de dados é definido como o Banco de Dados 2.

4.3 Metodologia de Comparação entre Classificadores

Os experimentos utilizados nesta dissertação para comparar o desempenho dos classificadores quanto a sua capacidade de detectar falhas seguem a metodologia ilustrada na figura 8. Ao finalizar todos os passos desta metodologia, ocorre a realização de determinado algoritmo. Para cada conjunto de passos determinados, foram feitas 50 realizações para cada classificador.

Deste modo, as estatísticas (máximo, mínimo, média, desvio padrão, mediana) referentes à capacidade de generalização¹ (taxa de acerto na detecção) de cada ferramenta pode ser observada. Essas estatísticas serão, no próximo capítulo, evidenciadas através de matrizes de confusão, tabelas e gráficos de caixa.

Inicialmente, define-se um banco de dados a ser tratado pelos classificadores. Nesta dissertação, pode-se optar pelo banco de dados 1 (com a quantidade desequilibrada de amostras de motores com ou sem falhas) ou pelo banco de dados 2 (com equilíbrio entre a quantidade de amostra com ou sem falhas). É importante salientar que, em ambos, cada amostra possui 6 atributos (0,5fs 1,5fs 2,5fs 3fs 5fs 7fs). Após esta definição, cada amostra de entrada é normalizada. Isto é feito para que nenhum componente do vetor de atributos se sobreponha aos outros durante o processo de classificação.

¹ Entende-se por generalização como o poder de um classificador, após o seu treinamento, rotular corretamente novas amostras.

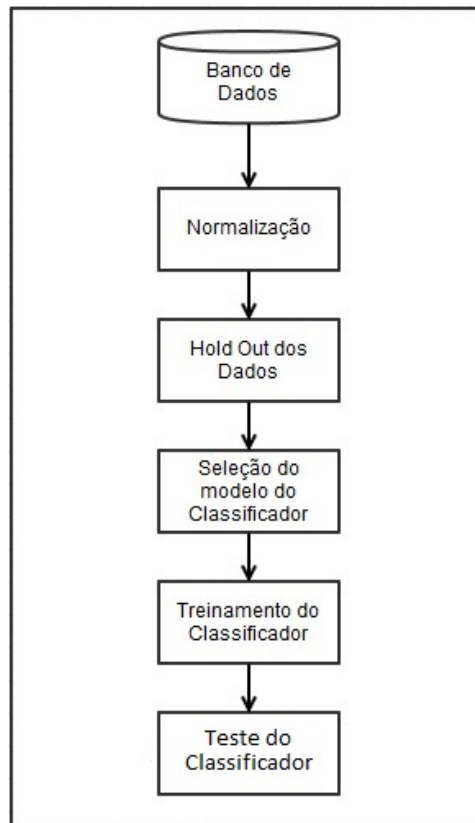


Figura 8 – Metodologia Dos Experimentos

Existem diversas formas de normalizar os dados, dentre elas, podem ser citadas a normalização no intervalo entre $[0 e 1]$, a normalização no intervalo entre $[-1 e +1]$ e a normalização pela média e pelo desvio padrão de cada atributo. As duas últimas citadas foram utilizadas (uma por vez) juntamente com os outros passos da metodologia. Estas normalizações podem ser resumidas, respectivamente, na equação

$$x_{nj}^* = 2 \left(\frac{x_{nj} - x_{nj}^{max}}{x_{nj}^{max} - x_{nj}^{min}} \right) - 1, \quad (4.1)$$

onde x_{nj}^{max} e x_{nj}^{min} são, respectivamente, o valor máximo e mínimo de um determinado atributo, dentre todas as amostras disponíveis; e na equação

$$x_{nj}^* = \frac{x_{nj} - \mu_j}{\sigma_j}, \quad (4.2)$$

em que μ_j é a média do atributo j e σ_j é o desvio padrão deste mesmo atributo. Vale salientar que esta normalização faz com que cada atributo do problema tenha média zero e variância unitária. Os demais passos da metodologia aplicada são descritos a seguir.

4.3.1 Holdout

Esta etapa consiste em, aleatoriamente, separar algumas amostras do banco de dados, para que sejam utilizadas apenas para testar os classificadores.

Assim, o conjunto de dados com N amostras é dividido entre $N1$ amostras de treinamento e $N2$ amostras de testes, tal que $N = N1 + N2$.

Nesta dissertação, foram utilizados dois métodos de *Hold Out*:

- **Holdout 01**: de todo o banco de dados utilizado, são retirados, aleatoriamente, (utilizando a função *rand()* do Matlab) 20% das amostras. Estas amostras só são apresentadas ao classificador para gerar as taxas de acerto, após o modelo deste ter sido determinado (ou seja, na etapa de teste do classificador). Já as 80% das amostras restantes são utilizadas para selecionar o modelo do classificador e treiná-lo (ou seja, para a etapa de treinamento).
- **Holdout 02**: as amostras são separadas por classe e, da classe que possui a menor quantidade de amostras, são separadas, aleatoriamente, 80% destas para seleção do modelo e treinamento do classificador. Das demais classes, uma quantidade equivalente de amostras é separada, de modo que, durante o treinamento, o número de amostras por classe seja equilibrado. O restante das amostras é, então, utilizado para testar os classificadores.

Este último método foi aplicado ao Banco de Dados 1, visto que, neste caso, o desequilíbrio entra a quantidade de amostras de motores com ou sem falhas é muito grande. Após a separação de dados para treinamento e teste, é feita a seleção do modelo do classificador.

4.3.2 Seleção do modelo

Esta etapa só está presente se o algoritmo em questão possuir hiperparâmetros que devem se ajustados antes do treinamento. No caso da rede MLP, por exemplo, um dos hiperparâmetros a ser ajustado é o número de neurônios da camada oculta.

Esta seleção ocorre apenas com os dados separados para treinamento. Isto se deve ao fato de que o modelo do classificador também carrega informações dos dados, e para que se meça a capacidade de generalização deste classificador, não deve haver interferência dos dados de teste.

Com isso, esta etapa consiste dos seguintes passos:

- **Busca em grade** (*grid search*): Define-se uma faixa de valores para determinados parâmetros de um classificador. A partir disso, é fornecido, ao passo seguinte, todas as combinações possíveis de valores dos parâmetros.

- **Validação cruzada:**

Divide-se, aleatoriamente, as amostras de treinamento em r partes iguais (*r-fold cross-validation*).

A partir dessa divisão, são realizadas r rodadas de treinamento e teste, para cada conjunto de valores dos parâmetros disponibilizados pela busca em grade.

Com isso, é calculada a taxa de acerto médio, no teste, de cada conjunto de valores de parâmetros.

- **Escolha dos hiperparâmetros:**

O conjunto de valores de parâmetros que obtiverem a maior taxa de acerto médio é selecionado para a próxima etapa.

Primeiramente, levando em conta todos os classificadores, estes possuem os seguintes hiperparâmetros:

Tabela 3 – Hiperparâmetros dos classificadores avaliados nesta dissertação

Classificador	Hiperparâmetros	Definição
OLS	-	-
Gaussiano	-	-
MLP	q	Quantidade de Neurônios Ocultos
	λ	Fator de Momento
	η	Passo de Aprendizagem
ELM	q	Quantidade de Neurônios Ocultos
SVM	C	Termo de Regularização
	σ^2	hiperparâmetro do Kernel Gaussiano
LSSVM	γ	Termo de Regularização
	σ^2	hiperparâmetro do Kernel Gaussiano
MLM	k	Quantidade de pontos de Referência

Para a busca em grade, as seguintes faixas de valores foram definidas para os hiperparâmetros:

- OLS: Este classificador não possui hiperparâmetros, mas apenas parâmetros que são estimados diretamente dos dados.
- Gaussiano: Como o OLS, este não possui hiperparâmetros, mas apenas parâmetros que são estimados diretamente dos dados.

- MLP: os valores $\eta = 0,05$ e $\lambda = 0,75$ foram mantidos constantes, e variou-se o número de neurônios ocultos q no intervalo entre 2 e 20.
- ELM: o número de neurônios ocultos foi variado no intervalo entre 5 e 30.
- SVM: o parâmetro de regularização e o hiperparâmetro do kernel tiveram, para a busca em grade, respectivamente os seguintes valores: $C = [0,5 \ 5 \ 10 \ 15 \ 25 \ 50 \ 100 \ 250 \ 500 \ 1000]$ e $\sigma^2 = [0,01 \ 0,05 \ 0,1 \ 0,5 \ 1 \ 5 \ 10 \ 50 \ 100 \ 500]$.
- LSSVM: já para este classificador, o parâmetro de regularização e o hiperparâmetro do kernel tiveram para a busca em grade, respectivamente, os seguintes valores testados: $\gamma = 2^{-5}$ a 2^{20} $\sigma^2 = 2^{-10}$ a 2^{10} .
- MLM: por fim, o número de pontos de referência k foi variado no intervalo entre 2 e 20.

Em seguida, definiu-se uma divisão r de 5 partes no conjunto de treinamento, ou seja, foi feita uma validação cruzada de 5 partes (5-fold cross-validation).

Por fim, verificou-se, para as diversas realizações dos classificadores, quais valores de hiperparâmetros eram comumente selecionados como sendo ótimos para o conjunto de treinamento. Se estes valores permanecessem constantes ou variassem pouco, eles poderiam ser definidos como os hiperparâmetros ótimos e o algoritmo da validação cruzada juntamente com a busca em grade se tornariam desnecessários.

4.3.3 Treinamento dos Classificadores

Após a seleção dos modelos, cada algoritmo é treinado com todos os dados separados para treinamento, de modo que os parâmetros sejam atualizados. Nesta etapa, como todos os algoritmos utilizados nesta dissertação possuem o treinamento supervisionado, os rótulos das amostras são utilizados.

Levando em conta todos os classificadores, estes possuem os parâmetros listados na tabela 4. A partir desta tabela, pode-se notar que existem muitos parâmetros que são vetores ou matrizes. Para quantificar o total de parâmetros, deve-se saber a dimensão destes vetores e matrizes. Em resumo, a quantidade total de parâmetros de cada classificador é calculado a seguir.

- L_{OLS} : $K \times p + 1$ (referente a \mathbf{W})
- Gaussiano: Para todas as classes, existem $K \times (p + p \times p)$ parâmetros.
- MLP: dado um problema MLP(p,q,K), a quantidade de parâmetros se resume a $(p + 1) \times q + q \times K$

Tabela 4 – Parâmetros dos Classificadores

Classificador	Parâmetros	Definição
OLS	\mathbf{W}	Matriz de pesos
Gaussiano	$\boldsymbol{\mu}_i$	Vetor centróide da classe i
	$\boldsymbol{\Sigma}_i$	Matriz de covariância da classe i
MLP	\mathbf{W}	Matriz de pesos da camada oculta
	\mathbf{M}	Matriz de pesos da camada de saída
ELM	\mathbf{W}	Matriz de pesos da camada oculta
	\mathbf{M}	Quantidade de Neurônios Ocultos
SVM	n_{sv}	Número de vetores de suporte
	$\boldsymbol{\alpha}$	Multiplicadores de Lagrange
	\mathbf{X}	Matriz com vetores de suporte
LSSVM	n_{sv}	Número de vetores de suporte
	$\boldsymbol{\alpha}$	Multiplicadores de Lagrange
	\mathbf{X}	Matriz com vetores de suporte
MLM	\mathbf{B}	Matriz de transformação
	\mathbf{M}_k	Matriz de entrada dos pontos de referência

- ELM: dado um problema ELM(p,q,K), a quantidade de parâmetros se resume a $(p + 1) \times q + (q \times K)$
- SVM: A quantidade total de parâmetros deste classificador depende dos vetores de suporte selecionados durante o treinamento. Assim, esta quantidade é de: $n_{sv} \times (1+p)$, onde o n_{sv} corresponde à quantidade de multiplicadores de Lagrange se refere aos multiplicadores de lagrange e o p se refere às amostras de treinamento que devem ser armazenadas.
- LSSVM: Assim como no SVM, a quantidade total de parâmetros deste classificador depende dos vetores de suporte selecionados durante o treinamento. Assim, esta quantidade é de: $n_{sv} \times (1+p)$, onde o n_{sv} corresponde à quantidade de multiplicadores de Lagrange e o p se refere às amostras de treinamento que devem ser armazenadas.
- MLM: a quantidade de parâmetros deste classificador depende do hiperparâmetro K , definido este, esta quantidade se torna $(k \times k) + (k \times p)$

4.3.4 Teste do Classificador

Finalmente, após cada classificador ser devidamente treinado com as $N1$ amostras de treino, a cada realização deste, apresenta-se as $N2$ amostras restantes a este. Para calcular a taxa de erro/acerto de cada classificador, compara-se a saída destes com o rótulo original das amostras. Se a saída do classificador for igual ao rótulo da amostra, uma classificação correta é computada, caso contrário, um erro é computado a este.

Assim, a taxa de acerto de cada classificador é a razão entre quantidade de amostras rotuladas corretamente pelo número total de amostras. Para comparar a acurácia dos classificadores, utilizaram-se, inicialmente, diagramas de caixa (*boxplot*) e matrizes de confusão.

A utilização de diagramas de caixas é um meio muito útil para visualizar e comparar desempenho entre classificadores, visto que, através destes, a distribuição da taxa de acerto para um conjunto de realizações de cada algoritmo pode ser observada, evidenciando informações úteis, tais como outliers (desempenhos atípicos). Além disso, cada quartil² desta distribuição pode ser observado.

Já através das matrizes de confusão F , pode-se verificar, em uma realização do algoritmo, entre quais classes estão ocorrendo os maiores erros do classificador.

Estas matrizes tem dimensão $c \times c$, onde as linhas representam a classe C_i a qual a amostra pertence, e as colunas representam as saídas \hat{y} estimadas pelo classificador, ou seja, a qual classe o classificador atribuiu as amostras.

Deste modo, o elemento f_{12} , por exemplo, indica a quantidade de amostras pertencentes à classe 1, que foram atribuídas à classe 2. Assim, a soma dos elementos da diagonal da matriz de confusão indica a quantidade de dados classificados corretamente.

4.4 Problema Multiclasses

Além de tratar o problema da detecção de falhas como binário, algumas metodologias são aplicadas aos classificadores, de modo a investigar o desempenho destes na detecção de falhas multiclasses. Assim, a maior modificação destas metodologias em relação à principal, está na rotulação geral do banco de dados. Nestas, começa-se a tratar o problema de detecção de falhas, durante o treinamento, como multiclasse, e considera-se este problema como binário durante o teste.

Assim, foram feitas três metodologias adicionais:

- Multiclasses 7: problema composto de 7 classes, onde cada combinação de tipo e nível de falha é considerada uma classe (AI1, AI2, AI3, BI1, BI2, BI3). Neste problema, cada classe de falha possui 42 amostras, e a classe de funcionamento normal pode possuir 42 amostras (caso fosse utilizado o banco de dados 1) ou 252 amostras (caso o banco de dados utilizado fosse o 2).
- Multiclasses 3: problema composto por 3 classes, onde cada tipo de falha é considerado uma classe (AI e BI). Neste problema, cada classe de falha possui 126 amostras, e

² Quartis são medidas de posição para uma dada distribuição. Estes dividem os resultados em 4 partes iguais. Ou seja, o primeiro quartil indica que 25% dos dados estão abaixo deste. Já o segundo, indica que 50% dos dados estão abaixo deste (mediana), e assim sucessivamente.

a classe de funcionamento normal pode possuir 42 amostras (caso fosse utilizado o banco de dados 1) ou 252 amostras (caso o banco de dados utilizado fosse o 2).

- Multiclasses 4: problema composto de 4 classes, onde cada nível de falha é considerado uma classe (AI1 BI1 X AI2 BI2 X AI3 BI3). Neste caso, cada classe de falha possui 84 amostras, e a classe de funcionamento normal pode possuir 42 amostras (caso fosse utilizado o banco de dados 1) ou 252 amostras (caso o banco de dados utilizado fosse o 2).

Vale salientar que estas duas últimas metodologias foram aplicadas para verificar se os classificadores conseguiam diferenciar melhor "níveis de falhas" ou "tipos de falhas".

Em todas estas metodologias, ajusta-se os parâmetros, durante o treinamento, a partir de um problema multiclasses. Por outro lado, durante o teste, a saída de cada classificador é comparada ao rótulo de cada amostra. Se, para determinada amostra, o classificador fizer confusão entre níveis ou tipo de falhas, isto não é considerado um erro. Assim, apenas quando o classificador rotula uma amostra normal como falha (ou vice-versa) um erro é computado.

4.5 Opção de Rejeição

Em problemas de classificação, normalmente, atribui-se à uma nova amostra o rótulo da classe para qual a sua função discriminante obteve o maior valor, mesmo quando a probabilidade a posteriori de duas classes é muito próxima. Esta característica pode levar ao aumento na taxa de erros de classificadores, visto que existem muitos casos onde há padrões difíceis de serem classificados.

Este é um caso recorrente na detecção de falhas incipientes por curto-circuito entre espiras, visto que as amostras de motores em funcionamento normal são muito próximas das amostras em início de falha. Assim, a aplicação da técnica da opção de rejeição, pode fazer com que os classificadores rejeitem algumas amostras difíceis de serem classificadas, de modo a reduzir a quantidade de falsos positivos e falsos negativos.

Por definição, opção de rejeição é qualquer estratégia que permita ao classificador não classificar determinada amostra de entrada caso a avaliação das saídas seja ambígua. Assim, a classificação não se resume apenas a separar as amostras entre classes, mas, também, converter potenciais erros em rejeição (CHOW, 1970).

Sem a utilização desta estratégia, ao se avaliar classificadores, apenas leva-se em conta a taxa de erros destes quanto a solução de determinado problema de classificação. Ao incluí-la, também deve-se medir a taxa de rejeição dos dados. Assim, o melhor classificador não será simplesmente o que atingiu a menor taxa de erros, mas, sim, o que obteve o melhor balanceamento entre esta métrica e a quantidade de dados rejeitados.

Existem várias formas de adicionar a opção de rejeição aos classificadores, podendo esta ser incorporada diretamente ao treinamento (NETO et al., 2011) ou pode-se medir apenas as saídas do classificador e aplicar alguma heurística (FILHO; SANTOS; BARRETO, 2013). Como exemplo desta última forma, pode-se definir um limiar para aceitar as saídas de dado classificador, ou seja, não basta apenas selecionar a classe de acordo com o índice da saída com maior valor, mas a maior saída também deve ser maior que este limiar.

Nesta dissertação, utilizou-se o método da definição de um limiar ótimo para aceitar a saída de um classificador. Esta abordagem é, a seguir, descrita em maior detalhe.

Assumindo que cada dado de entrada é representado por um vetor $\mathbf{x}_n \in \mathbb{R}^p$, e que o classificador não fornece saídas probabilísticas, então deve ser utilizado um limiar de rejeição β para a saída do classificador. Assim, rejeita-se a classificação do vetor \mathbf{x}_n se

$$\max_i \{y_i\} < \beta, \quad (4.3)$$

em que y_i é a i -ésima saída do classificador, $i = 1, 2, \dots, K$. Neste método, a região de rejeição é determinada depois da fase de treinamento, de uma forma heurística ou baseada na otimização de algum critério que pondera os custos de uma classificação errônea e da rejeição realizada.

Segundo Neto et al. (2011), o projeto de um classificador com opção de rejeição requer os seguintes passos:

- Passo 1: Treinar o classificador, para ajustar seus parâmetros e modelo.
- Passo 2: Apresentar as amostras de teste ao classificador ajustado no passo anterior.
- Passo 3: Encontrar um valor ótimo para o limiar de rejeição β , através da minimização do risco empírico (CHOW, 1970)

$$\widehat{R} = c_r R + E \quad (4.4)$$

em que R e E são, respectivamente as taxas de padrões rejeitados e de erros de classificação, enquanto c_r é o custo de rejeição, cujo valor deve ser especificado pelo usuário.

- Passo 3.1: Para um dado custo de rejeição c_r varia-se β de um valor inicial β_i até um valor final β_f em incrementos fixos $\Delta\beta$. Nesta dissertação, utilizaram-se os valores $\beta_i = 0.25$, $\beta_f = 1.00$ e $\Delta\beta = 0,05$.
- Passo 3.2: Para cada valor de β , calcular \widehat{R} usando a equação 4.4, de modo que:

$$R(\beta) = \frac{n^\circ \text{ de amostras rejeitadas}}{n^\circ \text{ total de amostras}}. \quad (4.5)$$

$$E(\beta) = \frac{n^\circ \text{ de amostras classificadas erroneamente}}{n^\circ \text{ total de amostras} - n^\circ \text{ de amostras rejeitadas}}. \quad (4.6)$$

- Passo 3.3: Selecionar o limiar de rejeição ótimo β_o de acordo com a regra:

$$\beta_o = \operatorname{argmin}_{\beta} \left\{ \widehat{R}(\beta) \right\}. \quad (4.7)$$

- Passo 4: Estabelecer novos rótulos para os dados de teste, de acordo com a regra:
SE $\max_i \{y_i\} < \beta$
ENTÃO colocar as amostras em uma nova classe (classe de rejeição)
SENÃO manter os rótulos originais das amostras.
- Passo 5: Apresentar os dados de teste mais uma vez ao classificador, usando o valor de β_o encontrador. Calcular as novas taxas de rejeição e de classificação, além dos erros cometidos.

O procedimento anterior é genérico e pode ser utilizado por qualquer classificador de padrões que forneça uma saída real (contínua).

4.6 Comparação entre Classificadores

Além das ferramentas previamente citadas (tais como *Boxplot* e taxa de acerto) para comparar classificadores, foram utilizadas mais duas ferramentas estatísticas: o Coeficiente de Variação e o Coeficiente de Correlação de Matthews.

O coeficiente de variação é uma medida de variabilidade relativa da taxa de acerto do classificador (BOSLAUGH; WATTERS, 2008). Este é calculado dividindo-se o desvio padrão (sd) da taxa de acerto pelo seu valor médio (md), ou seja:

$$cv = \left(100 \times \frac{sd}{md}\right)\% \quad (4.8)$$

Numa comparação entre classificadores que apresentam desempenhos médios, equivalentes, quanto menor o valor de cv , melhor é o classificador.

Além disso, o coeficiente de correlação de Matthews (*Matthews Correlation Coefficient* - MCC) (MATTHEWS, 1975) é utilizado em aprendizado de máquina para medir a qualidade de classificações binárias. Esta medida é geralmente considerada como equilibrada e pode ser utilizada mesmo se as classes são de tamanhos muito diferentes. Por isso, visto que o banco de dados utilizado nesta dissertação é desbalanceado (para cada uma amostra de motores em condições normais há seis amostras de motores com falhas), este foi utilizado.

O MCC é um coeficiente de correlação entre as saídas observadas e previstas. Este retorna um valor entre -1 e +1, onde +1 representa uma previsão perfeita, 0 indica

previsão aleatória e -1 indica total desacordo entre previsão e observação. Pode-se calculá-lo diretamente da matriz de confusão, utilizando a equação

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (4.9)$$

onde TP (*True Positives*) é o número de verdadeiros positivos, TN (*True Negatives*) o número de verdadeiros negativos, FP (*False Positives*) o número de falsos positivos e FN (*False Negatives*) o número de falsos negativos.

Também, nesta equação, se qualquer uma das quatro somas do denominador for igual a 0, o denominador pode ser arbitrariamente fixado em 1.

4.7 Conclusão

Neste capítulo, foram explicados a análise e a formação do banco de dados utilizado, a metodologia de comparação entre os classificadores e o tratamento do problema como multiclases. Além disso, a estratégia da opção de rejeição e as ferramentas estatísticas Coeficiente de Variação e Coeficiente de Correlação de Matthews foram apresentadas.

Assim, após explicados os passos e ferramentas utilizadas juntamente com os classificadores, no capítulo a seguir, serão discutidos os resultados gerais deste trabalho.

Análise dos Resultados

A partir da metodologia apresentada no capítulo anterior, aqui são indicados os resultados dos experimentos realizados com as técnicas e classificadores citados.

5.1 Análise dos Dados

Primeiramente, buscou-se um limiar de separação entre as classes. Este resultado poderia ser um ponto de partida para um sistema de detecção de falhas, visto que, se a amostra atingisse um valor definido em determinado atributo, se poderia diretamente considerar este dado como falha, sem a necessidade de aplicação dos classificadores.

Para fazer esta análise, foram utilizados diagramas de caixas. A partir das figuras 9 e 10, foi verificado que, para os dados não normalizados, existem faixas de valores onde apenas os motores com falhas apresentam amostras.

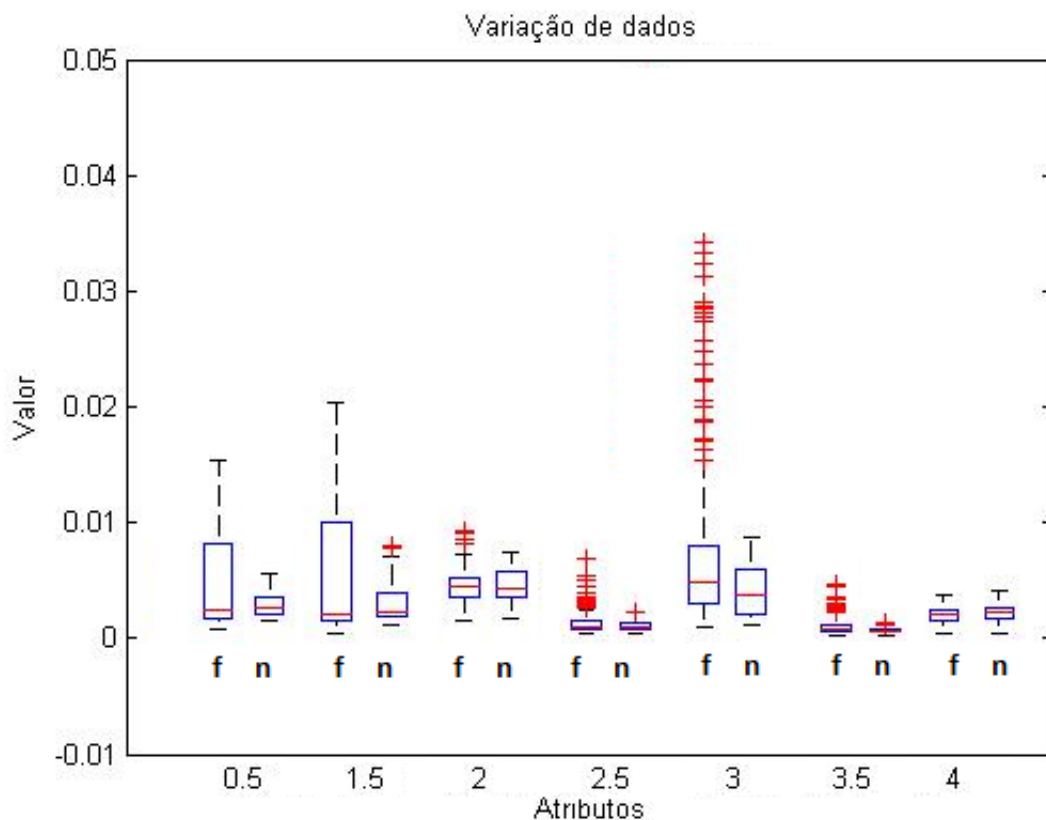


Figura 9 – Distribuição dos dados não-normalizados.

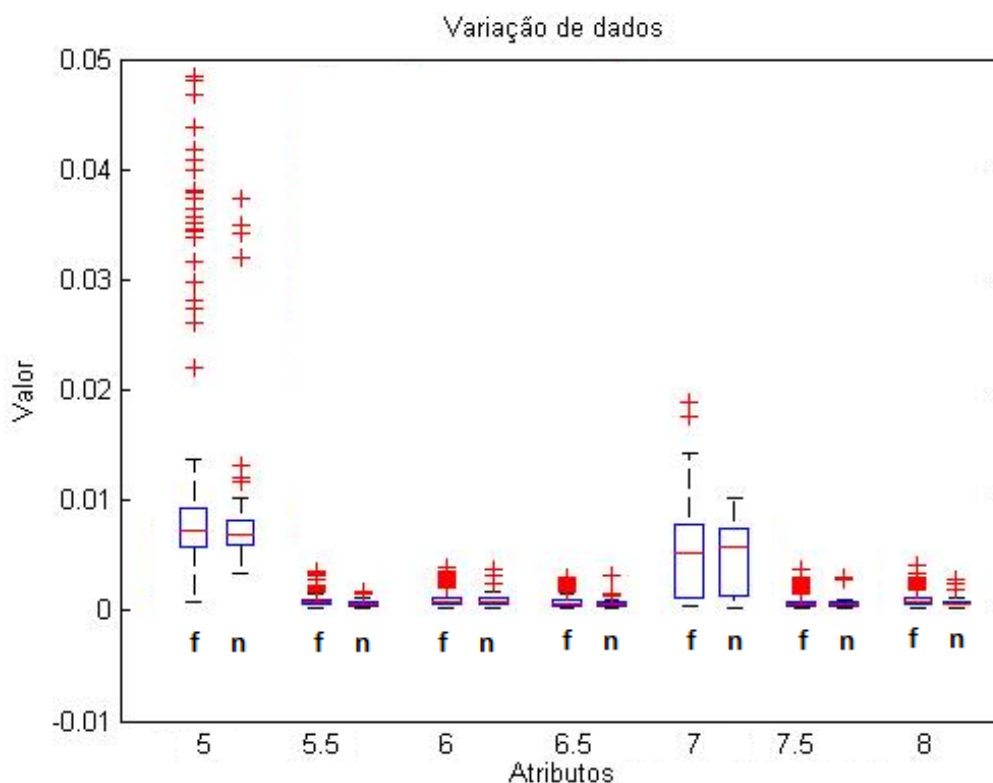


Figura 10 – Distribuição dos dados não-normalizados.

Porém, na faixa de valores de atributos onde se encontram as amostras do motor em funcionamento normal, também há amostras deste com falhas. Ou seja, não foi identificado um limiar que separasse as duas classes, mas sim um que demonstra a ocorrência de falhas.

Isto, como será mostrado a seguir, é um dos fatores que contribuíram na classificação errônea de vários motores em funcionamento normal.

Em seguida, as amostras foram normalizadas pela remoção da média e pela divisão pelo desvio padrão de cada atributo. Neste caso, como pode ser observado nas figuras 11 e 12, o mesmo padrão se repetiu: não há regiões bem distintas de dados do motor com ou sem falhas, apenas regiões onde só existem representantes de motores com falhas.

Da mesma forma, através de diversas combinações entre dois atributos do problema, não foi percebido nenhum padrão de separabilidade das classes, e nem uma distribuição de probabilidade intraclasses, fazendo com que fosse necessário a utilização de classificadores.

Nas figuras 13 e 14, são mostradas, respectivamente, combinações entre os atributos $0.5f_s$ e $1.5f_s$, e entre os atributos $3f_s$ e $5f_s$. Novamente pode-se observar o padrão anterior.

A partir deste resultado, ao desenvolver um sistema para detecção de falhas, se poderia mesclar uma árvore de decisão a um classificador de padrões. Se uma amostra tiver atributos que ultrapassem determinado limiar, esta já pode ser considerada uma amostra de falha no motor. Caso contrário, um classificador pode ser utilizado para determinar a

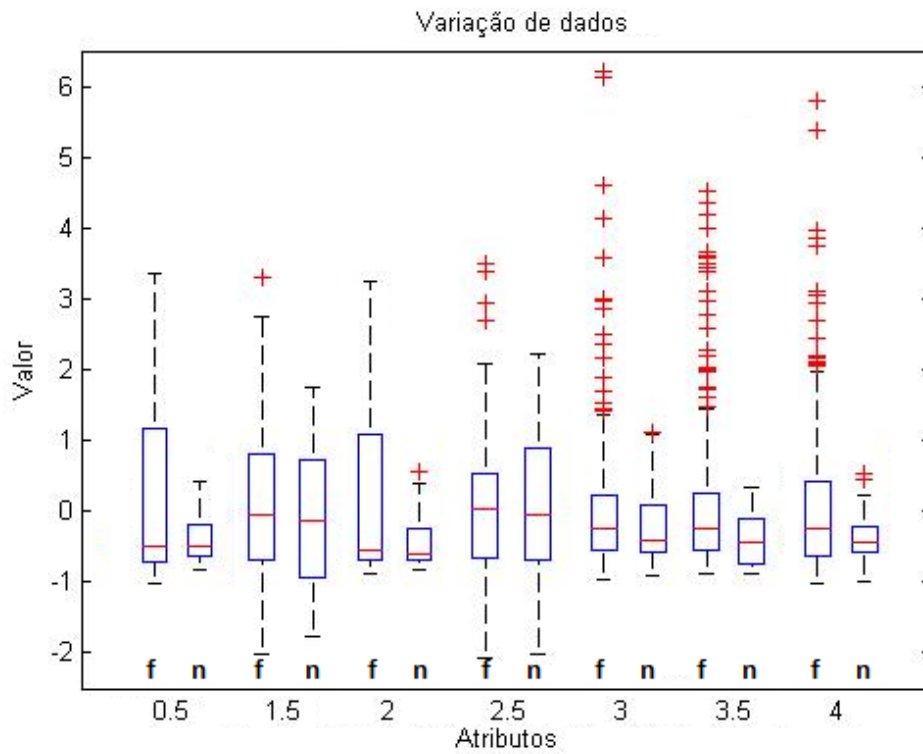


Figura 11 – Distribuição dos dados normais, normalizados.

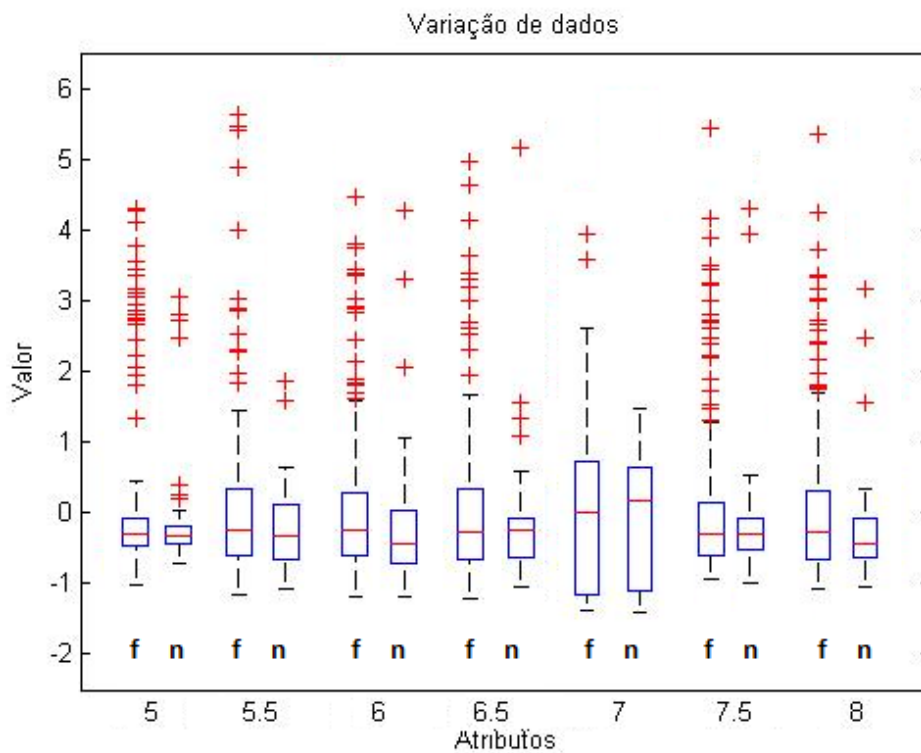


Figura 12 – Distribuição dos dados de falhas, normalizados.

condição deste motor.

Após a análise dos dados, os classificadores foram comparados utilizando a metodo-

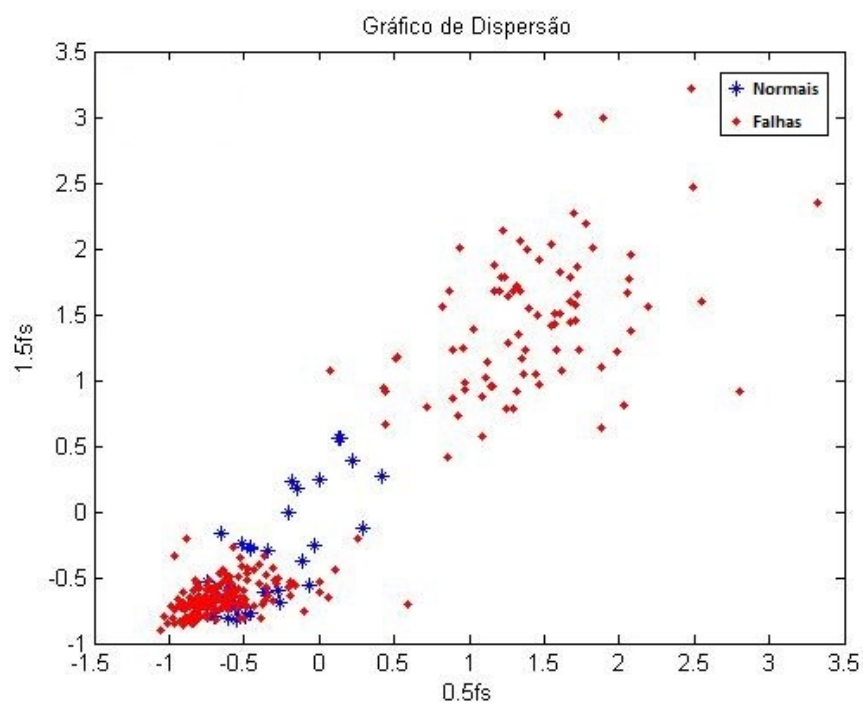


Figura 13 – Gráfico de Dispersão 1.

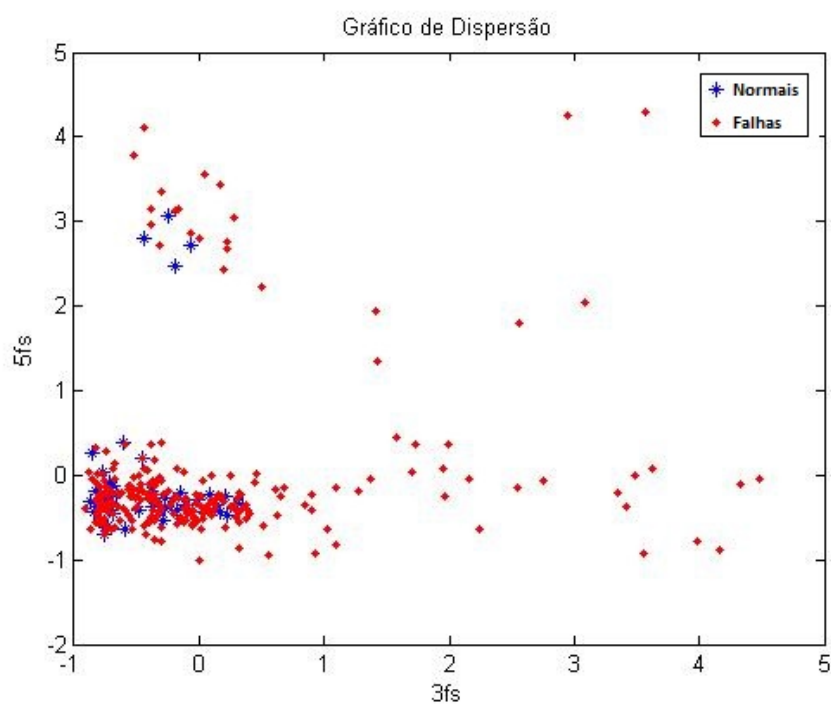


Figura 14 – Gráfico de Dispersão 3.

logia descrita na seção 4.3. Os resultados desta comparação são discutidos a seguir.

5.2 Comparação entre Classificadores

Inicialmente, tentou-se utilizar apenas os classificadores lineares (classificador OLS e Gaussiano linear - tipo 4), pois se fosse possível solucionar o problema com este tipo de classificador, não seria necessária a aplicação de classificadores não-lineares (algoritmos mais complexos).

Porém, mesmo abordando o problema de diversas formas, os resultados não se mostraram satisfatórios, como mostrado na Figura 15. Nesta figura, também pode-se observar um resultado ligeiramente melhor na taxa de acerto máxima, média e no desvio padrão do classificador OLS.

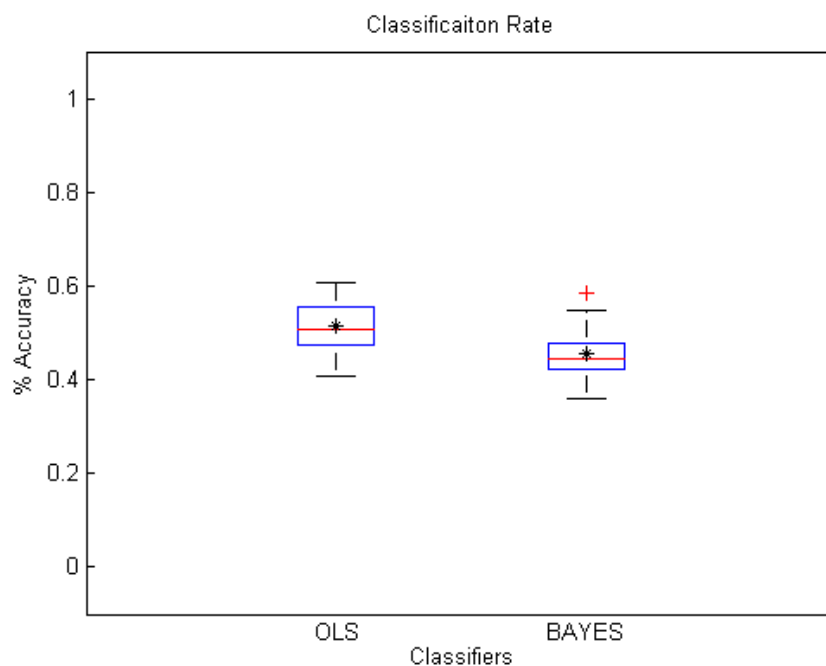


Figura 15 – Metodologia 01.

Devido a este resultado, tentou-se utilizar classificadores não lineares, iniciando pelos neurais (MLP e ELM), por serem métodos clássicos e serem muito utilizados em problemas de reconhecimento de padrões, e pelos gaussianos quadráticos (BAYES), por não haver um conhecimento prévio da distribuição intraclases do banco de dados utilizado. Também, como forma de comparação, manteve-se a utilização do classificador linear OLS, por este ter obtido os melhores resultados dentre os classificadores lineares.

Durante a comparação geral entre os classificadores, para avaliar o desempenho destes quanto a detecção de falhas, foram testadas diferentes formas de tratar o banco de dados. Cada forma será considerada uma metodologia distinta, e será numerada sequencialmente.

Assim, na metodologia 01 tratou-se o banco de dados da seguinte maneira para treinar e validar os classificadores:

- Metodologia 01:
 - Banco de dados utilizado: 01 (42 dados sem falha e 252 dados com falha).
 - Normalização: entre -1 e +1
 - Hold out: 80% dos dados para treinamento e 20% para validação
 - Seleção dos Hiperparâmetros: Validação Cruzada de 5 partes

onde os resultados desta primeira comparação são ilustrados no diagrama de caixas da figura 16.

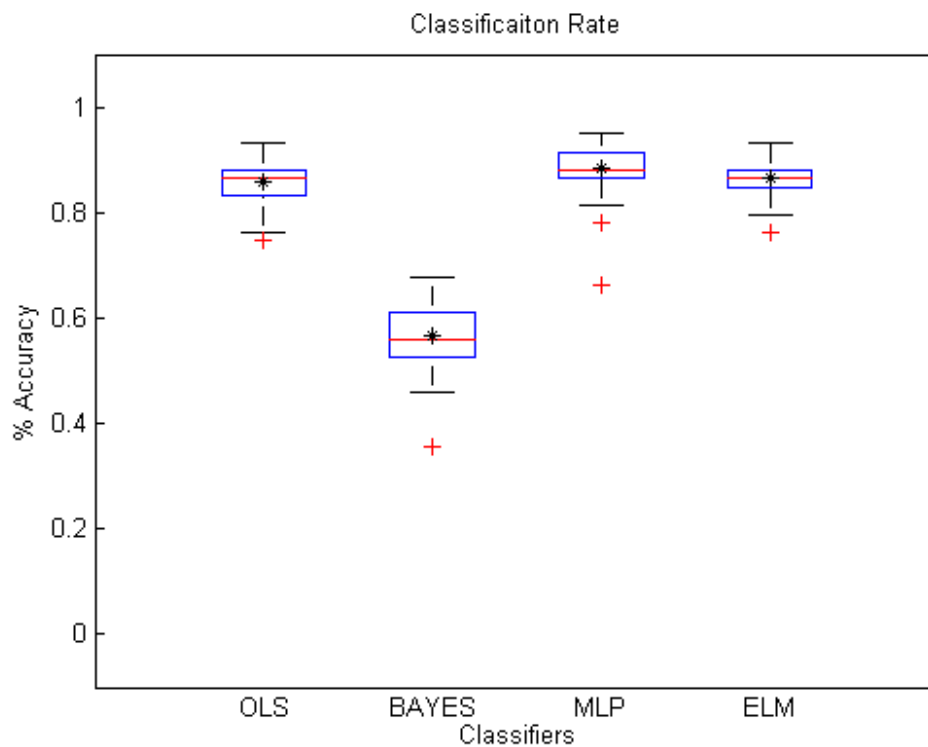


Figura 16 – Metodologia 01.

A partir desta, pode-se verificar que, excluindo o classificador gaussiano quadrático (que obteve uma taxa de acerto média de 55%), a taxa de acerto média de todos os classificadores se situou entre 85% e 90%, com uma pequena vantagem para o classificador MLP. Também pode-se notar que o classificador OLS, que é um classificador linear e bem menos complexo que os demais, obteve um resultado similar aos outros algoritmos.

Este resultado poderia levar à conclusão de que, através da aplicação de um classificador linear, seria possível verificar a condição do motor com até mais de 90% de acerto.

Porém, ao analisar as matrizes de confusão de cada classificador, pode-se perceber que os dados de teste estavam sendo, quase em todas as situações, rotulados como falhas. Isto se deve ao grande desbalanceamento das classes, pois 86% do banco de dados é composto por amostra de falhas.

Para exemplificar este resultado, na tabela 5 está representada a matriz de confusão de uma realização do classificador MLP. Já na tabela 6, está representada a matriz de confusão de uma realização do classificador OLS.

Tabela 5 – Matriz de Confusão MLP - Método 1.

MLP		Saída do Classificador	
		Normal	Falha
Rótulo	Normal	2	5
	Falha	0	52

Tabela 6 – Matriz de Confusão OLS - Método 1.

OLS		Saída do Classificador	
		Normal	Falha
Rótulo	Normal	0	10
	Falha	1	48

Se estes classificadores fossem aplicados em um ambiente real, ocorreriam vários falsos positivos (condição na qual um motor não tem falhas, e o classificador indica uma falha). Esta condição é indesejável, visto que a indicação de uma falha se converte em uma parada de processo, para que a equipe de manutenção possa atuar. Isto se refletiria em perda de tempo e capital para determinada empresa.

Para estes dois classificadores, a taxa média de ocorrência de falsos positivos foi de 66,5% e 100% respectivamente. Ou seja, o classificador OLS não conseguiu classificar corretamente amostras de motores em funcionamento normal. Para exemplificar este fato, o MCC do MLP e do OLS foi respectivamente de 0.51 e -0.06, demonstrando que, mesmo com taxas de acerto similares, os resultados do classificador MLP foram bem superiores segundo esta métrica.

Após utilizar a metodologia 01, investigou-se, na metodologia 02 (cujas forma de tratar os dados é listada a seguir), a influência da normalização dos dados na detecção de falhas. Nesta, os dados foram normalizados pela média e pelo desvio padrão.

- Metodologia 02
 - Banco de dados utilizado: 01 (42 dados sem falha e 252 dados com falha).
 - Normalização: pela média e desvio padrão dos atributos
 - Hold out: 80% dos dados para treinamento e 20% para teste

- Seleção dos Hiperparâmetros: Validação Cruzada

Como pode ser observado na figura 17 e nas matrizes de confusão, os resultados da metodologia anterior pouco se alteraram. Assim, tentou-se uma nova abordagem do problema.

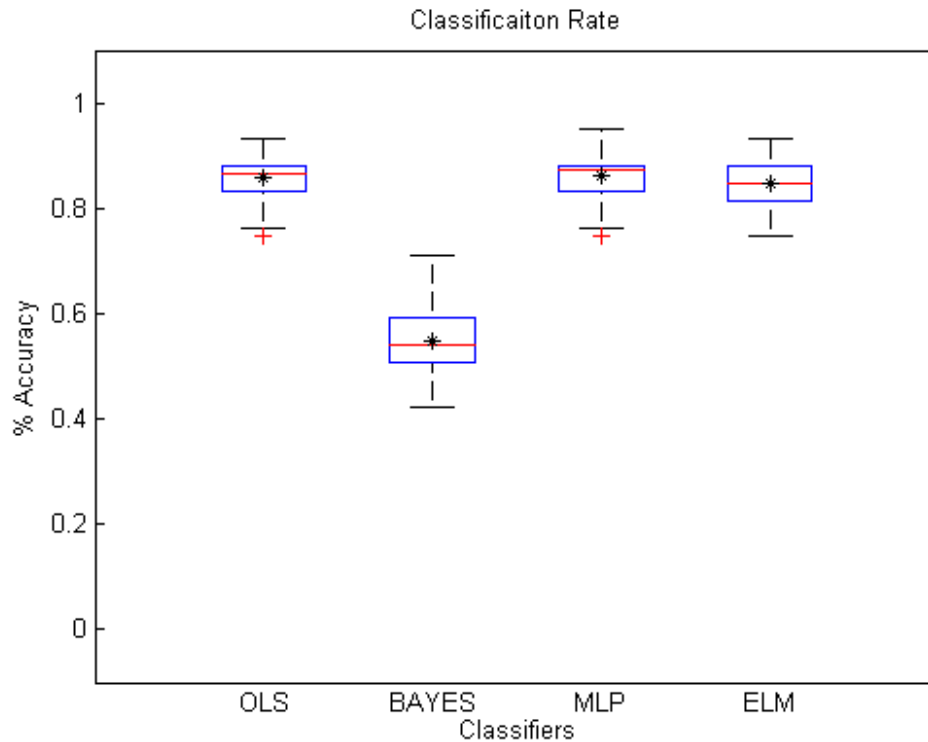


Figura 17 – Metodologia 02.

Tabela 7 – Matriz de Confusão MLP - Método 2.

MLP		Saída do Classificador	
		Normal	Falha
Rótulo	Normal	2	6
	Falha	1	50

Tabela 8 – Matriz de Confusão OLS - Método 2.

OLS		Saída do Classificador	
		Normal	Falha
Rótulo	Normal	0	8
	Falha	0	51

Para tentar mitigar a taxa de falsos positivos, modificou-se, na metodologia 03, o Holdout dos dados. Nesta nova etapa, os dados foram equilibrados durante o treinamento, de modo que os parâmetros dos classificadores fossem atualizados pela mesma quantidade

de dados de cada classe. As características desta metodologia estão listados a seguir, e os resultados são ilustradas na figura 18.

- Metodologia 03
 - Banco de dados utilizado: 01 (42 dados sem falha e 252 dados com falha).
 - Normalização: pela média e desvio padrão dos atributos
 - Hold out: 33 dados de cada classe para treinamento (33 dados corresponde a 80% da quantidade total de dados da classe normal) e o restante para validar.
 - Seleção dos Hiperparâmetros: Validação Cruzada

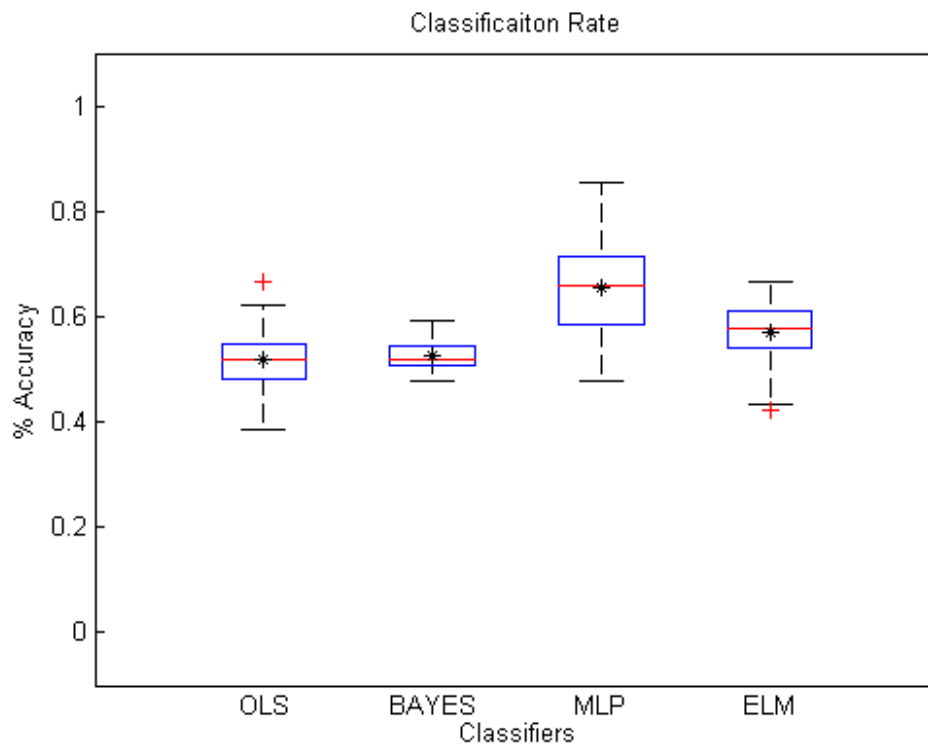


Figura 18 – Metodologia 03.

Como pode ser observado pelo Boxplot anterior, a taxa média de acerto de todos os classificadores reduziu, porém o classificador MLP atingiu uma taxa máxima de acerto de 84%.

Analisando as matrizes de confusão, observou-se uma redução geral na taxa de falsos positivos (de mais de 60% para menos de 30%). Já o MCC foi de 0.23 para o classificador MLP e 0.12 para o classificador OLS.

A taxa de falsos positivos reduziu bastante, porém um classificador com uma taxa de acerto média menor que 70% não se mostra satisfatório para a detecção de curto circuito

Tabela 9 – Matriz de Confusão MLP - Método 3.

MLP		Saída do Classificador	
		Normal	Falha
Rótulo	Normal	7	2
	Falha	55	164

Tabela 10 – Matriz de Confusão OLS - Método 3.

OLS		Saída do Classificador	
		Normal	Falha
Rótulo	Normal	6	3
	Falha	94	125

entre espiras. Esta redução na taxa de acerto também se deu pela baixa representatividade do conjunto de falha durante o treinamento, visto que, como haviam poucas amostras de motores sem falhas (42), apenas 33 amostras de cada classe foram utilizadas para treino, e o restante (maioria de amostras com falhas) foi utilizada para validação.

Este fato aumentou bastante a taxa de falsos negativos, que resultaria em um sistema de detecções de falhas com reduzida taxa de acerto para o acontecimento de falhas. Porém, isto pode não ser um grande problema, visto que, como a corrente de falhas do motor foi limitada durante a obtenção do banco de dados, todas as amostras de falha, na realidade, representam condições iniciais de curto circuito. Esta taxa de falsos negativos pode ser reduzida caso a falha comece a aumentar.

A partir deste momento, como os resultados ainda não se mostraram satisfatórios, os classificadores SVM, LSSVM e MLM (com outros paradigmas de aprendizado de máquina) foram utilizados para verificar se sua eficiência na detecção de falhas se mostrava superior aos demais utilizados nesta dissertação.

Os classificadores SVM e LSSVM estão sendo bastante utilizados em problemas de classificação e problemas de detecção de falhas em motores. Já o classificador MLM, por ser um algoritmo recente, ainda não foi aplicado a problemas de detecção de falhas.

Assim, utilizando a metodologia 01 (com 80% dos dados para treinamento), estes classificadores obtiveram resultados similares aos demais algoritmos (taxa de acerto média próxima a 85% e alta taxa de falsos positivos). Porém utilizando a metodologia 03 (33 dados de cada classe para treinamento e o restante para teste), o classificador SVM obteve taxa de acerto máxima superior a 90% (próximo ao classificador MLP), como mostrado na figura 19.

Porém, analisando a taxa média de acerto (72% para SVM e 57% para MLP) e o coeficiente de variação destes classificadores (variância pela taxa média de acerto), a capacidade de detecção de falhas não foi considerada satisfatória nas metodologias supracitadas. Assim, para solucionar o problema da pequena quantidade de amostras para

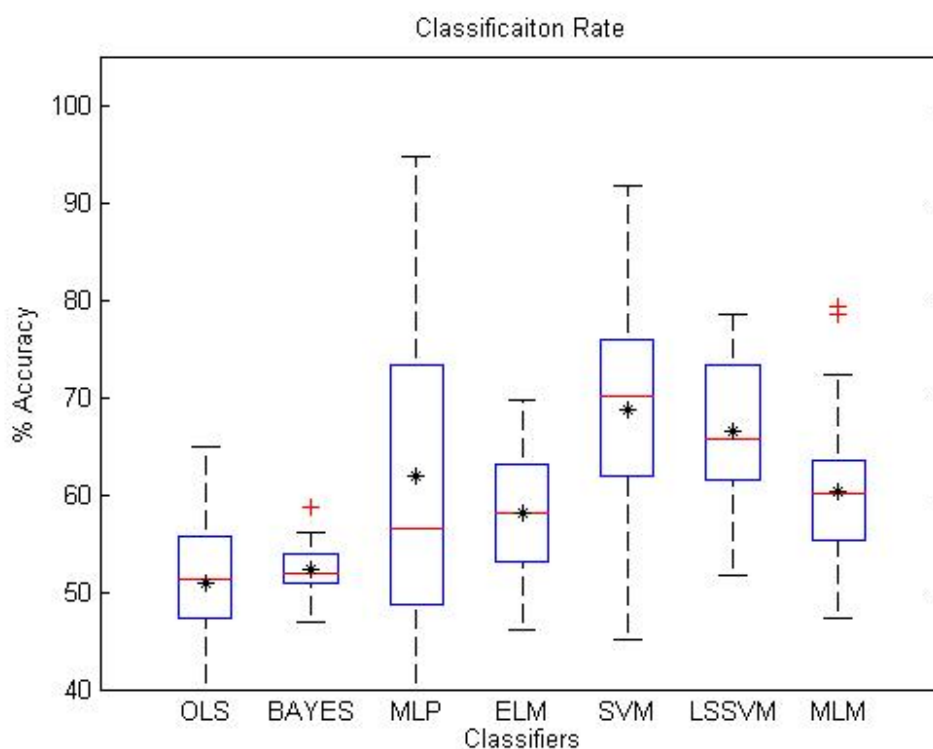


Figura 19 – Metodologia 03.

treinamento, e manter uma baixa taxa de falsos positivos, tentou-se uma nova abordagem. Na metodologia 04, cujas características estão descritas a seguir, utilizou-se o banco de dados 02. Este banco de dados possui várias amostras, geradas artificialmente, de motores sem falhas, fazendo com que este possua equilíbrio de amostras por classe e uma maior quantidade de dados.

- Metodologia 04
 - Banco de dados utilizado: 02 (252 dados sem falha e 252 dados com falha).
 - Normalização: pela média e desvio padrão dos atributos
 - Hold out: 80% dos dados para treinamento e 20% para validação
 - Seleção dos Hiperparâmetros: Validação Cruzada

Como pode ser visto na figura 20, os classificadores SVM e LSSVM conseguiram taxas médias de acerto próximas a 100% e o classificador baseado na rede MLP atingiu taxas médias de acerto de aproximadamente 90%. Juntamente a isso, houve uma redução na taxa de falsos positivos (atingiu menos de 20% em média) em relação à metodologia 01 e 02, que é uma situação desejada.

Uma justificativa para esta diferença de resultados de todos os algoritmos aos classificadores SVM e LSSVM se dá pela quantidade de informação (parâmetros) guardados

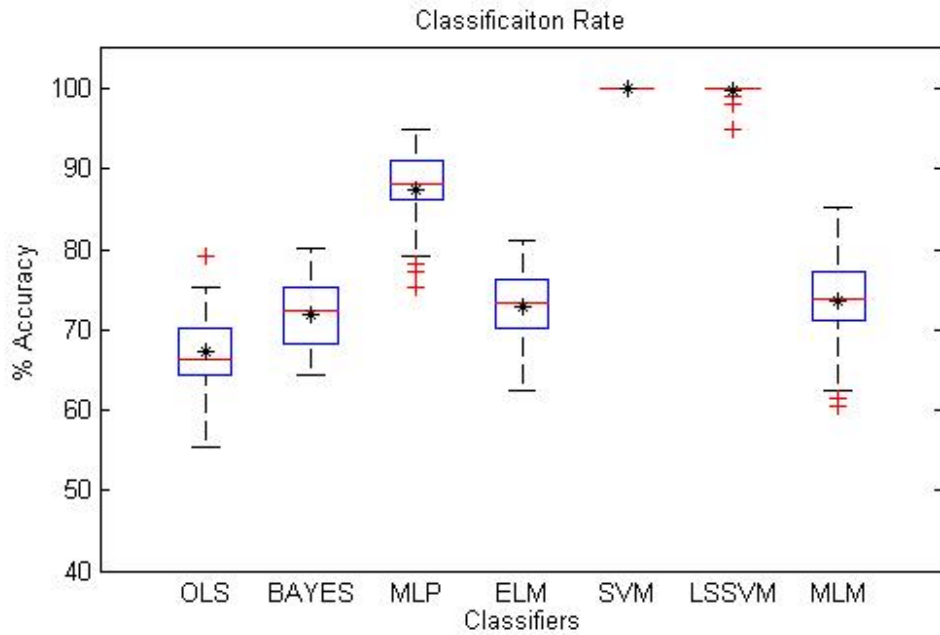


Figura 20 – Metodologia 04.

por estes classificadores.

Durante o treinamento, o classificador LSSVM acumula todas as amostras desta etapa, o que, no caso, equivale a 403 amostras (80% da quantidade total). Também, verificou-se a quantidade de vetores suporte que o classificador SVM estava acumulando, e este valor chegou, em média, a 60% do total de amostras (303 dados).

Para a aplicação destes algoritmos em um sistema embarcado, esta quantidade de parâmetros pode vir a ser um problema, visto que esta pode levar a um tempo de processamento excessivo e uma grande taxa de ocupação da memória do dispositivo.

Além dos métodos aplicados neste seção, tentou-se abordar o problema através da classificação multiclases e da opção de rejeição. Os resultados referentes a classificação multiclases são evidenciados na seção 5.4, já os referentes à opção de rejeição estão presentes na seção 5.5.

Na seção a seguir, a análise da seleção dos hiperparâmetros é indicada.

5.3 Seleção dos Hiperparâmetros

Como pode-se notar, na metodologia de comparação entre classificadores, a seleção dos hiperparâmetros ocorreu por meio de validação cruzada, ou seja, estes eram definidos e podiam ter valores variáveis a cada realização.

Com isso, após cada realização, guardava-se o valor de todos os hiperparâmetros selecionados para verificar se estes valores seguiam determinado padrão.

Na metodologia 1, na qual o banco de dados desequilibrado e o Hold Out do tipo 80% e 20% foram utilizados, os seguintes valores de hiperparâmetros foram atingidos:

- MLP: $q = 2$
- ELM: $q = 26$
- SVM: $C = 0,5$ ou 10 ; $\sigma^2 = 0,01$ ou 1
- LSSVM: $\gamma = 2$; $\sigma^2 = 0,5$
- MLM: $K = 10$;

Já na metodologia 3, na qual o banco de dados desequilibrado e o Hold Out equilibrando os dados durante o treinamento foram utilizados, os seguintes valores de hiperparâmetros foram atingidos:

- MLP: $q = 9$
- ELM: $q = 26$
- SVM: $C = 15$; $\sigma^2 = 5$
- LSSVM: $\gamma = 4$; $\sigma^2 = 0,5$
- MLM: $K = 9$;

Por fim, na metodologia 4, na qual o banco de dados equilibrado e o Hold Out do tipo 80% e 20% foram utilizados, os seguintes valores de hiperparâmetros foram atingidos:

- MLP: $q = 12$
- ELM: $q = 29$
- SVM: $C = 5$; $\sigma^2 = 0,01$
- LSSVM: $\gamma = 0,25$ ou $0,50$; $\sigma^2 = 128$
- MLM: $K = 10$;

5.4 Resultados Com Problema Multiclasses

Como os resultados dos classificadores SVM e LSSVM foram superiores aos demais para o problema binário, e estes necessitaram de uma grande quantidade de parâmetros para obter uma alta taxa de acerto na detecção de falhas, foram utilizadas duas técnicas visando o aumento de possibilidades para o desenvolvimento de um sistema embarcado com um algoritmo de aprendizado máquina: a opção de rejeição e a classificação multiclasses.

Esta última técnica foi utilizada pois, em suas formulações iniciais, os algoritmos OLS, MLP, ELM, MLM e Gaussianos não precisam de modificações para se adequar a problemas multiclasses.

Nesta metodologia, o problema de detecção de falhas é tratado como multiclasses somente durante o treinamento (atualização dos parâmetros de cada algoritmo). Já na etapa de teste, a confusão, por parte dos classificadores, entre as classes de falha não é considerada um erro. Erros são considerados apenas quando uma amostra da classes de motores em funcionamento normal é classificada como sendo de falta e vice-versa.

Primeiramente, tratou-se o problema a partir da divisão do banco de dados em 7 classes, onde cada combinação de tipo e nível de falha é considerada um erro.

Os resultado obtidos a partir deste método podem ser visualizados nas figuras 21 (utilizando o banco de dados 1 - desequilibrado) e 22 (utilizando o banco de dados 2 - com equilíbrio entre amostras normais e de falhas).

Em ambos os testes, não houve uma melhora significativa na taxa acerto dos classificadores, havendo, porém, uma redução na taxa de acerto do classificador MLP.

Deste modo, tentou-se um outra abordagem, onde o problema tratado neste trabalho foi dividido em 3 classes: normal, baixa impedância e alta impedância.

Assim como no método anterior, utilizou-se tanto o banco de dados 1 (42 amostras de motores sem falhas, 126 de baixa e 126 de alta impedância) como o banco de dados 2 (252 amostras de motores sem falha, 126 de baixa e 126 de alta impedância). Os resultados destes podem ser visualizados nas figuras 23 e 24.

Da mesma maneira que na classificação entre 7 classes, não houve uma melhora significativa na taxa de acerto média destes classificadores.

Por fim, após analisar algumas amostras do banco de dados utilizado neste dissertação, percebeu-se que, em alguns casos, amostras de alta e de baixa impedância, que correspondem o mesmo nível de curto circuito (número de espiras em curto), continham atributos com valores muito próximos.

Assim, levantou-se a hipótese de que o problema poderia ser melhor classificado se fosse dividido em 4 classes, uma correspondendo às amostras de motores em funcionamento

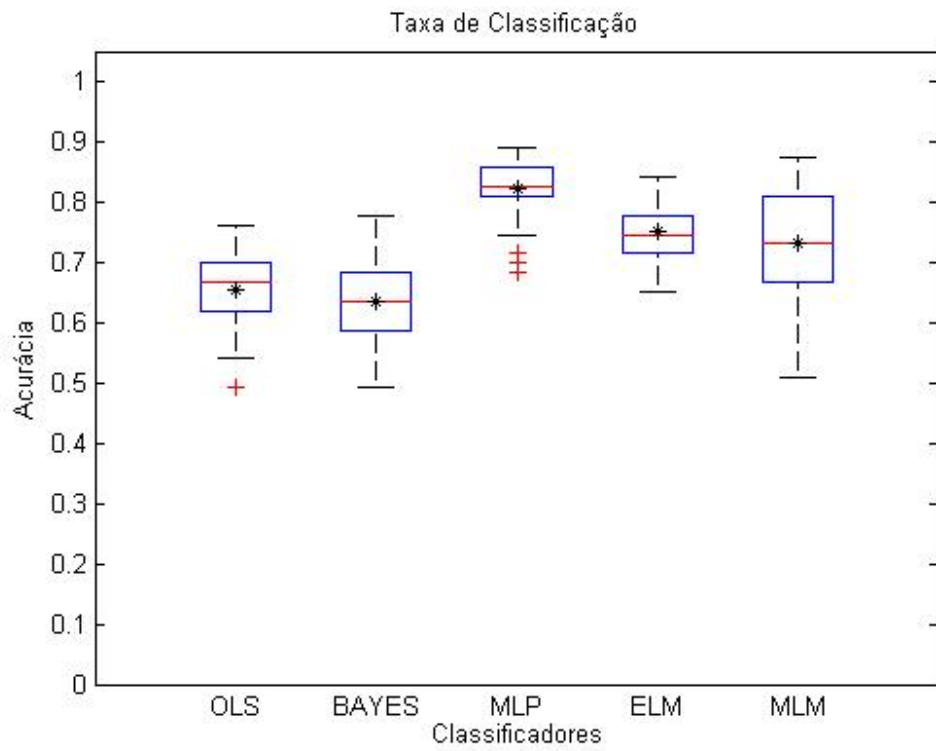


Figura 21 – Problema com 7 classes. Banco Desequilibrado de Dados.

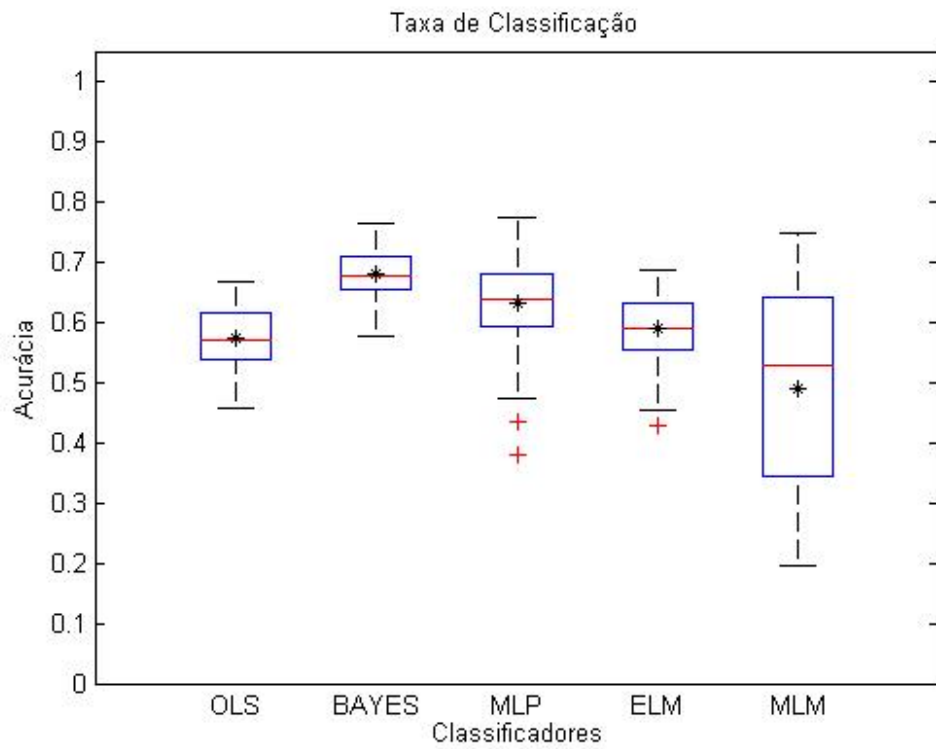


Figura 22 – Problema com 7 classes. Banco equilibrado de Dados.

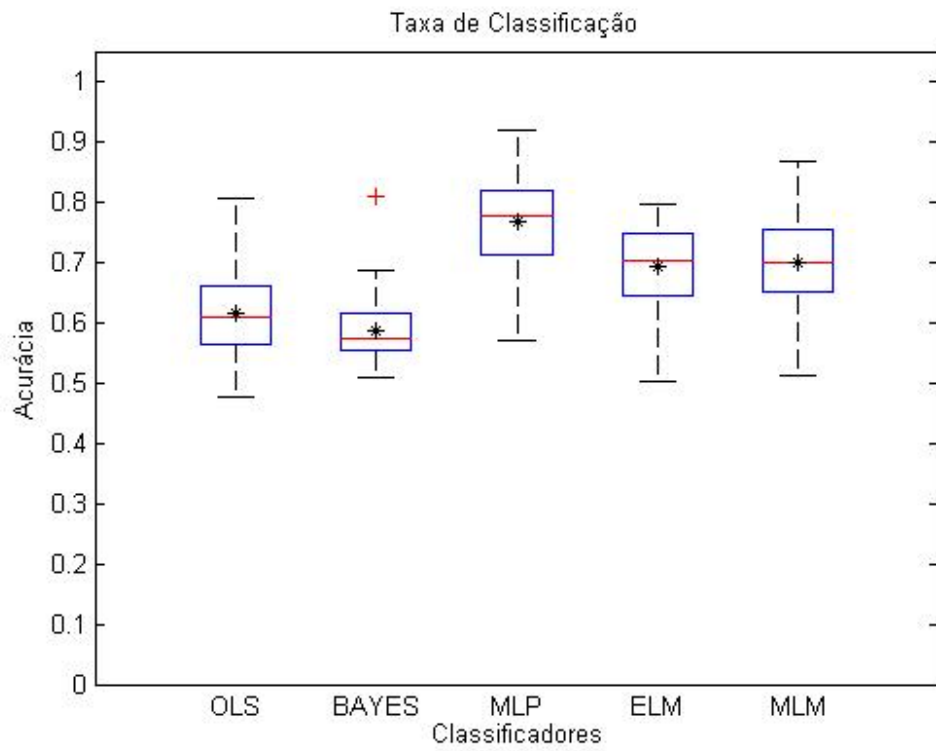


Figura 23 – Problema com 3 classes. Banco Desequilibrado de Dados.

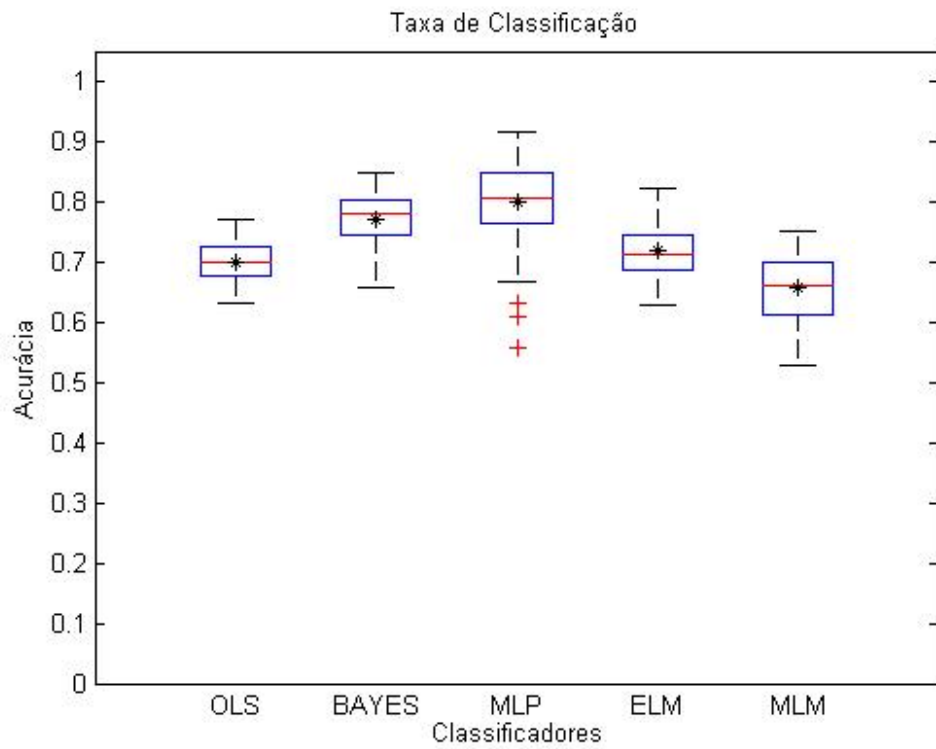


Figura 24 – Problema com 3 classes. Banco equilibrado de Dados.

normal, e as outras 3 representantes de cada nível de curto-circuito.

Os resultados desta abordagem estão dispostos nas figuras 25 e 26.

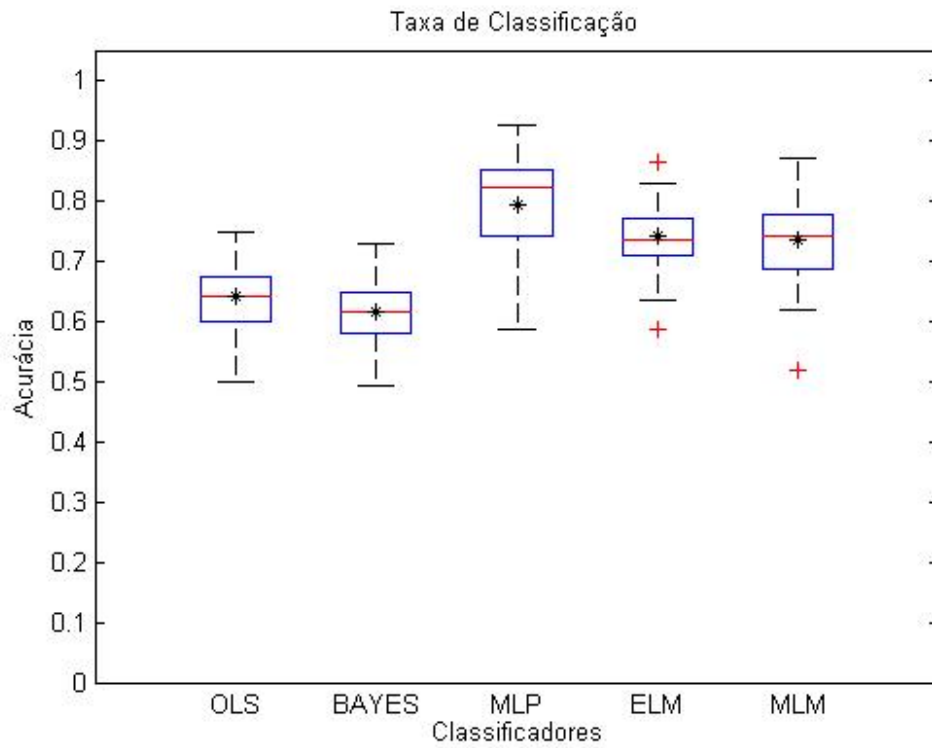


Figura 25 – Problema com 4 classes. Banco Desequilibrado de Dados.

Mesmo com este método, os resultados continuaram sendo considerados insatisfatórios. A pequena diferença entre os resultados da classificação com 4 (entre normal e níveis de falhas) e com 3 classes (entre normal, alta impedância e baixa impedância) demonstra que a diferença entre os conjuntos de alta e baixa impedância se mostra também pequena.

Novamente, isso se deve ao fato de que, durante a aquisição dos dados, a corrente do motor foi limitada para que este pudesse ser reutilizado diversas vezes para a formação do banco de dados. Assim, a dificuldade na classificação se deve à proximidade dos valores das amostras de motor em funcionamento normal em condição de falta.

Por isso, a técnica opção de rejeição foi adicionada, tanto para o problema binário como para os multiclases, visando a rejeição de amostras difíceis de serem classificadas, evitando, assim, erros durante classificação.

5.5 Resultados com Opção de Rejeição

Como já salientado anteriormente, a opção de rejeição consiste em rejeitar amostras que são difíceis de serem classificadas. Esta dificuldade é mensurada através do valor de

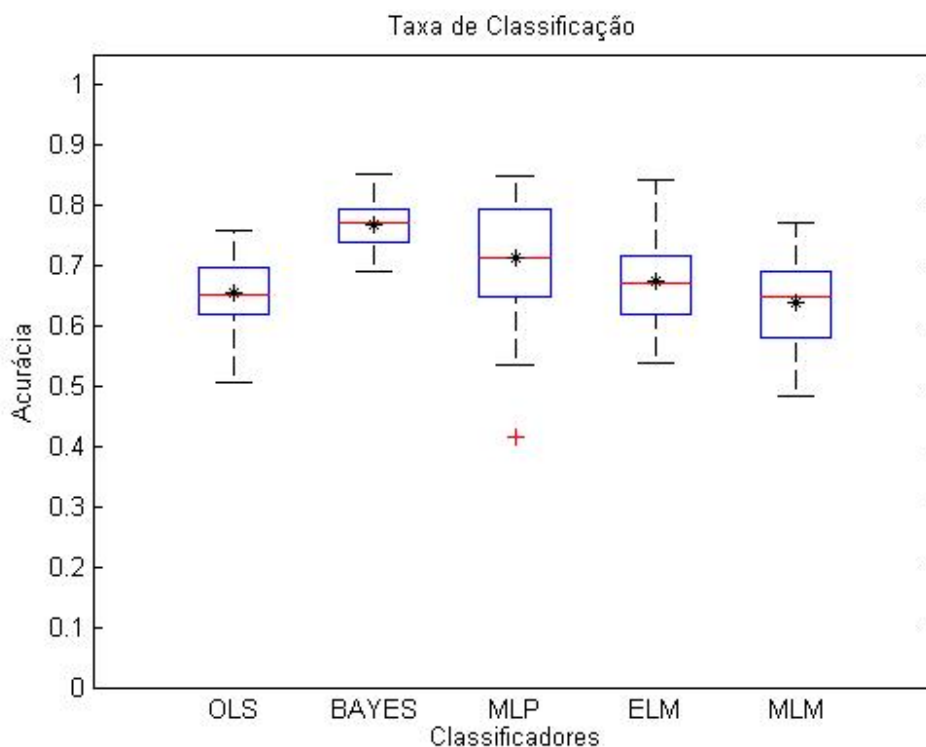


Figura 26 – Problema com 4 classes. Banco equilibrado de Dados.

saída da função discriminante de cada classificador. Se o valor máximo desta não for maior que determinado limiar, a amostra não é classificada.

A partir disto, espera-se melhorar a taxa de acerto de cada classificador, porém deve-se buscar um equilíbrio entre esta taxa e a quantidade de amostras rejeitadas por determinado classificador. No contexto da detecção de falhas, deve-se buscar um equilíbrio entre a quantidade de amostras rejeitadas (o que aumenta o tempo até que o classificador detecte alguma falha ou indique que o motor está em funcionamento normal), e a taxa de acerto global (preferencialmente com uma taxa de ocorrência de falsos positivos próxima a 0).

Para o problema binário, com um custo de rejeição de 0.25, os resultados com os bancos de dados 1 e 2 podem ser observados, respectivamente, nas figuras 27 e 28.

A taxa média de rejeição para os bancos de dados 1 e 2 foi respectivamente de [48% 0% 12% 31% 36%] e [63% 0% 4% 47% 44%] para os classificadores [OLS Gaussiano MLP ELM MLM].

Uma característica comum a todos os classificadores utilizando esta metodologia foi que, mesmo não obtendo altas taxas médias de acerto (menores que 90%), a quantidade de falsos positivos foi bastante reduzida.

Já os resultados da aplicação da opção de rejeição, utilizando o mesmo custo de

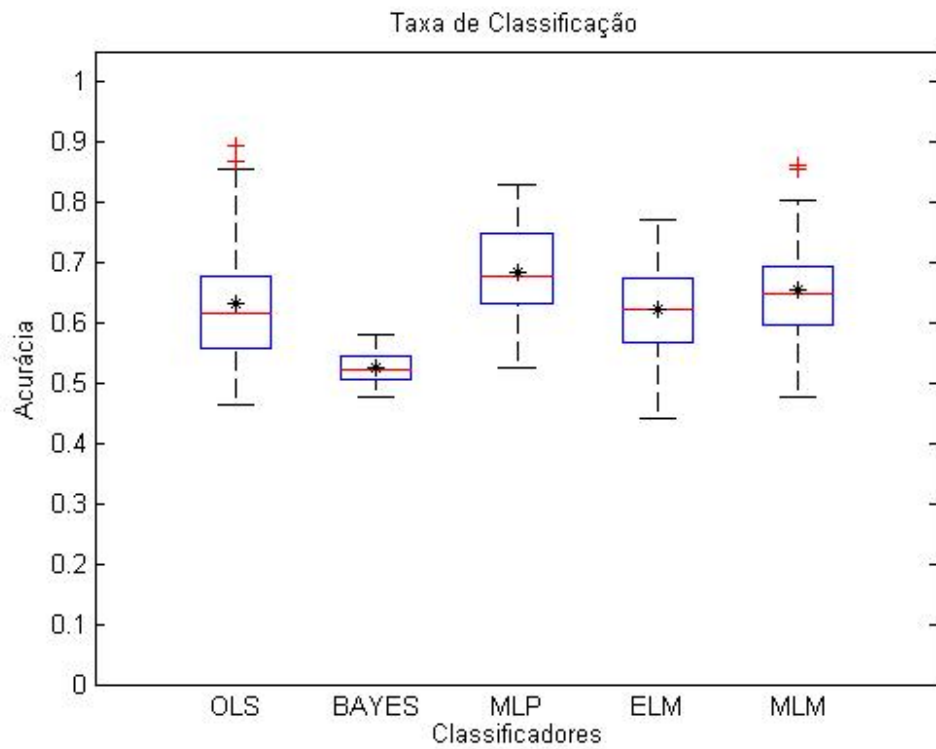


Figura 27 – Problema Binário. Com opção de Rejeição. Banco Desequilibrado de Dados.

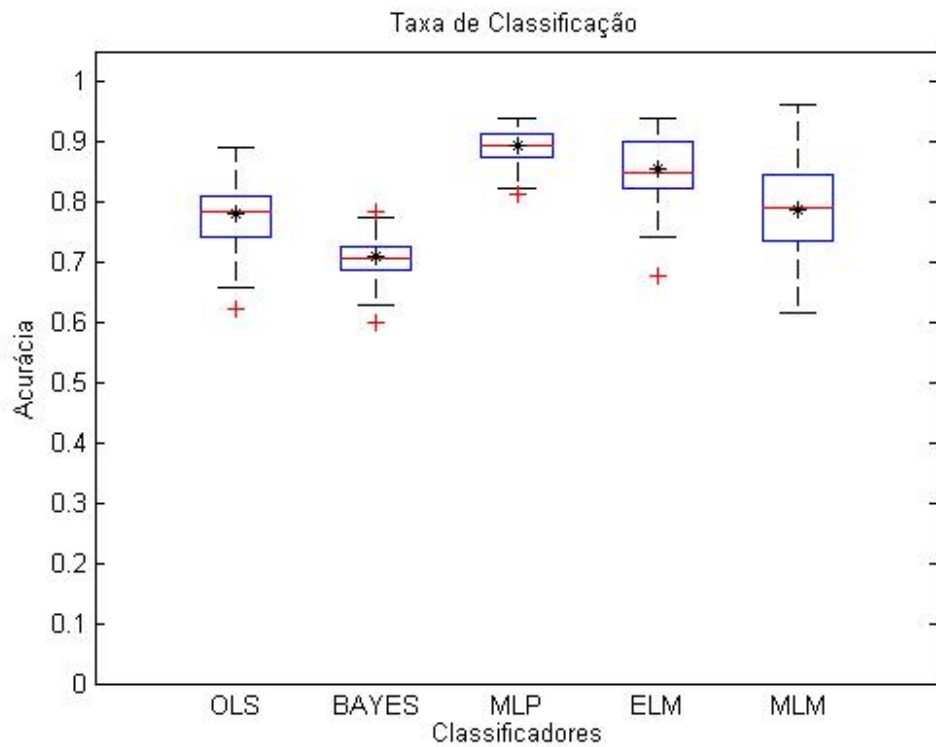


Figura 28 – Problema Binário. Com opção de Rejeição. Banco equilibrado de Dados.

rejeição anterior (0.25), aplicada ao problema com 7 classes está representado nas figuras 29 e 30.

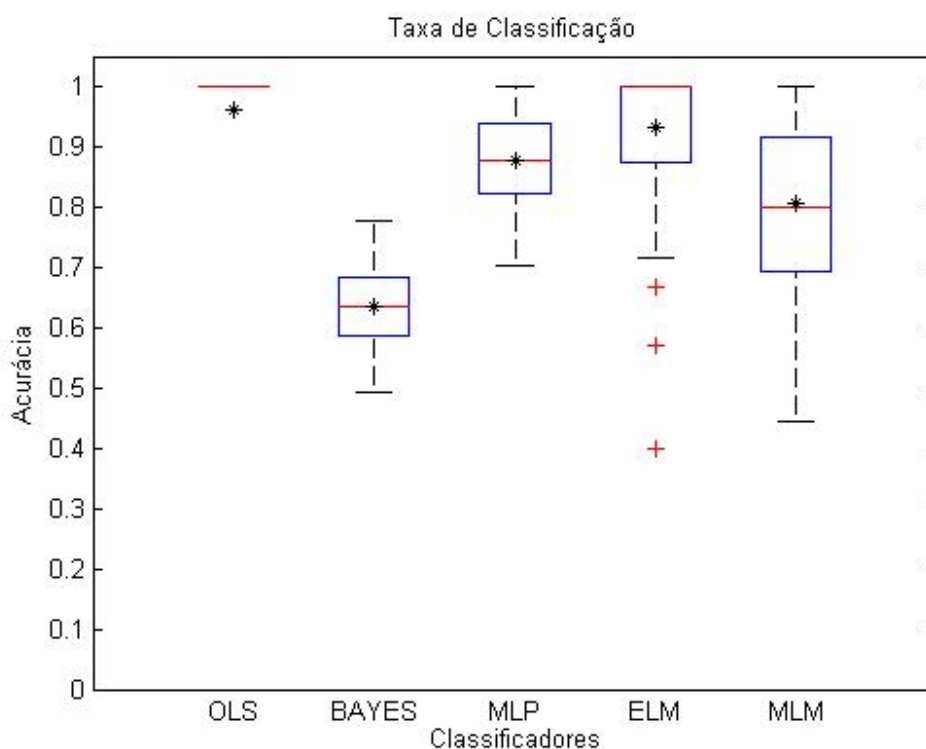


Figura 29 – Problema com 7 classes. Com opção de Rejeição. Banco Desequilibrado de Dados.

Adotando este método, a taxa média de rejeição para os bancos de dados 1 e 2 foi respectivamente de [93% 0 7% 91% 75%] e [98% 0 55% 97% 69%] para os classificadores [OLS Gaussiano MLP ELM MLM], e acertos de até 100% com os classificadores OLS, MLP, ELM e MLM são atingidos.

Mesmo com taxas de acerto bem superiores ao problema binário, pode-se perceber que as taxas médias de rejeição se encontram bastante elevadas.

Após estas verificações, a opção de rejeição também foi aplicada ao problema com 3 classes, o que levou aos resultados mostrados nas figuras 31 e 32

Adotando este método, a taxa média de rejeição para os bancos de dados 1 e 2 foi respectivamente de [83% 0 28% 59% 27%] e [96% 0 27% 75% 26%] para os classificadores [OLS Gaussiano MLP ELM MLM].

Tabmé, mesmo com uma taxa de rejeição de 75%, o classificador ELM obteve uma taxa de acerto média de 94%

Por fim, dado o problema com 4 classes, os seguintes resultados foram obtidos: 33 e 34

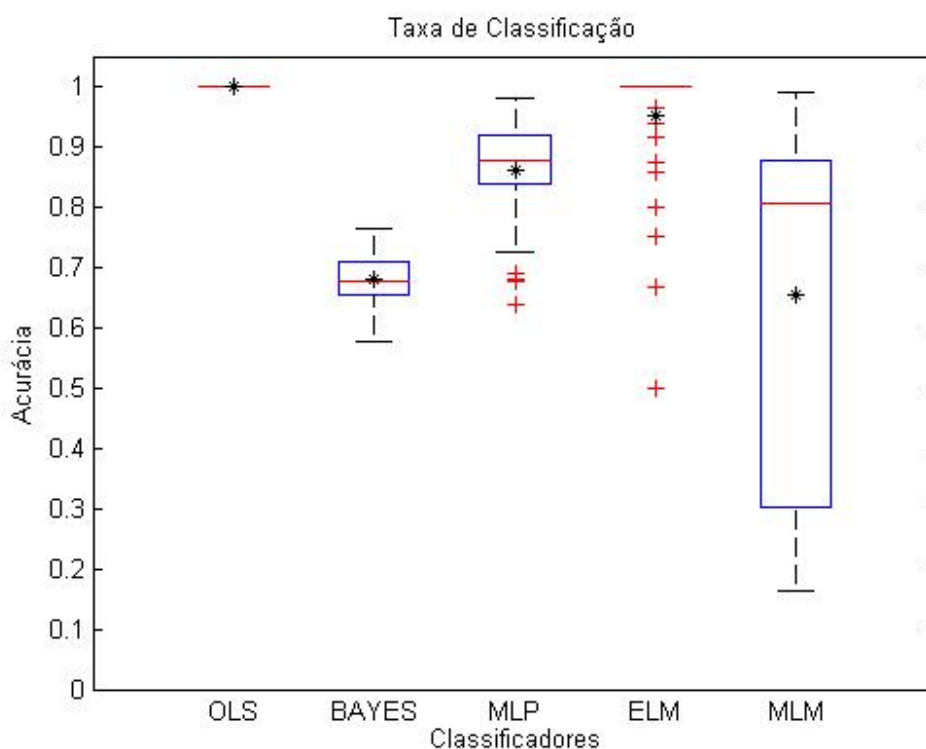


Figura 30 – Problema com 7 classes. Com opção de Rejeição. Banco equilibrado de Dados.

Adotando este método, a taxa média de rejeição para os bancos de dados 1 e 2 foi respectivamente de [0.91 0 0.43 0.75 0.37] e [98% 0 42% 87% 37%] para os classificadores [OLS Gaussiano MLP ELM MLM].

Em resumo, com a opção de rejeição, o classificador OLS rejeitou quase todas as amostras, mostrando que o problema não é facilmente separado linearmente.

Já o classificador MLP teve poucas diferenças com a aplicação desta técnica, evidenciando que a quantidade de amostras que são potenciais erros de classificação foram rejeitadas tanto quanto amostras que seriam classificadas corretamente.

Por outro lado, mesmo com uma taxa de rejeição elevada (75%), o classificador ELM obteve taxas de acerto médias superiores a 90% com o problema dividido em 4 classes.

Por fim, com os classificadores Gaussiano e MLM, houve uma melhora na taxa de acerto média, porém os resultados permaneceram abaixo de 90%.

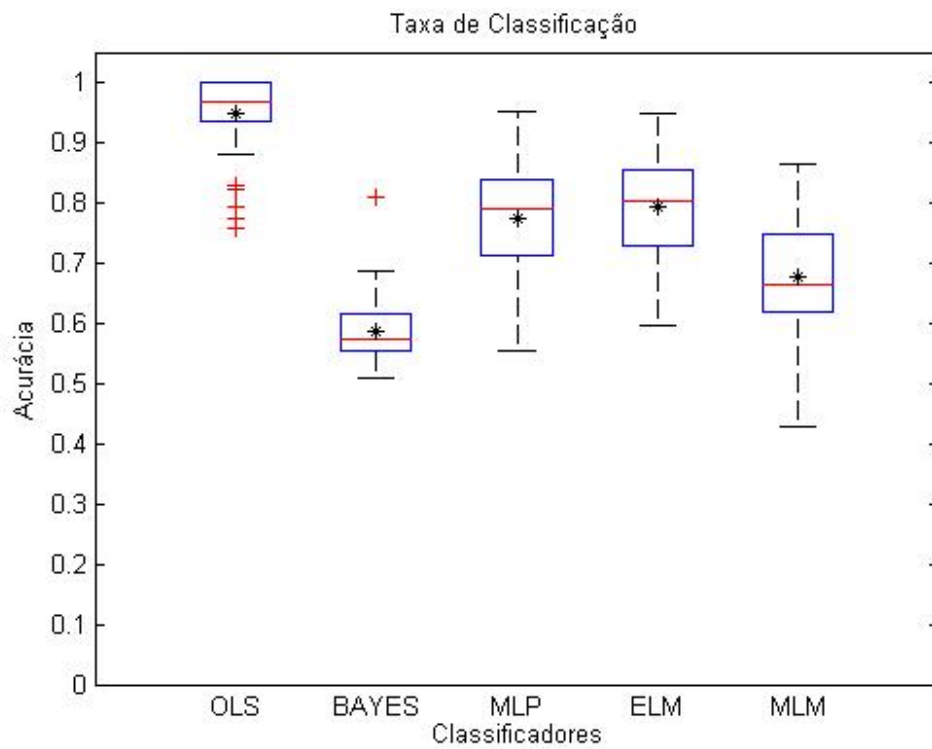


Figura 31 – Problema com 3 classes. Com opção de Rejeição. Banco Desequilibrado de Dados.

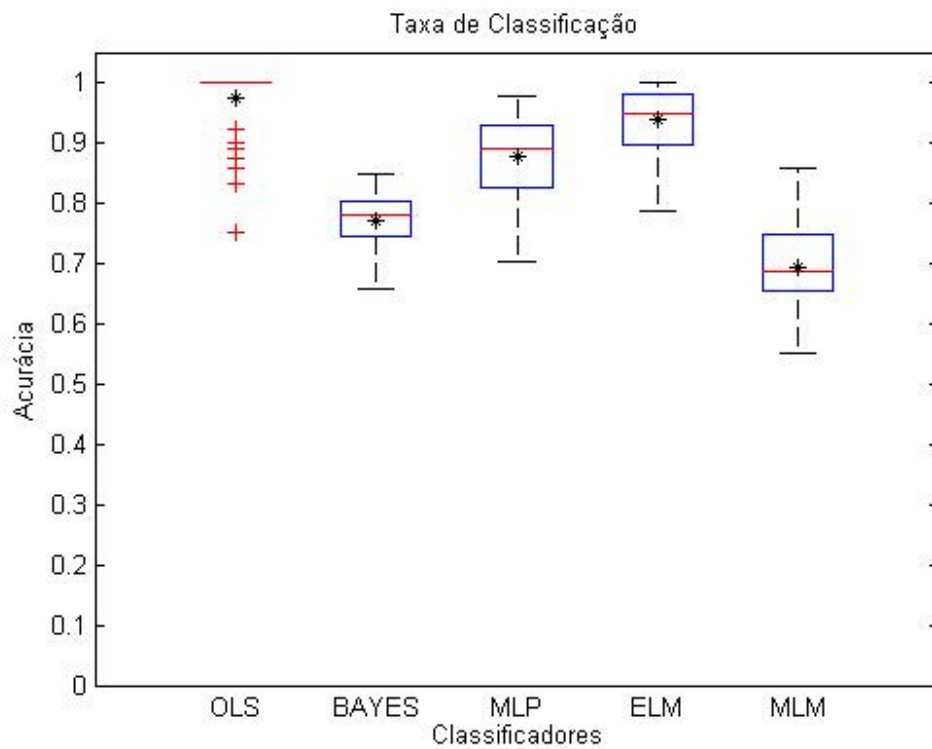


Figura 32 – Problema com 3 classes. Com opção de Rejeição. Banco equilibrado de Dados.

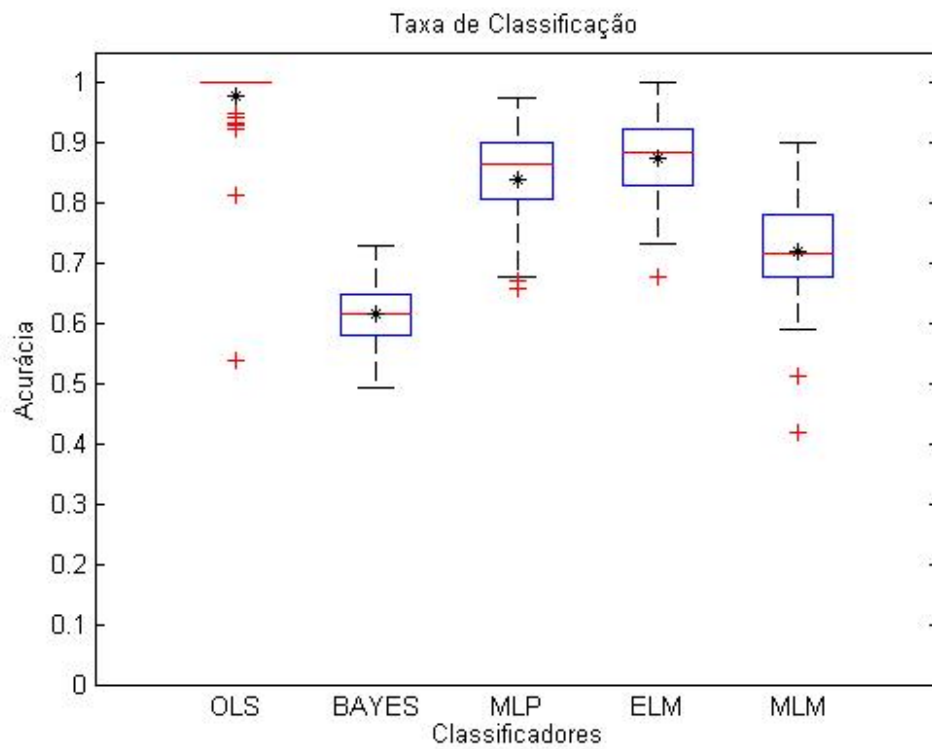


Figura 33 – Problema com 4 classes. Com opção de Rejeição. Banco Desequilibrado de Dados.

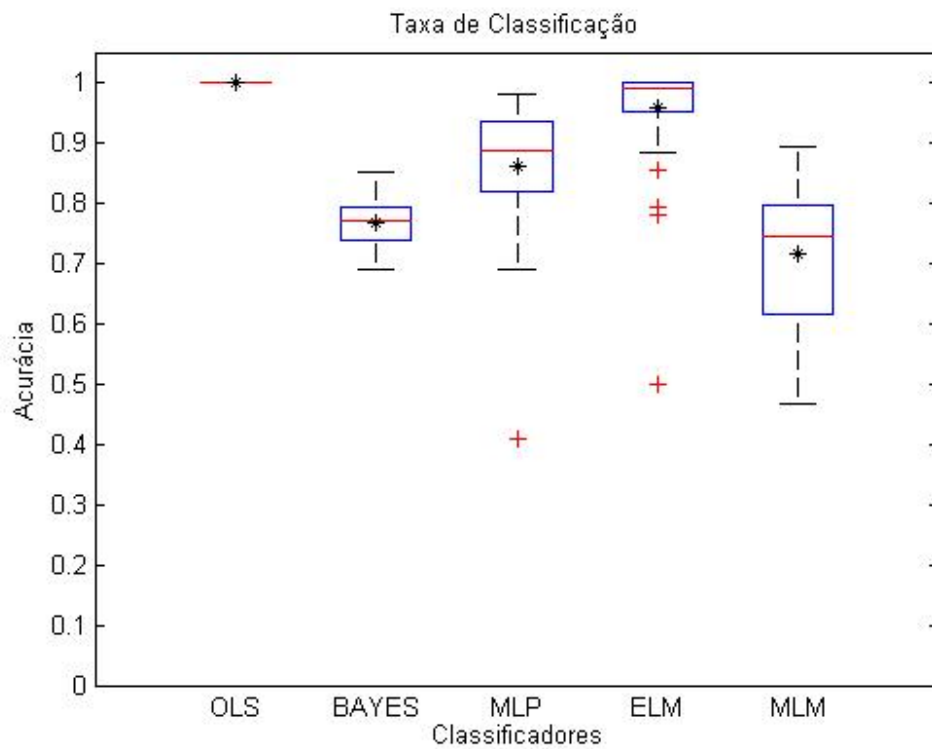


Figura 34 – Problema com 4 classes. Com opção de Rejeição. Banco equilibrado de Dados.

Conclusão

6.1 Objetivo Geral

Neste trabalho, buscou-se detectar falhas incipientes por curto-circuito entre espiras do motor de indução trifásico através de técnicas de aprendizado de máquina, e análise da assinatura de corrente do motor.

A partir do banco de dados reais gerado, através dos diagramas de caixa e gráficos de dispersão, pôde-se perceber, primeiramente, que na faixa de valores onde há variação dos atributos das amostras de motores em funcionamento normal, também há amostras de motores com falhas, fazendo com que seja difícil a separação destas duas classes por níveis.

Também, percebeu-se, pela baixa taxa de classificação dos classificadores lineares e gaussianos, que a divisão entre as classes deste problema não é linear e que a distribuição de probabilidade intraclasses não se mostrou gaussiana.

A partir da aplicação de algoritmos não-lineares, verificou-se que o desbalanceamento entre amostras de motores com falhas e em funcionamento normal prejudicou a taxa de acerto dos classificadores e, também, fez com que ocorresse uma alta taxa de falsos positivos (que é a indicação de uma falha, quando na verdade não há).

Para resolver o problema supracitado, a probabilidade a priori das classes foi equilibrada com a geração artificial de amostras de motores em funcionamento normal. Com isto, uma taxa média de acerto de 97% com o classificador SVM e de 99% com o classificador LSSVM, mostrando que a maior dificuldade para separar os dados estava tanto no desbalanceamento entre classes destes, como na pequena quantidade de amostras.

Porém, para obter as elevadas taxas de acerto, estes classificadores se utilizavam de muitos parâmetros em comparação com os demais classificadores. Como os resultados dos demais classificadores ainda não se mostravam satisfatórios, algumas técnicas no treinamento, juntamente com opção de rejeição foram utilizadas.

Primeiramente, tentou-se tratar o problema como multiclases, porém não houve aumento significativo na taxa de acerto média dos classificadores.

Já unindo a técnica supracitada com a opção de rejeição, mesmo com uma taxa de rejeição elevada, o classificador ELM obteve taxas de acerto média superiores a 90%.

Por fim, em um futuro próximo, novos métodos de extração de características serão

testados juntamente com novos e os já utilizados classificadores, de modo a escolher o melhor método (extrator, classificador) para ser embarcado em um processador. Dado os resultados atuais, o melhor classificador a ser embarcado é o SVM, visto que este possui taxas de acerto equivalentes ao LSSVM e utiliza menos dados de treinamento para atingir estas taxas.

6.2 Objetivos Específicos

Todos os objetivos específicos foram atingidos, visto que o banco de dados foi analisado estatisticamente; cinco paradigmas de classificadores foram implementados (incluindo sete algoritmos distintos); a eficácia de cada classificador foi estudada com o auxílio de ferramentas estatísticas, tais como gráfico de caixas e matriz de confusão; a estratégia de opção de rejeição foi utilizada e, com esta, pode-se notar uma diminuição na taxa de erros dos classificadores.

6.3 Trabalhos Futuros

Mesmo com os diversos estudos dessa dissertação, o problema de detecção de falhas ainda está longe de ser esgotado. Algumas sugestões para estudos futuros são listadas a seguir.

Primeiramente, o único método utilizado para a extração de atributos foi a FFT, a partir da qual algumas harmônicas da alimentação do motor foram utilizadas como atributos do problema. Uma nova análise seria aplicar outras técnicas para extração de atributos, tais como estatísticas de alta ordem.

Além disso, existem diversas falhas em motores que não foram estudadas, tais como quebra de barras e falhas em rolamentos. Pode-se verificar as similaridades e diferenças dos métodos de extração de atributos e detecção destas falhas.

Em relação ao banco de dados já gerado, percebeu-se que as classes de falha e normal estavam muito desequilibradas. Com isso, foi necessário gerar dados artificialmente. Neste trabalho, apenas um método foi aplicado para esta geração, porém outros métodos, tais como os baseados em clusters, podem ser aplicados.

Em relação aos classificadores SVM e LSSVM, pode-se estudar os métodos de poda destes e os meios para utilizá-los em classificação multi-classes.

Por fim, em relação às técnicas de classificação, como as classes de falha representam valores crescentes de falhas, pode ser verificada a utilização da classificação ordinal na detecção de falhas em motores.

Implementações

A.1 Implementação do Classificador ELM

Como mencionado anteriormente, o treinamento da rede ELM é mais rápido e de implementação mais simples que o MLP. Para fins de ilustração, segue a implementação deste no software Matlab.

Primeiramente, vamos assumir que o número de neurônios ocultos está definido em q e que a dimensão dos vetores de atributos \mathbf{x}_μ está definida em p . Além disso, vamos considerar que os vetores de atributos \mathbf{x}_μ , $\mu = 1, \dots, N_1$, usados no treinamento do classificador ELM estão dispostos ao longo das colunas da matriz \mathbb{X} e que os rótulos correspondentes estejam dispostos ao longo das colunas da matriz \mathbb{D} .

Assim, a sequência de comandos que leva à determinação da matriz \mathbf{H} e à estimação da matriz de pesos de saída β via Equação (3.41) é apresentada abaixo:

```

■ M = 0.1*rand(q,p+1);
■ U = M*X;
■ H = 1./(1+exp(-U));
■ B = D*H'*inv(H*H');

```

em que B denota a estimativa da matriz de pesos β . É importante salientar que, como visto acima, o comando `rand()` do Matlab foi utilizado em todas as rotinas que necessitavam de geração de números aleatórios, e este comando, por padrão, implementa o algoritmo Mersenne Twister ¹.

Além disso, como já mencionado na Seção 3.3, esta forma de se estimar β não é recomendada por ser muito susceptível a erros numéricos, além de não escalar bem. Neste caso, recomenda-se usar o operador *barra* (`/`) ou, preferencialmente, o comando `pinv`, por fornecer a solução de menor norma mesmo quando o posto da matriz H não é completo. Assim,

```

■ B = D/H;
■ B = D*pinv(H);

```

¹ <http://www.mathworks.com/help/matlab/ref/randstream.list.html>

Para a versão regularizada do classificador ELM, também é possível estimar $\hat{\beta}$ através da escrita direta no prompt do Matlab:

- `l = 0.01;`
- `I = ones(size(H*H'));`
- `B = D*H'/(H*H' + l*I);`

Referências

AMEL, B. et al. Classification and diagnosis of broken rotor bar faults in induction motor using spectral analysis and svm. In: *Ecological Vehicles and Renewable Energies (EVER), 2013 8th International Conference and Exhibition on*. [S.l.: s.n.], 2013. p. 1–5. Cited on page 17.

ASFANI, D. et al. Temporary short circuit detection in induction motor winding using combination of wavelet transform and neural network. *Expert Systems with Applications*, Elsevier, v. 39, n. 5, p. 5367–5375, 2012. Cited on page 17.

AVELAR, V. S.; BACCARINI, L. M. R.; AMARAL, G. F. V. Desenvolvimento de um sistema inteligente para diagnostico de falhas nos enrolamentos do estator de motores de indução. *X SABI-Simpósio Brasileiro de Automação Inteligente. São João Del Rei-MG-Brasil*, 2011. Cited on page 17.

BACHA, K. et al. Induction machine fault detection using stray flux emf measurement and neural network-based decision. *Electric Power Systems Research*, Elsevier, v. 78, n. 7, p. 1247–1255, 2008. Cited on page 16.

BONNETT, A. H. Root cause failure analysis for ac induction motors in the petroleum and chemical industry. In: *2010 Record of Conference Papers Industry Applications Society 57th Annual Petroleum and Chemical Industry Conference (PCIC)*. [S.l.: s.n.], 2010. Cited on page 17.

BOSLAUGH, S.; WATTERS, P. *Statistics in a Nutshell: A Desktop Quick Reference (In a Nutshell (O'Reilly))*. [S.l.]: O'Reilly Media, Inc., illustrated edition ed, 2008. Cited on page 63.

BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, v. 2, n. 2, p. 121–167, 1998. Cited on page 44.

CHOW, C. K. On optimum recognition error and reject tradeoff. *Information Theory, IEEE Transactions on*, v. 16, n. 1, p. 41–46, 1970. Cited 2 times on pages 61 e 62.

COELHO, D.; MEDEIROS, C. Short circuit incipient fault detection and supervision in a three-phase induction motor with a som-based algorithm. In: *Advances in Self-Organizing Maps*. [S.l.]: Springer, 2013. p. 315–323. Cited 4 times on pages 9, 22, 23 e 25.

- DAS, S.; PURKAIT, P.; CHAKRAVORTI, S. Separating induction motor current signature for stator winding faults from that due to supply voltage unbalances. In: *Power and Energy in NERIST (ICPEN), 2012 1st International Conference on*. [S.l.: s.n.], 2012. p. 1–6. Cited on page 22.
- DENG, W.; ZHENG, Q.; CHEN, L. Regularized extreme learning machine. In: *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM'09)*. [S.l.: s.n.], 2009. p. 389–395. Cited on page 42.
- EFTEKHARI, M. et al. Review of induction motor testing and monitoring methods for inter-turn stator winding faults. In: IEEE. *Electrical Engineering (ICEE), 2013 21st Iranian Conference on*. [S.l.], 2013. p. 1–6. Cited on page 18.
- FILHO, E. Q. S.; SANTOS, J. D. A.; BARRETO, G. A. Estudo comparativo de métodos de extração de características para classificação da qualidade de peles de caprinos com opção de rejeição. *Simpósio Brasileiro de Automação Inteligente (SBAI)*, p. 6, 2013. Cited on page 62.
- GHATE, V. N.; DUDUL, S. V. Optimal mlp neural network classifier for fault detection of three phase induction motor. *Expert Systems with Applications*, Elsevier, v. 37, n. 4, p. 3468–3481, 2010. Cited 2 times on pages 16 e 18.
- GHATE, V. N.; DUDUL, S. V. Cascade neural-network-based fault classifier for three-phase induction motor. *Industrial Electronics, IEEE Transactions on*, IEEE, v. 58, n. 5, p. 1555–1563, 2011. Cited on page 17.
- HAYKIN, S. S. Redes neurais artificiais: principios e praticas. *2a Edicao, Bookman, Sao Paulo, Brasil*, 2000. Cited 3 times on pages 37, 45 e 48.
- HOERL, A. E.; KENNARD, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, Taylor & Francis Group, v. 12, n. 1, p. 55–67, 1970. Cited on page 36.
- HORATA, P.; CHIEWCHANWATTANA, S.; SUNAT, K. Robust extreme learning machine. *Neurocomputing*, v. 102, p. 31–44, 2012. Cited on page 42.
- HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural networks*, Elsevier, v. 2, n. 5, p. 359–366, 1989. Cited on page 37.
- HUANG, G.-B.; WANG, D. H.; LAN, Y. Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, v. 2, p. 107–122, 2011. Cited on page 42.
- HUANG, G. B.; ZHU, Q. Y.; ZIEW, C. K. Extreme learning machine: Theory and applications. *Neurocomputing*, v. 70, n. 1–3, p. 489–501, 2006. Cited on page 41.
- JUNIOR, C. D. Sistema de simulação de cargas mecânicas para motor de indução acionado por inversor de frequência. Curitiba, 2013. Cited on page 16.
- LEE, J. et al. Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. *Mechanical Systems and Signal Processing*, Elsevier, v. 42, n. 1, p. 314–334, 2014. Cited 2 times on pages 18 e 21.

- LIU, N.; WANG, H. Ensemble based extreme learning machine. *IEEE Signal Processing Letters*, v. 17, n. 8, p. 754–757, 2010. Cited on page 42.
- MARTINS, J. F.; PIRES, V. F.; PIRES, A. Unsupervised neural-network-based algorithm for an on-line diagnosis of three-phase induction motor stator fault. *Industrial Electronics, IEEE Transactions on*, IEEE, v. 54, n. 1, p. 259–264, 2007. Cited 2 times on pages 17 e 18.
- MATSUOKA, K. Noise injection into inputs in back-propagation learning. *Industrial Electronics, IEEE Transactions on*, IEEE, v. 22, n. 3, p. 436–440, 1992. Cited on page 54.
- MATTHEWS, B. W. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, Elsevier, v. 405, n. 2, p. 442–451, 1975. Cited on page 63.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943. Cited on page 37.
- MICHE, Y. et al. OP-ELM: Optimally pruned extreme learning machine. *IEEE Transactions on Neural Networks*, v. 21, n. 1, p. 158–162, 2010. Cited on page 42.
- MICHE, Y. et al. TROP-ELM: a double-regularized ELM using LARS and Tikhonov regularization. *Neurocomputing*, v. 74, n. 16, p. 2413–2421, 2011. Cited on page 42.
- MOHAMMED, A. et al. Human face recognition based on multidimensional PCA and extreme learning machine. *Pattern Recognition*, v. 44, n. 10–11, p. 2588–2597, 2011. Cited on page 42.
- MONTEIRO, I. Q. *Métodos de Aprendizagem de Máquina para Reconhecimento de Faces: Uma Comparação de Desempenho*. Departamento de Engenharia de Teleinformática: [s.n.], 2009. Cited 2 times on pages 7 e 35.
- MOORE, E. H. On the reciprocal of the general algebraic matrix. *Bull. Amer. Math. Soc.*, v. 26, p. 394–395, 1920. Cited on page 35.
- NANDI, S.; TOLYAT, H. A.; LI, X. Condition monitoring and fault diagnosis of electrical motors-a review. *Energy Conversion, IEEE Transactions on*, IEEE, v. 20, n. 4, p. 719–729, 2005. Cited 2 times on pages 16 e 17.
- NATARAJAN, R. Failure identification of induction motors by sensing unbalanced stator currents. *Energy Conversion, IEEE Transactions on*, IEEE, v. 4, n. 4, p. 585–590, 1989. Cited on page 21.
- NETO, A. R. da R. et al. Diagnostic of pathology on the vertebral column with embedded reject option. In: *Pattern Recognition and Image Analysis*. [S.l.]: Springer, 2011. p. 588–595. Cited on page 62.
- NEUMANN, K.; STEIL, J. Optimizing extreme learning machines via ridge regression and batch intrinsic plasticity. *Neurocomputing*, v. 102, p. 23–30, 2013. Cited on page 42.
- NIRALI, R.; SHAH, S. Fuzzy decision based soft multi agent controller for speed control of three phase induction motor. *Transformation*, v. 2, p. 3Φ, 2011. Cited on page 16.

OLIVEIRA, A. G.; MEDEIROS, C. M. S. Stator winding interturns short circuit fault detection in a three phase induction motor driven by frequency converter using neural networks. In: *Energy Efficiency in Motor Driven Systems 2013 Conference, EEMODS*. [S.l.: s.n.], 2013. Cited 5 times on pages [7](#), [17](#), [23](#), [26](#) e [27](#).

PENMAN, J. et al. Detection and location of interturn short circuits in the stator windings of operating motors. *Energy conversion, IEEE transactions on*, v. 9, n. 4, p. 652–658, 1994. Cited on page [22](#).

PENROSE, R. A generalized inverse for matrices. In: *Mathematical proceedings of the Cambridge philosophical society*. [S.l.]: Cambridge Univ Press, 1955. p. 406–413. Cited on page [35](#).

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Cited on page [38](#).

RUMELHART, D.; HINTON, G.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, p. 533–536, 1986. Cited on page [39](#).

SAWA, T.; KUME, T. Motor drive technology-history and visions for the future. In: IEEE. *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*. [S.l.], 2004. v. 1, p. 2–9. Cited on page [16](#).

SAWITRI, D. et al. Early detection of unbalance voltage in three phase induction motor based on svm. In: IEEE. *Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED), 2013 9th IEEE International Symposium on*. [S.l.], 2013. p. 573–578. Cited on page [17](#).

SESHADRINATH, J.; SINGH, B.; PANIGRAHI, B. K. Incipient interturn fault diagnosis in induction machines using an analytic wavelet-based optimized bayesian inference. *Neural Networks and Learning Systems, IEEE Transactions on*, IEEE, v. 25, n. 5, p. 990–1001, 2014. Cited 3 times on pages [16](#), [17](#) e [21](#).

Souza Jr., A. H. et al. Minimal learning machine: A novel supervised distance-based approach for regression and classification. *Neurocomputing*, v. 164, n. 34–44, 2015. Cited 3 times on pages [7](#), [50](#) e [52](#).

SUYKENS, J. A. K.; VANDEWALLE, J. Least squares support vector machine classifiers. *Neural Processing Letters*, p. 293–300, 1999a. Cited on page [49](#).

TALLAM, R. M. et al. A survey of methods for detection of stator related faults in induction machines. In: *Diagnostics for Electric Machines, Power Electronics and Drives, 2003. SDEMPED 2003. 4th IEEE International Symposium on*. [S.l.: s.n.], 2003. p. 35–46. Cited on page [21](#).

THOMSON, W. T.; FENGER, M. Current signature analysis to detect induction motor faults. *Industry Applications Magazine, IEEE, Ieee*, v. 7, n. 4, p. 26–34, 2001. Cited 4 times on pages [16](#), [17](#), [21](#) e [22](#).

THORSEN, O.; DALVA, M. Condition monitoring methods, failure identification and analysis for high voltage motors in petrochemical industry. *International Conference on Electrical Machines and Drives*, v. 8, p. 109–113, 1997. Cited on page [22](#).

- TIKHONOV, A. N.; ARSENIN, V. I. *Solutions of ill-posed problems*. [S.l.]: Vh Winston, 1977. Cited on page 36.
- VAPNIK, V. N. *Statistical learning theory*. [S.l.]: Wiley New York, 1998. Cited on page 43.
- VAPNIK, V. N. *The nature of statistical learning theory*. [S.l.]: Springer Science & Business Media, 2000. Cited 2 times on pages 43 e 45.
- VENKADESAN, A.; HIMAVATHI, S.; MUTHURAMALINGAM, A. Performance comparison of neural architectures for on-line flux estimation in sensor-less vector-controlled im drives. *Neural Computing and Applications*, Springer, v. 22, n. 7-8, p. 1735–1744, 2013. Cited on page 16.
- VICO, J.; HUNT, R. Protection principles for electrical motors in the cement industry. In: IEEE. *Cement Industry Technical Conference, 2010 IEEE-IAS/PCA 52nd*. [S.l.], 2010. p. 1–13. Cited on page 16.
- YADAV, M.; WADHWANI, S. Automatic fault classification of rolling element bearing using wavelet packet decomposition and artificial neural network. *International journal of engineering and technology*, v. 3, n. 4, p. 270–276, 2011. Cited on page 17.
- ZAREI, J. Induction motors bearing fault detection using pattern recognition techniques. *Expert systems with Applications*, Elsevier, v. 39, n. 1, p. 68–73, 2012. Cited on page 17.
- ZHANG, G. P. Neural networks for classification: a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, IEEE, v. 30, n. 4, p. 451–462, 2000. Cited on page 29.
- ZONG, W.; HUANG, G.-B. Face recognition based on extreme learning machine. *Neurocomputing*, v. 74, n. 16, p. 2541–2551, 2011. Cited on page 42.