



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA DE TELEINFOMÁTICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA**  
**MESTRADO ACADÊMICO EM ENGENHARIA DE TELEINFORMÁTICA**

**DIEGO PERDIGÃO SOUSA**

**DETECÇÃO DE FALHAS DE CURTO-CIRCUITO EM MOTORES DE INDUÇÃO**  
**TRIFÁSICOS USANDO CLASSIFICADORES BASEADOS EM PROTÓTIPOS**

**FORTALEZA**

**2019**

DIEGO PERDIGÃO SOUSA

DETECÇÃO DE FALHAS DE CURTO-CIRCUITO EM MOTORES DE INDUÇÃO  
TRIFÁSICOS USANDO CLASSIFICADORES BASEADOS EM PROTÓTIPOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia de Teleinformática do Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia 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. Charles Casimiro Cavalcante

Coorientador: Prof. Dr. Guilherme de Alencar Barreto

FORTALEZA

2019

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

S696d Sousa, Diego Perdigão.  
Detecção de falhas de curto-circuito em motores de indução trifásicos usando classificadores baseados em protótipos / Diego Perdigão Sousa. – 2019.  
148 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2019.

Orientação: Prof. Dr. Charles Casimiro Cavalcante.

Coorientação: Prof. Dr. Guilherme de Alencar Barreto.

1. Detecção de falhas. 2. Motores de indução trifásicos. 3. Aprendizado baseado em protótipos. I. Título.  
CDD 621.38

---

DIEGO PERDIGÃO SOUSA

DETECÇÃO DE FALHAS DE CURTO-CIRCUITO EM MOTORES DE INDUÇÃO  
TRIFÁSICOS USANDO CLASSIFICADORES BASEADOS EM PROTÓTIPOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia de Teleinformática do Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas

Aprovada em: 11 de Fevereiro de 2019

BANCA EXAMINADORA

---

Prof. Dr. Charles Casimiro Cavalcante (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Guilherme de Alencar  
Barreto (Coorientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Cláudio Marques de Sá Medeiros  
Instituto Federal de Educação Ciência e Tecnologia  
do Ceará (IFCE)

---

Prof. Dr. José Daniel de Alencar Santos  
Instituto Federal de Educação Ciência e Tecnologia  
do Ceará (IFCE)

Dedico este projeto de pesquisa à minha esposa  
Alyne Morais cuja presença foi essencial para a  
conclusão deste trabalho. Grato pelo seu apoio  
incondicional. Te amo.

## **AGRADECIMENTOS**

A Deus por todas as oportunidades obtidas.

À minha esposa, Alyne Moraes, por todo incentivo, compreensão e, principalmente, também acreditar neste sonho.

Aos meus pais, Suerda e Sousa, pelos sacrifícios e investimentos destinados à minha educação e à minha formação moral, ética e profissional.

À minha tia, Aline, por estar presente em tantos momentos da minha vida.

À minha avó, Antônia, por toda dedicação e carinho.

Ao Prof. Dr. Guilherme de Alencar Barreto, pela excelente orientação desde a minha graduação.

Ao Prof. Dr. Charles Casimiro Cavalcante, pela excelente orientação.

Aos participantes da banca examinadora, Prof. Dr. Cláudio Marques de Sá Medeiros e Prof. Dr. José Daniel de Alencar Santos, pelo tempo, pelas valiosas colaborações e sugestões.

Aos amigos que convivi diariamente durante o mestrado, por todos os momentos de estudos e troca de experiências. Dentre eles, em especial, Júlio Peixoto, David Coelho e Polycarpo Neto.

Aos professores e funcionários do Departamento de Engenharia de Teleinformática que de forma direta ou indireta participaram do desenvolvimento deste trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

“O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis.”

(José de Alencar)

## RESUMO

Os motores de indução trifásicos constituem-se em um dos equipamentos mais importantes da indústria moderna. Contudo, em muitas situações, tais equipamentos ficam submetidos a condições de uso inadequadas, tais como em ambientes com temperatura e umidade elevadas, variações bruscas de carga acima das especificadas, vibrações excessivas, dentre outras. Estas condições deixam os motores mais susceptíveis a falhas diversas, seja de natureza externa ou interna, que são obviamente indesejadas nos processos industriais. Neste contexto, a manutenção preditiva desempenha papel relevante, em que a detecção e o correto diagnóstico de falhas em tempo hábil leva ao aumento da vida útil do motor e, conseqüentemente, à diminuição de custos com parada da produção com manutenção corretiva. Diante destes fatores, nesta dissertação propõe-se uma metodologia de detecção de falhas por curto-circuito em motores de indução trifásicos que envolve algoritmos baseados em protótipos. Para este fim, são usadas tanto técnicas não supervisionadas - como o algoritmo  $K$ -médias - quanto supervisionadas, como os classificadores da família LVQ (*Learning Vector Quantization*).

A metodologia inicia-se com a busca pelo número ótimo de protótipos a partir da análise não supervisionada de agrupamentos e de técnicas de validação de agrupamentos. Em seguida, os protótipos assim encontrados são utilizados no treinamento supervisionado de vários classificadores da família LVQ. Avalia-se com profundidade a influência que cada tipo de critério de validação de agrupamentos exerce sobre os vários classificadores LVQ implementados. Em particular, o classificador GRLVQ (*Generalized Relevance Learning Vector Quantization*) obteve os melhores resultados ao apresentar taxas de acerto máximo de 98,3%, com os critérios Dunn e Silhueta destacando-se como os mais eficientes na determinação da quantidade ótima de protótipos.

**Palavras-chave:** Detecção de falhas. Motores de indução trifásicos. Aprendizado baseado em protótipos.

## ABSTRACT

Three-phase induction motors are one of the most important equipment of modern industry. However, in many situations, these equipment are subject to inappropriate conditions such as in environments with high temperatures and humidity, abrupt variations of load above specified, excessive vibrations, among others. These conditions make motors more susceptible to various failures, whether external or internal, which are obviously undesirable in industrial processes. In this context, the predictive maintenance plays a relevant role, where the detection and correct diagnosis of failures in a timely manner leads to increasing the useful life of the motor and, consequently, to the reduction of costs with production stoppage due to corrective maintenance. Considering these factors, this dissertation proposes a methodology for detecting short-circuit failures in three-phase induction motors, which involves prototypes-based algorithms. To this end, both unsupervised techniques - such as the  $K$ -means and supervised algorithm, such as the LVQ (Learning Vector Quantization) family classifiers are used.

The methodology starts with the seeking of the optimal number of prototypes from the unsupervised analysis of clusters and techniques clustering validation. Then, the prototypes that were found are used in the supervised training of various classifiers of the LVQ family. The influence that each type of clustering validation criterion exerts on the various LVQ classifiers implemented is deeply evaluated. In particular, the GRLVQ (Generalized Relevance Learning Vector Quantization) classifier obtained the best results where it presented a maximum classification rate of 98.3%, with the Dunn and Silhouette criteria standing out as the most efficient in determining the optimal quantity of prototypes.

**Keywords:** Failures detection. Three-phase induction motors. Prototype-based learning.

## LISTA DE FIGURAS

Figura 1 – Ilustração pictórica das principais componentes de um motor de indução . . .	28
Figura 2 – Proporção entre tipos de falhas em motores de indução . . . . .	29
Figura 3 – Exemplo de falha por curto-circuito entre espiras . . . . .	30
Figura 4 – Bancada de testes utilizada na aquisição da base de dados . . . . .	37
Figura 5 – Características das amostras . . . . .	40
Figura 6 – Possíveis representações de dados bidimensionais através de protótipos . . .	45
Figura 7 – Projeção das RC em diferentes áreas . . . . .	46
Figura 8 – Ilustração das fronteiras teóricas bayesianas . . . . .	59
Figura 9 – Ilustração simplificada dos princípios de projeto de classificadores LVQ . .	59
Figura 10 – Projeção dos classificadores LVQ heurísticos . . . . .	60
Figura 11 – Exemplo unidimensional da janela de restrição . . . . .	63
Figura 12 – Família LVQ dos algoritmos de maximização da margem de separação . . .	65
Figura 13 – Ilustrações de margens . . . . .	66
Figura 14 – Família LVQ dos algoritmos de maximização da verossimilhança . . . . .	74
Figura 15 – Critérios de validação propostos para utilização . . . . .	79
Figura 16 – Diagrama da preparação dos dados . . . . .	80
Figura 17 – Diagrama da metodologia proposta . . . . .	80
Figura 18 – Frequência da moda das sugestões dos critérios de validação de agrupamentos utilizados . . . . .	90
Figura 19 – Frequência das sugestões dos critérios de validação de agrupamentos utilizados	92
Figura 20 – Frequência das sugestões dos critérios de validação de agrupamentos utilizados	93
Figura 21 – Frequência das sugestões dos critérios de validação de agrupamentos utilizados	94
Figura 22 – Frequência da moda das sugestões dos critérios de validação de agrupamentos utilizados para o banco de dados 2 . . . . .	95
Figura 23 – Frequência das sugestões dos critérios de validação de agrupamentos utilizados	98
Figura 24 – Frequência das sugestões dos critérios de validação de agrupamentos utilizados	99
Figura 25 – Frequência das sugestões dos critérios de validação de agrupamentos utilizados	100
Figura 26 – Taxas de acertos encontradas para classificação ternária do banco de dados 1	103
Figura 27 – Taxas de acertos mapeadas para classificação binária do banco de dados 1. .	104
Figura 28 – Taxas de sensibilidade encontradas para o banco de dados 1 . . . . .	114
Figura 29 – Taxas de especificidade encontradas para o banco de dados 1 . . . . .	115

Figura 30 – Taxas de acertos encontradas para classificação ternária do banco de dados 2	119
Figura 31 – Taxas de acertos mapeadas para classificação binária do banco de dados 2	120
Figura 32 – Taxas de sensibilidade encontradas para o banco de dados 2. . . . .	122
Figura 33 – Taxas de especificidade encontradas para o banco de dados 2. . . . .	123

## LISTA DE TABELAS

Tabela 1 – Características do MIT utilizado . . . . .	37
Tabela 2 – Condições de operação do motor . . . . .	40
Tabela 3 – Distribuição de classes por <i>cluster</i> entre 8 <i>clusters</i> . . . . .	41
Tabela 4 – Penalidades associadas aos diferentes critérios utilizados . . . . .	54
Tabela 5 – Principais algoritmos de maximização da margem de separação . . . . .	66
Tabela 6 – Principais algoritmos de maximização da margem de separação . . . . .	67
Tabela 7 – Principais algoritmos de maximização da verossimilhança . . . . .	75
Tabela 8 – Nomes por extenso dos principais algoritmos de maximização da verossimilhança . . . . .	75
Tabela 9 – Características dos classificadores LVQ envolvidos. . . . .	84
Tabela 10 – Apresentação de uma matriz de confusão binária . . . . .	86
Tabela 11 – Valores majoritários de $K_{opt}$ encontrados para BD1 e BD2 . . . . .	101
Tabela 12 – Melhor matriz de confusão do GLVQ linear em classificação binária do banco de dados 1 para metodologia de treinamento 1 . . . . .	107
Tabela 13 – Melhor matriz de confusão do GLVQ linear em classificação ternária do banco de dados 1 para metodologia de treinamento 1 . . . . .	108
Tabela 14 – Pior matriz de confusão do GLVQ linear em classificação binária do banco de dados 1 para metodologia de treinamento 1 . . . . .	108
Tabela 15 – Pior matriz de confusão do GLVQ linear em classificação ternária do banco de dados 1 para metodologia de treinamento 1 . . . . .	108
Tabela 16 – Melhor matriz de confusão do GLVQ linear em classificação binária do banco de dados 1 para metodologia de treinamento 2 . . . . .	108
Tabela 17 – Melhor matriz de confusão do GLVQ linear em classificação ternária do banco de dados 1 para metodologia de treinamento 2 . . . . .	108
Tabela 18 – Pior matriz de confusão do GLVQ linear em classificação binária do banco de dados 1 para metodologia de treinamento 2 . . . . .	109
Tabela 19 – Pior matriz de confusão do GLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 2 . . . . .	109
Tabela 20 – Melhor matriz de confusão do GLVQ linear em classificação binária do banco de dados 1 para metodologia de treinamento 3 . . . . .	109

Tabela 21 – Melhor matriz de confusão do GLVQ linear em classificação ternária do banco de dados 1 para metodologia de treinamento 3 . . . . .	109
Tabela 22 – Pior matriz de confusão do GLVQ linear em classificação binária do banco de dados 1 para metodologia de treinamento 3 . . . . .	109
Tabela 23 – Pior matriz de confusão do GLVQ linear em classificação ternária do banco de dados 1 para metodologia de treinamento 3 . . . . .	110
Tabela 24 – Melhor matriz de confusão do GRLVQ em classificação binária do banco de dados 1 para metodologia de treinamento 1 . . . . .	110
Tabela 25 – Melhor matriz de confusão do GRLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 1 . . . . .	110
Tabela 26 – Pior matriz de confusão do GRLVQ em classificação binária do banco de dados 1 para metodologia de treinamento 1 . . . . .	111
Tabela 27 – Pior matriz de confusão do GRLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 1 . . . . .	111
Tabela 28 – Melhor matriz de confusão do GRLVQ em classificação binária do banco de dados 1 para metodologia de treinamento 2 . . . . .	111
Tabela 29 – Melhor matriz de confusão do GRLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 2 . . . . .	111
Tabela 30 – Pior matriz de confusão do GRLVQ em classificação binária do banco de dados 1 para metodologia de treinamento 2 . . . . .	112
Tabela 31 – Pior matriz de confusão do GRLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 2 . . . . .	112
Tabela 32 – Melhor matriz de confusão do GRLVQ em classificação binária do banco de dados 1 para metodologia de treinamento 3 . . . . .	112
Tabela 33 – Melhor matriz de confusão do GRLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 3 . . . . .	113
Tabela 34 – Pior matriz de confusão do GRLVQ em classificação binária do banco de dados 1 para metodologia de treinamento 3 . . . . .	113
Tabela 35 – Pior matriz de confusão do GRLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 3 . . . . .	113
Tabela 36 – Sugestões de distribuição de protótipos para as taxas de acerto do banco de dados 1 . . . . .	116

Tabela 37 – Sugestões e moda de distribuição de protótipos para a maior taxa de acerto do banco de dados 1 para metodologia de treinamento 1 . . . . .	117
Tabela 38 – Sugestões e moda de distribuição de protótipos para a maior taxa de acerto do banco de dados 1 para metodologia de treinamento 2 . . . . .	117
Tabela 39 – Valores encontrados para os elementos do vetor de ponderação para o banco de dados 1 e metodologia de treinamento 1 . . . . .	118
Tabela 40 – Valores encontrados para os elementos do vetor de ponderação para o banco de dados 1 e metodologia de treinamento 2 . . . . .	118
Tabela 41 – Valores encontrados para os elementos do vetor de ponderação para o banco de dados 1 e metodologia de treinamento 3 . . . . .	118
Tabela 42 – Melhor matriz de confusão do GLVQ linear em classificação binária do banco de dados 2 para metodologia de treinamento 1 . . . . .	124
Tabela 43 – Melhor matriz de confusão do GLVQ linear em classificação ternária do banco de dados 2 para metodologia de treinamento 1 . . . . .	124
Tabela 44 – Pior matriz de confusão do GLVQ linear em classificação binária do banco de dados 2 para metodologia de treinamento 1 . . . . .	124
Tabela 45 – Pior matriz de confusão do GLVQ linear em classificação ternária do banco de dados 2 para metodologia de treinamento 1 . . . . .	125
Tabela 46 – Melhor matriz de confusão do GLVQ linear em classificação binária do banco de dados 2 para metodologia de treinamento 2 . . . . .	125
Tabela 47 – Melhor matriz de confusão do GLVQ linear em classificação ternária do banco de dados 2 para metodologia de treinamento 2 . . . . .	125
Tabela 48 – Pior matriz de confusão do GLVQ linear em classificação binária do banco de dados 2 para metodologia de treinamento 2 . . . . .	126
Tabela 49 – Pior matriz de confusão do GLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 2 . . . . .	126
Tabela 50 – Melhor matriz de confusão do GRLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 1 . . . . .	126
Tabela 51 – Melhor matriz de confusão do GRLVQ em classificação binária do banco de dados 2 para metodologia de treinamento 1 . . . . .	127
Tabela 52 – Pior matriz de confusão do GRLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 1 . . . . .	127

Tabela 53 – Pior matriz de confusão do GRLVQ em classificação binária do banco de dados 2 para metodologia de treinamento 1 . . . . .	127
Tabela 54 – Melhor matriz de confusão do GRLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 2 . . . . .	127
Tabela 55 – Melhor matriz de confusão do GRLVQ em classificação binária do banco de dados 2 para metodologia de treinamento 2 . . . . .	128
Tabela 56 – Pior matriz de confusão do GRLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 2 . . . . .	128
Tabela 57 – Pior matriz de confusão do GRLVQ em classificação binária do banco de dados 2 para metodologia de treinamento 2 . . . . .	128
Tabela 58 – Melhor matriz de confusão do LVQ1 em classificação ternária do banco de dados 2 para metodologia de treinamento 3 . . . . .	129
Tabela 59 – Melhor matriz de confusão do LVQ1 em classificação binária do banco de dados 2 para metodologia de treinamento 3 . . . . .	129
Tabela 60 – Pior matriz de confusão do LVQ1 em classificação ternária do banco de dados 2 para metodologia de treinamento 3 . . . . .	129
Tabela 61 – Pior matriz de confusão do LVQ1 em classificação binária do banco de dados 2 para metodologia de treinamento 3 . . . . .	130
Tabela 62 – Melhor matriz de confusão do GRLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 3 . . . . .	130
Tabela 63 – Melhor matriz de confusão do GRLVQ em classificação binária do banco de dados 2 para metodologia de treinamento 3 . . . . .	131
Tabela 64 – Pior matriz de confusão do GRLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 3 . . . . .	131
Tabela 65 – Pior matriz de confusão do GRLVQ em classificação binária do banco de dados 2 para metodologia de treinamento 3 . . . . .	131
Tabela 66 – Sugestões de distribuição de protótipos para as taxas de acerto do banco de dados 2 . . . . .	131
Tabela 67 – Sugestões e moda de distribuição de protótipos para a maior taxa de acerto do banco de dados 2 para metodologia de treinamento 1 . . . . .	132
Tabela 68 – Sugestões e moda de distribuição de protótipos para a maior taxa de acerto do banco de dados 2 para metodologia de treinamento 2 . . . . .	132

Tabela 69 – Valores encontrados para os elementos do vetor de ponderação para o banco de dados 2 e metodologia de treinamento 3 . . . . .	133
Tabela 70 – Tabela resumo de resultados encontrados para o BD1 . . . . .	133
Tabela 71 – Tabela resumo de resultados encontrados para o BD2 . . . . .	133
Tabela 72 – Tabela resumo dos valores ótimos de $K$ para metodologia 1 . . . . .	134
Tabela 73 – Estatísticas de taxas de acerto encontradas em classificação ternária do banco de dados 1 para metodologia de treinamento 1 . . . . .	144
Tabela 74 – Estatísticas de taxas de acerto encontradas em classificação ternária do banco de dados 1 para metodologia de treinamento 2 . . . . .	144
Tabela 75 – Estatísticas de taxas de acerto encontradas em classificação ternária do banco de dados 1 para metodologia de treinamento 3 . . . . .	144
Tabela 76 – Estatísticas de taxas de acerto encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 1 . . . . .	144
Tabela 77 – Estatísticas de taxas de acerto encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 2 . . . . .	145
Tabela 78 – Estatísticas de taxas de acerto encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 3 . . . . .	145
Tabela 79 – Estatísticas de taxas de sensibilidade encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 1 . . . . .	145
Tabela 80 – Estatísticas de taxas de sensibilidade encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 2 . . . . .	145
Tabela 81 – Estatísticas de taxas de sensibilidade encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 3 . . . . .	145
Tabela 82 – Estatísticas de taxas de especificidade encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 1 . . . . .	146
Tabela 83 – Estatísticas de taxas de especificidade encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 2 . . . . .	146
Tabela 84 – Estatísticas de taxas de especificidade encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 3 . . . . .	146
Tabela 85 – Estatísticas de taxas de acerto encontradas em classificação ternária do banco de dados 2 para metodologia de treinamento 1 . . . . .	146

Tabela 86 – Estatísticas de taxas de acerto encontradas em classificação ternária do banco de dados 2 para metodologia de treinamento 2 . . . . .	146
Tabela 87 – Estatísticas de taxas de acerto encontradas em classificação ternária do banco de dados 2 para metodologia de treinamento 3 . . . . .	147
Tabela 88 – Estatísticas de taxas de acerto encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 1 . . . . .	147
Tabela 89 – Estatísticas de taxas de acerto encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 2 . . . . .	147
Tabela 90 – Estatísticas de taxas de acerto encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 3 . . . . .	147
Tabela 91 – Estatísticas de taxas de sensibilidade encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 1 . . . . .	147
Tabela 92 – Estatísticas de taxas de sensibilidade encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 2 . . . . .	148
Tabela 93 – Estatísticas de taxas de sensibilidade encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 3 . . . . .	148
Tabela 94 – Estatísticas de taxas de especificidade encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 1 . . . . .	148
Tabela 95 – Estatísticas de taxas de especificidade encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 2 . . . . .	148
Tabela 96 – Estatísticas de taxas de especificidade encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 3 . . . . .	148

## LISTA DE ABREVIATURAS E SIGLAS

AIC	Akaike's Information Criterion
AKGLVQ	Accelerated Kernelized Generalized LVQ
AM	Aprendizado de Máquina
BD1	Banco de Dados 1
BD2	Banco de Dados 2
BIC	Bayesian Information Criterion
CH	Calinski-Harabasz
CSDLVQ	Cauchy-Schwarz Divergence LVQ
DB	Davies-Bouldin
ELM	Extreme Learning Machine
FDP	Função de Densidade de Probabilidade
FFT	Fast Fourier Transform
FN	Falso Negativo
FP	Falso Positivo
FPE	Final Prediction Error
GLVQ	Generalized LVQ
GMLVQ	Generalized Matrix LVQ
GRLVQ	Generalized Relevance LVQ
H2M-LVQ	Harmonic to Minimum LVQ
IT-LVQ	Information Theoretic LVQ
KGLVQ	Kernel Generalized LVQ
KRSLVQ	Kernel Robust Soft LVQ
LBG	Linde-Buzo-Gray
LGMLVQ	Localized Generalized Matrix LVQ
LGRLVQ	Local Generalized Relevance LVQ
LMRSLVQ	Local Matrix Robust Soft LVQ

LSSVM	Least Squares Support Vector Machine
LVQ1	Learning Vector Quantization 1
LVQ2	Learning Vector Quantization 2
LVQ2.1	Learning Vector Quantization 2.1
LVQ3	Learning Vector Quantization 3
MCSA	Motor Current Signature Analysis
MDL	Minimum Description Length
MFCC	Mel Frequency Cepstral Coefficients
MIT	Motores de Indução Trifásicos
MIGE	Motor de Indução Gaiola de Esquilo
MLM	Minimal Learning Machine
MLP	Multilayer Perceptron
MRSLVQ	Matrix Robust Soft LVQ
MSQE	Mean Squared Quantization Error
NG	Neural Gas
OLVQ1	Optimized-Learning-Rate LVQ1
QV	Quantização Vetorial
RC	Redes Competitivas
RGLVQ	Relational Generalized LVQ
RLVQ	Relevance LVQ
RNA	Redes Neurais Artificiais
RRSLVQ	Relational Robust Soft LVQ
RSLVQ	Robust Soft LVQ
SEQQ	Soma dos Erros Quadráticos de Quantização
SOM	Self-Organizing Maps
SGNG	Supervised Growing Neural Gas
SNG	Supervised Neural Gas
SOM	Self-Organizing Maps

SVM	Support Vector Machine
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
WTA	Winner Takes All

## LISTA DE SÍMBOLOS

$w_r$	Velocidade angular rotórica do motor (em RPM)
$f_s$	Frequência de alimentação do motor
$P_a$	Pares de de pólos do motor
$s$	Escorregamento do motor
$e_i$	Erro estimado para a $i$ -ésima amostra
$y_i$	Valor de saída conhecido do $i$ -ésimo dado
$f(\mathbf{x}_i)$	Valor estimado por um algoritmo para a $i$ -ésima amostra
$\mathbf{x}_i$	Vetor de características da $i$ -ésima amostra
$f_h$	Componentes harmônicas de frequência
$k_1$	Ordem dos harmônicos temporais
$k_2$	Ordem dos harmônicos espaciais
$\mathbf{x}_n$	Vetor de amostras
$N$	Número total de amostras
$p$	Dimensão do vetor de amostras
$\mathbf{W}$	Matriz formada por vetores protótipos
$K$	Número de grupos ( <i>clusters</i> )
$d(\mathbf{w}_j, \mathbf{x}_i)$	Distância entre a $i$ -ésima amostra e o $j$ -ésimo protótipo
$\eta$	Constante de taxa de aprendizagem
$\mathbf{w}^*(\mathbf{x}_i)$	Vetor protótipo mais próximo ao exemplo de entrada $\mathbf{x}_i$
$\mathbf{w}_c$	Protótipo vencedor
$\alpha$	Passo de aprendizagem
$\{\mathbf{w}_i\}_{i=1}^K$	Conjunto de protótipos ou vetores-código
$\mathbf{w}_i$	Vetor-código pertencente à $\{\mathbf{w}_i\}_{i=1}^K$
$V_i$	Conjunto de dados em $K$ regiões de Voronoi, $i = 1, \dots, K$
$\ \cdot\ $	Denota a norma euclidiana
$ u $	Denota o valor absoluto de $u$

$\varepsilon$	Limiar de distorção maior que zero
$N_i$	Número de exemplos selecionados para o conjunto do $i$ -ésimo protótipo
$t$	Iteração atual
$D(n)$	Valor do erro de quantização na $n$ -ésima rodada de ajustes dos protótipos
$CH(K)$	Função do índice CH
$B_K$	Soma dos quadrados entre <i>clusters</i>
$W_K$	Soma dos quadrados intracluster
$D(K)$	Função do índice Dunn
$\delta(S_i, S_j)$	Função de dissimilaridade entre $S_i$ e $S_j$
$\Delta(S_i)$	Distância intracluster de $S_i$
$DB(K)$	Função do índice DB
$S(K)$	Função do índice silhuetas
$S_i$	Representação para um <i>cluster</i>
$a(x_i)$	Dissimilaridade média do padrão $x_i$ em relação a todos os outros padrões do agrupamento $S_i$
$d(x_i, S_j)$	Dissimilaridade média do padrão $x_i$ em relação aos padrões do agrupamento $S_j$
$b(x_i)$	A menor dissimilaridade média de $x_i$ em relação a todos os demais agrupamentos
$CI$	Modelo geral de um critério de informação
$P$	Ordem do modelo
$RSS(P)$	Soma dos resíduos quadráticos para um modelo com $P$ parâmetros
$FPE(K)$	Função do critério FPE
$AIC(K)$	Função do critério AIC
$BIC(K)$	Função do critério BIC
$MDL(K)$	Função do critério MDL
$y_i$	Classe da $i$ -ésima amostra
$C$	Número total de classes

$c(\mathbf{w}_i)$	Classe do protótipo $\mathbf{w}_i$
$R^i$	Campo receptivo do protótipo $\mathbf{w}_i$
$s(t)$	Função sinal
$\alpha_c$	Passo de aprendizagem do protótipo $\mathbf{w}_c$
$\mathbf{w}^+$	Vetor protótipo mais próximo ao padrão de entrada $\mathbf{x}$ que o possui mesmo rótulo deste
$\mathbf{w}^-$	Vetor protótipo mais próximo ao padrão de entrada $\mathbf{x}$ que o possui diferente rótulo deste
$d^+$	Distância euclidiana entre $\mathbf{x}$ e $\mathbf{w}^+$
$d^-$	Distância euclidiana entre $\mathbf{x}$ e $\mathbf{w}^-$
$j$	Janela de restrição
$\mathbf{w}_a$	Vetor protótipo mais próximo ao padrão de entrada $\mathbf{x}$
$\mathbf{w}_b$	Vetor protótipo segundo mais próximo ao padrão de entrada $\mathbf{x}$
$d_a$	Distância euclidiana entre $\mathbf{x}$ e $\mathbf{w}_a$
$d_b$	Distância euclidiana entre $\mathbf{x}$ e $\mathbf{w}_b$
$\varepsilon$	Fator de ganho utilizado na atualização de vetores protótipos
$d_q^+$	Distância euclidiana quadrática entre $\mathbf{x}$ e $\mathbf{w}^+$
$d_q^-$	Distância euclidiana quadrática entre $\mathbf{x}$ e $\mathbf{w}^-$
$\mu(\mathbf{x})$	Índice de distância relativa
$sgd(a)$	Função logística
$\lambda$	Vetor ou matriz de pesos de relevância
$\ \cdot\ _\lambda^2$	Distância euclidiana ponderada quadrática
$d_{\lambda,q}^+$	Distância euclidiana ponderada quadrática entre $\mathbf{x}$ e $\mathbf{w}^+$
$d_{\lambda,q}^-$	Distância euclidiana ponderada quadrática entre $\mathbf{x}$ e $\mathbf{w}^-$
$\varepsilon_\lambda$	Fator de ganho utilizado na atualização de vetores de relevância

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>27</b>
<b>1.1</b>	<b>Motores de Indução</b>	<b>27</b>
<b>1.2</b>	<b>Falhas por Curto-Circuito em MIT</b>	<b>29</b>
<b>1.3</b>	<b>Histórico do Grupo de Pesquisas em Detecção de Falhas de Curto-Circuito em MIT</b>	<b>31</b>
<b>1.4</b>	<b>Objetivos</b>	<b>32</b>
<i>1.4.1</i>	<i>Objetivo Geral</i>	<i>32</i>
<i>1.4.2</i>	<i>Objetivos Específicos</i>	<i>33</i>
<b>1.5</b>	<b>Produção Científica</b>	<b>33</b>
<b>1.6</b>	<b>Organização Geral da Dissertação</b>	<b>34</b>
<b>2</b>	<b>DESCRIÇÃO DO BANCO DE DADOS UTILIZADO</b>	<b>35</b>
<b>2.1</b>	<b>Análise da Assinatura da Corrente</b>	<b>35</b>
<b>2.2</b>	<b>Descrição de Aquisição do Banco de Dados</b>	<b>36</b>
<i>2.2.1</i>	<i>Motor e Freio de Foucault</i>	<i>36</i>
<i>2.2.2</i>	<i>Sistema Auxiliar de Curto-Circuito</i>	<i>37</i>
<i>2.2.3</i>	<i>Conversor de Frequência</i>	<i>38</i>
<i>2.2.4</i>	<i>Sistema de Aquisição de Dados</i>	<i>38</i>
<i>2.2.5</i>	<i>Extração e Seleção de Atributos</i>	<i>38</i>
<b>2.3</b>	<b>Considerações Sobre o Banco de Dados</b>	<b>39</b>
<b>2.4</b>	<b>Análise Não Supervisionada Prévia do Banco de Dados</b>	<b>41</b>
<b>2.5</b>	<b>Conclusão</b>	<b>42</b>
<b>3</b>	<b>APRENDIZADO NÃO SUPERVISIONADO BASEADO EM PROTÓTIPOS</b>	<b>43</b>
<b>3.1</b>	<b>Definições Iniciais</b>	<b>43</b>
<b>3.2</b>	<b>Vetores Protótipos e Aprendizado Não Supervisionado</b>	<b>43</b>
<i>3.2.1</i>	<i>Análise de Agrupamentos</i>	<i>47</i>
<i>3.2.1.1</i>	<i>Algoritmo K-médias</i>	<i>48</i>
<b>3.3</b>	<b>Validação de Agrupamentos</b>	<b>49</b>
<i>3.3.1</i>	<i>Índices Baseados em Critérios Relativos</i>	<i>50</i>
<i>3.3.1.1</i>	<i>Índice Calinski-Harabasz (CH)</i>	<i>50</i>

3.3.1.2	<i>Família de Índices Dunn</i> . . . . .	51
3.3.1.3	<i>Índice Davies-Bouldin (DB)</i> . . . . .	51
3.3.1.4	<i>Silhuetas</i> . . . . .	52
3.3.2	<b>Índices Baseados em Critérios de Informação</b> . . . . .	53
3.3.2.1	<i>Critério do Erro Final de Predição</i> . . . . .	54
3.3.2.2	<i>Critério de Informação de Akaike</i> . . . . .	55
3.3.2.3	<i>Critério de Informação Bayesiano</i> . . . . .	55
3.3.2.4	<i>Critério do Comprimento Mínimo de Descrição</i> . . . . .	55
3.4	<b>Conclusão</b> . . . . .	56
4	<b>CLASSIFICADORES BASEADOS EM LEARNING VECTOR QUANTIZATION</b> . . . . .	57
4.1	<b>Definições Preliminares</b> . . . . .	57
4.2	<b>Família de Classificadores Baseados em LVQ</b> . . . . .	58
4.2.1	<b>Algoritmos Heurísticos</b> . . . . .	60
4.2.1.1	<i>Learning Vector Quantization 1 (LVQ1)</i> . . . . .	61
4.2.1.2	<i>Optimized Learning Vector Quantization 1 (OLVQ1)</i> . . . . .	62
4.2.1.3	<i>Learning Vector Quantization 2 (LVQ2) e Learning Vector Quantization 2.1 (LVQ2.1)</i> . . . . .	62
4.2.1.4	<i>Learning Vector Quantization 3 (LVQ3)</i> . . . . .	64
4.2.2	<b>Algoritmos de Maximização da Margem de Separação</b> . . . . .	65
4.2.2.1	<i>Generalized Learning Vector Quantization (GLVQ)</i> . . . . .	66
4.2.2.2	<i>Relevance Learning Vector Quantization (RLVQ)</i> . . . . .	70
4.2.2.3	<i>Generalized Relevance LVQ (GRLVQ) e Local Generalized Relevance LVQ (LGRLVQ)</i> . . . . .	71
4.2.3	<b>Algoritmos de Maximização da Verossimilhança</b> . . . . .	73
4.3	<b>Conclusão</b> . . . . .	76
5	<b>METODOLOGIA DOS EXPERIMENTOS</b> . . . . .	77
5.1	<b>Banco de Dados</b> . . . . .	77
5.2	<b>Heurística Proposta para Iniciação dos Vetores Protótipos</b> . . . . .	78
5.3	<b>Metodologia de Comparação Entre Classificadores LVQ</b> . . . . .	79
5.3.1	<i>Banco de Dados Escolhido</i> . . . . .	80
5.3.2	<i>Normalização dos Dados</i> . . . . .	81

5.3.3	<i>Separação dos Dados</i>	81
5.3.4	<i>Aprendizado Não Supervisionado do Conjunto de Treinamento</i>	81
5.3.5	<i>Treinamento dos Classificadores</i>	83
5.3.6	<i>Teste dos Classificadores</i>	85
5.4	<b>Conclusão</b>	88
6	<b>RESULTADOS DA ANÁLISE DE AGRUPAMENTOS</b>	89
6.1	<b>Banco de dados 1 (BD1)</b>	89
6.1.1	<i>Classe normal</i>	91
6.1.2	<i>Classe falha de alta impedância</i>	91
6.1.3	<i>Classe falha de baixa impedância</i>	91
6.2	<b>Banco de dados 2 (BD2)</b>	95
6.2.1	<i>Classe normal</i>	96
6.2.2	<i>Classe falha de alta impedância</i>	97
6.2.3	<i>Classe falha de baixa impedância</i>	97
6.3	<b>Conclusão</b>	101
7	<b>RESULTADOS DA COMPARAÇÃO ENTRE CLASSIFICADORES LVQ</b>	102
7.1	<b>Banco de dados 1 (BD1)</b>	102
7.1.1	<i>Taxas de acertos</i>	102
7.1.2	<i>Matrizes de confusão e estatísticas associadas</i>	106
7.1.3	<i>Resultados secundários</i>	116
7.2	<b>Banco de dados 2</b>	118
7.2.1	<i>Taxas de acertos</i>	118
7.2.2	<i>Matrizes de confusão e estatísticas associadas</i>	121
7.2.3	<i>Resultados secundários</i>	131
7.3	<b>Conclusão</b>	133
8	<b>CONCLUSÕES E PERSPECTIVAS</b>	135
8.1	<b>Objetivo Geral</b>	135
8.2	<b>Objetivos Específicos</b>	135
8.3	<b>Trabalhos Futuros</b>	136
	<b>REFERÊNCIAS</b>	138
	<b>APÊNDICES</b>	144
	<b>APÊNDICE A – Tabelas de Resultados</b>	144

**ANEXOS** ..... 144

## 1 INTRODUÇÃO

Neste capítulo é apresentada uma breve introdução sobre o motor de indução trifásico e a tarefa de identificação de falhas de curto-circuito deste tipo de motor nas Seções 1.1 e 1.2, respectivamente.

Além disso, este capítulo contém, na Seção 1.3, um histórico das ferramentas de aprendizado de máquina já utilizadas no estudo de solução do problema proposto que motivaram os objetivos e as contribuições deste trabalho.

### 1.1 Motores de Indução

Os motores elétricos, entre diversos tipos existentes no mercado, são utilizados no acionamento de cargas mecânicas por serem capazes de promover a transformação de energia elétrica em energia mecânica com pequenas perdas. Estes equipamentos são categorizados de acordo com o tipo de acionamento que pode ser através de corrente contínua e através de corrente alternada onde, entre estes modelos, os motores de indução (objetos analisados neste trabalho) pertencem ao segundo grupo.

A estrutura básica do motor de indução pode ser visualizada na Figura 1. Segundo Guedes (1994), o motor elétrico de indução, também conhecido como motor assíncrono, destaca-se por pouca necessidade de manutenção, baixo custo e possibilidade de emprego em praticamente qualquer aplicação. Entre os diferentes tipos de motores de indução, prefere-se o modelo trifásico ao modelo monofásico pelo fato deste ser capaz de alcançar apenas 60 a 70% da potência do primeiro (FRANCHI, 2018).

Os motores de indução trifásicos (MIT), devido a sua robustez, ciência e simplicidade, são a principal força motora da indústria (SESHADRINATH *et al.*, 2014). Dentre as diversas aplicações destas máquinas, pode-se citar o acionamento de ventiladores, bombas, bobinadeiras, esteiras transportadoras e elevadores. Segundo Franchi (2018), para a maioria das aplicações, a configuração mais econômica é conseguida através da utilização de um tipo específico de motor de corrente alternada assíncrono: o motor de indução gaiola de esquilo (MIGE), conhecido também como rotor em curto-circuito ou rotor de gaiola.

Construído por Nikola Tesla, o MIT baseia-se nas leis fundamentais do eletromagnetismo, entre elas a Lei de indução de Faraday e a Lei de Lenz. Primeiramente, é gerado um campo magnético girante, pela aplicação de tensões trifásicas nos enrolamentos do estator. Este

campo, ao atravessar o rotor, provoca uma variação de fluxo nos condutores deste. Como os condutores do rotor gaiola de esquilo estão em um circuito fechado, estes são percorridos por uma corrente induzida geradora de um campo magnético de mesmo sentido do campo girante produzido pelo estator. A interação entre os campos produz um conjugado do motor no mesmo sentido da velocidade do campo girante produzido pelo estator.

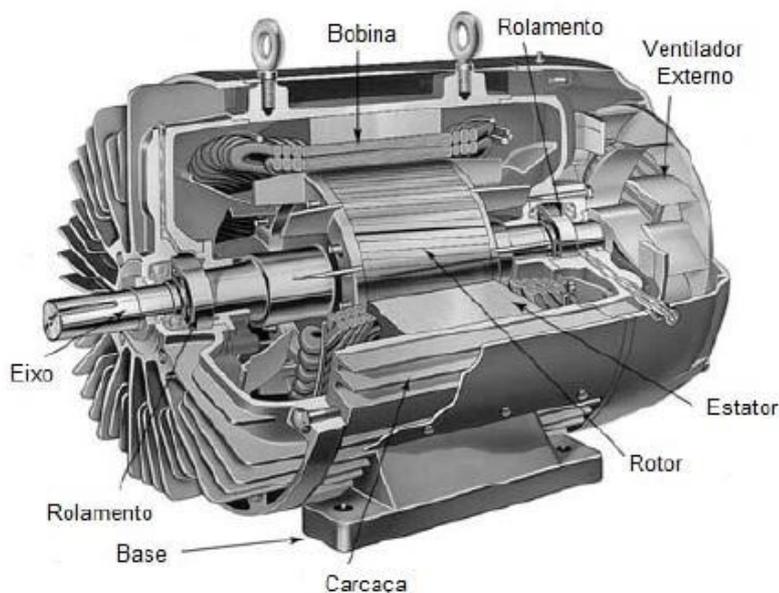
A equação que relaciona a velocidade do motor com a frequência da tensão de alimentação é dada por

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

em que  $w_r$  é a velocidade angular rotórica (em rpm),  $f_s$  é a frequência da rede elétrica de alimentação, em Hertz (Hz),  $P_a$  é o número de pares de pólos e  $s$  é o escorregamento deste.

Apesar de todos os estudos sobre esse tipo de máquina e a sua robustez comprovada, o motor de indução é, infelizmente, sujeito à diferentes tipos de falhas que são causadas devido ao envelhecimento da máquina, condições adversas do ambiente, aplicações inadequadas, ou falta de um programa de manutenção (GHATE; DUDUL, 2010; NANDI *et al.*, 2005). Logo, estudos sobre as diversas fontes de falhas que ocorrem nestes equipamentos são indispensáveis. Por fim, a próxima seção enumera os diferentes tipos de falhas que os MIT estão sujeitos.

Figura 1 – Ilustração pictórica das principais componentes de um motor de indução



Fonte: adaptada de Bonnett (2010).

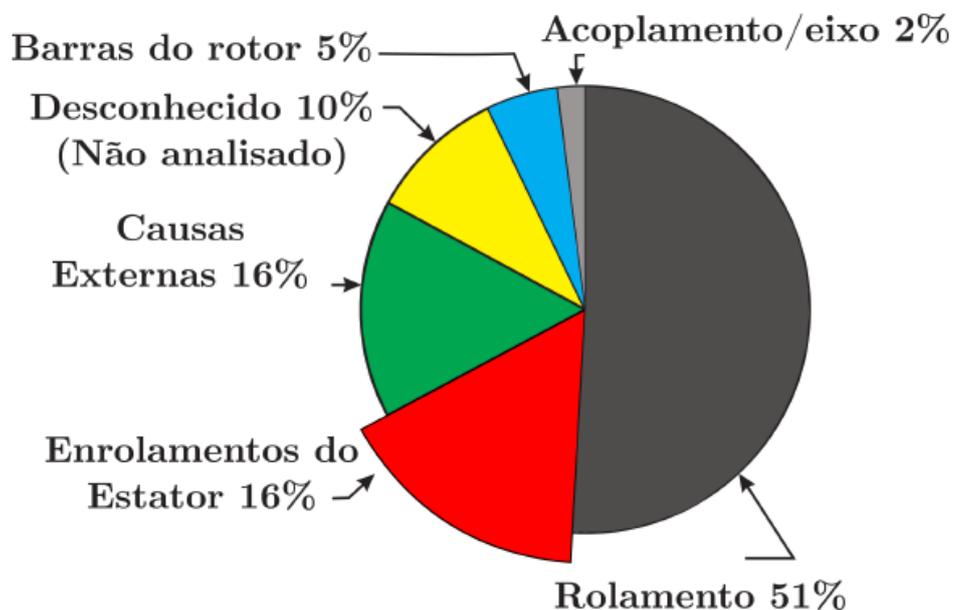
## 1.2 Falhas por Curto-Circuito em MIT

O trabalho de Bonnett (2010) reporta um registro histórico de 40 anos de falhas em máquinas de indução nas indústrias química e petroquímica e relata as falhas mais expressivas ocorridas nessa classe de equipamentos. Neste estudo, as falhas em MIT foram categorizadas entre acoplamentos do eixo, barras do rotor, rolamentos, enrolamentos do estator e, por último, causas externas. As falhas que possuem mais registros são as de rolamentos, representando 51% do total de ocorrências. Já as falhas no enrolamento do estator representaram 16% do total. A proporção encontrada dos diferentes tipos de falhas são apresentadas na Figura 2.

Em contraposição ao levantamento de Bonnett (2010), outros autores sugerem diferentes distribuições entre os tipos de falhas e apontam que as falhas de enrolamentos são mais relevantes. Por exemplo, Kliman *et al.* (1996) sugerem que este tipo de falha representam entre 30 a 40% do total.

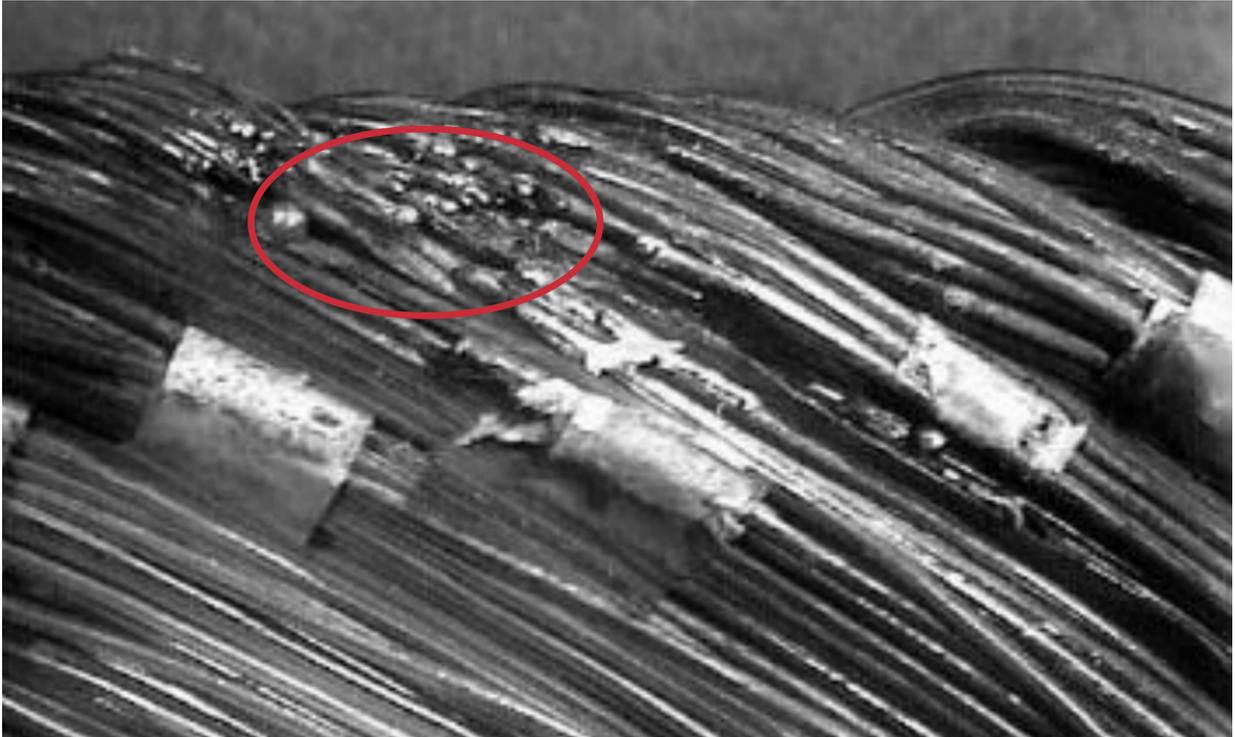
Considerando a relevância de falhas no enrolamento do estator, este trabalho e os listados na Seção 1.3 se limitaram à análise de falhas de motores de indução trifásicos por curto-circuito entre espiras do estator. Esse tipo de falha ocorre principalmente devido a *i*) alta temperatura do bobinamento ou núcleo do estator; *ii*) folga na laminação do núcleo magnético estatórico e perda da proteção das conexões de bobinas; *iii*) contaminação provocada por óleo, umidade ou sujeira; *iv*) descargas elétricas; e *v*) vazamentos no sistema de arrefecimento (NANDI *et al.*, 2002).

Figura 2 – Proporção entre tipos de falhas em motores de indução



Fonte: Bonnett (2010 apud NASCIMENTO, 2018).

Figura 3 – Exemplo de falha por curto-circuito entre espiras



Fonte: adaptada de Bonnett (2010).

Segundo Natarajan (1989), em decorrência do estresse, o processo de falha inicia-se primeiramente com uma falha de alta impedância entre espiras da mesma fase, entre fases ou para a terra. Depois desse estresse inicial, a corrente em estado de falha pode causar aquecimento localizado, provocando um rápido alastramento da falha de bobinamento (TALLAM *et al.*, 2007). Consequentemente, considera-se que a falha de alta impedância é incipiente a uma possível falha de baixa impedância.

É ilustrada na Figura 3 um exemplo de motor de indução que sofreu uma falha de curto-circuito entre espiras em uma mesma fase de seu estator. Esta falha ocorreu devido à perda de isolamento causada por temperatura excessiva dos enrolamentos.

A possibilidade da reutilização de motores de grande porte, após o rebobinamento do estator, caso a falha por curto-circuito seja detida em seu primeiro estágio (falha de alta impedância) motiva estudos de aprimoramento de técnicas de detecção de falhas por curto-circuito. Além disso, estudos sobre tais técnicas são motivados pela existência de cenários industriais em que a falha de curto-circuito é um potencial agente ocasionador de explosões. Consequentemente, é de grande interesse uma solução que possa detectar falhas na fase inicial de um curto-circuito visto que, entre outros fatores, elevaria o tempo de uso destas máquinas.

### 1.3 Histórico do Grupo de Pesquisas em Detecção de Falhas de Curto-Circuito em MIT

Baseado na importância da detecção de falhas em MIT, a presente dissertação justifica-se pela necessidade de garantir maior confiabilidade em aplicações que utilizem o MIGE. Além disso, oferta continuidade aos trabalhos iniciados nas instituições Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) e Universidade Federal do Ceará (UFC) sobre esta temática a partir de diferentes técnicas de aprendizado de máquina (AM).

As pesquisas em aprendizado de máquina têm por objetivo principal o desenvolvimento de ferramentas computacionais eficientes capazes de fornecer soluções satisfatórias, senão as melhores, a problemas complexos de classificação de padrões, tais como os de natureza não-linear, incerta e de alta dimensionalidade. Entre as propriedades almejadas, pode-se exemplificar a capacidade de construção de fronteiras de decisão arbitrárias onde se tenha o mínimo, ou até mesmo nenhuma, superposição acerca dos dados disponíveis e, também, a capacidade de generalização do conhecimento para novos dados (WEBB, 2002 apud ROCHA NETO, 2011).

As técnicas de AM são divididas nos grupos de aprendizagem supervisionada e não supervisionada. Os métodos de aprendizado supervisionado processam sinais coletados com o objetivo de minimizar erros das informações de saída, dados pela diferença entre valores estimados e conhecidos, conforme a relação abaixo:

$$e_i = \frac{1}{2} \left( y_i - f(\mathbf{x}_i) \right)^2, \quad (1.2)$$

em que  $e_i$  é o erro quadrático estimado para um dado  $i$  pela diferença entre um valor conhecido  $y_i$  e o valor estimado pela função  $f(\mathbf{x}_i)$  definida por cada método de AM, sendo  $\mathbf{x}_i$  a entrada para certo dado  $i$ .

Já os métodos de aprendizado não supervisionado, por sua vez, processam sinais de entrada com o objetivo de encontrar correlações e informações topológicas que permitam o agrupamento dos dados sem utilizar as informações de resposta da rede  $f(\mathbf{x}_i)$ . As informações de saída correspondentes aos sinais podem ser disponíveis ou não.

Os métodos de aprendizado supervisionado são os mais utilizados para sistemas de diagnósticos e estão associados a diversas abordagens de pré-processamento e, também, a diferentes extratores de parâmetros mais relevantes com o objetivo de reduzir a quantidade de informação a ser tratada pelos classificadores.

Dentre as diferentes abordagens de AM existentes, as Redes Neurais Artificiais (RNA) e *Support Vector Machine* (SVM) são as técnicas mais comumente utilizadas pela literatura. Conseqüentemente, estes métodos foram primeiramente utilizados no desenvolvimento de pesquisas em detecção de falhas de curto-circuito pelas instituições IFCE e UFC. Por exemplo, as redes *Multilayer Perceptron* (MLP), *Extreme Learning Machine* (ELM) e, também, a rede não supervisionada *Self-Organizing Maps* (SOM) foram utilizadas no trabalho de Oliveira *et al.* (2013b). Já as redes MLP, ELM, *Minimal Learning Machine* (MLM), SVM e *Least Squares Support Vector Machine* (LSSVM) foram utilizadas no trabalho de Coelho *et al.* (2014). Posteriormente, a rede SOM em versões baseadas em métodos de *kernel* fora utilizada no trabalho de Coelho *et al.* (2017). Destaca-se que os trabalhos mencionados acima utilizaram o mesmo conjunto de dados e este também será utilizado nesta dissertação.

Por fim, verificou-se que os métodos de AM baseados em ANFIS (*Adaptive Neuro-Fuzzy Inference System*) e LVQ (*Learning Vector Quantization*) ainda não foram explorados por estas instituições. Dado que os algoritmos da família LVQ são soluções populares e atrativas em problemas de classificação com múltiplas classes (WITOELAR *et al.*, 2007), isto motivou a utilização de métodos baseados em LVQ no desenvolvimento da presente pesquisa. Além disso, os algoritmos LVQ são facilmente projetados para problemas com múltiplas classes (NOVA; ESTÉVEZ, 2014) e suas complexidades podem ser modificadas durante a etapa de treinamento (BIEHL *et al.*, 2007).

## 1.4 Objetivos

Uma vez exposto o problema de interesse na seção anterior, os objetivos descritos a seguir foram determinados para esta dissertação.

### 1.4.1 Objetivo Geral

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

### 1.4.2 *Objetivos Específicos*

Os objetivos específicos desta dissertação estão listados a seguir:

- Implementar o algoritmo de *clustering* K-médias em diferentes configurações de protótipos para as diferentes classes do banco de dados utilizado;
- Implementar diferentes algoritmos de validação de agrupamentos baseados em critérios relativos (CH, DB, Dunn e Silhueta) e baseados em critérios de informação (AIC, BIC, FPE e MDL);
- Através das técnicas de validação de agrupamentos citadas acima, encontrar a melhor estrutura de agrupamentos em cada classe do banco de dados de um MIT.
- Implementar os algoritmos LVQ baseados em métodos heurísticos desenvolvidos por Teuvo Kohonen e os baseados em métodos de maximização da margem de separação no espaço dos dados (GLVQ linear, GLVQ não linear, RLVQ, GRLVQ e LGRLVQ);
- Avaliar os efeitos de contribuição ao iniciar os vetores protótipos de treinamento do LVQ de acordo com as técnicas de agrupamentos para determinar a quantidade ótima de protótipos e, também, suas respectivas posições de convergência;
- Estudar a eficácia destes classificadores supervisionados baseados em protótipos na detecção de falhas de motores de indução trifásicos e, comparar, através de testes estatísticos, os classificadores LVQ explorados nesta dissertação com outras técnicas de aprendizado de máquinas que já utilizaram este conjunto de dados previamente.

### 1.5 **Produção Científica**

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

SOUSA, D. P.; BARRETO, G. A.; MEDEIROS, C. M. Efficient Selection of Data Samples for Fault Classification by the Clustering of the SOM. In: **Congresso Brasileiro de Inteligência Computacional**. Niterói, RJ. 2017.

Além deste artigo, o autor desta dissertação também submeteu o seguinte artigo:

SOUSA, D. P.; BARRETO, G. A.; CAVALCANTE, C. C.; & MEDEIROS, C. M. LVQ-based Classifiers for Condition Monitoring of a Three-Phase Induction Motor: A Comprehensive Performance Comparison. In: **IEEE. 2019 13th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM)**. Barcelona. 2019.

## 1.6 Organização Geral da Dissertação

O restante desta dissertação está organizada na seguinte formatação:

O Capítulo 2, mostra um estudo sobre a falha por curto-circuito entre espiras do estator de um MIT e, também, sobre a identificação de falhas neste tipo de motor. Por fim, descreve-se o conjunto de dados utilizado neste trabalho.

No Capítulo 3, é apresentado o problema de aprendizado não supervisionado baseado em protótipos. É ainda realizada uma breve descrição do algoritmo e das métricas de validação de agrupamentos utilizados neste trabalho.

No Capítulo 4, apresenta-se o problema de aprendizado supervisionado baseado em protótipos. É ainda realizada uma breve descrição dos algoritmos e suas respectivas métricas de dissimilaridade utilizados neste trabalho.

No Capítulo 5, os detalhes da metodologia proposta, de comparação entre os diferentes algoritmos, e das técnicas computacionais utilizadas são discriminadas.

No Capítulo 6, os resultados da análise de agrupamentos dos conjuntos de treinamento são discutidos e analisados.

No Capítulo 7, os resultados da comparação entre classificadores LVQ utilizados são discutidos e analisados.

No Capítulo 8, são feitas as considerações finais e as perspectivas para trabalhos futuros.

Por fim, no Apêndice A, são mostradas as tabelas que contêm as estatísticas encontradas sobre as taxas de acertos.

## 2 DESCRIÇÃO DO BANCO DE DADOS UTILIZADO

Neste capítulo, apresenta-se a fundamentação dos procedimentos realizados para a formação do banco de dados analisado neste trabalho. Além disso, descreve-se uma análise não supervisionada do banco de dados.

Este capítulo está organizado da seguinte forma: na Seção 2.1 é apresentada a descrição do extrator de atributos utilizado no desenvolvimento deste trabalho; a Seção 2.2 apresenta a descrição de aquisição do banco de dados; na Seção 2.3, são realizadas considerações sobre o banco de dados formado e, por fim, a Seção 2.4 descreve uma análise não supervisionada do banco de dados com o objetivo de aferição da complexidade deste.

### 2.1 Análise da Assinatura da Corrente

Como descrito previamente na Seção 1.2, o prognóstico precoce de falhas por curto-circuito no estator do motor de indução trifásico possibilita que a equipe de manutenção possa atuar em prevenção de acréscimos aos custos de produção e em proteção do conversor de frequência (THOMSON; FENGER, 2001).

Para detectar este tipo de falha, primeiramente, realiza-se a etapa de extração de atributos a fim da obtenção das características mais relevantes dos sinais coletados. Dentre os diversos métodos, a análise da assinatura de corrente do motor (do inglês, *motor current signature analysis* - MCSA) tem sido vastamente utilizada, dado que não é um método invasivo, pois não precisa ser adaptado para áreas com risco de explosão e pode ser aplicado, sem restrição de potência, a qualquer máquina elétrica (THORSEN; DALVA, 1997).

Os estudos recentes de Mendonça *et al.* (2017), Aimer *et al.* (2017) e Sakhara *et al.* (2017) reportam que o método de extração de atributos MCSA é o mais adequado para a detecção de falhas por curto-circuito em motores de indução. Segundo Glowacz (2019), as técnicas de diagnósticos baseadas em MCSA possuem altas taxas de eficiência de reconhecimento justificadas pela facilidade de processamento dos sinais elétricos e que estes apresentam menores níveis de sobreposição que sinais acústicos.

Este método consiste em utilizar algumas componentes do espectro de frequência da corrente do motor como características para representar condições de operação do motor. Com o objetivo de seleção destas frequências, Penman *et al.* (1994) elaboraram o seguinte modelo que

relaciona as harmônicas geradas pelas falhas por curto-circuito:

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

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

De acordo com Coelho (2015), a utilização de um conversor de frequências na alimentação do motor, onde o valor da frequência fundamental é conhecido, possibilita a medição ou até mesmo a estimação do escorregamento e, conseqüentemente, pode-se extrair as demais componentes.

## 2.2 Descrição de Aquisição do Banco de Dados

Nesta seção, é apresentada a metodologia utilizada na modelagem do banco de dados que será utilizado neste trabalho considerando-se que o sucesso de uma aplicação de reconhecimento de padrões é dependente da construção de um modelo confiável para os dados. Para isto, descreve-se a sequência de passos realizados na obtenção da quantidade total dos dados, a distribuição dos exemplos entre as classes, as características de cada amostra e as técnicas de seleção e extração de atributos utilizadas.

### 2.2.1 Motor e Freio de Foucault

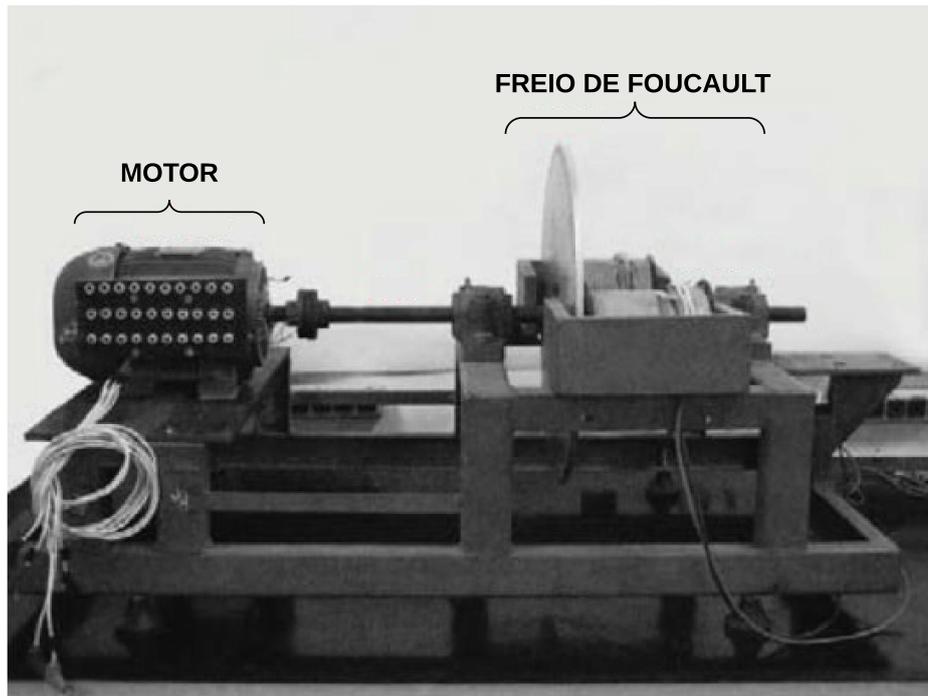
Os principais componentes da bancada de testes são ilustrados na Figura 4. Na configuração apresentada, tem-se um motor de indução trifásico assíncrono do tipo gaiola de esquilo, que é alimentado por um conversor de frequência, conectado a um freio de Foucault. As principais características do MIT utilizado na obtenção do banco de dados estão listadas na Tabela 1.

Normalmente, esse motor possui apenas dois terminais disponíveis por fase (para alimentação). 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. Neste caso,

<sup>1</sup> Decorrentes da saturação magnética de transformadores.

<sup>2</sup> Decorrentes da construção física do MIT e do posicionamento de seus enrolamentos.

Figura 4 – Bancada de testes utilizada na aquisição da base de dados



Fonte: adaptada de Coelho (2015).

Tabela 1 – Características do MIT utilizado

<b>Grandeza</b>	<b>Valor</b>	<b>Unidade</b>
Potência Nominal	0,75	kW
Tensão Nominal	220/380	V
Corrente Nominal (In)	3,02/1,75	A
Eficiência	79,5	%
Velocidade Nominal	1.720	rpm
Corrente de Pico	7,2In	A
Fator de Potência	0,82	-

Fonte: o autor.

pode-se emular diferentes níveis de curto-circuito entre espiras do mesmo enrolamento. Neste trabalho, falhas por curto-circuito entre fases não são consideradas.

Um Freio de Foucault fora utilizado para aplicar três níveis de carga ao motor. O primeiro nível utilizado representa a condição ausente de carga aplicada (0%). Já o segundo nível representa a condição metade da capacidade de carga nominal (50%). Por fim, o último nível representa a totalidade da carga nominal (100%).

### 2.2.2 Sistema Auxiliar de Curto-Circuito

Um sistema auxiliar de comando fora construído para executar dois tipos de curto-circuito discutidos no capítulo anterior: alta impedância (AI - emulando o início do processo de curto-circuito), e baixa impedância (BI - emulando a intensificação da falha).

Para estes dois tipos de curto-circuito, foram estabelecidos três níveis de curto-

circuito. 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 colocadas em curto-circuito. Conseqüentemente, existem seis condições diferentes de falha: AI1, AI2, AI3, BI1, BI2 e BI3.

### **2.2.3 Conversor de Frequência**

Para variar a velocidade do motor, utilizou-se um conversor de frequência de modelo CFW-09, fabricação WEG, 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 equipamento operou no modo escalar  $V/f$  para a geração da modulação por largura de pulsos do tipo senoidal com frequência de chaveamento de 10 kHz.

### **2.2.4 Sistema de Aquisição de Dados**

Três sensores de efeito Hall foram utilizados para aferir as correntes de linha do motor. Os dados foram adquiridos através de um sistema de aquisição Agilent de modelo U2352 o qual possui resolução de 16 bits, 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 cada amostra da base de dados, após escolhidas as condições de operação do motor (fase do conversor, nível de carga, frequência comandada e extensão da falha), realizou-se a aquisição do sinal durante o período de 10 segundos utilizando taxa de amostragem de 10 kHz. Assim, cada amostra armazenada é um sinal estacionário de 100.000 pontos.

### **2.2.5 Extração e Seleção de Atributos**

Considerando-se interesse apenas no regime estacionário dos sinais adquiridos, aplicou-se a transformada rápida de Fourier (do inglês, *fast Fourier transform* - FFT) sobre estes sinais para a obtenção do espectro de frequência de cada amostra coletada.

Em consequência do filtro analógico de 1 kHz utilizado, pode-se escolher componentes de frequência de até 500 Hz<sup>3</sup> como informação a ser considerada no problema. Além disso, devido à frequência máxima de alimentação do conversor utilizado ser limitada em 60 Hz,

<sup>3</sup> Restrição imposta pelo Teorema de Nyquist.

a maior componente múltipla desta frequência, considerando o limite superior de 500 Hz, é a oitava harmônica<sup>4</sup>.

Considerando o limite máximo de componente múltipla harmônica, Oliveira *et al.* (2013a) aplicaram a FFT aos sinais que compõem o banco de dados e verificaram, após aplicarem passos de  $0,5f_s$ , que as componentes do espectro de frequências que apresentaram os maiores valores de variância eram as seguintes:  $0,5f_s$ ;  $1,5f_s$ ;  $2,5f_s$ ;  $3f_s$ ;  $5f_s$ ; e  $7f_s$ . Logo, estas componentes foram utilizadas como atributos no conjunto de dados.

### 2.3 Considerações Sobre o Banco de Dados

É importante ressaltar que, em ambos os tipos de curto-circuito, as correntes de falha foram limitadas à corrente nominal do motor. Esta limitação é necessária para que o motor possa ser reutilizado após várias emulações de falhas sem ser danificado.

Além disso, utilizou-se o motor em conexão delta. Nesta configuração, duas correntes de linha do conversor de frequências foram diretamente conectadas à fase do motor que continha as falhas. Dado que a futura construção de um sistema capaz de detectar falhas utilizando apenas uma fase do conversor, independentemente do sensor estar conectado diretamente ou não à fase com falha, é um objetivo final em uma pesquisa de aplicação, apenas uma das duas fases é utilizada para que não haja informação redundante.

Considerando a utilização de apenas duas correntes de linha, o conjunto de dados é formado de um exemplar por combinação possível entre fase do conversor (2 possibilidades), nível de carga (3 possibilidades), frequência comandada (7 possibilidades) e extensão da falha (7 possibilidades). O campo extensão da falha é formado pela condição normal (NO) de operação, os 3 níveis da falha de alta impedância e os 3 níveis de falha de baixa impedância.

Logo, o banco de dados é composto por 294 amostras ( $2 \times 3 \times 7 \times 7$ ), onde cada amostra representa uma condição específica de operação. Além disso, pode-se verificar que o banco de dados é composto por 147 amostras da fase 1 (diretamente conectada à bobina com falha) e 147 amostras da fase 3 (indiretamente conectada à bobina com falha). Todas estas condições de operação estão listadas na Tabela 2.

A partir da Figura 5, pode-se verificar que 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 e BI3). Com esta configuração, cada classe possui 42 amostras.

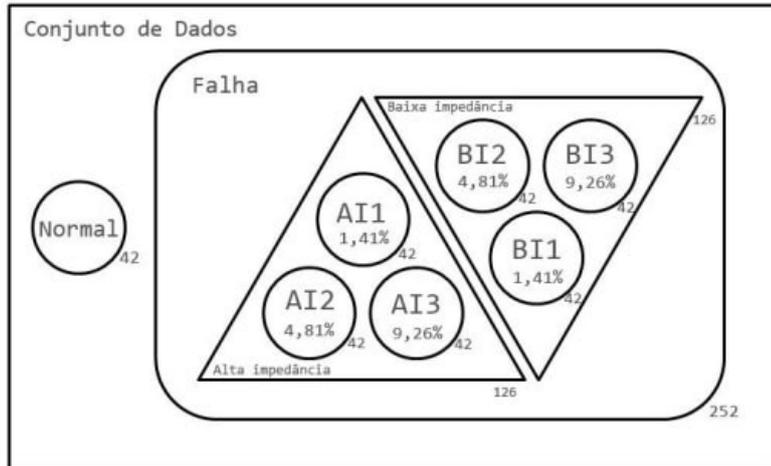
<sup>4</sup> Pois  $8 \times 60 \text{ Hz} = 480 \text{ Hz}$ .

Tabela 2 – Condições de operação do motor

Fase do Conversor	Nível de Carga	Frequência Comandada	Extensão da Falha
Fase 1; Fase 3	0%; 50%; 100%	30; 35; 40; 45; 50 ; 55; 60 Hz	NO; A1; A2; A3; B1; B2; B3

Fonte: o autor.

Figura 5 – Características das amostras



Fonte: Coelho (2015).

Além da classificação com 7 classes, pode-se classificar quanto ao estado de operação do motor (NO, AI e BI) e, ainda assim, o problema mantém-se multiclases, porém com apenas 3 classes desbalanceadas em que uma classe possui 42 amostras (normal), outra com 126 amostras (AI1, AI2 e AI3) e a última também com 126 amostras (BI1, BI2 e BI3).

Por último, ainda é possível classificar entres os estados normal e falha. Neste caso, tem-se classificação binária ao considerar que uma classe possui 42 amostras (normal) e a outra possui 252 amostras (AI1, AI2, AI3, BI1, BI2 e BI3).

Como explorado, o conjunto final de dados é composto por 294 amostras, onde cada um dos atributos corresponde a uma harmônica múltipla da frequência fundamental comandada pelo conversor e é normalizada pela frequência fundamental. Logo, cada amostra deste conjunto de dados pode ser representada por

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

em que  $\mathbf{x}_n$  é o vetor coluna (vetor de atributos) contendo valores de componentes do espectro de frequência da corrente do motor.

## 2.4 Análise Não Supervisionada Prévia do Banco de Dados

No trabalho de Sousa *et al.* (2017), resultado parcial desta dissertação mencionado na Seção 1.5, realizou-se uma análise não supervisionada do banco de dados. Nesta análise, utilizou-se a abordagem de *clustering* hierárquico definida como *Clustering of the SOM*. Esta abordagem heurística, proposta por Vesanto e Alhoniemi (2000), foi utilizada em conjunto com os critérios de validação de agrupamentos que serão descritos na Seção 3.3.

Neste resultado parcial da dissertação, verificou-se que o aprendizado não supervisionado guiado por critérios de validação de agrupamentos encontrou diferentes tipos de sobreposições entre as classes ao utilizar o conjunto de dados inteiro (todas as classes misturadas). Então, pode-se inferir que exemplares de classes distintas possuem correlações e, conseqüentemente, trata-se de um problema complexo de classificação.

As correlações encontradas de amostras pertencentes à classes distintas são mostradas na Tabela 3. Por exemplo, têm-se 35 amostras da classe falha de alta impedância que são agrupadas com 18 amostras da classe de falha de baixa impedância no *Cluster 6*. A sobreposição destas duas classes causadas por essas amostras dificulta o aprendizado supervisionado, pois estas amostras são estatisticamente semelhantes mesmo que possuam significados distintos.

Também nesta tabela, um exemplo distinto pode ser verificado nos *Clusters 2 e 7*. Estes agrupamentos são formados exclusivamente por falhas de baixa impedância e pode-se inferir, a partir desta verificação, que estas amostras serão "aprendidas" mais facilmente por técnicas de aprendizado supervisionado pois estas amostras constituem uma topologia bem definida.

Tabela 3 – Distribuição de classes por *cluster* entre 8 *clusters*

Rótulos	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
NO	14	0	9	9	4	0	0	6
AI	7	0	30	21	12	35	0	21
BI	5	20	23	20	10	18	16	14

Fonte: adaptada de Sousa *et al.* (2017).

## 2.5 Conclusão

Neste capítulo, apresentou-se as técnicas utilizadas na construção do banco de dados a ser utilizado nesta dissertação. Dado que exemplos de motores em funcionamento normal são mais frequentes que exemplos de falhas em ambiente industrial, conclui-se que a distribuição dos dados coletados é incomum. Esta característica já torna este conjunto de dados uma contribuição em si mesmo, pois foram geradas situações de falhas em uma frequência bem maior do que encontrada na prática industrial. Isto faz com que um problema de detecção de falhas, que normalmente é tratado como um problema de detecção de anomalias (*1-class problem*), seja tratado como um problema de classificação com múltiplas classes.

### 3 APRENDIZADO NÃO SUPERVISIONADO BASEADO EM PROTÓTIPOS

Neste capítulo, os métodos de aprendizado não supervisionado baseado em protótipos são abordados. Em seguida, o algoritmo de agrupamento de dados e as técnicas de validação de agrupamentos utilizados neste trabalho são apresentados.

Este capítulo está organizado da seguinte forma: a Seção 3.1 apresenta as definições iniciais necessárias para a compreensão do aprendizado não supervisionado baseado em protótipos; a Seção 3.2 apresenta as definições de vetores protótipos e aprendizado não supervisionado e, por fim, a Seção 3.3 apresenta as técnicas de validação de agrupamentos utilizadas neste trabalho.

#### 3.1 Definições Iniciais

Para o entendimento dos modelos a serem descritos neste trabalho, supõe-se a disponibilidade de um conjunto de  $N$  exemplos representados como vetores de dimensão  $p$ . Logo, um vetor exemplo  $\mathbf{x} \in \mathbb{R}^p$  é representado como

$$\mathbf{x}^\top = (x_1, \dots, x_p)^\top, \quad (3.1)$$

em que a  $j$ -ésima componente  $x_j(t)$  carrega alguma informação relevante para a análise em questão, sendo denominada por isso de **característica** (*feature*) ou **atributo** (*attribute*). Em aprendizado baseado em protótipos, as componentes de um vetor podem representar diferentes variáveis (massa, força, tensão, largura, etc.), ou podem representar a mesma variável deslocada espacialmente e/ou temporalmente (*pixels* de uma imagem ou amostras de sinais de voz).

Dessa forma, um vetor  $\mathbf{x}$  é, normalmente, chamado de **vetor de características** (*feature vector*) ou **vetor de atributos** (*attribute vector*) em análise estatística de dados (WEBB, 2002). Através do mapeamento dessas características é que os algoritmos não supervisionados constroem suas representações estatísticas dos dados de entrada.

#### 3.2 Vetores Protótipos e Aprendizado Não Supervisionado

O principal objetivo do aprendizado não supervisionado baseado em protótipos é a eficiente modelagem de vetores protótipos capazes de representar a frequência de observações no espaço de características. Este tipo de aprendizado pode ser aplicado para fins de redução de

armazenamento de observações, para reavaliar estruturas nos dados (ex: agrupamentos), ou para pré-processar grandes conjuntos de dados para futuras análises.

Em algoritmos de aprendizado não supervisionado, o objetivo dos vetores protótipos é a minimização, durante a etapa de treinamento, de uma função custo que seja capaz de quantificar a qualidade da representação de um dado conjunto com  $p$  atributos por um conjunto menor contendo  $K$  protótipos com, também,  $p$  atributos. Na abordagem *crisp* (a abordagem mais tradicional), cada exemplo de entrada é associado ao protótipo mais próximo<sup>1</sup>, definido como protótipo vencedor, entre a matriz de protótipos  $\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_K\}$ , onde  $\mathbf{W} \in \mathbb{R}^{p \times K}$ . Considera-se nesta dissertação, inicialmente, a distância euclidiana quadrática<sup>2</sup> como a métrica de dissimilaridade a ser utilizada no espaço de características. A função custo associada a esta métrica de dissimilaridade é chamada de soma dos erros quadráticos de quantização (SEQQ) e é definida como

$$SEQQ = \sum_{i=1}^N d^2(\mathbf{w}^*(\mathbf{x}_i), \mathbf{x}_i), \quad (3.2)$$

onde  $\mathbf{w}^*(\mathbf{x}_i)$  representa o vetor protótipo mais próximo ao exemplo de entrada  $\mathbf{x}_i$  e é definido como

$$d(\mathbf{w}^*(\mathbf{x}_i), \mathbf{x}_i) \leq d(\mathbf{w}_j, \mathbf{x}_i), \quad \forall j = 1, 2, \dots, K. \quad (3.3)$$

A partir das Equações (3.2) e (3.3), pode-se inferir que o SEQQ soma as contribuições de cada vetor de características aos seus respectivos protótipos mais próximos. Consequentemente, o SEQQ quantifica a magnitude da confiabilidade da representação dos dados de entrada pelo conjunto de protótipos.

Apesar do SEQQ ser uma função capaz de quantificar a qualidade da representação dos dados, esta função deve ser aplicada apenas para comparar sistemas que possuam a mesma quantidade de protótipos. Por exemplo, espera-se, ao comparar um mesmo sistema com diferentes quantidades de protótipos, que o cenário com maior disponibilidade de protótipos será beneficiado ao produzir menores valores de SEQQ. Também é verificado que se tem  $SEQQ = 0$  ao posicionar um protótipo em cada exemplo de entrada.

Segundo Biehl *et al.* (2016), quantização vetorial (QV) é frequentemente confundida ou, até mesmo, identificada como análise de agrupamentos ao se considerar que os protótipos representam agrupamentos distintos e o erro de quantização relaciona a distâncias intragrupos

<sup>1</sup> Segundo a métrica de dissimilaridade escolhida.

<sup>2</sup>  $d^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^\top (\mathbf{x} - \mathbf{y})$  para  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$

encontradas. Porém, é sabido que QV pode ser utilizada em tarefas de detecção e identificação de grupos com informações similares mesmo que não se tenha estruturas de grupos nos dados.

Conforme ilustrado na Figura 6, a minimização da função custo (ex: SEQQ) não consiste, necessariamente, na identificação de grupos previamente conhecidos e, também, pode-se observar que o resultado da QV é fortemente dependente da representação dos dados. A Figura 6a exibe um único grupo alongado formado por protótipos que resumem as propriedades essenciais e caracterizam a variação dos vetores de características observados sem necessariamente sugerir divisões intuitivamente significativas dos dados em grupos. Na Figura 6b, ilustra-se o cenário idealizado de dois aglomerados com semelhanças esféricas em que cada grupo é representado por um protótipo. Na Figura 6c, são mostrados agrupamentos alongados onde os protótipos não representam os centros destes mesmo que as posições dos protótipos minimizem o SEQQ. Finalmente, a Figura 6d ilustra a identificação de um agrupamento menor que é facilmente separado do maior e não é representado por protótipo(s) devido à sua baixa contribuição no cálculo do SEQQ.

O aprendizado não supervisionado baseado em protótipos é realizado através de redes neurais competitivas. Nesta dissertação, para fins de simplificação, estas redes serão chamadas de redes competitivas (RC). Segundo Grossberg (1976) e Rumelhart e Zipser (1985), os primeiros modelos de RC surgiram a partir de estudos que investigavam hipóteses sobre a organização dos neurônios em determinadas áreas do córtex cerebral, por exemplo, a partir de símbolos sensoriais e como esta organização neuronal está associada ao aprendizado.

A Figura 7 ilustra diferentes áreas em que as redes competitivas podem ser aplicadas. Qualquer que seja a abordagem adotada para construir uma RC, a ideia básica da competição entre os protótipos é a seguinte: os protótipos da camada de saída competem entre si pelo direito

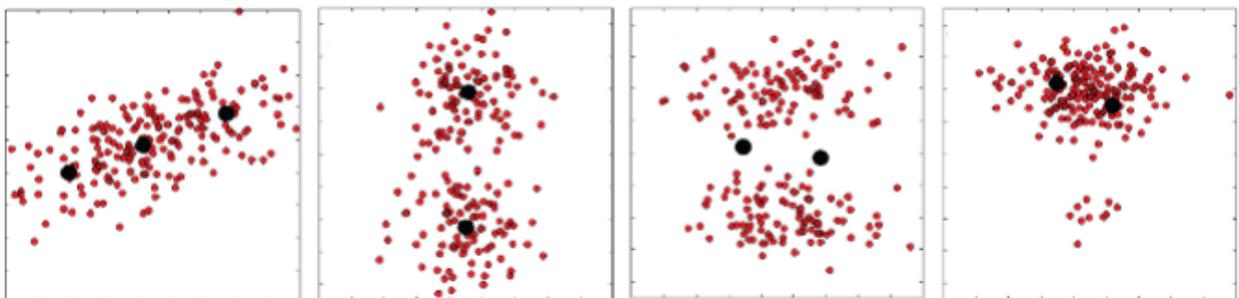
Figura 6 – Possíveis representações de dados bidimensionais através de protótipos

(a) Exemplo I

(b) Exemplo II

(c) Exemplo III

(d) Exemplo IV



Fonte: adaptada de Biehl *et al.* (2016).

de responder, ou seja, de permanecerem ativos para um determinado estímulo de entrada. Ao final da competição, apenas um protótipo (ou um pequeno grupo de protótipos) estará ativo em resposta àquela informação de entrada. Em geral, através de disparos de um protótipo ao longo do processo de ajuste das conexões sinápticas, este passará a indicar a ocorrência de um padrão de entrada que possua um determinado conjunto de características próprias do grupo a que o padrão de entrada pertence. Isto é possível porque, como explicado anteriormente, os vetores protótipos são capazes de representar a frequência de observações no espaço de características.

O aprendizado competitivo é formulado a partir de um algoritmo gradiente descendente e um estocástico (ou *online*) onde a função custo  $H_{QV}$  é responsável por um esquema de atualização muito intuitivo (ROBBINS; MONRO, 1985). A cada iteração, um único vetor de características  $\mathbf{x}$  é escolhido aleatoriamente do conjunto de dados em que todos vetores de características possuem a mesma probabilidade de seleção. Então, o atual protótipo vencedor  $\mathbf{w}_c(t)$ , associado ao vetor de características escolhido, é determinado através da Equação 3.3. Logo, o protótipo vencedor é atualizado como

$$\mathbf{w}_c(t+1) = \mathbf{w}_c(t) + \eta(\mathbf{x} - \mathbf{w}_c(t)), \quad (3.4)$$

onde a constante  $\eta$ , chamada de taxa de aprendizagem, é responsável pelo controle da magnitude das atualizações. Esta abordagem descrita acima é um exemplo de esquema o vencedor leva tudo (WTA, do inglês "*Winner Takes All*").

Figura 7 – Projeção das RC em diferentes áreas



Nesta dissertação, a aplicação em objetivo é a identificação de padrões em formações de agrupamentos que representem diferentes condições de operação de um motor trifásico. Portanto, para a obtenção de protótipos semelhantes ao cenário idealizado da Figura 6b, a abordagem de aprendizado não supervisionado baseado em protótipos neste trabalho envolverá análise de agrupamentos. Esta técnica de AM é apresentada na Seção 3.2.1.

### 3.2.1 *Análise de Agrupamentos*

A análise de agrupamentos, também conhecida como clusterização (do inglês, *clustering*), tem sido frequentemente utilizada em tarefas de mineração de dados e reconhecimento de padrões. (METZ, 2006).

O resultado obtido por meio dos algoritmos de *clustering* é um conjunto de agrupamentos de dados, nos quais cada agrupamento é denominado *cluster*. Os agrupamentos obtidos por meio dessa técnica de aprendizado de máquina não supervisionado são altamente dependentes da escolha de parâmetros, como as medidas de dissimilaridade e métodos de agrupamentos utilizados.

Segundo Webb (2011), pode-se caracterizar um *cluster* como um agrupamento de objetos em uma população com o propósito de encontrar estruturas nos dados. Nesta abordagem, é desejado que objetos pertencentes ao mesmo agrupamento estejam próximos ou sejam similares, mas diferentes e distantes de objetos de outros grupos. Portanto, um *cluster* é composto de um número não fixo de objetos similares de acordo com uma medida de similaridade ou dissimilaridade escolhida.

Em consequência de diversas interpretações encontradas na literatura, se faz necessário escolher definições que sejam gerais para qualquer aplicação. De acordo com Everitt (2011), as definições mais utilizadas são as seguintes:

**Definição 1:** Um *cluster* é um conjunto de entidades semelhantes e entidades pertencentes a *clusters* diferentes não são semelhantes.

**Definição 2:** Um *cluster* é um agrupamento de pontos no espaço de teste de tal maneira que a distância entre quaisquer dois pontos em um mesmo *cluster* é menor que a distância entre qualquer ponto desse *cluster* e um outro ponto qualquer não pertencente a ele.

**Definição 3:** *Clusters* podem ser descritos como regiões conectadas de um espaço multidimensional contendo uma alta densidade relativa de pontos, separadas de outras regiões por uma região contendo uma baixa densidade relativa de pontos.

Os algoritmos de *clustering* podem ser validados considerando diferentes aspectos. Assim, os métodos de validação destes algoritmos são descritos na Seção 3.3. Descreve-se, a seguir, um modelo clássico (não neural) de análise de agrupamentos.

### 3.2.1.1 Algoritmo $K$ -médias

O algoritmo  $K$ -médias (MACQUEEN *et al.*, 1967) é também conhecido no campo de quantização vetorial como algoritmo de Linde-Buzo-Gray (LBG) ou algoritmo de Lloyd generalizado (VASUKI; VANATHI, 2006). A aplicação do algoritmo  $K$ -médias a um conjunto de  $N$  vetores visa encontrar um conjunto de  $K$  protótipos,  $\{\mathbf{w}_i\}_{i=1}^K$ ,  $K \ll N$ , que particione os dados de entrada em exatamente  $K$  grupos distintos.

A região de influência de determinado protótipo é chamada de partição de Voronoi (ou Dirichlet) daquele protótipo, sendo definida como

$$V_i = \{\mathbf{x} \in \mathbb{R}^p \mid \|\mathbf{x} - \mathbf{w}_i\| < \|\mathbf{x} - \mathbf{w}_j\|, \forall j \neq i\}, \quad (3.5)$$

em que  $\|\cdot\|$  denota a norma euclidiana. Assim, com  $K$  protótipos o espaço de entrada é particionado em  $K$  regiões de Voronoi. O algoritmo  $K$ -médias provê um método simples para a obtenção de  $K$  protótipos que minimizem a SEQQ definida na Equação 3.2.

Também conhecida pelos seguintes nomes: erro de reconstrução ou ainda erro de distorção. Esta minimização é realizada através da implementação da seguinte sequência de passos:

**Passo 1** - Seleção aleatória de  $K$  vetores do conjunto de dados para funcionar como protótipos iniciais;

**Passo 2** - Separação do conjunto de dados em  $K$  regiões de Voronoi  $V_i$ ,  $i = 1, \dots, K$ , de acordo com a Equação (3.5);

**Passo 3** - Os novos protótipos são recalculados como as médias aritméticas (centróides) dos dados alocados a cada região de Voronoi  $V_i$ , ou seja

$$\mathbf{w}_i = \frac{1}{N_i} \sum_{\mathbf{x} \in V_i} \mathbf{x}, \quad (3.6)$$

onde  $N_i$  é o número de vetores pertencentes à célula de Voronoi do  $i$ -ésimo protótipo; ou equivalentemente, é o número de vetores de dados para os quais o protótipo  $\mathbf{w}_i$  é o mais próximo, segundo a métrica euclidiana.

Os passos 2 e 3 devem ser repetidos até que não haja mudanças substanciais no valor da SEQQ, ou um determinado número máximo de iterações tenha sido alcançado. Um critério

de parada comumente usado verifica se a taxa de variação da SEQQ está abaixo de um limiar de distorção  $0 < \varepsilon \ll 1$  preestabelecido, ou seja

$$\left| \frac{SEQQ(t+1) - SEQQ(t)}{SEQQ(t+1)} \right| < \varepsilon, \quad (3.7)$$

em que o operador  $|u|$  denota o valor absoluto de  $u$  e  $SEQQ(t)$  corresponde ao valor da soma dos erros quadráticos de quantização na  $t$ -ésima rodada de ajuste dos protótipos.

O interesse do algoritmo  $K$ -médias *batch* para este trabalho é utilizá-lo em uma etapa de pré-processamento de uma tarefa de classificação baseada em protótipos.

### 3.3 Validação de Agrupamentos

Conforme descrito na Seção 3.2, a função custo SEQQ não é uma métrica apropriada para comparar o posicionamento de protótipos quando se deseja avaliar relações intragrupos (ver Figura 6) e quando se compara os mesmos dados de entrada com diferentes quantidades de protótipos. Logo, faz-se necessário a utilização de validação de agrupamentos. A técnica validação de agrupamentos utiliza métricas baseadas no cálculo de relação intragrupos e entre grupos. Estas métricas são definidas como critérios de validação.

Os critérios de validação indicam a maneira pela qual um índice é aplicado para validar um agrupamento. Existem três tipos de critérios para investigar a validade de um agrupamento, a saber: critérios internos, critérios externos e critérios relativos.

Os critérios internos quantificam a qualidade de um agrupamento com base apenas nos dados originais (matriz de dados, matriz de distância ou matriz de covariância). Por exemplo, um critério interno pode avaliar o grau em que uma partição obtida por um dado algoritmo de agrupamento é justificado pela matriz de covariância.

Já os critérios externos quantificam um agrupamento julgando-o a partir de uma estrutura pré-definida, que é imposta ao conjunto de dados e que reflete a intuição do usuário sobre a estrutura presente nos dados. Essa estrutura pré-definida pode ser uma partição que se sabe previamente existir nos dados, ou um agrupamento construído por um especialista da área com base em conhecimento prévio. Um critério externo pode avaliar o grau de correspondência entre o número de agrupamentos obtidos com o agrupamento e os rótulos dos dados conhecidos previamente.

Por fim, os chamados critérios relativos comparam diversos agrupamentos para decidir qual deles é o melhor em algum aspecto, por exemplo, qual é o mais estável ou qual é o

mais adequado aos dados. Esses critérios podem ser utilizados para comparar diversos algoritmos de agrupamento ou para determinar o valor mais apropriado de algum parâmetro do algoritmo aplicado, como o número de agrupamentos. Por exemplo, pode-se medir quantitativamente qual dentre dois algoritmos melhor se ajusta aos dados ou determinar o número de agrupamentos mais apropriado para um agrupamento feito com um determinado algoritmo.

Segundo Webb (2011), as duas primeiras abordagens de validação de agrupamentos são baseadas em testes estatísticos ao utilizar modelos para diferentes estruturas definidas *a priori*, enquanto que a terceira abordagem objetiva a descoberta da melhor estrutura de agrupamento a partir de uma métrica pré-estabelecida.

Dentre esses, os critérios relativos são os que se adequam aos objetivos deste trabalho. Consequentemente, a tarefa de validação de agrupamentos neste trabalho é realizada através de critérios relativos, os quais são descritos na próxima seção.

Além disso, o presente trabalho também realizou a validação de agrupamentos a partir de uma modificação da teoria de critérios de informação para a estimação da ordem do modelo. Os índices de validação de agrupamentos a partir desta metodologia são descritos na Seção 3.3.2.

### 3.3.1 Índices Baseados em Critérios Relativos

#### 3.3.1.1 Índice Calinski-Harabasz (CH)

O índice CH, proposto por Caliński e Harabasz (1974), é representado pela seguinte expressão:

$$CH(K) = \frac{\text{traço}(\mathbf{B}_K)/(K-1)}{\text{traço}(\mathbf{W}_K)/(N-K)}, \quad (3.8)$$

onde  $\mathbf{B}_K = \sum_{l=1}^K \#C^l (\mathbf{x}^l - \mathbf{x})(\mathbf{x}^l - \mathbf{x})'$  é a soma dos quadrados entre *clusters*,  $N$  é total de amostras,  $K$  é o número de *clusters*,  $\mathbf{W}_K = \sum_{l=1}^K \sum_{i=1}^{\#C^l-1} \sum_{j=i+1}^{\#C^l} (\mathbf{x}_i^l - \mathbf{x}_j^l)(\mathbf{x}_i^l - \mathbf{x}_j^l)'$  é a soma dos quadrados *intracluster*,  $\mathbf{x}_i^l$  e  $\mathbf{x}_j^l$  são, respectivamente, os  $i$ -ésimos e  $j$ -ésimos itens do *cluster*  $C^l$ ,  $\mathbf{x}^l$  é o centróide do  $l$ -ésimo *cluster*,  $\#C^l$  é a cardinalidade do  $l$ -ésimo *cluster* e  $\mathbf{x}$  é o centróide do banco de dados inteiro.

A escolha da quantidade de *clusters*  $K$  é dada pelo argumento que maximiza o índice CH. Perceba que  $CH(K)$  não é definido para  $K = 1$  e portanto não pode ser usado.

### 3.3.1.2 Família de Índices Dunn

A família de índices Dunn, proposta por Dunn (1973) é representada genericamente pela seguinte expressão:

$$D(K) = \frac{\min_{i \neq j} \{\delta(S_i, S_j)\}}{\max_{1 \leq l \leq k} \{\Delta(S_l)\}}, \quad (3.9)$$

onde  $\delta(S_i, S_j)$  denota uma função de dissimilaridade (ex: distância euclidiana) entre os agrupamentos  $S_i$  e  $S_j$ , e  $\Delta(S_i)$  é uma medida da dispersão dos dados dentro do agrupamento  $S_i$ .

No índice Dunn original,  $\delta(S_i, S_j)$  é definido pela equação:

$$\delta(S_i, S_j) = \min_{\mathbf{x} \in S_i, \mathbf{y} \in S_j} \{d(\mathbf{x}, \mathbf{y})\}, \quad (3.10)$$

e  $\Delta(S_i)$  é dada pela equação:

$$\Delta(S_i) = \max_{\mathbf{x}, \mathbf{y} \in S_i} \{d(\mathbf{x}, \mathbf{y})\}. \quad (3.11)$$

O trabalho de Pakhira *et al.* (2004) analisou o índice Dunn como a razão da separação entre os agrupamentos (*intercluster*) e dentro dos agrupamentos (*intracluster*). O gráfico índice  $D(K)$  não apresenta nenhuma tendência em relação ao número de agrupamentos. O ponto máximo no gráfico de  $D(K)$  contra  $K$  pode ser uma indicação do número de agrupamentos que mais se ajusta aos dados.

De acordo com Halkidi *et al.* (2002), o índice Dunn apresenta bons resultados na identificação de agrupamentos compactos e bem separados. Os valores altos do índice sugerem a presença desse tipo de agrupamento. As principais limitações do índice Dunn original são a sua complexidade e sensibilidade a ruído. Esse índice também não é apropriado para agrupamentos de formas arbitrárias.

### 3.3.1.3 Índice Davies-Bouldin (DB)

O índice Davies-Bouldin (DB), proposto por Davies e Bouldin (1979), é formulado pela razão da soma da dispersão dentro dos agrupamentos pela dispersão entre os agrupamentos (PAKHIRA *et al.*, 2004).

No índice DB, a métrica  $R_{i,j}$  representa uma medida de similaridade entre dois agrupamentos  $S_i$  e  $S_j$  dada pela equação:

$$R_{j,k} = \frac{e_{S_j} + e_{S_k}}{d(S_j, S_k)}, \quad (3.12)$$

em que  $e_{S_j}$  e  $e_{S_k}$  são, respectivamente, os erros médios para os agrupamentos  $S_j$  e  $S_k$ , e  $d(S_j, S_k)$  denota a distância euclidiana entre os protótipos dos dois agrupamentos.

Seja o índice para o  $k$ -ésimo agrupamento dado indicado por:

$$R_k = \max_{j \neq k} \{R_{j,k}\}. \quad (3.13)$$

O índice de DB é representado pela seguinte expressão:

$$DB(K) = \frac{1}{K} \sum_{k=1}^K R_k. \quad (3.14)$$

O melhor agrupamento é dado pelo menor valor do índice DB para uma sequência de valores calculados para diferentes número de agrupamentos  $K$ . É importante notar que o valor do índice DB é nulo para o agrupamento trivial, em que cada padrão pertence a um agrupamento individual, devendo ser computado apenas quando cada agrupamento contiver um número razoável (ou seja, maior do que 1) de padrões.

De acordo com Jain e Dubes (1988), o índice DB é mais indicado para dados que se organizam em agrupamentos hiperesféricos, não sendo apropriado para agrupamentos de formas arbitrárias.

#### 3.3.1.4 Silhuetas

A medida de silhuetas (do inglês, *silhouettes*), proposta por Rousseeuw (1987), é calculada para cada padrão de entrada que faz parte de um agrupamento. As silhuetas medem a validade dos agrupamentos com base na proximidade entre os padrões de um agrupamento e na distância dos padrões de um agrupamento ao agrupamento mais próximo.

Formalmente, denota-se por  $a(x_i)$  a dissimilaridade média do padrão  $x_i$  em relação a todos os outros padrões do agrupamento  $S_i$  e por  $d(x_i, S_j)$  a dissimilaridade média do padrão  $x_i$  em relação aos padrões do agrupamento  $S_j$ .

Seja  $b(x_i)$  a menor dissimilaridade média de  $x_i$  em relação a todos os demais agrupamentos quantificada pela seguinte equação:

$$b(x_i) = \min_{C_i \neq C_j} \{d(x_i, C_j)\}. \quad (3.15)$$

Assim, a silhueta de um dado padrão,  $\delta(x_i)$ , empregando uma medida de dissimilari-

dade, é dada por:

$$\delta(x_i) = \begin{cases} 1 - a(x_i)/b(x_i), & a(x_i) < b(x_i) \\ 0, & a(x_i) = b(x_i) \\ b(x_i)/a(x_i) - 1, & a(x_i) > b(x_i) \end{cases} \quad (3.16)$$

Então, o índice silhueta  $S(K)$  é representado pela seguinte expressão:

$$S(K) = \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x}_i) \quad (3.17)$$

O critério silhuetas visa indicar quais padrões estão bem situados dentro dos seus agrupamentos e quais estão fora de um agrupamento apropriado. Elas podem ser calculadas para agrupamentos realizados utilizando tanto medidas de similaridade (e.g. produto escalar) quanto medidas de dissimilaridade (e.g. distância euclidiana).

Este método de validação pode ser aplicado tanto a agrupamentos de formatos hiperesféricos, quanto a agrupamentos de formas arbitrárias. A principal limitação do método de validação de agrupamentos por silhuetas está no elevado custo computacional.

### 3.3.2 Índices Baseados em Critérios de Informação

A seguir são descritos alguns critérios amplamente utilizados em identificação de sistemas e séries temporais para seleção e validação de modelos. O objetivo dos critérios de informação é encontrar o modelo que melhor explique o processo gerador dos dados. De forma simplificada, em seleção de modelos busca-se pelo modelo que melhor *explique* os dados.

No âmbito da identificação de sistemas, existem diversos procedimentos que permitem estimar a ordem de modelos dinâmicos a partir de dados medidos. Entre tais procedimentos, destacam-se o critério do erro final de predição (*Final Prediction Error*, FPE), o critério de informação de Akaike (*Akaike's Information Criterion*, AIC), o critério de informação bayesiano (*Bayesian Information Criterion*, BIC) e o critério do comprimento mínimo de descrição (*Minimum Description Length*, MDL).

A expressão geral de um critério de informação  $CI$  é apresentada abaixo:

$$CI(P) = N \ln \left( \frac{RSS(P)}{N} \right) + penalidade(P), \quad (3.18)$$

em que  $N$  é o número de amostras,  $P$  é a ordem do modelo e  $RSS(P)$  é a soma dos resíduos quadráticos para um modelo com  $P$  parâmetros.

É proposta neste trabalho uma simples modificação da Equação 3.18 para que os critérios de informação sejam capazes de avaliar agrupamentos formados através de realizações de *clustering*. Para esta finalidade, substituiu-se a função  $RSS(P)$  pela função erro quadrático médio de quantização (*Mean Squared Quantization Error*, MSQE) para todas a  $N$  amostras de treinamento:

$$MSQE(P) = \frac{1}{N} \sum_{i=1}^K \sum_{l=1}^{n_i} d^2(\mathbf{w}^*(\mathbf{x}_i), \mathbf{x}_i), \quad (3.19)$$

e a ordem do modelo será definida como  $P = K \times p$ , onde  $K$  é a quantidade de protótipos e  $p$  é a quantidade de atributos do vetor de características.

Consequentemente, a expressão geral apresentada na Equação 3.18 é adaptada para:

$$CI(K) = N \ln \left( \frac{MSQE(P)}{N} \right) + \text{penalidade}(P), \quad (3.20)$$

e os critérios descritos a seguir serão apresentados nesta formulação adaptada e suas respectivas funções de penalidades são mostradas na Tabela 4.

Tabela 4 – Penalidades associadas aos diferentes critérios utilizados

<b>Critério:</b>	FPE	AIC	BIC	MDL
<b>Penalidade:</b>	$N \ln \left( \frac{N+P}{N-P} \right)$	$2p$	$p \ln N$	$\frac{p}{2} \ln N$

Fonte: o autor.

### 3.3.2.1 Critério do Erro Final de Predição

O critério FPE, proposto por Akaike (1969), seleciona, de modo a minimizar a variância do erro médio de predição ao mesmo tempo que penaliza o excesso de parâmetros do modelo. Matematicamente, o critério FPE pode ser descrito pela seguinte expressão:

$$FPE(K) = N \ln \left( \frac{MSQE(P)}{N} \right) + N \ln \left( \frac{N+P}{N-P} \right), \quad (3.21)$$

onde  $N$  é o número de amostras,  $P$  é a ordem do modelo com  $K$  protótipos.

A primeira parte do lado direito da Equação 3.21 representa uma função com uma tendência exponencial decrescente à medida que o valor de  $P$  aumenta. Por outro lado, a segunda parte dessa equação deve atuar como um termo de penalização para o excesso de parâmetros e, por isso, exibe uma tendência crescente à medida que  $P$  aumenta. Assim, acredita-se que a função  $FPE(K)$  é convexa e que o seu ponto de mínimo indica a ordem mais adequada do modelo de ordem  $P$ , para aquele conjunto de dados.

### 3.3.2.2 Critério de Informação de Akaike

O critério AIC, proposto também por Akaike (1974), determina a ordem  $P$  do modelo minimizando uma função custo obtida a partir de conceitos oriundos da teoria da informação. A função custo associada à este critério é modelada da seguinte forma:

$$AIC(K) = N \ln \left( \frac{MSQE(P)}{N} \right) + 2P, \quad (3.22)$$

onde o termo  $2P$  representa uma função de tendência linear usada para penalização dos parâmetros excedentes, os quais não resultam na redução do erro quadrático de predição.

De acordo com Kashyap (1980), em função de  $N$ , os critérios AIC e FPE são assintoticamente equivalentes, apresentando o mesmo comportamento para sinais reais, já que a possibilidade de erro na escolha da ordem correta não tende a zero à medida que  $N$  aumenta. A tendência, então, é de subestimar a ordem dos dados à medida que aumenta número de amostras.

### 3.3.2.3 Critério de Informação Bayesiano

O critério BIC é uma outra estatística para seleção de modelo, também chamado do Critério de Informação de Schwarz (SIC), pela interpretação bayesiana dada por Schwarz *et al.* (1978). Matematicamente, este critério é descrito pela seguinte equação:

$$BIC(K) = N \ln \left( \frac{MSQE(P)}{N} \right) + P \ln N. \quad (3.23)$$

De acordo com a Equação (3.23), dados quaisquer dois modelos cujos parâmetros foram estimados, o modelo com o menor valor de BIC é aquele a ser selecionado. De modo semelhante ao critério AIC, o critério BIC é uma função decrescente de  $MSQE$ , adicionada a uma função crescente de  $P$ , contudo o critério BIC penaliza mais os parâmetros excedentes do que o AIC.

### 3.3.2.4 Critério do Comprimento Mínimo de Descrição

O critério MDL, proposto por Rissanen (1978) é obtido a partir de uma variante da função-custo baseada em teoria da informação utilizada pelo critério AIC:

$$MDL(K) = N \ln \left( \frac{MSQE(P)}{N} \right) + \frac{P}{2} \ln N. \quad (3.24)$$

É importante destacar a semelhança entre os critérios BIC e MDL, contudo neste último o termo  $\frac{P}{2} \ln N$  aumenta mais rápido com relação à  $N$  do que com  $P$ .

### 3.4 Conclusão

Neste capítulo, foram apresentadas as definições e a formulação do aprendizado não supervisionado baseado em protótipos. Além disso, abordou-se o clássico algoritmo particional  $K$ -médias e também critérios de validação dos agrupamentos encontrados. Por fim, verifica-se que os resultados obtidos por meio de métodos não supervisionados são altamente dependentes da escolha de parâmetros como as métricas de similaridade e métodos de agrupamentos utilizados.

Após a descrição do algoritmo clássico de aprendizado não supervisionado e dos critérios para determinação do valor mais adequado de agrupamentos, as técnicas destes juntamente com as de algoritmos LVQ (métodos de aprendizado supervisionado baseados em protótipos), que serão detalhadas no próximo capítulo, serão utilizadas no modelo proposto descrito no Capítulo 5.

## 4 CLASSIFICADORES BASEADOS EM *LEARNING VECTOR QUANTIZATION*

Neste capítulo, os métodos de aprendizado supervisionado baseado em protótipos são abordados. Em seguida, os algoritmos da família de classificadores *Learning Vector Quantization* (LVQ) utilizados neste trabalho são apresentados.

Este capítulo está organizado da seguinte forma: a Seção 4.1 apresenta as definições iniciais necessárias para a compreensão do aprendizado supervisionado baseado em protótipos; a Seção 4.2 apresenta a família de classificadores LVQ e, por fim, a Seção 4.3 enumera os classificadores escolhidos para modelagem da solução do problema de detecção de falhas.

### 4.1 Definições Preliminares

Assim como em aprendizado não supervisionado, os métodos de aprendizado supervisionado baseado em protótipos são formulados a partir da suposição de disponibilidade de um conjunto de  $N$  exemplos. Um vetor exemplo deste conjunto, definido como **vetor de atributos**, é representado como mostrado na Equação 3.1 e é portador da informação de  $p$  atributos.

Outra semelhança entre os aprendizados supervisionado e não supervisionado baseados em protótipos é a associação de cada exemplo de entrada à matriz de protótipos  $\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_K\}$ , onde  $\mathbf{W} \in \mathbb{R}^{p \times K}$ .

O treinamento desses algoritmos é realizado por um determinado conjunto de vetores protótipos e seus respectivos rótulos conhecidos. O modelo geral para os classificadores *Learning Vector Quantization* (LVQ) é formalmente descrito a seguir.

Seja  $\mathbf{X} = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^p \times \{1, \dots, C\} \mid i = 1, \dots, N\}$  um conjunto de dados de treinamento, onde  $\mathbf{x} = (x_1, \dots, x_p) \in \mathbb{R}^p$  são dados de entrada  $p$ -dimensionais, com cardinalidade  $\#\mathbf{X} = N$ ;  $y_i \in \{1, \dots, C\}_{i=1, \dots, N}$  são os rótulos das amostras, e  $C$  é o número de classes. Esta rede neural consiste em uma quantidade de protótipos que são caracterizados pelos vetores  $\mathbf{w}_i \in \mathbb{R}^p$ , para  $i = 1, \dots, K$  e seus rótulos  $c(\mathbf{w}_i) \in \{1, \dots, C\}$  com  $\mathbf{Y} = \{c(\mathbf{w}_j) \in \{1, \dots, C\} \mid j = 1, \dots, K\}$ .

Têm-se, em todos os modelos LVQ, que a cardinalidade de classes é sempre menor que a cardinalidade de protótipos. Como consequência, diferentes protótipos podem representar a mesma classe.

O modelo de classificação é baseado na escolha da melhor unidade correspondente (estratégia *winner take all*). O campo receptivo dos protótipos  $\mathbf{w}_i$  é definido a seguir:

$$R^i = \{\mathbf{x} \in \mathbf{X} \mid d(\mathbf{x}, \mathbf{w}_i) < d(\mathbf{x}, \mathbf{w}_j), \forall j \neq i\}, \quad (4.1)$$

onde  $d(\mathbf{w}_i, \mathbf{x})$  é uma medida de dissimilaridade dependente do modelo LVQ escolhido. O treinamento objetiva a determinação dos vetores de peso (protótipos), logo os dados de treinamento são mapeados em seus rótulos correspondentes.

## 4.2 Família de Classificadores Baseados em LVQ

Os classificadores baseados em LVQ pertencem à classe de algoritmos de classificação estatística de padrões formados a partir do cálculo de distâncias. Este tipo de algoritmo é realizado a partir do aprendizado dos protótipos, e estes são os representantes de regiões de classes, conforme explicados na seção anterior.

O objetivo principal do LVQ é a representação de muitos pontos a partir de poucos protótipos. Embora a identificação de agrupamentos não seja o objetivo direto, este é implícito durante a realização desta classe de algoritmos (SEO; OBERMAYER, 2003).

Segundo Kohonen *et al.* (2001 apud NOVA; ESTÉVEZ, 2014), os classificadores LVQ almejam a aproximação das fronteiras teóricas bayesianas e podem, diretamente, tratar problemas multiclases. As denominadas fronteiras teóricas bayesianas são ilustradas na Figura 8. Neste exemplo, a distribuição conhecida dos dados entre as três classes é ilustrada na Figura 8a e a respectiva representação das fronteiras de decisão, após o treinamento dos protótipos, é ilustrada na Figura 8b.

Existem diversas aplicações que utilizam as técnicas de LVQ. Por exemplo, têm-se os trabalhos de Blume e Ballard (1997), Bashyal e Venayagamoorthy (2008), Liu *et al.* (2015) em processamento de sinais e de imagens. Têm-se os trabalhos de Pradhan *et al.* (1996), Pang *et al.* (2014), Belka *et al.* (2018) em aplicações biomédicas e da medicina. Há, também, os trabalhos de Lieberman e Patil (1997), Ahn e Nguyen (2007), Yu *et al.* (2013) em aplicações industriais.

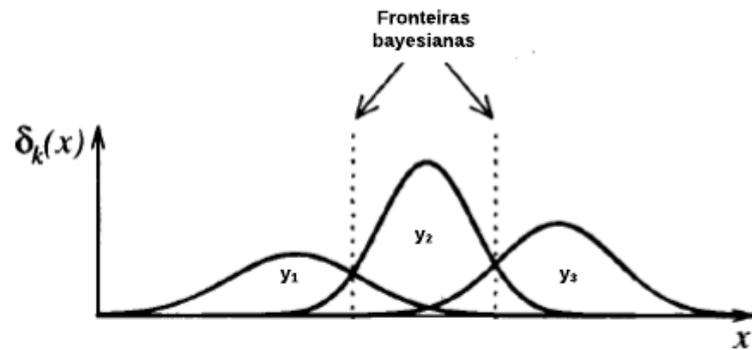
O primeiro algoritmo LVQ é o LVQ1 e foi desenvolvido através do trabalho de Teuvo Kohonen. A primeira versão desta classe de algoritmos foi publicada em 1988. Após a divulgação deste algoritmo, diversas variantes do LVQ1 foram modeladas. Uma busca realizada no ISI *Web of Science* em dezembro de 2018 apontou 1.339 artigos com a palavra-chave “LVQ” em seus títulos ou *abstracts* e, para a palavra-chave “*Learning Vector Quantization*”, são encontrados 2.661 artigos.

Os algoritmos LVQ são agrupados em 3 (três) famílias. A primeira família é denominada de algoritmos heurísticos e é composta pelos métodos propostos por Kohonen. A segunda família é denominada de algoritmos de maximização da margem de separação. Por fim, a terceira

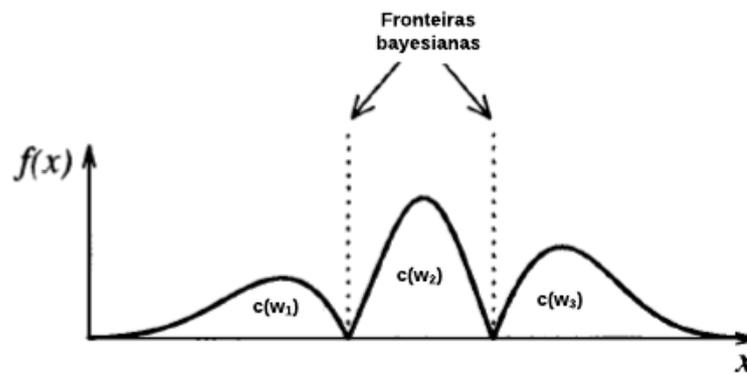
família é denominada de algoritmos de maximização da verossimilhança. As diferentes famílias LVQ são ilustradas na Figura 9 e serão detalhadas a seguir.

Figura 8 – Ilustração das fronteiras teóricas bayesianas

(a) Distribuição dos dados em três classes



(b) Representação das fronteiras de decisão



Fonte: adaptada de Kohonen *et al.* (2001).

Figura 9 – Ilustração simplificada dos princípios de projeto de classificadores LVQ



Fonte: o autor.

### 4.2.1 Algoritmos Heurísticos

Os denominados algoritmos heurísticos foram todos propostos por Teuvo Kohonen. Estes são denominados heurísticos em consequência de não possuírem uma função custo associada que garanta a convergência do algoritmo. Os algoritmos pertencentes a esta família são ilustrados na Figura 10.

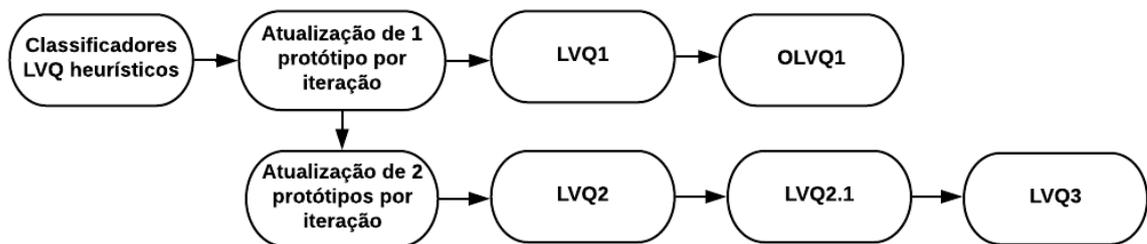
Para a avaliação do protótipo vencedor, todos os algoritmos desta família utilizam a métrica de dissimilaridade distância euclidiana.

Dentre eles, o primeiro algoritmo a ser desenvolvido fora o LVQ1 e é detalhado nos trabalhos de Kohonen (1988) e Kohonen *et al.* (2001). Posteriormente, o algoritmos LVQ2, LVQ2.1, LVQ3 e OLVQ1 foram desenvolvidos com os propósitos de velocidades de convergência mais eficientes e melhores aproximações das fronteiras bayesianas e são detalhados nos trabalhos de Kohonen (1990) e Kohonen *et al.* (2001).

A modelagem dos algoritmos LVQ1 e OLVQ1 é baseada na atualização de 1 protótipo por iteração. Já a modelagem dos algoritmos LVQ2, LVQ2.1 e LVQ3 é baseada na atualização de 2 protótipos por iteração.

Nesta dissertação, os algoritmos heurísticos LVQ1, OLVQ1, LVQ2.1 e LVQ3 foram utilizados na tarefa de detecção de falhas de curto-circuito. Devido a semelhança entre o LVQ2 e LVQ2.1, utilizou-se apenas a variante LVQ2.1 por ser apresentada por Kohonen (1990) como a opção mais estável. Os algoritmos selecionados serão descritos a seguir.

Figura 10 – Projeção dos classificadores LVQ heurísticos



Fonte: o autor.

#### 4.2.1.1 Learning Vector Quantization 1 (LVQ1)

Considerando o conjunto de dados de treinamento  $\mathbf{X}$  e a matriz de vetores protótipos  $\mathbf{W}$  ambos definidos na Seção 4.1, e, também, que a matriz  $\mathbf{W}$  tenha sido adequadamente<sup>1</sup> iniciada, o treinamento do algoritmo LVQ1 é realizado ao longo da apresentação destes dados de treinamento à matriz de vetores protótipos durante uma quantidade de épocas previamente determinada. A cada época, a ordenação de apresentação dos dados torna-se uma nova sequência aleatória.

Ao logo da realização de uma época, ocorre uma iteração para cada padrão de treinamento. A cada  $t$ -ésima iteração deste algoritmo realiza uma atualização na matriz  $\mathbf{W}$  seguindo o procedimento descrito abaixo:

Seja:

$$c = \arg \min_i \{ \| \mathbf{x}(t) - \mathbf{w}_i(t) \| \} \quad (4.2)$$

o índice do protótipo  $\mathbf{w}_i(t)$  mais próximo ao vetor de treinamento  $\mathbf{x}(t)$  entre todos os protótipos disponíveis. Consequentemente, o protótipo  $\mathbf{w}_c(t)$  é definido como vencedor.

Após a determinação do protótipo vencedor, este é atualizado segundo a equação abaixo:

$$\mathbf{w}_c(t+1) = \begin{cases} \mathbf{w}_c(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{w}_c(t)], & \text{se } y(t) = c(\mathbf{w}_c(t)); \\ \mathbf{w}_c(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{w}_c(t)], & \text{caso contrário.} \end{cases} \quad (4.3)$$

Consequentemente, o restante dos protótipos (os considerados não vencedores) não serão atualizados e, para a próxima iteração, permanecerão sem atualizações conforme mostrado na equação abaixo:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t), \quad \forall i \neq c. \quad (4.4)$$

O passo de aprendizagem  $\alpha(t)$  deve decair ao decorrer das iterações e deve ser restrito ao intervalo  $0 < \alpha(t) < 1$ . Segundo Kohonen *et al.* (2001), o passo de aprendizado deve ser iniciado com valores de baixa magnitude, como menores que 0,1.

<sup>1</sup> Considera-se iniciação adequada aquela que supostamente reduz erros de classificação.

#### 4.2.1.2 *Optimized Learning Vector Quantization 1 (OLVQ1)*

Assim como o LVQ1, a variante *Optimized Learning Vector Quantization 1 (OLVQ1)* é também baseada na modificação apenas de um único protótipo vencedor durante a realização de uma iteração.

Neste algoritmo, o protótipo vencedor  $\mathbf{w}_c(t)$  também é definido segundo o critério estabelecido pela Equação 4.2. Os protótipos não vencedores na  $(t + 1)$ -ésima iteração permanecem sem atualização assim como mostrado na Equação 4.4.

O modelo de atualização do protótipo vencedor  $\mathbf{w}_c(t)$  é apresentado abaixo:

$$\mathbf{w}_c(t + 1) = \mathbf{w}_c(t) + s(t)\alpha_c(t)[\mathbf{x}(t) - \mathbf{w}_c(t)], \quad (4.5)$$

onde  $s(t)$  representa uma função sinal e  $\alpha_c(t)$  o passo de aprendizagem associado ao protótipo  $\mathbf{w}_c$  durante a  $t$ -ésima iteração. A função sinal será  $s(t) = +1$  caso a classificação esteja correta<sup>2</sup> ou  $s(t) = -1$  caso a classificação esteja incorreta.

Diferentemente do passo de aprendizagem do LVQ1, o passo de aprendizagem para o OLVQ1 é atualizado segundo o modelo abaixo:

$$\alpha_c(t) = \frac{\alpha_c(t - 1)}{1 + s(t)\alpha_c(t - 1)}. \quad (4.6)$$

Considerando a restrição  $0 < \alpha(t) < 1$  apresentada na Seção 4.2.1.1 e, também, o modelo proposto na Equação 4.6, uma imposição deve ser adotada no desenvolvimento deste algoritmo para que o valor de  $\alpha_c(t)$  não ultrapasse o limite superior estabelecido. Kohonen *et al.* (2001) recomendam que os valores iniciais dos passos de aprendizagem sejam  $\alpha_i = 0,3$  para todos os protótipos.

#### 4.2.1.3 *Learning Vector Quantization 2 (LVQ2) e Learning Vector Quantization 2.1 (LVQ2.1)*

Nos algoritmos LVQ2 e LVQ2.1, o critério utilizado para a decisão do protótipo vencedor é idêntico ao utilizado no LVQ1. Porém, estes algoritmos utilizam modelagens distintas de aprendizado que serão mostradas a seguir.

Nestes algoritmos, são utilizados dois vetores protótipos próximos ao padrão de entrada  $\mathbf{x}$ . Estes vetores são denominados  $\mathbf{w}^+$  e  $\mathbf{w}^-$ , onde  $\mathbf{w}^+$  representa o vetor mais próximo ao padrão de entrada  $\mathbf{x}$  que possui o mesmo rótulo que este padrão e  $\mathbf{w}^-$  representa o vetor mais próximo ao padrão de entrada  $\mathbf{x}$  que possui rótulo diferente deste padrão.

<sup>2</sup> Caso o  $\mathbf{x}(t)$  e  $\mathbf{w}_c(t)$  possuam o mesmo rótulo.

A atualização dos vetores  $\mathbf{w}^+$  e  $\mathbf{w}^-$  é dependente de uma condição  $j$  chamada de restrição da janela que é formulada a partir das distâncias  $d^+$  e  $d^-$ . A métrica  $d^+$  representa a distância euclidiana entre  $\mathbf{x}$  e  $\mathbf{w}^+$ , já a medida  $d^-$  representa a distância euclidiana entre  $\mathbf{x}$  e  $\mathbf{w}^-$ . A restrição da janela é atendida quando um dado de entrada é encontrado dentro da janela calculada conforme o exemplo genérico mostrado na Figura 11. Esta figura ilustra um caso de janela de restrição atendida em um espaço unidimensional qualquer.

Por fim, o cálculo da restrição  $r$  imposta é mostrado abaixo:

$$\min\left(\frac{d^+}{d^-}, \frac{d^-}{d^+}\right) > r, \text{ onde } r = \frac{1-j}{1+j}, \quad (4.7)$$

onde Kohonen *et al.* (2001) recomendam que os valores atribuídos à variável  $j$  pertençam ao intervalo  $0,2 \leq j \leq 0,3$ .

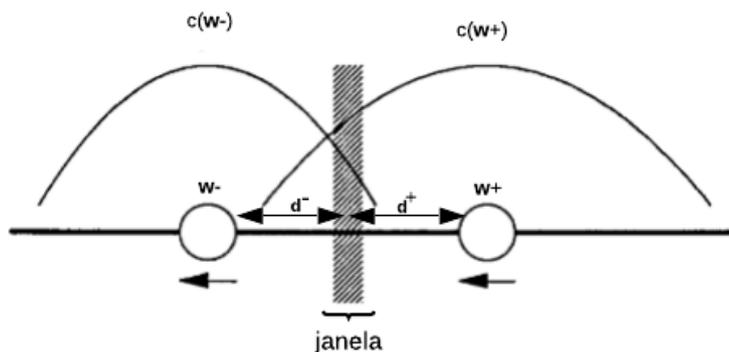
Os algoritmos LVQ2 e LVQ2.1 são diferenciados somente por uma imposição aos protótipos  $\mathbf{w}^+$  e  $\mathbf{w}^-$ . No algoritmo LVQ2, o protótipo incorreto deve ser o protótipo mais próximo ao dado  $\mathbf{x}$  e o protótipo correto deve ser o segundo protótipo mais próximo ao dado  $\mathbf{x}$ . Conseqüentemente, no algoritmo LVQ2, há também a restrição de atualização para os protótipos  $\mathbf{w}^+$  e  $\mathbf{w}^-$  de que valor de  $d^-$  é obrigatoriamente menor que o valor de  $d^+$  para a ocorrência de atualizações entre iterações deste algoritmo.

No LVQ2.1, não há a imposição descrita no parágrafo anterior. Logo, para a ocorrência de atualizações dos vetores  $\mathbf{w}^+$  e  $\mathbf{w}^-$  durante uma iteração  $t$ , é suficiente que  $\mathbf{w}^+$  e  $\mathbf{w}^-$  obedeam à restrição da janela.

Finalmente, o modelo de atualização de  $\mathbf{w}^+$  e  $\mathbf{w}^-$  é apresentado abaixo:

$$\begin{aligned} \mathbf{w}^+(t+1) &= \mathbf{w}^+(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{w}^+(t)], \\ \mathbf{w}^-(t+1) &= \mathbf{w}^-(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{w}^-(t)]. \end{aligned} \quad (4.8)$$

Figura 11 – Exemplo unidimensional da janela de restrição



Fonte: adaptada de Kohonen (1990).

#### 4.2.1.4 Learning Vector Quantization 3 (LVQ3)

Conforme exposto na Seção 4.2.1.3, os algoritmos LVQ2 e LVQ2.1 são desenvolvidos com o objetivo do deslocamento das fronteiras de decisão dos limites bayesianos. Ou seja, estes algoritmos tratam apenas o cenário em que os dois protótipos mais próximos ao padrão de entrada pertencem a classes distintas e nem sempre isto ocorre. Assim, os cenários onde os dois protótipos mais próximos compartilham o mesmo rótulo é desconsiderado.

Diferentemente do LVQ2 e LVQ2.1, a variante LVQ3 considera a opção dos dois protótipos compartilharem o mesmo rótulo. A atualização do segundo protótipo mais próximo é justificada pela continuidade da aproximação da distribuição da classe.

A variante LVQ3 é definida como uma extensão da variante LVQ2.1, ou seja, ele trata do aprendizado da matriz  $\mathbf{W}$  quando os dois protótipos mais próximos possuem classes distintas e quando os dois protótipos mais próximos possuem o mesmo rótulo. Portanto, o treinamento é composto por duas possíveis modelagens de atualizações dos protótipos durante a realização de cada iteração de treinamento. Conseqüentemente, durante o treinamento do LVQ3, realiza-se uma verificação dos rótulos dos dois protótipos vencedores. Além disso, assim como na variante LVQ anterior, a atualização também será dependente da janela de restrição  $j$ :

$$\min \left( \frac{d_a}{d_b}, \frac{d_b}{d_a} \right) > r, \text{ onde } r = \frac{1-j}{1+j}. \quad (4.9)$$

Para o caso em que os dois protótipos possuem rótulos distintos, os índices destes protótipos são definidos como  $a$  e  $b$  para o mais próximo correto e para o mais próximo incorreto respectivamente. Por fim, as atualizações dos protótipos  $\mathbf{w}_a$  e  $\mathbf{w}_b$  são modeladas como:

$$\begin{aligned} \mathbf{w}_a(t+1) &= \mathbf{w}_a(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{w}_a(t)], \\ \mathbf{w}_b(t+1) &= \mathbf{w}_b(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{w}_b(t)]. \end{aligned} \quad (4.10)$$

Desta maneira,  $\mathbf{w}_a$  é aproximado do padrão de entrada enquanto que  $\mathbf{w}_b$  é afastado.

Já para o caso onde os dois protótipos mais próximos possuem classes idênticas, os índices destes protótipos também são definidos como  $a$  e  $b$ . Porém, as atualizações destes protótipos são modeladas como:

$$\mathbf{w}_c(t+1) = \mathbf{w}_c(t) + \varepsilon s(t) \alpha(t) [\mathbf{x}(t) - \mathbf{w}_c(t)], \text{ para } c \in \{a, b\}, \quad (4.11)$$

em que  $s(t)$  representa uma função sinal e a variável  $\varepsilon$  é definida como ganho e deve pertencer ao intervalo  $0, 1 \leq \varepsilon \leq 0,5$  enquanto que a variável  $j$  pertença ao intervalo  $0, 2 \leq j \leq 0,3$ . Por fim, para este caso, os protótipos  $\mathbf{w}_a$  e  $\mathbf{w}_b$  são aproximados do padrão de entrada.

#### 4.2.2 Algoritmos de Maximização da Margem de Separação

Após o desenvolvimento dos algoritmos heurísticos propostos por Teuvo Kohonen, o primeiro algoritmo a utilizar uma função custo em sua formulação é pertencente a família da maximização da margem de separação. A variante pioneira desta família é a *Generalized Learning Vector Quantization* (GLVQ) e foi proposta por Sato e Yamada (1996).

Segundo Nova e Estévez (2014), as abordagens dessa família solucionam algumas limitações do LVQ1 como convergência lenta, iniciação sensível dos protótipos, e limitações em dados multidimensionais com correlações entre os atributos.

Conforme ilustrado na Figura 12, os algoritmos da família maximização da margem de separação são subdivididos nos segmentos espaço dos dados, similaridades e dissimilaridades e, por último, kernelizados.

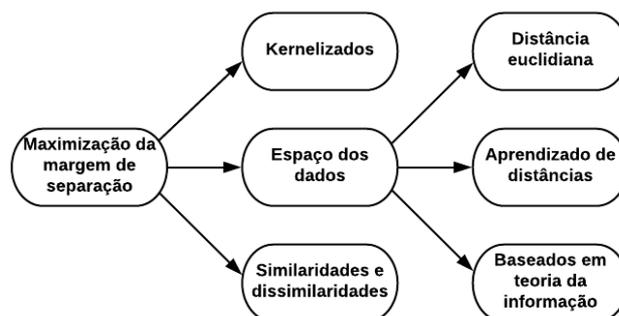
Nesta dissertação, cada segmento desta família será brevemente explorado. Porém, apenas os algoritmos do segmento espaço dos dados serão utilizados na solução do problema proposto e, conseqüentemente, cada variante utilizada desta família será discutida. Especificamente, um algoritmo baseado em distância euclidiana e outros algoritmos baseados no aprendizado de distâncias serão discutidos.

Antes da descrição dos algoritmos desta família, é necessário compreender os conceitos de margens. Os conceitos de margens mais utilizados pela literatura são exploradas em Cramer *et al.* (2003) e são descritos a seguir:

**Margem da amostra** : Calcula o quanto uma amostra pode se deslocar antes de tocar a fronteira de decisão. Esta definição é ilustrada na Figura 13a.

**Margem hipotética** : Calcula o quanto uma hipótese pode se deslocar antes de tocar uma

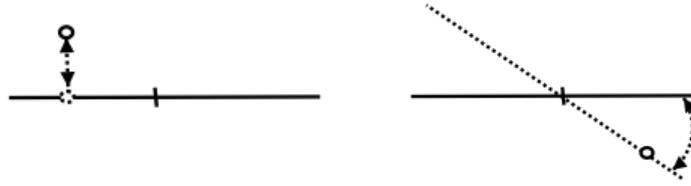
Figura 12 – Família LVQ dos algoritmos de maximização da margem de separação



Fonte: o autor.

Figura 13 – Ilustrações de margens

(a) Exemplo de margem da amostra (b) Exemplo de margem hipotética

Fonte: adaptada de Crammer *et al.* (2003).

amostra. Esta definição é ilustrada na Figura 13b.

Esta família de algoritmos LVQ é inspirada na maximização da margem de separação entre diferentes classes dado que as margens estão associadas aos erros de generalização. Consequentemente, a maximização da margem de separação é equivalente a minimizar os erros de generalização.

Os principais algoritmos desta família e seus respectivos autores são apresentados na Tabela 5. Os nomes destes algoritmos por extenso são apresentados na Tabela 6. Por fim, os algoritmos utilizados no desenvolvimento deste trabalho são destacados nestas tabelas.

Tabela 5 – Principais algoritmos de maximização da margem de separação

Segmento	Métrica	Algoritmo	Autores
Espaço dos dados	Distância euclidiana	GLVQ	Sato e Yamada (1996)
		H2M-LVQ	Qin <i>et al.</i> (2004)
		SNG	Hammer <i>et al.</i> (2005)
		SGNG	Jirayusakul e Auwatanamongkol (2007)
	Aprendizado de distâncias	RLVQ	Bojer <i>et al.</i> (2001)
		GRLVQ	Strickert <i>et al.</i> (2001) e Hammer e Villmann (2002)
		LGRLVQ	Strickert <i>et al.</i> (2001) e Hammer e Villmann (2002)
		GMLVQ	Biehl <i>et al.</i> (2006)
	Baseados em teoria da informação	LGMLVQ	Biehl <i>et al.</i> (2006)
		IT-LVQ	Torkkola e Campbell (2000)
		CSDLVQ	Villmann <i>et al.</i> (2008) e Villmann e Haase (2011)
Similaridades e dissimilaridades	-	RGLVQ	Hammer <i>et al.</i> (2011)
Kernelizados		KGLVQ	Qin e Suganthan (2004)
		AKGLVQ	Schleif <i>et al.</i> (2011)

Fonte: o autor.

#### 4.2.2.1 Generalized Learning Vector Quantization (GLVQ)

Os algoritmos LVQ2.1 e LVQ3, anteriormente desenvolvidos, são modelos construídos com a finalidade da aproximação contínua dos protótipos para as distribuições das classes. Porém, estes algoritmos não apresentam funções custo estáveis.

Tabela 6 – Principais algoritmos de maximização da margem de separação

Algoritmo	Nome por extenso	Algoritmo	Nome por extenso
<b>GLVQ</b>	<i>Generalized LVQ</i>	GMLVQ	<i>Generalized Matrix LVQ</i>
H2M-LVQ	<i>Harmonic to Minimum LVQ</i>	LGMLVQ	<i>Localized Generalized Matrix LVQ</i>
SNG	<i>Supervised Neural Gas</i>	IT-LVQ	<i>Information Theoretic LVQ</i>
SGNG	<i>Supervised Growing Neural Gas</i>	CSDLVQ	<i>Cauchy-Schwarz Divergence LVQ</i>
<b>RLVQ</b>	<i>Relevance LVQ</i>	RGLVQ	<i>Relational Generalized LVQ</i>
<b>GRLVQ</b>	<i>Generalized Relevance LVQ</i>	KGLVQ	<i>Kernel Generalized LVQ</i>
<b>LGRLVQ</b>	<i>Local Generalized Relevance LVQ</i>	AKGLVQ	<i>Accelerated Kernelized Generalized LVQ</i>

Fonte: o autor.

Já o algoritmo *Generalized LVQ* (GLVQ) é um método proposto para a minimização de uma função custo capaz de fazer com que os vetores protótipos se aproximem das distribuições das classes. De acordo com Hammer e Villmann (2002), o sucesso do GLVQ depende da métrica euclidiana ser apropriada para os dados e dos diferentes atributos de entradas possuírem magnitudes semelhantes e que sejam igualmente importantes.

Apesar dos algoritmos heurísticos não possuírem função custo associada, conforme explorado por Kohonen (1990), é comum encontrar na literatura a seguinte função custo para o algoritmo LVQ2.1:

$$f_{LVQ2.1} = d_q^+ + d_q^-, \quad (4.12)$$

onde  $d_q^+$  representa a distância euclidiana quadrática entre o vetor de entrada  $\mathbf{x}$  e o vetor protótipo  $\mathbf{w}^+$  e  $d_q^-$  representa a distância euclidiana quadrática entre o vetor de entrada  $\mathbf{x}$  e o vetor protótipo  $\mathbf{w}^-$ .

É comumente verificado que a função custo acima pode variar abruptamente entre  $t$ -ésima e  $(t + 1)$ -ésima iterações. Diante desta limitação, o algoritmo GLVQ propõe a formulação da função custo descrita a seguir:

Sejam  $\mathbf{w}^+$  e  $\mathbf{w}^-$  os protótipos mais próximos à amostra  $\mathbf{x}$  e que representam, respectivamente, o mais próximo correto e o mais próximo incorreto, formula-se o índice de distância relativa  $\mu(\mathbf{x})$  a partir das variáveis  $d_q^+$  e  $d_q^-$ :

$$\mu(\mathbf{x}) = \frac{d_q^+ - d_q^-}{d_q^+ + d_q^-}. \quad (4.13)$$

Verifica-se que o valor de  $\mu(\mathbf{x})$  é limitado ao intervalo  $-1 \leq \mu(\mathbf{x}) \leq +1$ . O valor de  $\mu(\mathbf{x})$  é negativo se  $\mathbf{x}$  é classificado corretamente. Já o valor de  $\mu(\mathbf{x})$  é positivo se  $\mathbf{x}$  é classificado incorretamente. Conseqüentemente, para a obtenção de melhores taxas de classificação, o valor de  $\mu(\mathbf{x})$  deve decair para todos os dados de entrada no treinamento. Então, o método de

aprendizado do GLVQ é baseado na minimização da função custo  $f_{GLVQ}$  definida como:

$$f_{GLVQ} = \sum_{i=1}^N f(\mu(\mathbf{x}_i)), \quad (4.14)$$

onde  $N$  é quantidade de vetores de entrada para o treinamento, e  $f(\mu)$  é uma função monotônica crescente. Para minimizar  $f_{GLVQ}$ , os vetores  $\mathbf{w}^+$  e  $\mathbf{w}^-$  são atualizados baseados no método descida mais íngreme (do inglês, *steepest descent*) com um passo de aprendizagem  $\alpha$  como demonstrado abaixo:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) - \alpha(t) \frac{\partial f_{GLVQ}(t)}{\partial \mathbf{w}_i(t)}, \quad i = \mathbf{w}^+, \mathbf{w}^-. \quad (4.15)$$

Considerando a métrica de dissimilaridade distância euclideana quadrática, obtém-se o modelo abaixo de convergência formulado por Sato e Yamada (1998):

$$\begin{aligned} \frac{\partial f_{GLVQ}(t)}{\partial \mathbf{w}^+(t)} &= \frac{\partial f_{GLVQ}(t)}{\partial \mu(\mathbf{x}(t))} \frac{\partial \mu(\mathbf{x}(t))}{\partial d_q^+} \frac{\partial d_q^+}{\partial \mathbf{w}^+(t)} \\ &= - \frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} \frac{4d_q^-}{(d_q^+ + d_q^-)^2} (\mathbf{x}(t) - \mathbf{w}^+(t)), \end{aligned} \quad (4.16)$$

$$\begin{aligned} \frac{\partial f_{GLVQ}(t)}{\partial \mathbf{w}^-(t)} &= \frac{\partial f_{GLVQ}(t)}{\partial \mu(\mathbf{x}(t))} \frac{\partial \mu(\mathbf{x}(t))}{\partial d_q^-} \frac{\partial d_q^-}{\partial \mathbf{w}^-(t)} \\ &= + \frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} \frac{4d_q^+}{(d_q^+ + d_q^-)^2} (\mathbf{x}(t) - \mathbf{w}^-(t)). \end{aligned} \quad (4.17)$$

Consequentemente, a regra de aprendizado do GLVQ é definida como:

$$\mathbf{w}^+(t+1) = \mathbf{w}^+(t) + \alpha(t) \frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} \frac{4d_q^-}{(d_q^+ + d_q^-)^2} (\mathbf{x}(t) - \mathbf{w}^+(t)), \quad (4.18)$$

$$\mathbf{w}^-(t+1) = \mathbf{w}^-(t) - \alpha(t) \frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} \frac{4d_q^+}{(d_q^+ + d_q^-)^2} (\mathbf{x}(t) - \mathbf{w}^-(t)). \quad (4.19)$$

Considera-se, então, o valor de  $f(\mu(\mathbf{x}(t)))$ . Verifica-se, através das Equações 4.18 e 4.19, que  $\partial f(\mu(\mathbf{x}(t)))/\partial \mu(\mathbf{x}(t))$  possui o papel de um fator de ganho para a atualização e o seu valor é dependente da entrada  $\mathbf{x}(t)$ .

Como mencionado anteriormente, a função  $f(\mu(\mathbf{x}(t)))$  deve ser uma monotônica crescente. Considerando esta restrição, têm-se as versões GLVQ linear ao utilizar a função identidade e GLVQ não linear ao utilizar a função logística.

Consequentemente, a regra de aprendizado do GLVQ linear é obtida como:

$$\begin{aligned}\mathbf{w}^+(t+1) &= \mathbf{w}^+(t) + \alpha(t) \frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} \frac{4d_q^-}{(d_q^+ + d_q^-)^2} (\mathbf{x}(t) - \mathbf{w}^+(t)) \\ &= \mathbf{w}^+(t) + \alpha(t) \frac{4d_q^-}{(d_q^+ + d_q^-)^2} (\mathbf{x}(t) - \mathbf{w}^+(t)),\end{aligned}\quad (4.20)$$

$$\begin{aligned}\mathbf{w}^-(t+1) &= \mathbf{w}^-(t) - \alpha(t) \frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} \frac{4d_q^+}{(d_q^+ + d_q^-)^2} (\mathbf{x}(t) - \mathbf{w}^-(t)) \\ &= \mathbf{w}^-(t) - \alpha(t) \frac{4d_q^+}{(d_q^+ + d_q^-)^2} (\mathbf{x}(t) - \mathbf{w}^-(t)).\end{aligned}\quad (4.21)$$

Considerando a versão não linear do GLVQ, tem-se:

$$f(\mu(\mathbf{x}(t))) = \frac{1}{1 + e^{\mu(\mathbf{x}(t))}} = (1 + e^{-\mu(\mathbf{x}(t))})^{-1}. \quad (4.22)$$

A equação acima pode ser diferenciada através da regra da cadeia. Logo:

$$\begin{aligned}\frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} &= -(1 + e^{-\mu(\mathbf{x}(t))})^{-2} (-e^{-\mu(\mathbf{x}(t))}) \\ &= \frac{e^{-\mu(\mathbf{x}(t))}}{(1 + e^{-\mu(\mathbf{x}(t))})^2} \\ &= \underbrace{\left( \frac{1}{1 + e^{-\mu(\mathbf{x}(t))}} \right)}_{f(\mu(\mathbf{x}(t)))} \underbrace{\left( \frac{e^{-\mu(\mathbf{x}(t))}}{1 + e^{-\mu(\mathbf{x}(t))}} \right)}_{1 - f(\mu(\mathbf{x}(t)))} \\ &= f(\mu(\mathbf{x}(t))) [1 - f(\mu(\mathbf{x}(t)))].\end{aligned}\quad (4.23)$$

Consequentemente, a regra de aprendizado do GLVQ não linear é obtida como:

$$\begin{aligned}\mathbf{w}^+(t+1) &= \mathbf{w}^+(t) + \alpha(t) \left[ \frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} \right] \frac{4d_q^-}{(d_q^+ + d_q^-)^2} (\mathbf{x}(t) - \mathbf{w}^+(t)) \\ &= \mathbf{w}^+(t) + \alpha(t) \{f(\mu(\mathbf{x}(t))) [1 - f(\mu(\mathbf{x}(t)))]\} \frac{4d_q^-}{(d_q^+ + d_q^-)^2} (\mathbf{x}(t) - \mathbf{w}^+(t)),\end{aligned}\quad (4.24)$$

$$\begin{aligned}\mathbf{w}^-(t+1) &= \mathbf{w}^-(t) - \alpha(t) \left[ \frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} \right] \frac{4d_q^+}{(d_q^+ + d_q^-)^2} (\mathbf{x}(t) - \mathbf{w}^-(t)) \\ &= \mathbf{w}^-(t) - \alpha(t) \{f(\mu(\mathbf{x}(t))) [1 - f(\mu(\mathbf{x}(t)))]\} \frac{4d_q^+}{(d_q^+ + d_q^-)^2} (\mathbf{x}(t) - \mathbf{w}^-(t)).\end{aligned}\quad (4.25)$$

#### 4.2.2.2 Relevance Learning Vector Quantization (RLVQ)

O algoritmo *Relevance LVQ* (RLVQ) fora o primeiro método proposto para a determinação automática da relevância dos atributos dos vetores de entrada. Conforme ilustrado na Figura 12, o RLVQ e suas respectivas variantes são pertencentes à subdivisão denominada aprendizado de distâncias.

Na subdivisão aprendizado de distâncias, a métrica de dissimilaridade utilizada no campo receptivo dos protótipos é adaptada durante o treinamento dos algoritmos. Pode-se exemplificar dois benefícios em utilizar dissimilaridades adaptativas (BOJER *et al.*, 2001):

1. A utilização de fatores de relevância eleva as taxas de acertos destas variantes LVQ e adapta a métrica aos dados específicos utilizados no treinamento;
2. Este método realiza atenuação dos atributos que não contribuem para as taxas de classificação.

O algoritmo RLVQ é, então, apresentado a seguir:

Sejam  $\mathbf{x}, \mathbf{w} \in \mathbb{R}^p$  dois vetores escolhidos anteriormente, o vetor de pesos  $\lambda \in \mathbb{R}^p$  onde  $\lambda_1, \dots, \lambda_p \geq 0$  é inserido no cálculo da distância euclidiana entre  $\mathbf{x}$  e  $\mathbf{w}$  conforme mostrado abaixo:

$$\|\mathbf{x} - \mathbf{w}\|_{\lambda}^2 = \sum_{l=1}^p \lambda_l (\mathbf{x}_l - \mathbf{w}_l)^2. \quad (4.26)$$

A métrica ilustrada acima é denominada de distância euclidiana ponderada quadrática e é utilizada na construção do campo receptivo de cada protótipo  $\mathbf{w}_i$  para este algoritmo. Conseqüentemente, a Equação 4.1 é modificada e obtêm-se:

$$R_{\lambda}^i = \{\mathbf{x} \in \mathbf{X} \mid \forall \mathbf{w}_j (j \neq i) \rightarrow \|\mathbf{x} - \mathbf{w}_i\|_{\lambda} < \|\mathbf{x} - \mathbf{w}_j\|_{\lambda}\}. \quad (4.27)$$

Após a formulação da métrica a ser utilizada, o treinamento do algoritmo é utilizado com o objetivo, além da atualização das posições dos vetores protótipos, de encontrar os pesos  $\lambda_i$  apropriados aos dados.

Logo, a realização de uma  $t$ -ésima iteração do RLVQ é descrita a seguir:

**Passo 1** - Determinação do vetor vencedor  $\mathbf{w}_c(t)$  ao qual  $\mathbf{x}(t) \in R_{\lambda}^c$ .

**Passo 2** - Realizar a atualização de todas as componentes do vetor de ponderação conforme o modelo abaixo:

$$\lambda_l(t+1) = \begin{cases} \max(\lambda_l(t) - \alpha(t) |x_l(t) - w_{c_l}(t)|, 0), & \text{se } y(t) = c(\mathbf{w}_c(t)); \\ \lambda_l(t) + \alpha(t) |x_l(t) - w_{c_l}(t)|, & \text{caso contrário.} \end{cases} \quad (4.28)$$

**Passo 3** - Realizar normalização do vetor de ponderação conforme mostrado abaixo:

$$\lambda_l(t+1) = \lambda_l(t+1) / |\boldsymbol{\lambda}(t+1)|. \quad (4.29)$$

**Passo 4** - Realizar a atualização do vetor protótipo:

$$\mathbf{w}_c(t+1) = \begin{cases} \mathbf{w}_c(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{w}_c(t)], & y(t) = c(\mathbf{w}_c(t)); \\ \mathbf{w}_c(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{w}_c(t)], & \text{caso contrário.} \end{cases} \quad (4.30)$$

O algoritmo RLVQ pode ser aplicado em três metodologias distintas. Na primeira, o treinamento do RLVQ é realizado após a realização de poucas épocas do LVQ. Na segunda, o treinamento do RLVQ é realizado após um treinamento completo do LVQ. Na terceira, o treinamento do RLVQ é realizado sem uma etapa prévia do LVQ.

#### 4.2.2.3 Generalized Relevance LVQ (GRLVQ) e Local Generalized Relevance LVQ (LGRLVQ)

O algoritmo *Generalized Relevance* LVQ, formulado por Hammer e Villmann (2002), apresenta uma adaptação da métrica de dissimilaridade na construção dos campos receptivos do algoritmo GLVQ. Assim como o RLVQ, o algoritmo GRLVQ utiliza a distância euclidiana ponderada apresentada na Equação 4.26 para obter os campos receptivos dos protótipos conforme a Equação 4.27.

Assim como no algoritmo GLVQ, os vetores  $\mathbf{w}^+$  e  $\mathbf{w}^-$  no GRLVQ representam os protótipos mais próximos à amostra  $\mathbf{x}$  em que caracterizam, respectivamente, o mais próximo correto e o mais próximo incorreto. Logo, o índice de distância relativa  $\mu(\mathbf{x})$  para este algoritmo se torna:

$$\mu(\mathbf{x}) = \frac{d_{\lambda_q}^+ - d_{\lambda_q}^-}{d_{\lambda_q}^+ + d_{\lambda_q}^-}, \quad (4.31)$$

onde  $d_{\lambda_q}^+$  e  $d_{\lambda_q}^-$  são distâncias ponderadas quadráticas de  $\mathbf{x}$  para  $\mathbf{w}^+$  e  $\mathbf{w}^-$ , respectivamente.

A função custo do GRLVQ a ser minimizada é bastante semelhante ao de sua variante anterior (GLVQ) e é mostrada abaixo:

$$f_{GRLVQ} = \sum_{i=1}^N f(\mu(\mathbf{x}_i)), \quad (4.32)$$

esta função é estabelecida pelo acúmulo do índice distância relativa  $\mathbf{x}$  de todos os padrões de entrada de treinamento. Porém, enquanto o GLVQ também possui sua versão linear, a função de transformação  $f$  sobre este índice é exclusivamente não linear para o algoritmo GRLVQ. A função escolhida para  $f$  é a função logística  $sgd(a) = (1 + \exp(-a))^{-1}$ .

Durante a realização de uma  $t$ -ésima iteração, calcula-se primeiramente o índice de distância relativa  $\mu(\mathbf{x}(t))$  associada ao padrão de entrada  $\mathbf{x}(t)$ . Posteriormente, calcula-se o valor do vetor de ponderação para a próxima iteração conforme mostrado abaixo:

$$\boldsymbol{\lambda}(t+1) = \boldsymbol{\lambda}(t) - \varepsilon_{\lambda}(t) \text{sgd}' \left( \frac{d_{\lambda_q}^-}{(d_{\lambda_q}^+ + d_{\lambda_q}^-)^2} (\mathbf{x}(t) - \mathbf{w}^+)^2 - \frac{d_{\lambda_q}^+}{(d_{\lambda_q}^+ + d_{\lambda_q}^-)^2} (\mathbf{x}(t) - \mathbf{w}^-)^2 \right), \quad (4.33)$$

onde  $\varepsilon_{\lambda}(t)$  representa um fator de ganho e deve ser menor que a escala de  $\alpha(t)$  em uma ordem. Após o cálculo da Equação 4.33, o vetor de ponderação é submetido às restrições  $\lambda_l \geq 0$  e  $\sum_l^p \lambda_l = 1$  através da normalização explícita a cada iteração.

Após a determinação do vetor de ponderação  $\boldsymbol{\lambda}$  para a próxima iteração, calcula-se a atualização dos vetores  $\mathbf{w}^+$  e  $\mathbf{w}^-$  conforme exposto abaixo:

$$\mathbf{w}^+(t+1) = \mathbf{w}^+(t) + \alpha(t) \frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} \frac{4d_{\lambda_q}^-}{(d_{\lambda_q}^+ + d_{\lambda_q}^-)^2} (\mathbf{x}(t) - \mathbf{w}^+(t)) \quad (4.34)$$

$$\mathbf{w}^-(t+1) = \mathbf{w}^-(t) - \alpha(t) \frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} \frac{4d_{\lambda_q}^+}{(d_{\lambda_q}^+ + d_{\lambda_q}^-)^2} (\mathbf{x}(t) - \mathbf{w}^-(t)) \quad (4.35)$$

Já o algoritmo *Local Generalized Relevance* LVQ, formulado por Hammer e Villmann (2002), apresenta uma adaptação da métrica de dissimilaridade na construção dos campos receptivos do algoritmo GRLVQ inserindo uma associação distinta a cada protótipo. Logo, o vetor de ponderações apresentado na formulação do algoritmo RLVQ se torna uma matriz formada por um vetor de ponderações para cada protótipo conforme mostrado abaixo:

$$\boldsymbol{\lambda} = \left[ \boldsymbol{\lambda}_{1[1 \times p]} \mid \cdots \mid \boldsymbol{\lambda}_{K[1 \times p]} \right]. \quad (4.36)$$

No algoritmo LGRLVQ, a distância euclidiana ponderada apresentada na Equação 4.26 se torna:

$$\|\mathbf{x} - \mathbf{w}\|_{\boldsymbol{\lambda}_i}^2 = \sum_{l=1}^p \lambda_{li} (\mathbf{x}_l - \mathbf{w}_l)^2. \quad (4.37)$$

Consequentemente, o cálculo do campo receptivo dos protótipos mostrado na Equação 4.27 é modificado e obtém-se:

$$R_{\boldsymbol{\lambda}}^i = \{\mathbf{x} \in \mathbf{X} \mid \forall \mathbf{w}_j (j \neq i) \rightarrow \|\mathbf{x} - \mathbf{w}_i\|_{\boldsymbol{\lambda}_i} < \|\mathbf{x} - \mathbf{w}_j\|_{\boldsymbol{\lambda}_j}\}. \quad (4.38)$$

Logo, este algoritmo considera informações locais do espaço de características e, para isto, utiliza os perfis de relevância locais. Isto é evidenciado a cada iteração do algoritmo visto que apenas os

vetores de ponderações vinculados aos protótipos  $\mathbf{w}^+$  e  $\mathbf{w}^-$  são atualizados. Consequentemente, este algoritmo permite decisões de contorno mais complexas do que as realizadas por métricas de parâmetros globais.

O treinamento do LGRLVQ também é realizado pelos vetores  $\mathbf{x}$ ,  $\mathbf{w}^+$  e  $\mathbf{w}^-$  que são atualizados a cada iteração. O cálculo do índice de distância relativa  $\mu(\mathbf{x})$  é mostrado abaixo:

$$\mu(\mathbf{x}) = \frac{d_{\lambda_{\mathbf{w}^+q}}^+ - d_{\lambda_{\mathbf{w}^-q}}^-}{d_{\lambda_{\mathbf{w}^+q}}^+ + d_{\lambda_{\mathbf{w}^-q}}^-}, \quad (4.39)$$

Então, a função custo deste algoritmo a ser minimizada é apresentada abaixo:

$$f_{LGRLVQ} = \sum_{i=1}^N sgd \left( \frac{d_{\lambda_{\mathbf{w}^+q}}^+ - d_{\lambda_{\mathbf{w}^-q}}^-}{d_{\lambda_{\mathbf{w}^+q}}^+ + d_{\lambda_{\mathbf{w}^-q}}^-} \right). \quad (4.40)$$

Os vetores de ponderação  $\lambda_{\mathbf{w}^+}$  e  $\lambda_{\mathbf{w}^-}$  são atualizados como:

$$\lambda_c(t+1) = \lambda_c(t) - \varepsilon_\lambda(t) sgd' \left( \frac{d_{\lambda_{\mathbf{w}^-q}}^-}{(d_{\lambda_{\mathbf{w}^+q}}^+ + d_{\lambda_{\mathbf{w}^-q}}^-)^2} (\mathbf{x}(t) - \mathbf{w}^+)^2 - \frac{d_{\lambda_{\mathbf{w}^+q}}^+}{(d_{\lambda_{\mathbf{w}^+q}}^+ + d_{\lambda_{\mathbf{w}^-q}}^-)^2} (\mathbf{x}(t) - \mathbf{w}^-)^2 \right), \quad (4.41)$$

em que  $c \in \{\lambda_{\mathbf{w}^+}, \lambda_{\mathbf{w}^-}\}$  e, assim como no algoritmo GRLVQ, estes vetores são submetidos às restrições  $\lambda_{c_l} \geq 0$  e  $\sum_l^p \lambda_{c_l} = 1$  através da normalização explícita a cada iteração. Já  $\varepsilon_\lambda$  representa o passo de atualização dos vetores de ponderação e deve ser menor que o passo de aprendizagem  $\alpha$  em uma ordem de grandeza.

Após a determinação do vetores de ponderação  $\lambda_{\mathbf{w}^+}$  e  $\lambda_{\mathbf{w}^-}$  para a próxima iteração, calcula-se a atualização dos vetores  $\mathbf{w}^+$  e  $\mathbf{w}^-$  conforme exposto abaixo:

$$\mathbf{w}^+(t+1) = \mathbf{w}^+(t) + \alpha(t) \frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} \frac{4d_{\lambda_{\mathbf{w}^-q}}^-}{(d_{\lambda_{\mathbf{w}^+q}}^+ + d_{\lambda_{\mathbf{w}^-q}}^-)^2} (\mathbf{x}(t) - \mathbf{w}^+(t)), \quad (4.42)$$

$$\mathbf{w}^-(t+1) = \mathbf{w}^-(t) - \alpha(t) \frac{\partial f(\mu(\mathbf{x}(t)))}{\partial \mu(\mathbf{x}(t))} \frac{4d_{\lambda_{\mathbf{w}^+q}}^+}{(d_{\lambda_{\mathbf{w}^+q}}^+ + d_{\lambda_{\mathbf{w}^-q}}^-)^2} (\mathbf{x}(t) - \mathbf{w}^-(t)). \quad (4.43)$$

### 4.2.3 Algoritmos de Maximização da Verossimilhança

Após o desenvolvimento dos algoritmos heurísticos propostos por Teuvo Kohonen e dos primeiros algoritmos da família de maximização da margem de separação, iniciou-se o desenvolvimento de algoritmos LVQ da família de maximização da verossimilhança. A primeira

variante desta família é a *Robust Soft Learning Vector Quantization* (RSLVQ) e foi proposta por Seo e Obermayer (2003) e Seo *et al.* (2003).

Conforme ilustrado na Figura 14, os algoritmos da família maximização da verossimilhança são subdivididos nos segmentos espaços dos dados, similaridades e dissimilaridades e, por último, kernelizados.

Os algoritmos desta família não serão utilizados no desenvolvimento deste trabalho. Conseqüentemente, apenas o algoritmo RSLVQ será brevemente explorado com o objetivo de apresentação desta família de algoritmos LVQ.

Primeiramente, assume-se uma que cada classe do conjunto de dados possa ser descrita por uma Função de Densidade de Probabilidade (FDP). Logo, considera-se que o conjunto de dados possa ser modelado como um modelo de misturas de gaussianas. Então, a FDP de uma amostra é gerada por modelo de mistura de gaussianas da classe correta e é comparada com a FDP desta amostra gerada pelos modelos de misturas de gaussianas das classes incorretas.

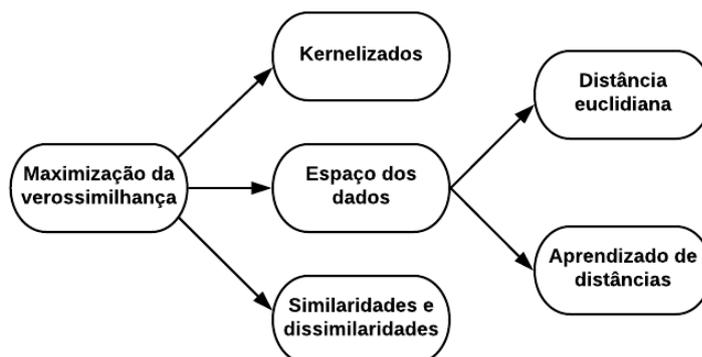
Esta família de algoritmos LVQ é inspirada na maximização do logaritmo da proporção entre o modelo correto de mistura de gaussianas e os modelos incorretos de misturas de gaussianas.

Logo, seja a matriz de protótipos  $\mathbf{W}$  com rótulos conhecidos, a FDP de uma amostra  $\mathbf{x}$  é calculada como:

$$p(\mathbf{x} | \mathbf{W}) = \sum_{y=1}^C \sum_{\{j:c(\mathbf{w}_j=y)\}} p(\mathbf{x} | j)P(j), \quad (4.44)$$

em que  $C$  é o número de classes e  $y$  é o rótulo das amostras geradas pela componente  $j$ .  $P(j)$  é a probabilidade das amostras geradas pela componente  $j$  da mistura e estas podem ser

Figura 14 – Família LVQ dos algoritmos de maximização da verossimilhança



igualmente escolhidas por cada protótipo  $\mathbf{w}_j$ .  $p(\mathbf{x} | j)$  é a FDP condicional da componente  $j$  gerar especificamente a amostra  $\mathbf{x}$  e o protótipo  $\mathbf{w}_j$ .

Os principais algoritmos desta família e seus respectivos autores são apresentados na Tabela 7. Os nomes destes algoritmos por extenso são apresentados na Tabela 8.

Para o algoritmo RSLVQ, é proposta a seguinte razão de verossimilhança como função custo a ser maximizada:

$$f_{RSLVQ} = \sum_{i=1}^N \log \left( \frac{p(\mathbf{x}_i, y | \mathbf{W})}{p(\mathbf{x}_i | \mathbf{W})} \right), \quad (4.45)$$

onde  $p(\mathbf{x}_i, y | \mathbf{W})$  é a FDP da amostra  $\mathbf{x}_i$  que é gerada pelo modelo de mistura da classe correta  $y$ , e  $p(\mathbf{x}_i | \mathbf{W})$  é a densidade de probabilidade total da amostra  $\mathbf{x}_i$ . Estas probabilidades são definidas a seguir:

$$p(\mathbf{x}_i, y_i | \mathbf{W}) = \sum_{\{j:c(\mathbf{w}_j=y)\}} p(\mathbf{w}_i | j)P(j) \quad (4.46)$$

$$p(\mathbf{x}_i | \mathbf{W}) = \sum_j p(\mathbf{x}_i | j)P(j). \quad (4.47)$$

As PDFs condicionais são definidas como  $p(\mathbf{w} | j) = K(j) \cdot \exp f(\mathbf{x}, \mathbf{w}_j, \sigma_j^2)$ . Logo:

$$f(\mathbf{x}, \mathbf{w}_j, \sigma_j^2) = -\frac{d(\mathbf{x}, \mathbf{w})}{2\sigma^2}, \quad (4.48)$$

em que  $d(\mathbf{x}, \mathbf{w})$  é a distância euclidiana entre  $\mathbf{x}$  e  $\mathbf{w}$ .

Por fim, outras variantes podem ser escolhidas a partir da escolha da métrica  $d$  definida na Equação 4.48.

Tabela 7 – Principais algoritmos de maximização da verossimilhança

Segmento	Métrica	Algoritmo	Autores
Espaço dos dados	Distância euclidiana	RSLVQ	Seo e Obermayer (2003) e Seo <i>et al.</i> (2003)
	Aprendizado de distâncias	MRSLVQ	Schneider <i>et al.</i> (2009)
		LMRSLVQ	Schneider <i>et al.</i> (2009)
Similaridades e dissimilaridades	-	RRSLVQ	Hammer <i>et al.</i> (2011)
Kernelizados	-	KRSLVQ	Hofmann <i>et al.</i> (2013)

Fonte: o autor.

Tabela 8 – Nomes por extenso dos principais algoritmos de maximização da verossimilhança

Algoritmo	Nome por extenso	Algoritmo	Nome por extenso
RSLVQ	<i>Robust Soft LVQ</i>	RRSLVQ	<i>Relational Robust Soft LVQ</i>
MRSLVQ	<i>Matrix Robust Soft LVQ</i>	KRSLVQ	<i>Kernel Robust Soft LVQ</i>
LMRSLVQ	<i>Local Matrix Robust Soft LVQ</i>		

Fonte: o autor.

### 4.3 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: LVQ1, OLVQ1, LVQ2.1, LVQ3, GLVQ, GRLVQ e LGRLVQ.

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.

## 5 METODOLOGIA DOS EXPERIMENTOS

Neste capítulo, são apresentadas as técnicas utilizadas para analisar o conjunto de dados e comparar os resultados encontrados na etapa de análise de agrupamentos dos conjuntos de treinamento. Além disso, são apresentadas as técnicas utilizadas para comparar os classificadores supervisionados da família LVQ na tarefa de detecção de falhas nos motores de indução trifásicos.

### 5.1 Banco de Dados

Como descrito no Capítulo 2, o banco de dados reais utilizado nesta dissertação possui 294 amostras representantes de possíveis condições de operação de um motor trifásico. No presente trabalho, esta configuração de banco de dados é definida como *banco de dados 1* (BD1) e é composta por 42 amostras de normalidade, 126 amostras de falhas de alta impedância e 126 amostras de baixa impedância. Percebe-se facilmente que o BD1 é desequilibrado, dado que as amostras que representam condições faltosas (85,7% do total) equivalem a seis vezes a quantidade de amostras da classe de motores em operação normal.

Vale mencionar que o BD1 não reflete a realidade em cenários industriais, que são caracterizados pela escassez de exemplos de falhas em comparação com casos de normalidade. Este desequilíbrio dificulta a tarefa de identificação de falhas por inserir um viés de classificar todas as amostras de teste como casos de falha.

Assim, com o objetivo de solucionar os efeitos indesejados do viés ao utilizar o BD1, foram geradas amostras artificiais de condições de normalidade do motor utilizando o método de inserção de ruído, originalmente proposto por Matsuoka (1992), descrito a seguir:

**Passo 1** - Seleção aleatória de um exemplo pertencente à classe de funcionamento normal;

**Passo 2** - Adiciona-se um valor aleatório de distribuição uniforme com intervalo  $[-0,01 \times x, 0,01 \times x]$  a cada atributo do exemplo selecionado.

Estes passos foram realizados até que fossem gerados 84 exemplos artificiais de condições de motores em funcionamento normal. Conseqüentemente, o novo banco de dados, denominado de *banco de dados 2* (BD2), é composto por 378 amostras contendo 126 amostras de normalidade, 126 amostras de falhas de alta impedância e 126 amostras de falha de baixa impedância.

## 5.2 Heurística Proposta para Iniciação dos Vetores Protótipos

Como visto anteriormente no Capítulo 4, os classificadores LVQ tentam aproximar as fronteiras teóricas bayesianas e podem tratar, diretamente, problemas de classificação multi-classes. Porém, os algoritmos LVQ heurísticos sofrem de problemas de sensibilidade na iniciação dos protótipos, convergência lenta, e instabilidades (KOHONEN *et al.*, 2001). Soluções de convergência e estabilidade são propostas nos algoritmos LVQ baseados em maximização da margem de separação e em maximização da verossimilhança. Já o problema de sensibilidade de iniciação dos protótipos é solucionado através da modelagem de heurísticas.

Uma popular estratégia de iniciação dos protótipos é colocá-los próximos aos centros de massas de cada classe do conjunto de dados. Uma vez que reproduzem uma representação do conjunto de dados, os protótipos devem representar suas respectivas classes assumindo posições típicas da classe no espaço de características.

Esta heurística de iniciação dos protótipos pode ser realizada através do aprendizado não supervisionado de cada classe do conjunto de treinamento do banco de dados. Assim, a presente dissertação propõe a utilização do algoritmo  $K$ -médias (descrito na Seção 3.2.1.1) devido ser um modelo simples<sup>1</sup> com a finalidade de provar a viabilidade da heurística proposta<sup>2</sup>.

Com o sucesso deste algoritmo mais simples, pode-se induzir que resultados ainda melhores serão encontrados ao utilizar redes neurais como a *Self-Organizing Maps* (SOM) e a *Neural Gas* (NG).

Porém, a dificuldade desta abordagem configura-se na definição da quantidade de protótipos a ser utilizada para representar cada classe. Deseja-se que a quantidade de protótipos  $K$  seja bastante inferior à quantidade de amostras do conjunto de dados, porém é uma tarefa desafiadora definir um limiar inferior que seja suficiente para representar os dados de maneira satisfatória. No presente trabalho, a quantidade total de protótipos  $K$  é obtida através da soma da quantidade de protótipos da classe normal ( $K_{NO}$ ), com a quantidade de protótipos da classe falha de alta impedância ( $K_{AI}$ ) e, por último, com a quantidade de protótipos da classe falha de baixa impedância ( $K_{BI}$ ).

Nesta dissertação, propõe-se utilizar os índices de validação relativos e os critérios de informação, ambos descritos na Seção 3.3, com o objetivo de determinar a quantidade ótima de protótipos por classe para o problema de detecção de curto-cirto em motores trifásicos.

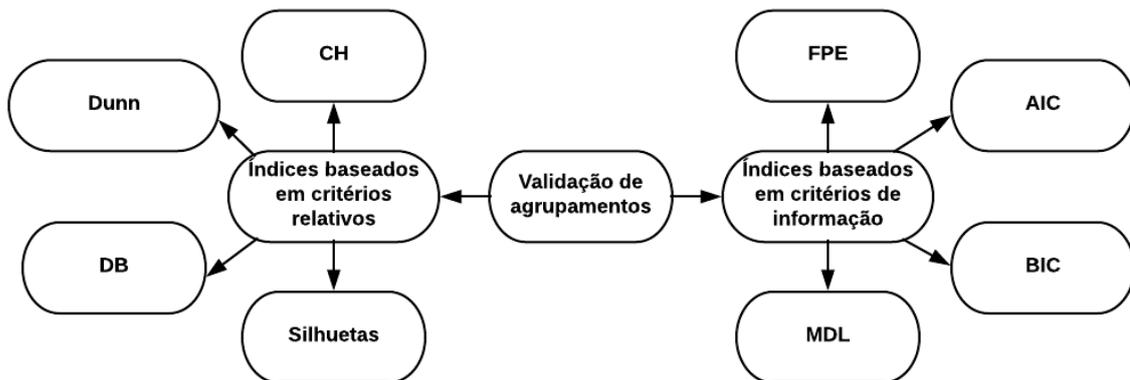
<sup>1</sup> Não apresenta função de vizinhança em sua modelagem.

<sup>2</sup> Realizar iniciação de protótipos adequada para a classificação de um motor trifásico.

Como visto no Capítulo 3, os critérios de validação de agrupamentos utilizam diferentes variáveis no cálculo de suas respectivas métricas. Consequentemente, é natural encontrar valores distintos reportados como a melhor configuração de representação.

Considerando este fato, o presente trabalho propõe a definição da melhor quantidade de protótipos através da moda dos valores reportados pelas métricas apresentadas na Figura 15.

Figura 15 – Critérios de validação propostos para utilização



Fonte: o autor.

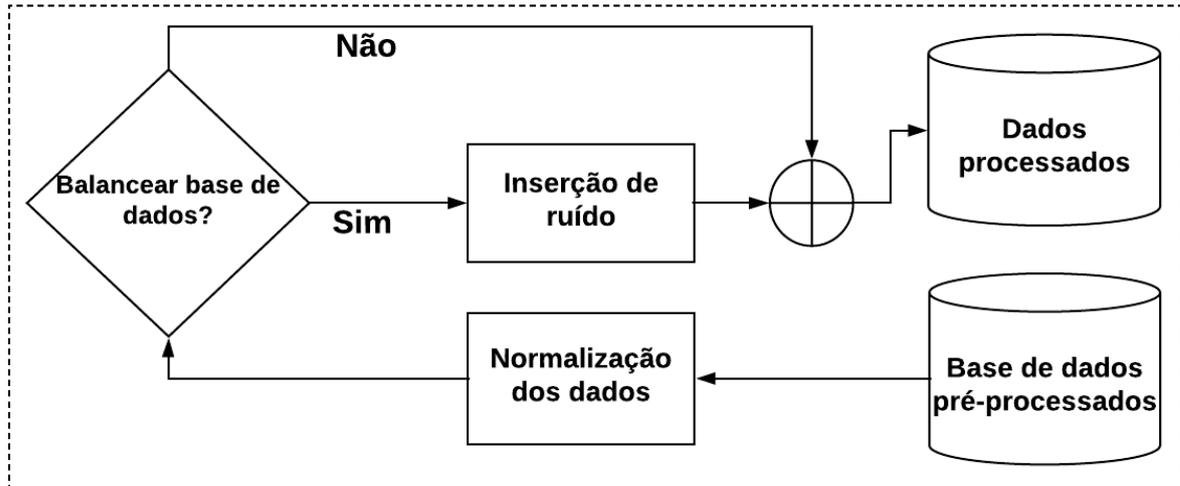
### 5.3 Metodologia de Comparação Entre Classificadores LVQ

Os experimentos realizados nesta dissertação para comparar o desempenho de algoritmos da família de classificadores LVQ quanto à sua capacidade de detectar falhas, são ilustrados nas Figuras 16 e 17 e são descritos a seguir.

O conjunto de dados utilizado neste trabalho é preparado nas etapas ilustradas na Figura 16 e descritas nas Subseções 5.3.1 e 5.3.2. Após a preparação destes dados, as etapas da metodologia proposta são executadas em conformidade da sequência ilustrada na Figura 17. Ao final de todos os passos desta metodologia, ocorre uma realização de cada algoritmo paralelamente. Para cada variante LVQ implementada nesta dissertação, foram executadas 100 realizações da metodologia proposta.

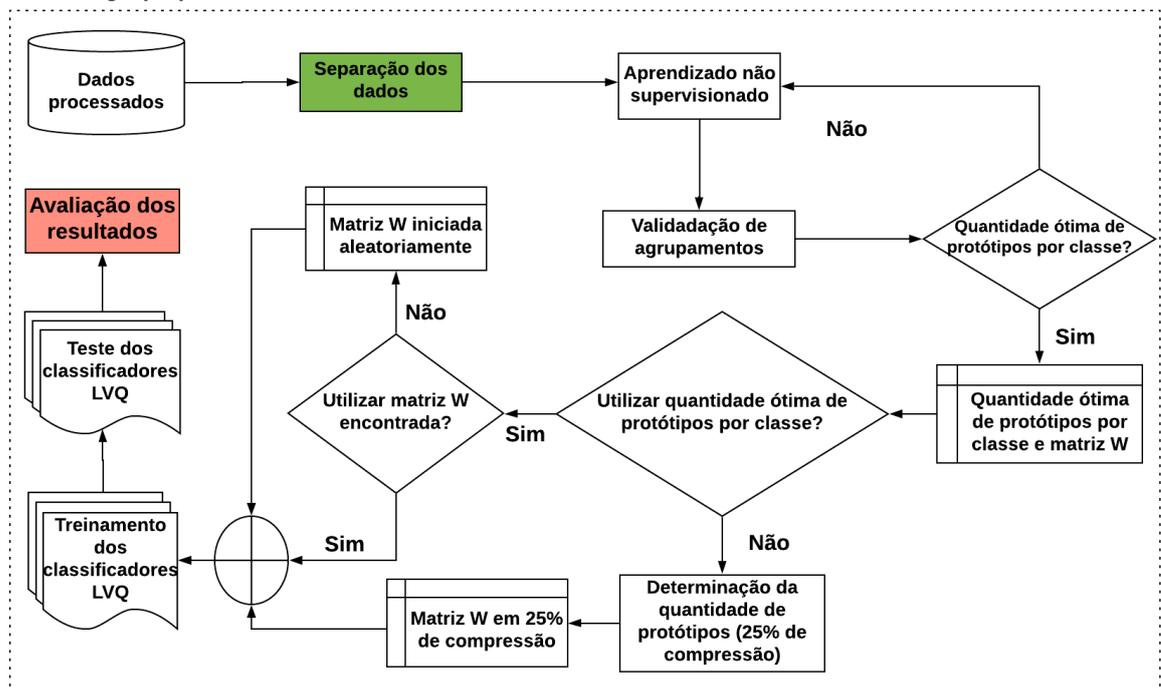
Após as 100 realizações de cada classificador, as estatísticas de máximo, mínimo, média, mediana e desvio padrão sobre as taxas de acerto na detecção de falhas são aferidas com o objetivo de determinar a melhor configuração LVQ para o problema abordado. No próximo capítulo, estas estatísticas serão interpretadas através de matrizes de confusão, tabelas e *boxplots*.

Figura 16 – Diagrama da preparação dos dados

**Preparação dos dados**

Fonte: o autor.

Figura 17 – Diagrama da metodologia proposta

**Metodologia proposta**

Fonte: o autor.

**5.3.1 Banco de Dados Escolhido**

Inicialmente, o banco de dados a ser utilizado é selecionado. Nesta dissertação, pode-se optar entre o BD1 (que apresenta proporções desequilibradas entre amostras de motores sem falha, falha de alta impedância e falha de baixa impedância) e o BD2 (que apresenta proporção

equilibrada entre as três possíveis classes).

### 5.3.2 *Normalização dos Dados*

Em classificação de padrões, a etapa de normalização dos dados é indispensável para a atenuação do viés gerado pelas diferentes magnitudes dos atributos presentes nos dados. Os dados podem ser normalizados através de diversas formas. Dentre elas, as mais utilizadas são a normalização entre o intervalo  $[0, 1]$  e a z-escore, que representa os dados no intervalo  $[-1, 1]$ . Nesta dissertação, o banco de dados selecionado é normalizado através da normalização z-escore. Esta normalização é apresentada abaixo:

$$x_j = \frac{x_j - \bar{x}_j}{s_j}, \quad (5.1)$$

onde  $\bar{x}_j$  é a média amostral do atributo  $j$  e  $s_j$  é o desvio padrão amostral deste mesmo atributo. Ao final desta transformação, a distribuição de cada atributo terá média zero e variância unitária.

### 5.3.3 *Separação dos Dados*

Nesta etapa, as amostras do banco de dados selecionado são divididas aleatoriamente entre os subconjuntos treinamento e teste. Considerando o conjunto de dados selecionado com  $N$  amostras, a separação realizada obedece a proporção  $0,8N$  para a etapa de treinamento e  $0,2N$  para a etapa de teste. Por fim, os subconjuntos são armazenados com o propósito de serem utilizados em todos classificadores LVQ a fim de que se garanta comparação justa destes.

Após a realização desta tarefa, é feita a etapa de aprendizado não supervisionado do conjunto de treinamento.

### 5.3.4 *Aprendizado Não Supervisionado do Conjunto de Treinamento*

Após a separação dos dados, o conjunto de treinamento é processado pelo algoritmo  $K$ -médias sob diferentes valores de  $K$  clusters com o objetivo da obtenção da melhor representação destes dados por protótipos.

Inicialmente, as amostras de cada classe existente no conjunto de treinamento são separadas. Assim, o conjunto de treinamento é separado entre as classes normal (NO), alta impedância (AI) e baixa impedância (BI). Devido à formação aleatória do conjunto de treinamento, a

quantidade de amostras por classe *a priori* é desconhecida. Pode-se apenas afirmar que:

$$N_{Tr} = N_{NO_{Tr}} + N_{HI_{Tr}} + N_{BI_{Tr}}, \quad (5.2)$$

onde  $N_{Tr}$  é a quantidade de amostras de treinamento,  $N_{NO_{Tr}}$  é a quantidade de amostras de treinamento que pertencem à classe normal,  $N_{AI_{Tr}}$  é a quantidade de amostras de treinamento que pertencem à classe falha de alta impedância e  $N_{BI_{Tr}}$  é a quantidade de amostras de treinamento que pertencem à classe falha de baixa impedância.

Após a separação do conjunto de treinamento entre as classes conforme a Equação (5.2), cada subconjunto é, então, submetido ao método proposto descrito abaixo:

**Passo 01** - Seleciona-se o subconjunto (classe) a ser utilizado neste procedimento;

**Passo 02** - Definição do valor do hiperparâmetro  $k$ ;

**Passo 03** - O subconjunto selecionado é processado pelo algoritmo  $K$ -médias com  $k$  partições.

O SEQQ é calculado e armazenado. Este procedimento é executado 10 vezes;

**Passo 04** - A matriz de pesos da realização do  $K$ -médias com  $k$  partições que apresentou o menor SEQQ e suas respectivas células de Voronoi são armazenadas. Esta matriz de pesos passa a ser definida como a melhor matriz de pesos do subconjunto selecionado ao utilizar  $k$  partições;

**Passo 05** - A melhor matriz de pesos e suas respectivas células de Voronoi armazenadas são submetidas às técnicas de validação de agrupamentos ilustradas na Figura 15;

**Passo 06** - Os resultados encontrados pelas métricas de validação de agrupamentos são armazenados;

**Passo 07** - Finalmente, com a matriz de pesos e índices de validação de agrupamentos calculados e armazenados, descartam-se as células de Voronoi e o SEQQ;

**Passo 08** - Os Passos 2 - 7 são repetidos para diferentes valores do hiperparâmetro  $k$  para todos os valores do intervalo  $2 \leq k \leq 10$ ;

**Passo 09** - Após calcular o índices de validação para todos os hiperparâmetros, verifica-se o  $K_{opt}$  sugerido por cada índice. Assim, obtém-se um vetor de sugestões a ser tratado no próximo passo;

**Passo 10** - O valor de  $k$ , que está contido no vetor de sugestões, a ser definido como  $K_{opt}$  do subconjunto selecionado será determinado pela moda dos valores indicados pelos diferentes índices de validação que são ilustrados na Figura 15. Em caso de empate entre

os candidatos a  $K_{opt}$  em qualquer classe, será escolhida a sugestão de menor quantidade de protótipos;

**Passo 11** - Finalmente, obtém-se a matriz de pesos associada ao valor de  $K_{opt}$  encontrado para o subconjunto selecionado.

Conforme apresentado na Seção 3.3, o vetor de sugestões  $\delta$  utilizado no Passo 10 é definido como:

$$\delta = \left[ \begin{array}{l} \min(DB), \max(Dunn), \max(S), \max(CH), \\ \min(AIC), \min(BIC), \min(FPE), \min(MDL) \end{array} \right], \quad (5.3)$$

de tal forma que o valor de  $k$  sugerido pelo índice DB é o que apresentou o menor valor obtido pelo cálculo de sua métrica no intervalo  $[2, 10]$ . Já os índices Dunn, Silhuetas e CH utilizam os valores máximos em suas respectivas métricas para este mesmo intervalo. Por fim, os critérios de informação AIC, BIC, FPE e MDL utilizam os valores mínimos obtidos através dos cálculos de suas respectivas métricas no intervalo  $[1, 10]$ .

Após a conclusão do procedimento proposto para os três subconjuntos, concatenam-se as matrizes de pesos associadas ao  $K_{opt}$  de cada subconjunto conforme a equação abaixo:

$$\mathbf{W} = \left[ \mathbf{W}_{NO}_{[p \times K_{optNO}]} \mid \mathbf{W}_{HI}_{[p \times K_{optHI}]} \mid \mathbf{W}_{BI}_{[p \times K_{optBI}]} \right]_{[p \times N_{Tr_{final}}]}, \quad (5.4)$$

em que a  $p$  representa a quantidade de atributos e  $N_{Tr_{final}}$  representa a quantidade de vetores protótipos da matriz  $\mathbf{W}$ .

Após da determinação da quantidade ótima de vetores protótipos e sua matriz  $\mathbf{W}$  associada, dá-se início à etapa de treinamento dos classificadores LVQ detalhados neste trabalho.

### 5.3.5 Treinamento dos Classificadores

Após a determinação da matriz de protótipos  $\mathbf{W}$  e de posse dos rótulos das classes que cada vetor protótipo representa, estes vetores protótipos serão atualizados através da etapa de treinamento dos classificadores supervisionados LVQ. Neste tipo de aprendizado, todos os classificadores considerados nesta etapa utilizam os rótulos dos vetores protótipos encontrados na etapa anterior.

Almejando a análise da relevância da quantidade de protótipos e a iniciação adequada destes, são consideradas as seguintes opções de treinamento dos classificadores LVQ:

**Treinamento 01** - Utiliza a quantidade ótima de protótipos definida na etapa descrita na Seção 5.3.4 e sua matriz  $\mathbf{W}$  associada como posições iniciais dos protótipos para todos os classificadores LVQ a serem comparados;

**Treinamento 02** - Utiliza a quantidade ótima de protótipos definida na etapa descrita na Seção 5.3.4 e sua matriz  $\mathbf{W}$  associada com posições iniciais aleatórias dos protótipos para todos os classificadores LVQ a serem comparados;

**Treinamento 03** - Define a quantidade de protótipos como 25% da quantidade de amostras de cada classe do conjunto de treinamento. A matriz  $\mathbf{W}$  associada é iniciada utilizando a configuração de menor SEQQ (entre 10 realizações) do algoritmo  $K$ -médias ao utilizar  $K$  protótipos correspondentes aos 25% de compressão.

Conforme explorado no capítulo anterior, as características dos classificadores LVQ são listadas na Tabela 9. A partir desta tabela, pode-se verificar que apenas os classificadores LVQ, OLVQ1 e GLVQ não possuem restrições durante o treinamento.

Os modelos de algoritmos LVQ utilizados nos três modelos de treinamentos propostos para classificação dos bancos de dados apresentados na Seção 5.1 são descritos a seguir.

Em todos classificadores presentes neste trabalho, foram utilizados 500 épocas na etapa de treinamento. Exceto o algoritmo OLVQ1 que fora treinado com passo de aprendizado inicial  $\alpha_{inicial} = 0,3$ , o restante dos algoritmos foram treinados utilizando decaimento linear do passo de aprendizado com  $\alpha_{inicial} = 0,1$  e  $\alpha_{final} = 0,001$ . Já o ganho  $\varepsilon$  utilizado no algoritmo no LVQ3 fora 0,5. Por último, o LGRLVQ fora treinado com passo de atualização dos vetores de ponderação onde  $\varepsilon_{\lambda_{inicial}} = 0,01$  e  $\varepsilon_{\lambda_{final}} = 0,0001$ .

Tabela 9 – Características dos classificadores LVQ envolvidos.

Classificador	Princípio Norteador	Parâmetros	Distância	Restrição
LVQ1	Heurístico	$\{\alpha, \mathbf{W}\}$	Euclidiana	-
LVQ 2.1	Heurístico	$\{\alpha, j, \mathbf{W}\}$	Euclidiana	$\min \left( \frac{d(\mathbf{x}, \mathbf{w}^+)}{d(\mathbf{x}, \mathbf{w}^-)}, \frac{d(\mathbf{x}, \mathbf{w}^-)}{d(\mathbf{x}, \mathbf{w}^+)} \right) < r, r = \frac{1-j}{1+j}$
LVQ 3	Heurístico	$\{\alpha, j, \varepsilon, \mathbf{W}\}$	Euclidiana	$\min \left( \frac{d(\mathbf{x}, \mathbf{w}^+)}{d(\mathbf{x}, \mathbf{w}^-)}, \frac{d(\mathbf{x}, \mathbf{w}^-)}{d(\mathbf{x}, \mathbf{w}^+)} \right) < s, s = \frac{1-\omega}{1+\omega}$
OLVQ1	Heurístico	$\{\alpha, \mathbf{W}\}$	Euclidiana	-
GLVQ	Maximização da margem de separação	$\{\alpha, \mathbf{W}\}$	Euclidiana	-
RLVQ	Maximização da margem de separação	$\{\alpha, \mathbf{W}\}$	Adaptativa	$\sum_{i=1}^p \lambda_i = 1$
GRLVQ	Maximização da margem de separação	$\{\alpha, \mathbf{W}\}$	Adaptativa	$\sum_{i=1}^p \lambda_i = 1$
LGRLVQ	Maximização da margem de separação	$\{\alpha, \varepsilon_{\lambda}, \mathbf{W}\}$	Adaptativa	$\sum_{i=1}^p \lambda_{k_i} = 1, \forall k \in \{1, \dots, K\}$

Neste trabalho, os modelos linear e não linear do algoritmo GLVQ foram utilizados na classificação do problema proposto com o objetivo de aferir os ganhos obtidos ao utilizar o algoritmo GLVQ em sua versão com maior peso computacional.

Os autores do algoritmo RLVQ, Bojer *et al.* (2001), recomendam que sejam utilizadas poucas épocas (ex: 100), do algoritmo LVQ1 como etapa de pré-processamento com o propósito de evitar instabilidades durante o treinamento do RLVQ. Logo, esta metodologia fora utilizada no treinamento do RLVQ no desenvolvimento deste trabalho e é apresentada como LVQ+RLVQ nos resultados apresentados no próximo capítulo.

Os autores dos algoritmos GRLVQ e LGRLVQ, Hammer e Villmann (2002), recomendam que sejam utilizados poucas épocas (ex:100) do algoritmo GLVQ não linear como etapa de pré-processamento com o propósito de evitar instabilidades durante o treinamento do GRLVQ e LGRLVQ. Logo, esta metodologia fora utilizada no treinamento dos algoritmos GRLVQ e LGRLVQ no desenvolvimento deste trabalho. Com o objetivo de avaliar os efeitos práticos do pré-processamento, utilizou-se os algoritmos GLVQ (sem a etapa prévia do GLVQ), GLVQ+GRLVQ e GLVQ+LGRLVQ no desenvolvimento deste trabalho.

### 5.3.6 *Teste dos Classificadores*

Finalmente, após o treinamento de todos os classificadores LVQ em condições semelhantes<sup>3</sup> para os três métodos de treinamento apresentados na seção anterior, dá-se início à etapa de testes dos classificadores.

Na fase de teste, a matriz de protótipos  $W$  é associada às amostras contidas no conjunto de teste através das regras de construções dos campos receptivos de cada algoritmo, ou seja, a saída de cada exemplo de teste será predita de acordo com a métrica de dissimilaridade particular de cada algoritmo utilizado.

Após encontrar a classe predita em cada classificador para todos os exemplos de teste, as taxas de acerto de cada classificador são estimadas. Para isso, as saídas dos classificadores são comparadas com os rótulos originais dos exemplos de teste. Caso a saída do classificador seja igual ao rótulo do exemplo de teste, uma classificação correta é computada. Caso contrário, um erro é computado para o exemplo atual de teste.

A taxa de acertos de um classificador é calculada através da razão entre a quantidade

<sup>3</sup> A cada nova realização entre as 100 repetições, todos os classificadores LVQ receberam a mesma separação aleatória do conjunto de dados.

de exemplos computados como corretos e o número total de exemplos de teste. A taxa de acertos pode ser representada através de estatísticas (exemplos: valores mínimo, máximo, média, mediana e desvio padrão) na construção de tabelas e diagramas de caixa (*boxplots*) para então ser comparada com a taxa de acertos de outros classificadores.

*Boxplots* (ou diagrama de caixas) são ferramentas utilizadas para visualização e comparação de desempenho entre classificadores onde apresentam as distribuições das taxas de acertos obtidas durante um conjunto de realizações de cada algoritmo. Estas ferramentas são capazes de evidenciar informações importantes reportadas através de estatísticas mostradas em cada quartil<sup>4</sup> e, também, ilustram desempenhos atípicos denominados de *outliers*.

No desenvolvimento desta dissertação, foram calculadas as taxas de acertos para a classificação entre as três classes definidas anteriormente. Logo, tem-se classificação ternária entre as classes normal, falha de alta impedância e falha de baixa impedância. Porém, o verdadeiro objetivo em aplicações é a detecção de falhas, ou seja, o diagnóstico de ocorrências de falhas de curto-circuito independentemente do estágio da falha.

Consequentemente, os rótulos preditos e os previamente conhecidos de falhas de AI e de BI são mapeados para um rótulo genérico denominado falha. Assim, tem-se classificação binária ao utilizar os rótulos normal e falha. Logo, as estatísticas de taxas de acertos de classificação ternária e de classificação binária são disponíveis para análise.

Além da taxa de acertos, a comparação de classificadores pode ser realizada através de matrizes de confusão. Esta matriz é ilustrada na Tabela 10 e é descrita a seguir.

Tabela 10 – Apresentação de uma matriz de confusão binária

		Classe verdadeira	
		Falha	Normal
Classe predita	Falha	Verdadeiro Positivo (VP)	Falso Positivo (FP)
	Normal	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Estas matrizes possuem dimensão  $[c \times c]$ , em que as linhas representam a classe  $C_i$  a qual o exemplo pertence, e as colunas representam as saídas estimadas (preditas) pelo classificador, ou seja, a qual classe a classe fora atribuída pelo classificador. Logo, a matriz de

<sup>4</sup> Quartis são medidas representadas em posições em uma dada distribuição. Estes dividem os resultados em 4 partes igualmente concentradas. Logo, o primeiro quartil indica que 25% das informações estão contidas até o fim deste. Já o segundo, indica que 50% dos exemplos estão abaixo do final deste, e assim sucessivamente.

confusão em um problema de classificação binária possui dimensão  $[2 \times 2]$  e em classificação ternária possui dimensão  $[3 \times 3]$ .

No presente problema abordado, a hipótese nula é definida pela condição do motor estar em estado de falha, ou seja, o termo positivo define o motor estar em falha e o termo negativo define o motor estar operando corretamente. Deste modo, o elemento Verdadeiro Positivo (VP) apresenta a quantidade de exemplos computados que pertencem à classe falha e foram classificados corretamente. Já o elemento Falso Positivo (FP), também definido como alarme falso, apresenta a quantidade de exemplos computados que pertencem à classe normal mas foram classificados como em estados de falha. Por sua vez, o elemento Falso Negativo (FN) apresenta a quantidade de exemplos computados que pertencem à classe falha mas foram classificados como em estados de normalidade de operação do motor. Finalmente, o elemento Verdadeiro Negativo (VN) apresenta a quantidade de exemplos computados que pertencem à classe normal e foram classificados corretamente.

A matriz de confusão é utilizada para, além de verificar a taxa geral de acertos do classificador, avaliar os diferentes erros presentes na classificação. Para este segundo objetivo, as métricas sensibilidade e especificidade são calculadas a partir dela.

A métrica sensibilidade reporta a relação entre verdadeiros positivos e falsos negativos, ou seja, calcula a capacidade do classificador detectar corretamente exemplos da classe de falha. Já a métrica especificidade reporta a relação entre verdadeiros negativos e falsos positivos, ou seja, calcula a capacidade do classificador identificar corretamente exemplos da classe normal. O cálculo destas métricas é mostrado abaixo:

$$\text{sensibilidade} = \frac{VP}{VP + FN}, \quad (5.5)$$

$$\text{especificidade} = \frac{VN}{VN + FP}. \quad (5.6)$$

Por exemplo, em uma classificação sem a presença de erros, têm-se  $FN = FP = 0$  e, conseqüentemente, sensibilidade e especificidade são igual à 1. Logo, em classificação binária é desejado que essas métricas possuam valores próximos de 1.

Finalmente, os classificadores LVQ apresentados na Tabela 9 serão comparados a partir de estatísticas extraídas das taxas de acerto e, posteriormente, analisa-se suas matrizes de confusão através das métricas sensibilidade e especificidade.

## 5.4 Conclusão

Neste capítulo, foram explicadas as definições dos bancos de dados utilizados e a heurística proposta de aprendizado não supervisionado utilizada como etapa de pré-processamento para iniciação dos protótipos. Os resultados encontrados da análise de agrupamentos serão discutidos no Capítulo 6.

Além disso, foram explicadas as metodologias utilizadas no treinamento dos classificadores e as ferramentas estatísticas utilizadas para comparação entre as variantes LVQ. Os resultados obtidos da comparação entre os classificadores LVQ serão discutidos no Capítulo 7.

## 6 RESULTADOS DA ANÁLISE DE AGRUPAMENTOS

Neste capítulo, buscou-se encontrar a melhor quantidade de protótipos por classe segundo os critérios de validação de agrupamentos discutidos na Seção 3.3 para cada conjunto de dados apresentado na Seção 5.1. Este resultado é o ponto de partida para os classificadores LVQ devido a dependência destes sobre a quantidade de protótipos e, também, da localização dos protótipos durante o início do treinamento.

A partir da metodologia apresentada no capítulo anterior, aqui são indicados os resultados dos experimentos realizados com o algoritmo  $K$ -médias e as técnicas de validação de agrupamentos citadas. Logo, a análise de agrupamentos dos dados de treinamento do BD1 é realizada na Seção 6.1. Já a análise do BD2 é realizada na Seção 6.2.

### 6.1 Banco de dados 1 (BD1)

A Figura 18 ilustra o histograma da moda das sugestões encontradas ao utilizar os critérios de validação de agrupamentos utilizados entre as realizações de treinamento de cada conjunto de treinamento formado aleatoriamente do BD1. Verifica-se, na Figura 18a, que as sugestões obtidas para a classe de condição normal variam entre o intervalo  $[1, 3]$ . É importante ressaltar que apenas os índices baseados em critérios de informação são capazes de sugerir  $K_{NO} = 1$ . Além disso, verifica-se que a votação majoritária mais frequente fora a sugestão de 2 protótipos a serem utilizados para a representação desta classe durante o treinamento dos algoritmos LVQ.

Verifica-se, na Figura 18b, que as sugestões obtidas para a classe de condição de falha de alta impedância variam no intervalo  $[3, 6]$ , onde observa-se que a votação majoritária de maior ocorrência fora a sugestão de 3 protótipos a serem utilizados para a representação desta classe durante o treinamento dos algoritmos LVQ.

Verifica-se, na Figura 18c, que as sugestões obtidas para a classe de condição de falha de baixa impedância variam entre os intervalos  $[2, 6]$  e  $[8, 9]$ . Apesar de uma maior dispersão ser encontrada para esta classe, observa-se que a votação majoritária mais frequente fora a sugestão de 4 protótipos a serem utilizados para a representação desta classe durante o treinamento dos algoritmos LVQ.

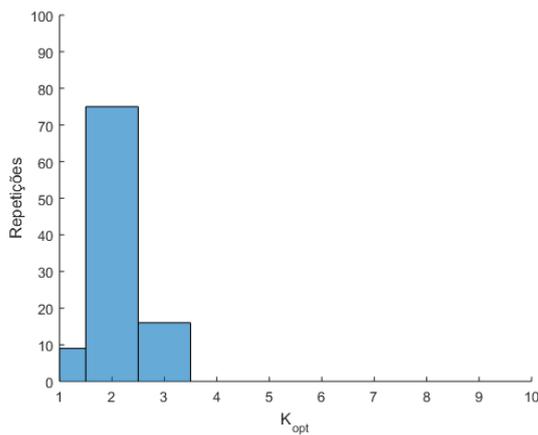
Logo, ao final desta etapa de treinamento, pode-se afirmar que o intervalo possível do parâmetro quantidade de protótipos  $K = (K_{NO} + K_{AI} + K_{BI})$  passou de  $[3, 30]$  (1 protótipo

por classe até 10 protótipos por classe) para [6, 18] ao considerar  $K_{min} = (1 + 3 + 2)$  e  $K_{max} = (3 + 6 + 9)$ . Apesar das muitas possibilidades para o valor de  $K$ , espera-se que a melhor configuração encontrada após o treinamento supervisionado seja  $K = (2 + 3 + 4) = 9$  devido ser a combinação da maior frequência de cada classe separadamente.

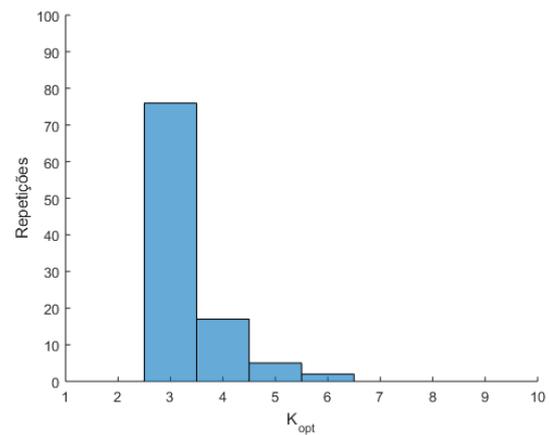
Verificou-se, posteriormente, as sugestões de cada métrica de validação de agrupamentos em cada classe a fim de aferir divergências significativas entre estas. As sugestões para a classe normal são ilustradas na Figura 19. Já as sugestões para a classe falha de alta impedância são ilustradas na Figura 20. Por fim, as sugestões para a classe falha de baixa impedância são ilustradas na Figura 21.

Figura 18 – Frequência da moda das sugestões dos critérios de validação de agrupamentos utilizados para o banco de dados 1

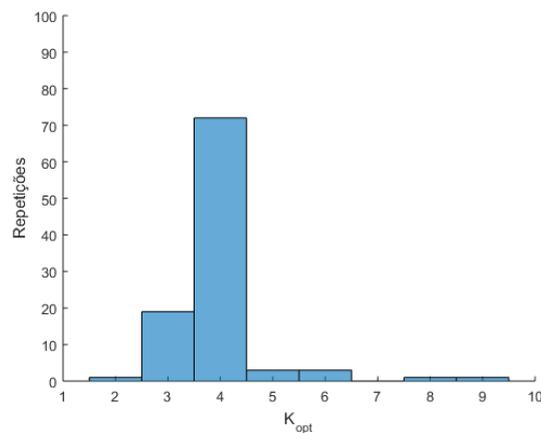
(a) Classe normal



(b) Classe falha AI



(c) Classe falha BI



Fonte: o autor.

### 6.1.1 Classe normal

Analisou-se, primeiramente, a classe normal. Verifica-se, a partir da Figura 18a, a ocorrência de moda para esta classe no intervalo  $[1, 3]$  onde os critérios de informação são as únicas métricas utilizadas em que  $K_{opt} = 1$  é possível.

Entre os índices relativos, apenas o índice CH apresentou votações variadas por praticamente todo o intervalo considerado, ou seja, excluiu apenas a possibilidade de  $K_{opt} = 10$ . Já os critérios DB, Dunn e Silhuetas sugeriram  $K_{opt} = 2$ .

Entre os critérios de informação, apenas o critério BIC se estabilizou majoritariamente em  $K_{opt} = 1$  conforme mostrado na Figura 19f. Os demais critérios de informação apresentaram votações variadas onde  $K_{opt} = 2$  no critério FPE e  $K_{opt} = 3$  nos critérios AIC e MDL.

### 6.1.2 Classe falha de alta impedância

Analisou-se, posteriormente, a classe falha de alta impedância. Verifica-se, a partir da Figura 18b, a não ocorrência de moda para  $K_{opt} = 1$  e entre os valores  $K_{opt} = 7$  e  $K_{opt} = 10$ . Observa-se, também, que  $K_{opt} = 3$  possui a ocorrência de moda mais relevante.

Entre os índices relativos, os índices Dunn e CH apresentaram votações fora do intervalo da moda, onde o índice Dunn indicou  $K_{opt} = 7$  e o índice CH indicou  $K_{opt} = 2$  mesmo que em frequências irrelevantes. Os índices DB e Silhuetas apresentaram sugestões semelhantes, porém o índice Silhuetas não apresentou sugestões de  $K_{opt} = 5$ .

Entre os critérios de informação, apenas o critério BIC sugeriu  $K_{opt} = 3$  onde este valor possui maior ocorrência de moda entre todas as métricas para esta classe. Ao avaliar a Figura 20f, percebe-se que o critério BIC apresentou sugestões semelhantes ao da moda dos índices, porém não apresentou sugestões de  $K_{opt} = 6$ . Os demais critérios apresentaram sugestões a partir de  $K_{opt} = 5$ .

### 6.1.3 Classe falha de baixa impedância

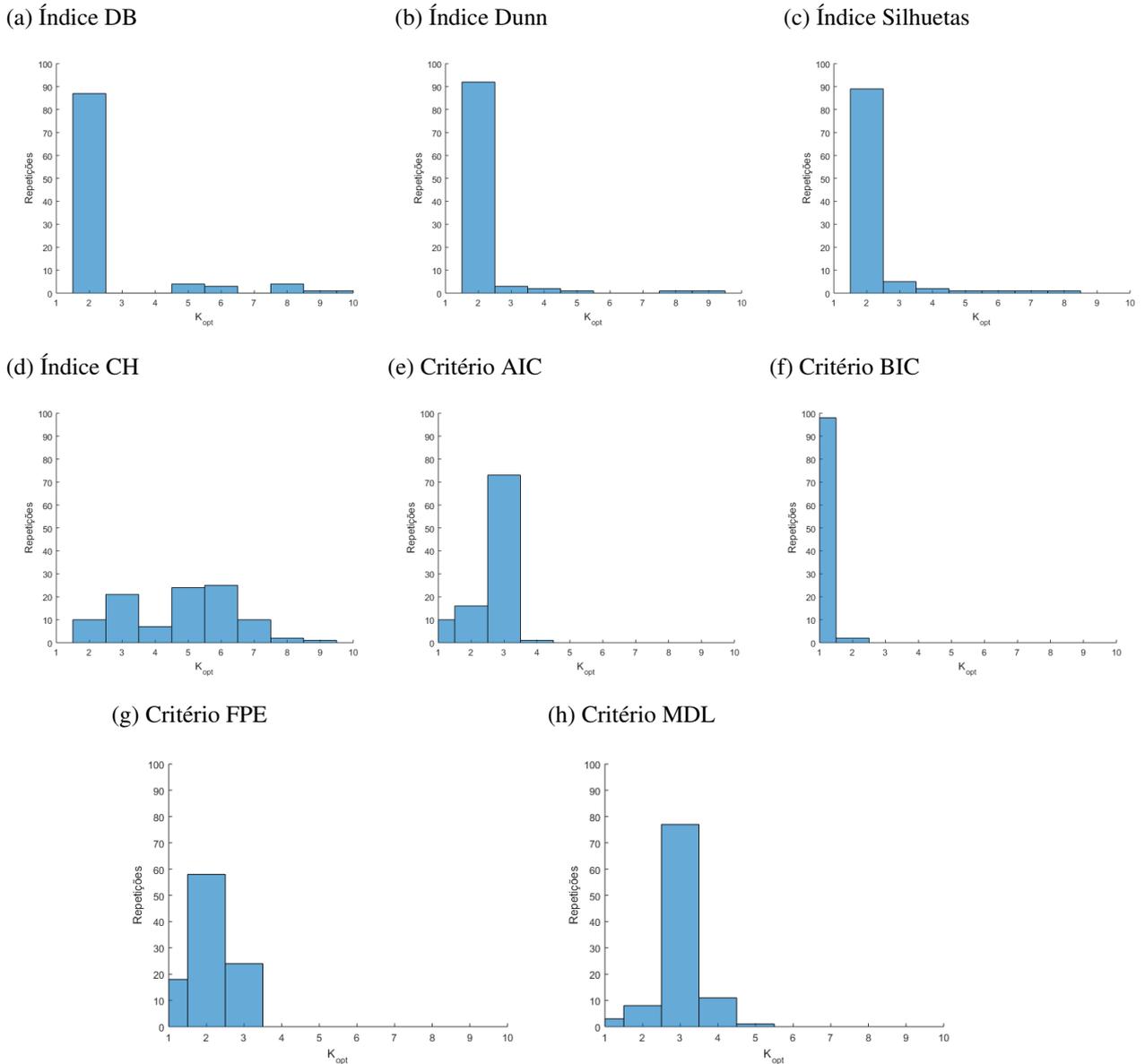
Analisou-se, por último, a classe falha de baixa impedância. Verifica-se, a partir da Figura 18c, a ocorrência de moda para esta classe no intervalo  $[3, 6]$ .

Entre os índices relativos, os índices Dunn e CH apresentaram votações fora do intervalo da moda, onde o índice Dunn indicou  $K_{opt} = 7$  e o índice CH indicou  $K_{opt} = 2$  mesmo

que em frequências irrelevantes. Os índices DB e Silhuetas apresentaram sugestões semelhantes, porém o índice Silhuetas não apresentou sugestões de  $K_{opt} = 5$ .

Entre os critérios de informação, apenas o critério BIC apresentou distribuição de sugestões semelhante à da moda de todos os índices, porém não apresentou sugestões para os valores sugeridos em menor relevância.

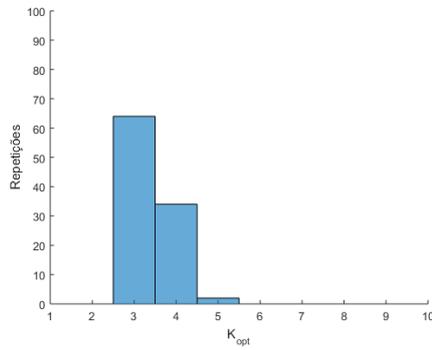
Figura 19 – Frequência das sugestões dos critérios de validação de agrupamentos utilizados para a classe normal do banco de dados 1



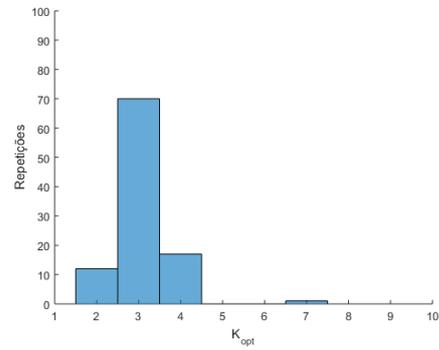
Fonte: o autor.

Figura 20 – Frequência das sugestões dos critérios de validação de agrupamentos utilizados para a classe falha de alta impedância do banco de dados 1

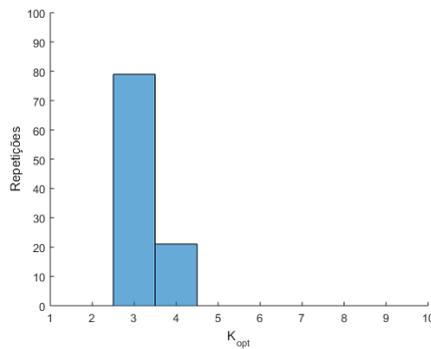
(a) Índice DB



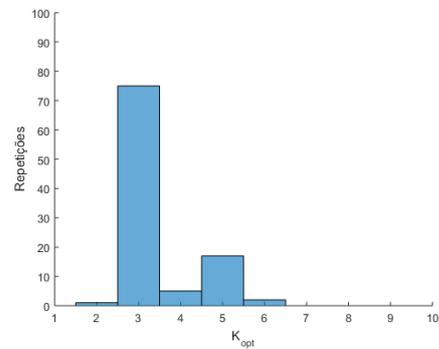
(b) Índice Dunn



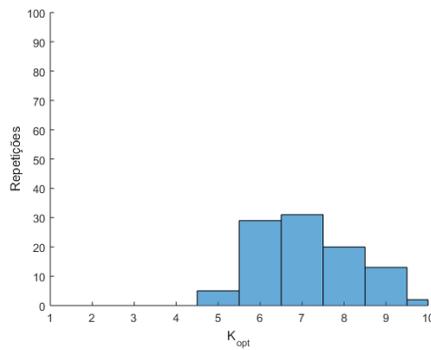
(c) Índice Silhuetas



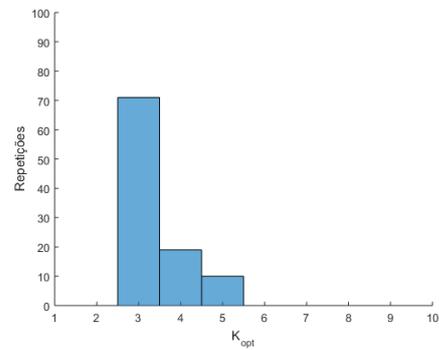
(d) Índice CH



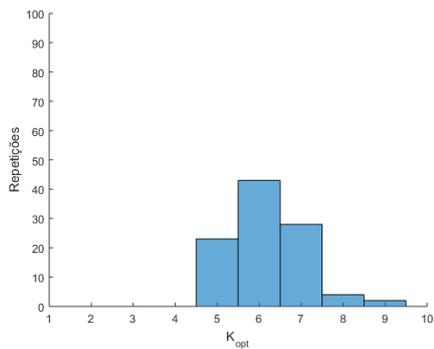
(e) Critério AIC



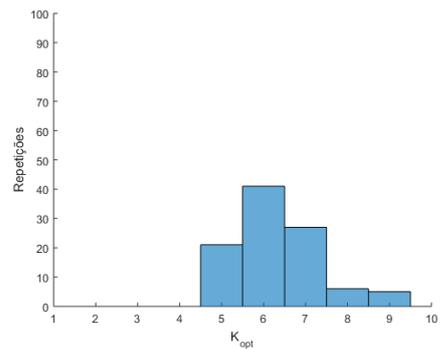
(f) Critério BIC



(g) Critério FPE



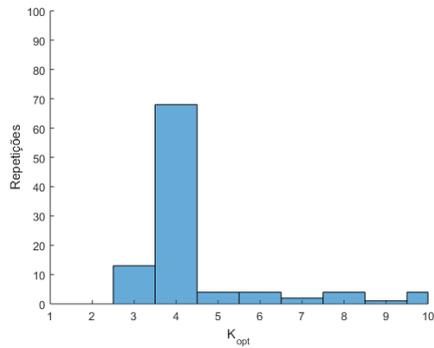
(h) Critério MDL



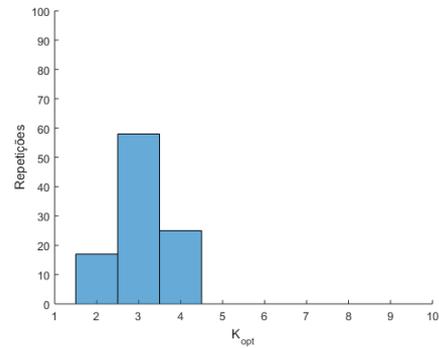
Fonte: o autor.

Figura 21 – Frequência das sugestões dos critérios de validação de agrupamentos utilizados para a classe falha de baixa impedância do banco de dados 1

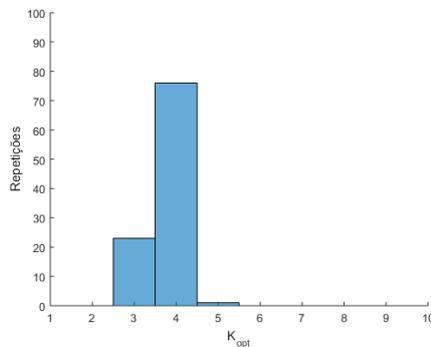
(a) Índice DB



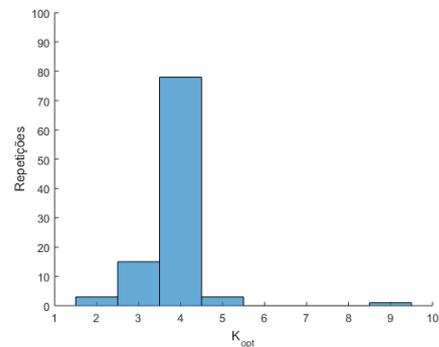
(b) Índice Dunn



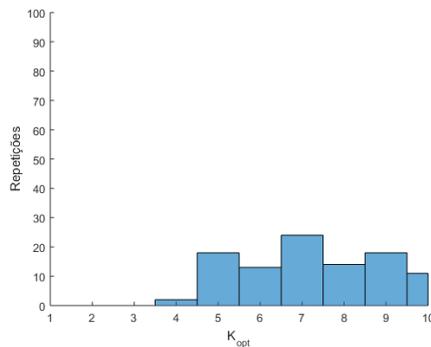
(c) Índice Silhuetas



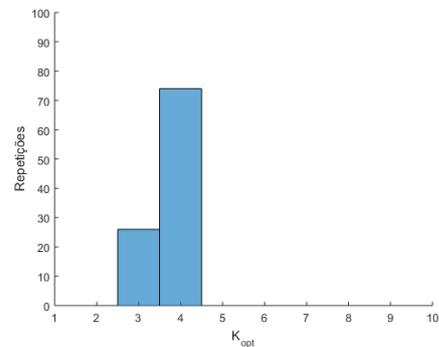
(d) Índice CH



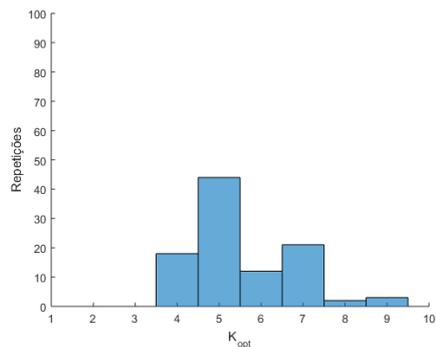
(e) Critério AIC



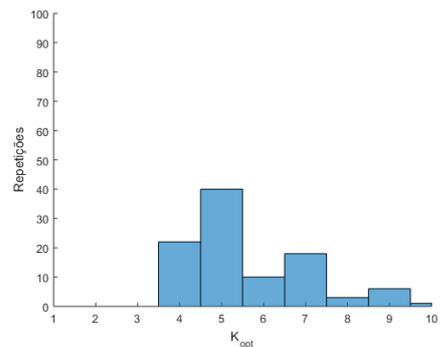
(f) Critério BIC



(g) Critério FPE



(h) Critério MDL



Fonte: o autor.

## 6.2 Banco de dados 2 (BD2)

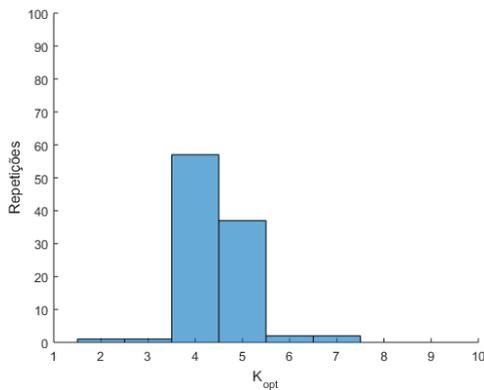
A Figura 22 ilustra o histograma da moda das sugestões encontradas ao utilizar os critérios de validação de agrupamentos utilizados entre as realizações de treinamento de cada conjunto de treinamento formado aleatoriamente do BD2.

Verifica-se, na Figura 22a, que as sugestões obtidas para a classe de condição normal variam no intervalo  $[2, 7]$ . Apesar do grande intervalo encontrado, as sugestões  $K_{NO} = 4$  e  $K_{NO} = 5$  são predominantes. Entre estes dois valores mais frequentes, a sugestão de 4 protótipos ocorrerá mais expressivamente. Além disso, verifica-se que mais protótipos para esta classe são demandados ao incluir exemplos artificiais desta classe no banco de dados 1 ao comparar as Figuras 18a e 22a.

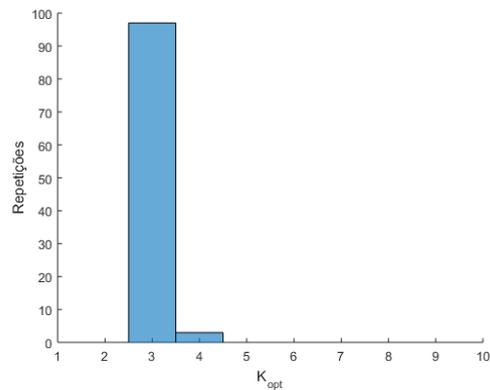
Verifica-se, na Figura 22b, que as sugestões obtidas para a classe de condição de falha de alta impedância variam no intervalo  $[3, 4]$ , onde observa-se que a votação majoritária

Figura 22 – Frequência da moda das sugestões dos critérios de validação de agrupamentos utilizados para o banco de dados 2

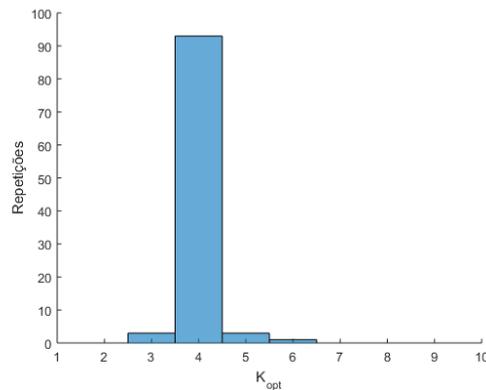
(a) Classe normal



(b) Classe falha AI



(c) Classe falha BI



Fonte: o autor.

mais frequente fora a sugestão de 3 protótipos a serem utilizados para a representação desta classe durante o treinamento dos algoritmos LVQ Comparando esta figura com a Figura 18b, é percebido que a votação majoritária de  $K_{AI} = 3$  se tornou mais expressiva.

Verifica-se, na Figura 22c, que as sugestões obtidas para a classe de condição de falha de baixa impedância variam entre o intervalo  $[3, 6]$ . Apesar da grande dispersão encontrada para esta classe, observa-se que o intervalo de possibilidades sofreu uma compressão ao comparar com o intervalo ilustrado na Figura 22c. Além disso, percebe-se a votação majoritária de  $K_{BI} = 4$  se tornou mais expressiva.

Logo, ao final desta etapa de treinamento, pode-se afirmar que o intervalo possível do parâmetro quantidade de protótipos  $K = (K_{NO} + K_{AI} + K_{BI})$  passou de  $[3, 30]$  (1 protótipo por classe até 10 protótipos por classe) para  $[8, 17]$  ao considerar  $K_{min} = (2 + 3 + 3)$  e  $K_{max} = (7 + 4 + 6)$ . Apesar das muitas possibilidades para o valor de  $K$ , observa-se que o intervalo de possibilidades para o BD2 é inferior ao do BD1 e espera-se que a melhor configuração encontrada após o treinamento supervisionado seja:

$$K = \begin{cases} 4 + 3 + 4 = 11 \\ 5 + 3 + 4 = 12 \end{cases} \quad (6.1)$$

devido estas combinações possuírem maiores frequências de cada classe separadamente.

Verificou-se, posteriormente, as sugestões de cada métrica de validação de agrupamentos em cada classe a fim de aferir divergências significativas entre estas. As sugestões para a classe normal são ilustradas na Figura 23. Já as sugestões para a classe falha de alta impedância são ilustradas na Figura 24. Por fim, as sugestões para a classe falha de baixa impedância são ilustradas na Figura 25.

### 6.2.1 Classe normal

Analisou-se, primeiramente, a classe normal. Verifica-se, a partir da Figura 22a, a ocorrência de maior relevância de moda para esta classe para  $K_{opt} = 4$  e  $K_{opt} = 5$ . As sugestões de diferentes valores a estas possuem a mesma ordem de grandeza.

Entre os índices relativos, o índice DB deu maior relevância para  $K_{opt} = 5$ . Já os índices Dunn e Silhuetas indicaram suas sugestões mais relevantes assim como ao da moda desta classe. Por último, o CH reportou exclusivamente  $K_{opt} = 4$  na maioria das repetições.

Entre os critérios de informação, apenas o critério BIC se estabilizou majoritari-

amente em  $K_{opt} = 1$  conforme mostrado na Figura 23f. Os demais critérios de informação apresentaram votações com concentração em  $K_{opt} = 4$  e  $K_{opt} = 5$ . Entre todos os critérios de validação utilizados, apenas o critério AIC sugeriu  $K_{opt} = 7$  mesmo que em frequência irrelevante.

A partir da comparação entre os dois bancos de dados, verificou-se que a inserção de exemplos artificiais desta classe contribuiu para a elevação da demanda de protótipos para representação desta classe.

### 6.2.2 Classe falha de alta impedância

Analisou-se, posteriormente, a classe falha de alta impedância. Verifica-se, a partir da Figura 22b, a ocorrência de moda entre os valores  $K_{opt} = 3$  e  $K_{opt} = 4$  onde  $K_{opt} = 3$  possui a frequência mais relevante. A distribuição da moda desta classe é bastante semelhante à do BD1, porém não apresenta as sugestões  $K_{opt} = 5$  e  $K_{opt} = 6$  que são reportados no banco de dados anterior.

Entre os índices relativos, verifica-se que todos concentraram suas sugestões em  $K_{opt} = 3$  e apenas o índice DB apresentou exclusivamente sugestões de  $K_{opt} = 3$ . Os índices Dunn e Silhuetas apresentaram sugestões de  $K_{opt} = 4$  mesmo que inexpressivamente. Já o índice CH, além de  $K_{opt} = 4$ , também sugeriu  $K_{opt} = 2$ .

Entre os critérios de informação, apenas o critério BIC apresentou frequência de sugestões para  $K_{opt} = 3$ . Este valor possui maior ocorrência de moda entre todas as métricas para esta classe. Ao avaliar a Figura 24f, percebe-se que o critério BIC apresentou sugestões semelhantes ao da moda dos índices relativos. Assim como a moda de todas as métricas, este critério também apresenta sugestões de menor frequência para  $K_{opt} = 4$ . Os demais critérios não apresentaram sugestões para  $K_{opt} = 3$  e possuem intervalos maiores.

### 6.2.3 Classe falha de baixa impedância

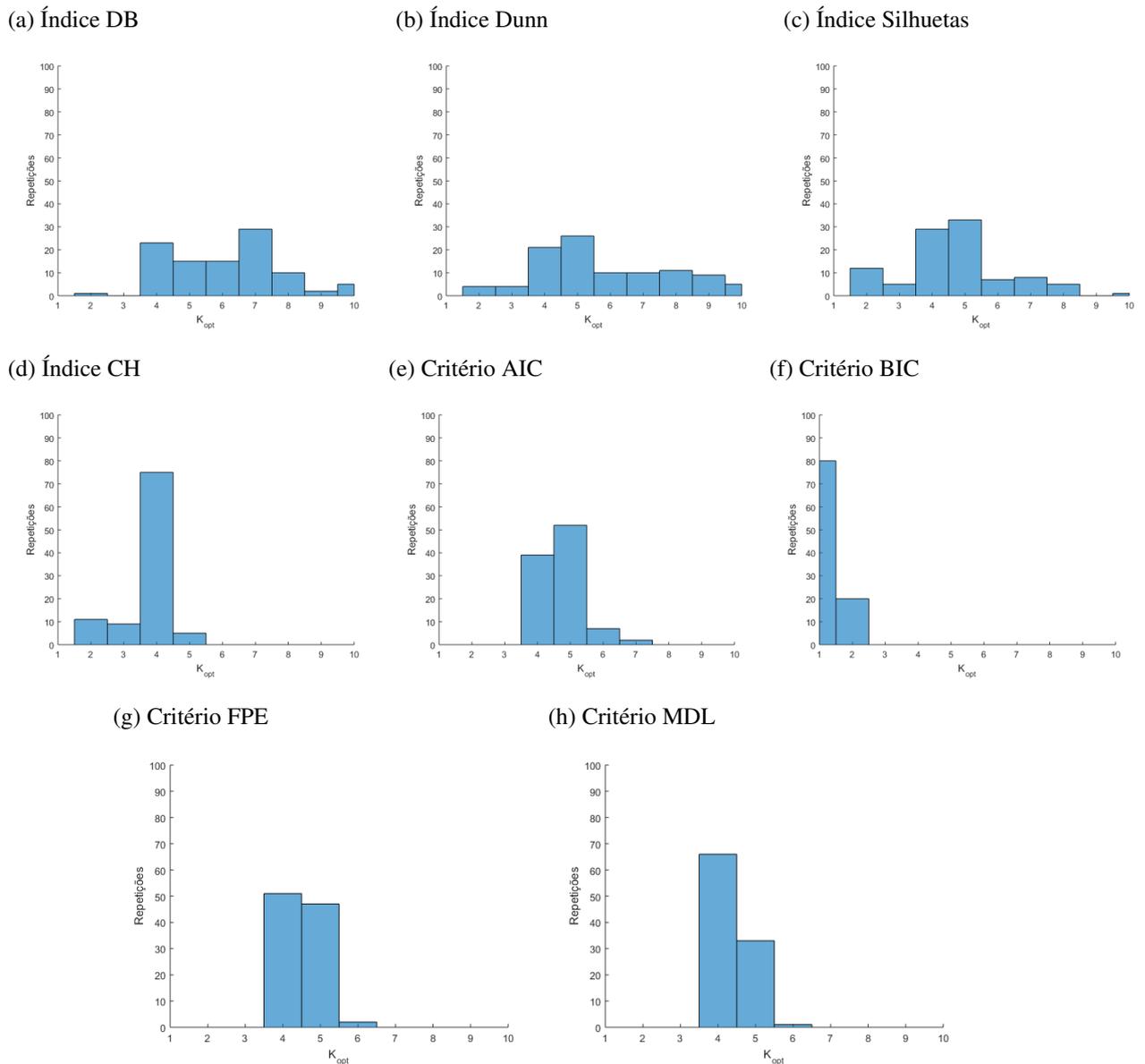
Analisou-se, por último, a classe falha de baixa impedância. Verifica-se, a partir da Figura 22c, a ocorrência de moda para esta classe no intervalo  $[3, 6]$  e este é o mesmo intervalo obtido no BD1.

Entre os índices relativos, o índice Dunn fora o único a não apresentar  $K_{opt} = 4$  como o mais votado entre as repetições, porém, para este índice,  $K_{opt} = 4$  fora o segundo mais votado e bem próximo do  $K_{opt}$  mais relevante segundo este índice. Os índices DB, Silhuetas e

CH concentraram suas sugestões em  $K_{opt} = 4$  onde o intervalo do índice DB é  $[3, 6]$  e  $K_{opt} = 10$ , e o intervalo dos índices Silhuetas e CH é  $[3, 5]$ .

Entre os critérios de informação, apenas o critério BIC apresentou distribuição de sugestões semelhante à da moda de todos os índices, porém não apresentou sugestões para  $K_{opt} = 5$  e  $K_{opt} = 6$ .

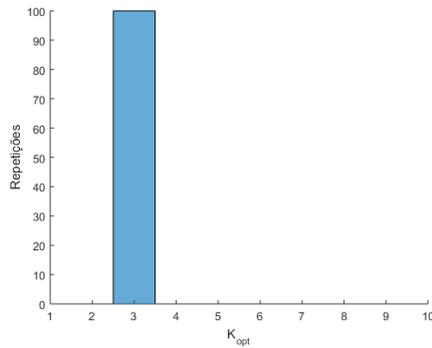
Figura 23 – Frequência das sugestões dos critérios de validação de agrupamentos utilizados para a classe normal do banco de dados 2



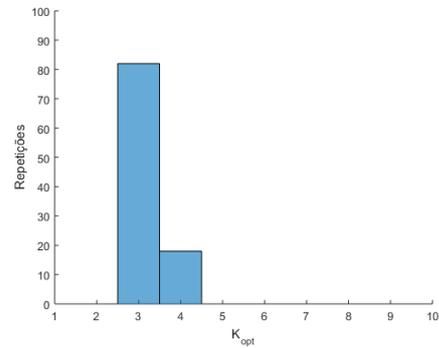
Fonte: o autor.

Figura 24 – Frequência das sugestões dos critérios de validação de agrupamentos utilizados para a classe falha de alta impedância do banco de dados 2

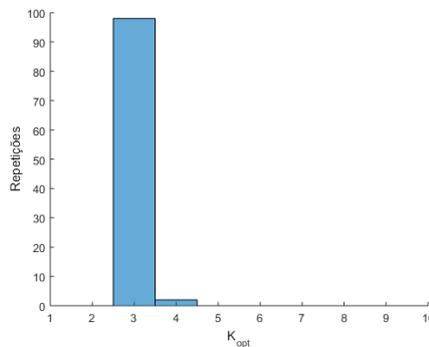
(a) Índice DB



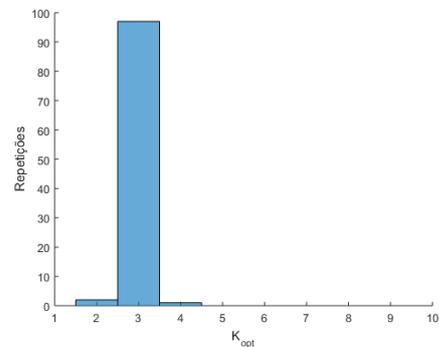
(b) Índice Dunn



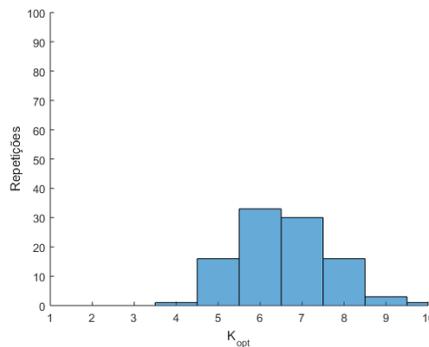
(c) Índice Silhuetas



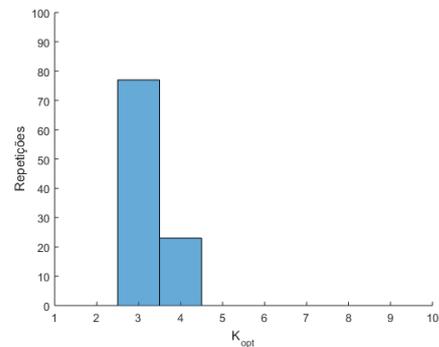
(d) Índice CH



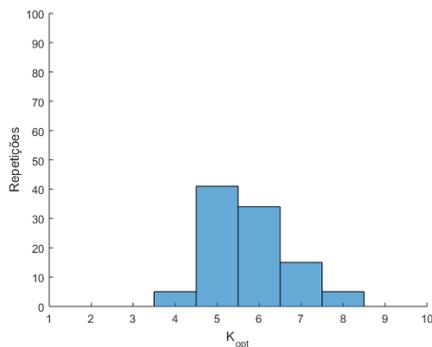
(e) Critério AIC



(f) Critério BIC



(g) Critério FPE



(h) Critério MDL

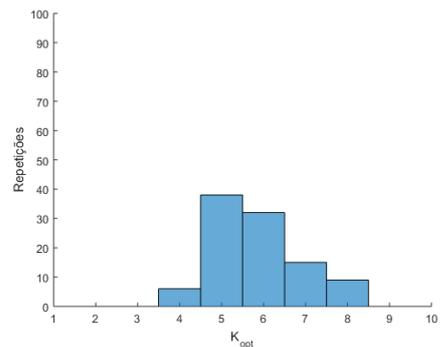
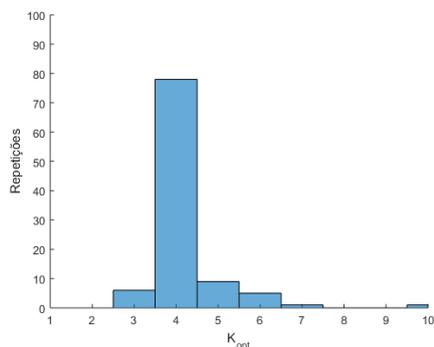
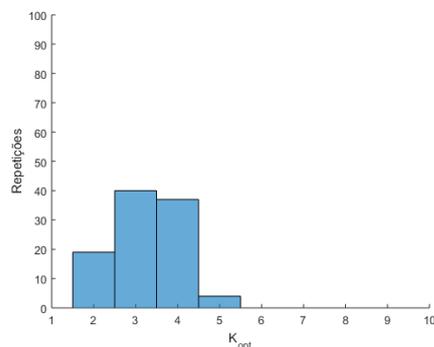


Figura 25 – Frequência das sugestões dos critérios de validação de agrupamentos utilizados para a classe falha de baixa impedância do banco de dados 2

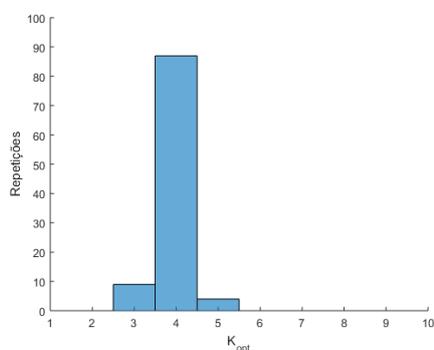
(a) Índice DB



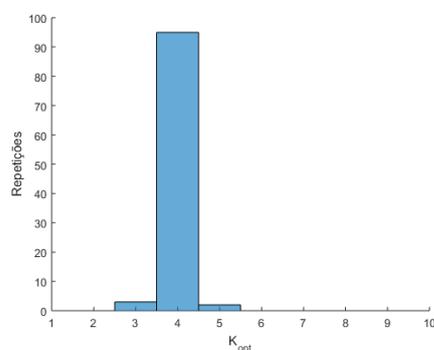
(b) Índice Dunn



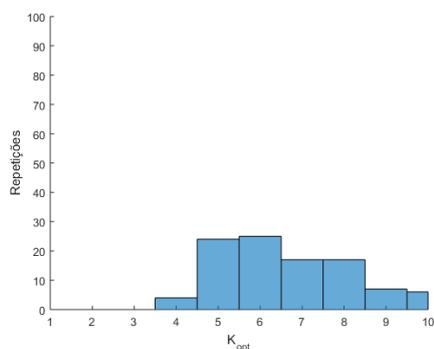
(c) Índice Silhuetas



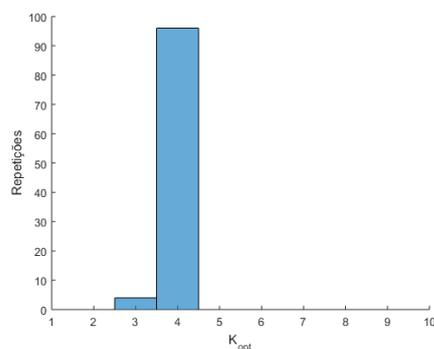
(d) Índice CH



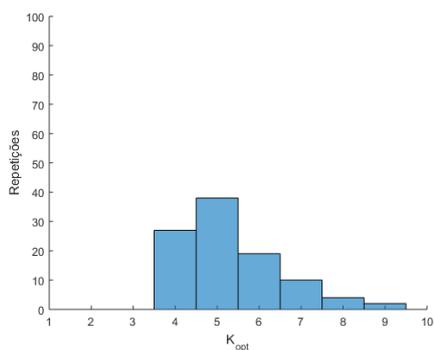
(e) Critério AIC



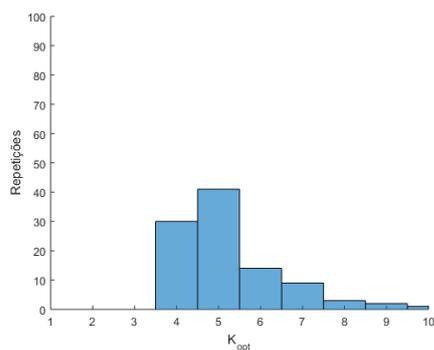
(f) Critério BIC



(g) Critério FPE



(h) Critério MDL



Fonte: o autor.

### 6.3 Conclusão

Neste capítulo, fora determinado o valor de  $K_{opt}$  de cada classe em cada realização da metodologia proposta baseada em aprendizado não supervisionado para os conjuntos de treinamento dos bancos de dados BD1 e BD2. As configurações que mais ocorreram através da votação majoritária dos critérios de validação de agrupamentos para os dois bancos de dados são mostrados na Tabela 11.

Tabela 11 – Valores majoritários de  $K_{opt}$  encontrados para BD1 e BD2

	$K_{NO}$	$K_{AI}$	$K_{BI}$	$K$
<b>BD1</b>	2	3	4	<b>9</b>
<b>BD2</b>	4–5	3	4	<b>11–12</b>

Fonte: o autor.

Após a determinação dos valores ótimos de  $K$ , estes serão utilizados no treinamento dos classificadores LVQ a partir dos métodos Treinamento 1 e Treinamento 2 propostos na Seção 5.3.5. Finalmente, os resultados da comparação entre os classificadores LVQ serão discutidos no próximo capítulo.

## 7 RESULTADOS DA COMPARAÇÃO ENTRE CLASSIFICADORES LVQ

Após a análise de agrupamentos dos dados de treinamento apresentada no capítulo anterior, os resultados de  $K_{opt}$  obtidos por cada classe são utilizados no treinamento dos algoritmos supervisionados propostos nesta dissertação. Logo, a análise dos classificadores ao utilizar os métodos de treinamento propostos para o banco de dados 1 é realizada na Seção 7.1. Já a análise dos classificadores ao utilizar os métodos de treinamento propostos para o banco de dados 2 é realizada na Seção 7.2.

Ressalta-se que o treinamento dos algoritmos LVQ fora realizado em caráter multi-objetivo, ou seja, o treinamento fora desenvolvido com o aprendizado do classificador para as classes normal, falha de AI e falha de BI e, posteriormente, estes resultados são mapeados para a classificação binária entre os estados normal e falha.

### 7.1 Banco de dados 1 (BD1)

Os resultados obtidos dos classificadores LVQ utilizados neste trabalho para o banco de dados 1 encontram-se divididos em quatro seções. A Seção 7.1.1 apresenta, em diagramas de caixa (*boxplots*), e discute sobre as taxas de acertos encontradas por cada algoritmo LVQ em classificação ternária e a representação destas em classificação binária. As matrizes de confusão e suas estatísticas associadas são apresentadas na Seção 7.1.2 para os classificadores de melhores rendimentos na classificação. Os resultados específicos sobre o classificador de melhor aptidão são apresentados na Seção 7.1.3. Para evitar sobrecarga de informação, as estatísticas encontradas, em formato de tabelas, sobre as taxas de acertos em classificação ternária e binária, sensibilidade e especificidade são mostradas apenas no Apêndice A.

#### 7.1.1 Taxas de acertos

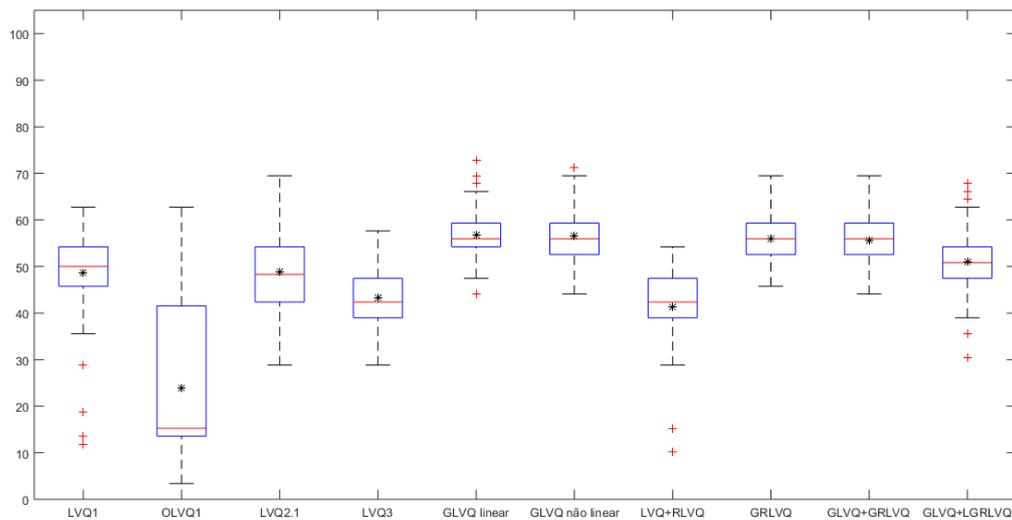
Inicialmente, verificou-se o desempenho dos classificadores em função das taxas de acertos encontradas por cada classificador LVQ em cada metodologia de treinamento proposta. Os resultados encontrados na classificação ternária<sup>1</sup> são ilustrados na Figura 26. Já o mapeamento destes resultados para classificação binária são ilustrados na Figura 27.

As estatísticas encontradas a partir da classificação ternária ao utilizar a metodologia de treinamento 1 são mostradas na Tabela 73 e estes resultados mapeados em classificação

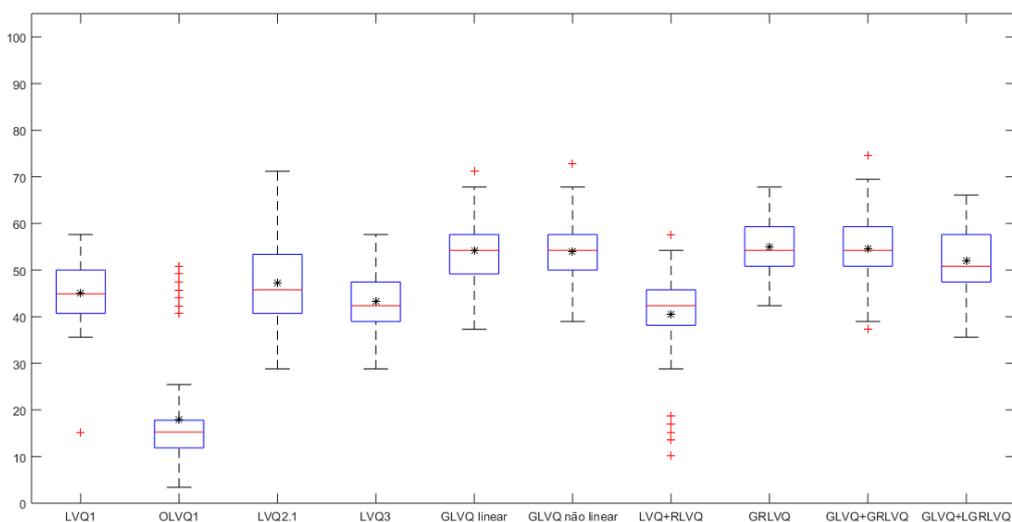
<sup>1</sup> Entre as classes normal, falha de alta impedância e falha de baixa impedância.

Figura 26 – Taxas de acertos encontradas para classificação ternária do banco de dados 1

(a) Metodologia de Treinamento I



(b) Metodologia de Treinamento II



(c) Metodologia de Treinamento III

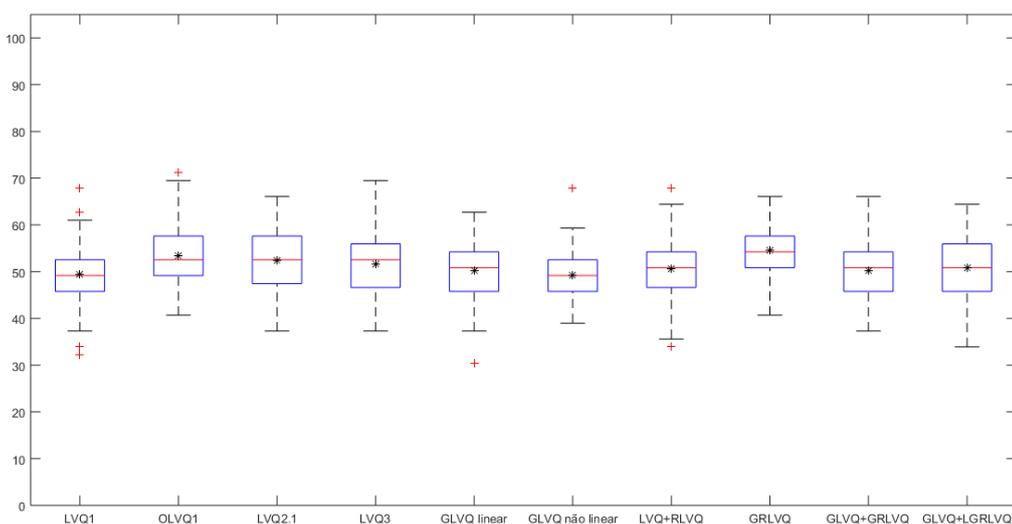
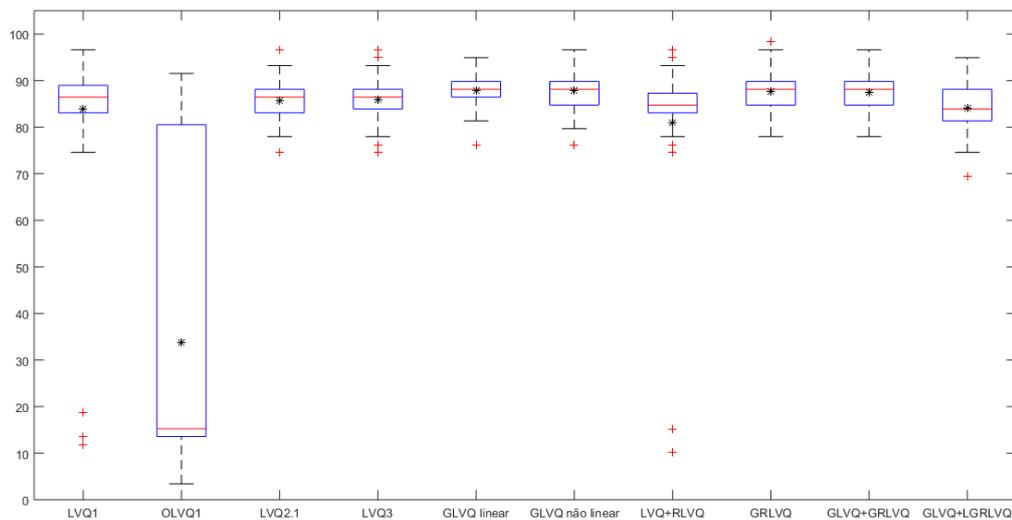
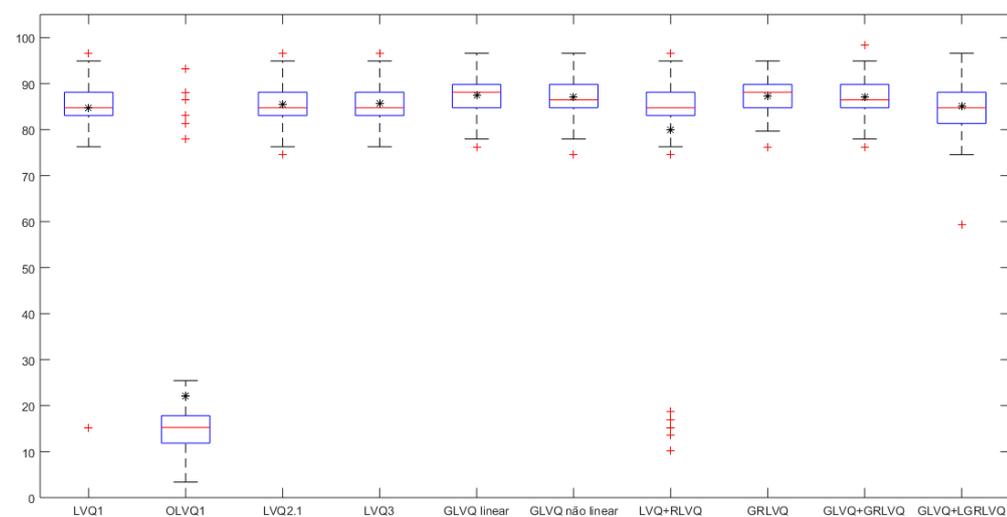


Figura 27 – Taxas de acertos mapeadas para classificação binária do banco de dados 1.

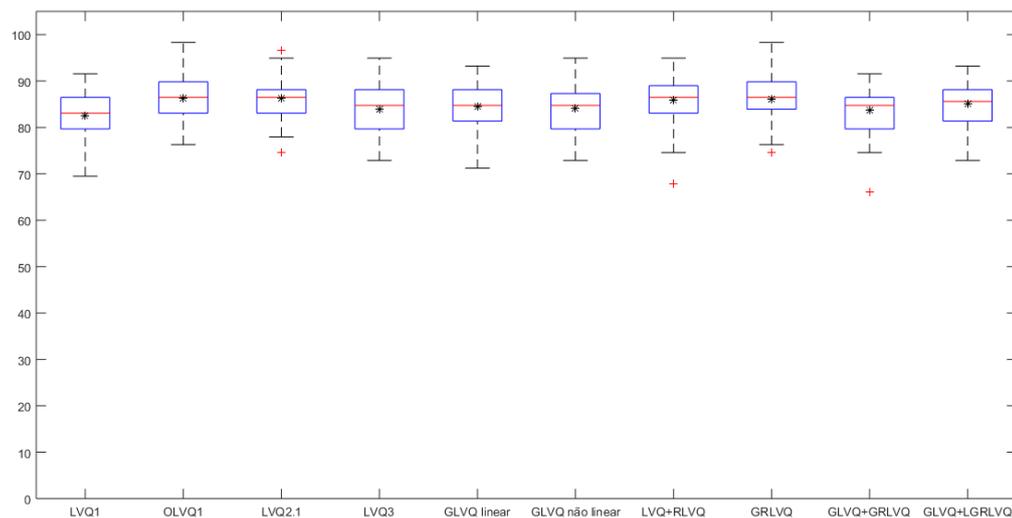
(a) Metodologia de Treinamento I



(b) Metodologia de Treinamento II



(c) Metodologia de Treinamento III



binária são mostradas na Tabela 76. Da mesma forma, as estatísticas das taxas de acertos para a metodologia de treinamento 2 são mostradas na Tabela 74 para classificação ternária e na Tabela 77 para classificação binária. Enquanto para a metodologia de treinamento 3, as respectivas estatísticas são mostradas nas Tabelas 75 e 78.

Através da comparação geral entre os classificadores para as três metodologias, verifica-se que o OLVQ1 é o mais sensível para o parâmetro quantidade de protótipos, onde verifica-se, em classificação ternária, que as taxas mínimas de acerto encontradas são de apenas 3,3898% nas metodologias 1 e 2 e saltaram para 40,678% na metodologia 3. Este comportamento também ocorreu para as taxas máxima, média e mediana, onde pode-se exemplificar as taxas máximas encontradas de 62,7119% (91,5254% em classificação binária) na metodologia 1, 50,8475% (93,2203% em classificação binária) na metodologia 2 e 71,1864% (98,3051% em classificação binária) na metodologia 3.

Observa-se, ao comparar os resultados encontrados dos algoritmos LVQ heurísticos (LVQ1, OLVQ1, LVQ2.1 e LVQ3) nas metodologias 1 e 2, que a heurística proposta de iniciação dos protótipos melhorou as taxas de acertos máximas, medianas e médias, porém piorou as taxas mínimas. Isto pode ser aferido através das Figuras 26a e 26b e, também, a partir das Tabelas 73 e 74. Em consequência da elevação das taxas máximas, medianas e médias e atenuação das taxas mínimas, o cenário que utilizou as posições iniciais dos protótipos sugeridas através da etapa de aprendizado não supervisionado apresentaram maiores valores de desvio padrão. Já para os algoritmos da família maximização da margem de separação, devido possuírem funções custo que garantem a convergência (diferentemente dos algoritmos LVQ heurísticos), não apresentaram elevações das taxas citadas no caso dos LVQ heurísticos, porém apresentaram melhorias nas taxas mínimas encontradas.

Comparou-se, posteriormente, as metodologias que possuem iniciações aleatórias dos protótipos, onde a metodologia 2 utiliza a quantidade de protótipos estabelecida pelos critérios de validação de agrupamentos enquanto que a metodologia 3 possui a quantidade fixa de 25% da quantidade de amostras do conjunto de treinamento. Ou seja, têm-se, na metodologia 2, aproximadamente 10 protótipos enquanto que a metodologia 3 possui aproximadamente<sup>2</sup> 58 protótipos.

Obtém-se, através da análise entre estas duas metodologias, que a metodologia que utiliza mais protótipos apresentou taxas de acertos consideravelmente melhores para os

<sup>2</sup> Utiliza-se o termo aproximadamente devido a quantidade de protótipos utilizada pode variar entre diferentes realizações. Porém, permanecem com as respectivas magnitudes.

algoritmos heurísticos em classificação ternária. Porém, este comportamento não é observado ao verificar a classificação binária, onde as taxas máxima, média e mediana se tornam bastante próximas excetuando o algoritmo OLVQ1.

Já para os algoritmos LVQ da família margem de separação, ao comparar as metodologias 2 e 3, excetuando o algoritmo LVQ+RLVQ, todos os algoritmos desta família apresentaram melhores taxas de acerto na metodologia 2 para a classificação ternária. Para classificação binária, a maior taxa máxima para a metodologia fora obtida pelo GLVQ+GRLVQ para a metodologia 2 enquanto o GRLVQ obtivera a maior taxa máxima na metodologia 3.

Ao comparar o desempenho dos algoritmos GLVQ linear e GLVQ não linear, verificou-se que a utilização da função não linear na atualização dos protótipos não gerou resultados que justificassem benefício suficiente ao utilizar este algoritmo em sua versão que possui maior custo computacional e que ambos estão entre os classificadores que obtiveram melhores taxas de acertos. Então, selecionou-se o GLVQ linear para melhor análise.

Além disso, verifica-se que o GRLVQ possui a maior frequência entre os algoritmos que apresentam as estatísticas de acerto encontradas mais elevadas e está entre os algoritmos que apresentam os menores valores de desvio padrão.

Logo, concluiu-se que os algoritmos GLVQ linear e GRLVQ são os melhores candidatos, entre as variantes utilizadas, para representarem o banco de dados 1 em todas as metodologias de treinamento propostas e compará-los com o resultados obtidos do banco de dados 2.

### **7.1.2 Matrizes de confusão e estatísticas associadas**

Após a seleção dos algoritmos GLVQ linear e GRLVQ como classificadores mais adequados, estes foram analisados através das matrizes de confusão obtidas entre as diversas realizações de treinamento deste algoritmo.

Primeiramente, analisou-se o algoritmo GLVQ linear:

As matrizes de confusão obtidas a partir da maior taxa de acerto binária na metodologia 1 são mostradas nas Tabelas 12 e 13. Já as matrizes de confusão obtidas da menor taxa de acerto são mostradas nas Tabelas 14 e 15.

A partir das Tabelas 12 e 13, verifica-se que ocorrem 2 falsos positivos onde 2 exemplos da classe normal são preditos como falha de alta impedância. Verifica-se, também, que há uma ocorrência de falso negativo onde este erro ocorre em um exemplo de falha de baixa

impedância.

A partir das Tabelas 14 e 15, verifica-se a ocorrência de 1 falso positivo onde este exemplo da classe normal é predito como falha de baixa impedância. Verifica-se, também, que ocorrem 13 falsos negativos onde 12 destes erros ocorrem em exemplos de falha de alta impedância e apenas um erro de falha de baixa impedância.

Observa-se, também, que a elevação da quantidade de erros de classificação ocorre apenas no sentido de falsos estados de normalidade ao comparar as taxas de máximo e de mínimo.

Já na metodologia 2, as matrizes de confusão obtidas a partir maior taxa de acerto binária são mostradas nas Tabelas 16 e 17. Já as matrizes de confusão obtidas da menor taxa de acerto são mostradas nas Tabelas 18 e 19.

A partir das Tabelas 16 e 17, verifica-se que há ocorrência de apenas um erro de classificação de falso positivo onde uma amostra normal fora classificada como falha de AI. Verifica-se, também, que há uma ocorrência de erro do tipo falso negativo onde este erro ocorre em um exemplo de falha de BI.

A partir das Tabelas 18 e 19, verifica-se que há ocorrência de apenas um falso positivo onde uma amostra normal fora classificada como falha de BI. Verifica-se, também, que há ocorrência de 13 falsos negativos onde 12 destes erros ocorrem em exemplos de falha de alta impedância e apenas um exemplo de falha de baixa impedância.

Observa-se, ao comparar as piores matrizes de confusão das metodologias 1 e 2, que estão são idênticas inclusive no tipo de erro de falso positivo onde uma amostra normal é classificada como falha de BI. Já entre as melhores matrizes de confusão entre estas metodologias, verifica-se há um erro a mais de falso positivo na metodologia 1.

Por último, na metodologia 3, as matrizes de confusão obtidas a partir maior taxa de acerto binária são mostradas nas Tabelas 20 e 21. Já as matrizes de confusão obtidas da menor taxa de acerto são mostradas nas Tabelas 22 e 23.

A partir das Tabelas 20 e 21, verifica-se que há um falso positivo onde um exemplo

Tabela 12 – Melhor matriz de confusão do GLVQ linear em classificação binária do banco de dados 1 para metodologia de treinamento 1

GLVQ linear		Classe Vedadeira	
		Normal	Falha
Classe Predita	Normal	2	1
	Falha	2	54

Fonte: o autor.

Tabela 13 – Melhor matriz de confusão do GLVQ linear em classificação ternária do banco de dados 1 para metodologia de treinamento 1

GLVQ linear		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	2	0	1
	AI	2	14	9
	BI	0	13	18

Fonte: o autor.

Tabela 14 – Pior matriz de confusão do GLVQ linear em classificação binária do banco de dados 1 para metodologia de treinamento 1

GLVQ linear		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	2	13
	Falha	1	43

Fonte: o autor.

Tabela 15 – Pior matriz de confusão do GLVQ linear em classificação ternária do banco de dados 1 para metodologia de treinamento 1

GLVQ linear		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	2	12	1
	AI	0	21	3
	BI	1	13	6

Fonte: o autor.

Tabela 16 – Melhor matriz de confusão do GLVQ linear em classificação binária do banco de dados 1 para metodologia de treinamento 2

GLVQ linear		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	2	1
	Falha	1	55

Fonte: o autor.

Tabela 17 – Melhor matriz de confusão do GLVQ linear em classificação ternária do banco de dados 1 para metodologia de treinamento 2

GLVQ linear		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	2	0	1
	AI	1	17	7
	BI	0	14	17

Fonte: o autor.

normal fora classificado como falha de AI. Verifica-se, também, que há três falsos negativos onde um exemplo pertence à classe falha de AI e dois pertencem à classe de falha de BI.

A partir das Tabelas 22 e 23, verifica-se que há 12 falsos positivos onde 6 destas amostras normais foram preditas como de falha de AI e outras 6 como falha de BI. Verifica-se,

Tabela 18 – Pior matriz de confusão do GLVQ linear em classificação binária do banco de dados 1 para metodologia de treinamento 2

GLVQ linear		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	2	13
	Falha	1	43

Fonte: o autor.

Tabela 19 – Pior matriz de confusão do GLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 2

GLVQ linear		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	2	12	1
	AI	0	21	3
	BI	1	13	6

Fonte: o autor.

Tabela 20 – Melhor matriz de confusão do GLVQ linear em classificação binária do banco de dados 1 para metodologia de treinamento 3

GLVQ linear		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	3	3
	Falha	1	52

Fonte: o autor.

Tabela 21 – Melhor matriz de confusão do GLVQ linear em classificação ternária do banco de dados 1 para metodologia de treinamento 3

GLVQ linear		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	3	1	2
	AI	1	18	9
	BI	0	9	16

Fonte: o autor.

também, que há ocorrência de 5 falsos negativos onde 4 destes erros ocorrem em exemplos de falha de AI e 1 em exemplo de falha de BI.

Tabela 22 – Pior matriz de confusão do GLVQ linear em classificação binária do banco de dados 1 para metodologia de treinamento 3

GLVQ linear		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	3	5
	Falha	12	39

Fonte: o autor.

Por fim, compara-se as matrizes de confusão das metodologias 1 e 3. Observa-se, para as piores matrizes, que a metodologia 3 apresentou 17 erros onde 12 deles são falsos

Tabela 23 – Pior matriz de confusão do GLVQ linear em classificação ternária do banco de dados 1 para metodologia de treinamento 3

GLVQ linear		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	3	4	1
	AI	6	12	11
	BI	6	5	11

Fonte: o autor.

positivos e 5 deles são falsos negativos e que a metodologia 1 apresentou 14 erros onde apenas 1 é falso positivo e 13 deles são falsos negativos.

Comparou-se, posteriormente, o algoritmo GRLVQ. As matrizes de confusão obtidas a partir da maior taxa de acerto binária na metodologia 1 são mostradas nas Tabelas 24 e 25. Já as matrizes de confusão obtidas da menor taxa de acerto são mostradas nas Tabelas 26 e 27.

A partir das Tabelas 24 e 25, verifica-se que não há ocorrência de falsos positivos. Verifica-se, também, que há uma ocorrência de falso negativo onde este erro ocorre em exemplo de falha de alta impedância.

Tabela 24 – Melhor matriz de confusão do GRLVQ em classificação binária do banco de dados 1 para metodologia de treinamento 1

GRLVQ		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	1	1
	Falha	0	57

Fonte: o autor.

Tabela 25 – Melhor matriz de confusão do GRLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 1

GRLVQ		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	1	1	0
	AI	0	21	4
	BI	0	13	19

Fonte: o autor.

A partir das Tabelas 26 e 27, verifica-se que não há ocorrência de falsos positivos. Verifica-se, também, que há ocorrência de 13 falsos negativos onde 12 destes erros ocorrem em exemplos de falha de alta impedância e apenas um exemplo de falha de baixa impedância.

Observa-se, também, que a elevação da quantidade de erros de classificação ocorre apenas no sentido de falsos estados de normalidade ao comparar as taxas de máximo e de

Tabela 26 – Pior matriz de confusão do GRLVQ em classificação binária do banco de dados 1 para metodologia de treinamento 1

GRLVQ		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	2	13
	Falha	0	44

Fonte: o autor.

Tabela 27 – Pior matriz de confusão do GRLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 1

GRLVQ		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	2	12	1
	AI	0	20	4
	BI	0	13	7

Fonte: o autor.

mínimo.

Já na metodologia 2, as matrizes de confusão obtidas a partir da maior taxa de acerto binária são mostradas nas Tabelas 28 e 29. Já as matrizes de confusão obtidas da menor taxa de acerto são mostradas nas Tabelas 30 e 31.

A partir das Tabelas 28 e 29, verifica-se que há ocorrência de apenas um erro de falso positivo onde uma amostra normal fora classificada como falha de BI. Verifica-se, também, que há duas ocorrências de falsos negativos onde estes erros ocorrem em exemplos de falha de AI.

Tabela 28 – Melhor matriz de confusão do GRLVQ em classificação binária do banco de dados 1 para metodologia de treinamento 2

GRLVQ		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	3	2
	Falha	1	53

Fonte: o autor.

Tabela 29 – Melhor matriz de confusão do GRLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 2

GRLVQ		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	3	2	0
	AI	0	20	4
	BI	1	17	12

Fonte: o autor.

A partir das Tabelas 30 e 31, verifica-se que há ocorrência de apenas um erro de

Tabela 30 – Pior matriz de confusão do GRLVQ em classificação binária do banco de dados 1 para metodologia de treinamento 2

GRLVQ		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	2	13
	Falha	1	43

Fonte: o autor.

Tabela 31 – Pior matriz de confusão do GRLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 2

GRLVQ		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	2	12	1
	AI	0	20	4
	BI	1	12	7

Fonte: o autor.

falso positivo onde uma amostra normal fora classificada como falha de BI. Verifica-se, também, que há ocorrência de 13 falsos negativos onde 12 destes erros ocorrem em exemplos de falha de alta impedância e apenas um exemplo de falha de baixa impedância.

Observa-se, ao comparar as piores matrizes de confusão das metodologias 1 e 2, que a única diferença entre elas é um erro de falso positivo onde uma amostra normal é classificada como falha de BI. Já entre as melhores matrizes de confusão entre estas metodologias, verifica-se a existência de um falso positivo na metodologia 2, enquanto este tipo de erro não ocorre na primeira metodologia.

Por último, na metodologia 3, as matrizes de confusão obtidas a partir maior taxa de acerto binária são mostradas nas Tabelas 32 e 33. Já as matrizes de confusão obtidas da menor taxa de acerto são mostradas nas Tabelas 34 e 35.

A partir das Tabelas 32 e 33, verifica-se que não há falsos positivos. Verifica-se, também, que há uma ocorrência de falso negativo onde este erro ocorre para uma predição de falha de BI.

Tabela 32 – Melhor matriz de confusão do GRLVQ em classificação binária do banco de dados 1 para metodologia de treinamento 3

GRLVQ		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	2	1
	Falha	0	56

Fonte: o autor.

A partir das Tabelas 34 e 35, verifica-se que há 8 falsos positivos onde estas amostras

Tabela 33 – Melhor matriz de confusão do GRLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 3

GRLVQ		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	2	0	1
	AI	0	18	7
	BI	0	14	17

Fonte: o autor.

Tabela 34 – Pior matriz de confusão do GRLVQ em classificação binária do banco de dados 1 para metodologia de treinamento 3

GRLVQ		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	2	7
	Falha	8	42

Fonte: o autor.

Tabela 35 – Pior matriz de confusão do GRLVQ em classificação ternária do banco de dados 1 para metodologia de treinamento 3

GRLVQ		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	2	5	2
	AI	8	13	8
	BI	0	7	14

Fonte: o autor.

normais foram preditas como de falha de AI. Verifica-se, também, que há ocorrência de 7 falsos negativos onde 5 destes erros ocorrem em exemplos de falha de AI e 2 em exemplos de BI.

Por fim, compara-se as matrizes de confusão das metodologias 1 e 3. Observa-se, para as piores matrizes, que a metodologia 3 apresentou falsos positivos enquanto isso não ocorre na primeira, porém a metodologia 3 errou menos que a primeira em falsos negativos. Já entre as melhores matrizes destas metodologias, ambas apresentaram apenas um falso negativo onde a metodologia 1 classificou como normal um exemplo de falha de AI e a metodologia 3 classificou como normal um exemplo de falha de BI.

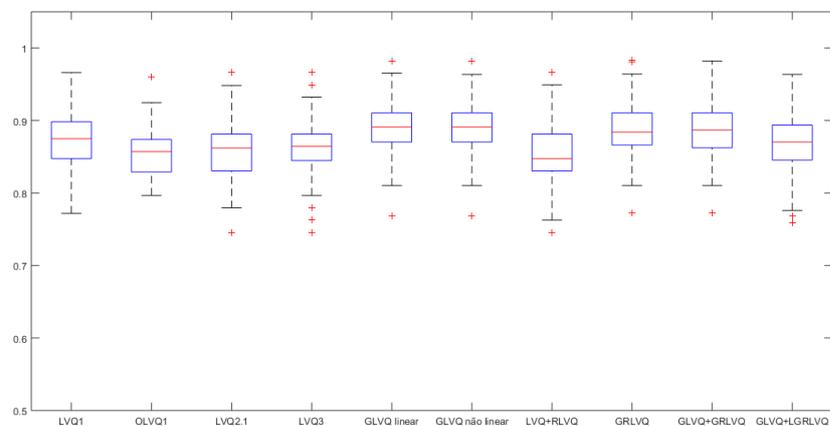
Após a análise destas matrizes, percebe-se que a sensibilidade apresentou altas taxas nas 3 metodologias. Já a especificidade variou demasiadamente devido ao viés causado pelo desbalanceamento entre classes.

Então, verificou-se as estatísticas sensibilidade e especificidade obtidas pelos classificadores nas 3 metodologias. As taxas encontradas de sensibilidade são ilustradas na Figura 28 e comprova-se os elevados índices em todas as classificações. Já a especificidade, ilustrada na Figura 29, apresentou valores indesejados por todos os classificadores onde o GRLVQ se

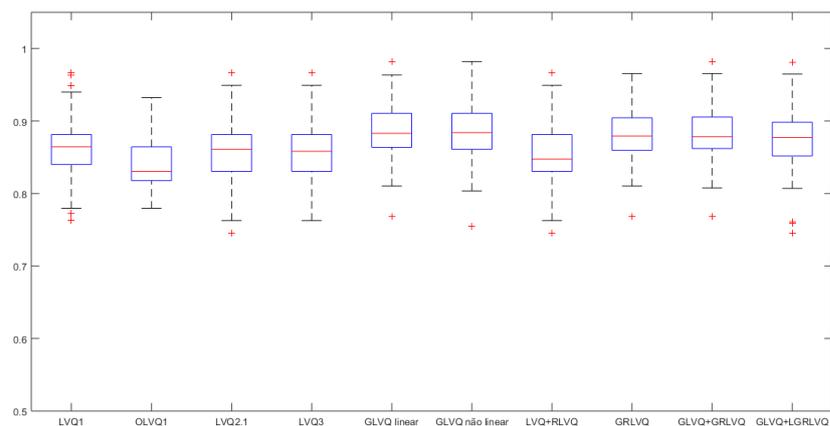
apresentou entre os melhores.

Figura 28 – Taxas de sensibilidade encontradas para o banco de dados 1

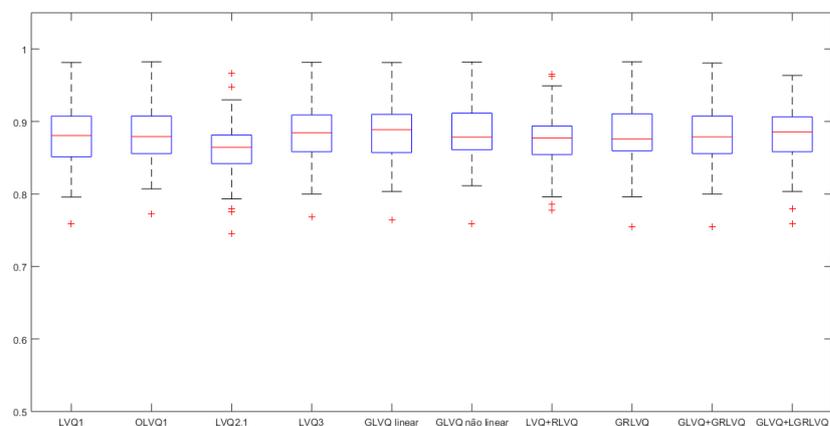
(a) Metodologia de Treinamento I



(b) Metodologia de Treinamento II



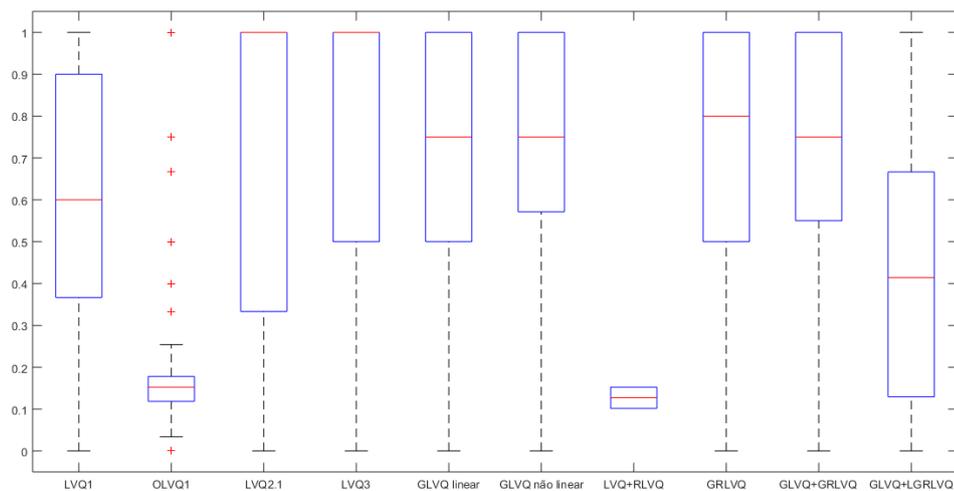
(c) Metodologia de Treinamento III



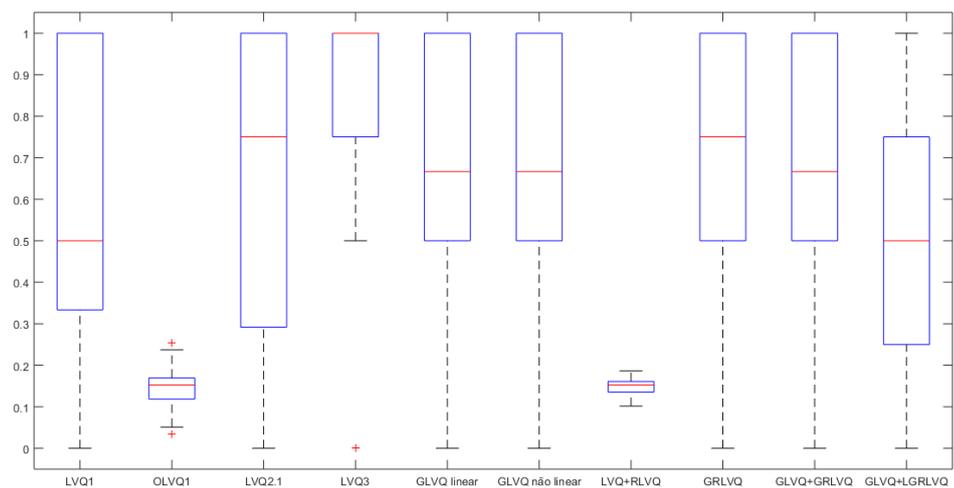
Fonte: o autor.

Figura 29 – Taxas de especificidade encontradas para o banco de dados 1

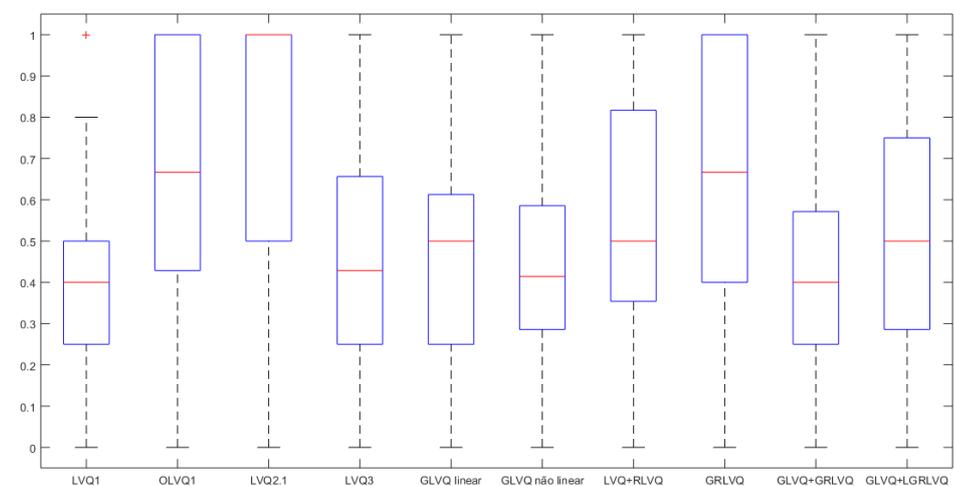
(a) Metodologia de Treinamento I



(b) Metodologia de Treinamento II



(c) Metodologia de Treinamento III



Fonte: o autor.

Sobre a especificidade, percebe-se, ao comparar as matrizes de confusão entre o GIVQ linear e GRLVQ, que o algoritmo GLVQ linear apresentou mais erros do tipo falso positivo e isto exerce influência nesta métrica.

Conseqüentemente, concluiu-se que, entre os dois algoritmos considerados como melhores candidatos para a classificação do BD1, o GRLVQ é o de melhor aptidão.

A deficiência dos resultados encontrados na especificidade é o principal incentivador para a inserção de amostras artificiais tal como feito no BD2.

Após a definição do GRLVQ como o melhor algoritmo para o BD1, os valores dos parâmetros que foram obtidos a partir das estatísticas dos resultados serão avaliadas na próxima seção. Para a etapa não supervisionada, os valores dos critérios de validação de agrupamentos serão avaliados. Já para a etapa de treinamento supervisionado, os vetores de relevância utilizados na métrica de dissimilaridade deste algoritmo serão avaliados.

### 7.1.3 Resultados secundários

Nesta seção, verificou-se os valores de parâmetros que foram obtidos a partir dos resultados encontrados das taxas de acerto binárias máxima, mediana e mínima do GRLVQ.

Primeiramente, aferiu-se as distribuições de protótipos por classe utilizada em cada metodologia que influenciaram as taxas de acertos citadas. Estas distribuições são mostradas na Tabela 36.

Tabela 36 – Sugestões de distribuição de protótipos para as taxas de acerto do banco de dados 1

<b>Met. 1</b>	<b>NO</b>	<b>AI</b>	<b>BI</b>	<b>Met. 2</b>	<b>NO</b>	<b>AI</b>	<b>BI</b>
<b>Máxima</b>	3	4	3	<b>Máxima</b>	2	4	3
<b>Mediana</b>	2	3	4	<b>Mediana</b>	2	3	4
<b>Mínima</b>	1	4	4	<b>Mínima</b>	1	4	4

Fonte: o autor.

A partir desta tabela, verifica-se que a distribuição por classe que gerou as taxas mínimas e medianas, entre estas duas metodologias, são idênticas. Porém, observa-se que a metodologia 1 (que apresentou melhores taxas de acertos) utilizou a mesma quantidade de protótipos para as classes de falha de AI e de BI e, para a classe normal, utilizou um protótipo a mais.

Além disso, verificou-se a sugestão reportada por cada critério de validação com o objetivo de aferir quais sugeriram a quantidade de protótipos que gerou o melhor resultado

de classificação. Estas sugestões são mostradas nas Tabelas 37 e 38 para, respectivamente, as metodologias 1 e 2.

Tabela 37 – Sugestões e moda de distribuição de protótipos para a maior taxa de acerto do banco de dados 1 para metodologia de treinamento 1

	<b>DB</b>	<b>Dunn</b>	<b>S</b>	<b>CH</b>	<b>AIC</b>	<b>BIC</b>	<b>FPE</b>	<b>MDL</b>	<b>MODA</b>
<b>Normal</b>	6	3	3	6	3	1	3	3	<b>3</b>
<b>Alta imp</b>	4	4	4	5	8	4	5	5	<b>4</b>
<b>Baixa imp</b>	3	3	3	3	7	3	5	5	<b>3</b>

Fonte: o autor.

Tabela 38 – Sugestões e moda de distribuição de protótipos para a maior taxa de acerto do banco de dados 1 para metodologia de treinamento 2

	<b>DB</b>	<b>Dunn</b>	<b>S</b>	<b>CH</b>	<b>AIC</b>	<b>BIC</b>	<b>FPE</b>	<b>MDL</b>	<b>MODA</b>
<b>Normal</b>	2	2	2	2	3	1	2	3	<b>2</b>
<b>Alta imp</b>	4	4	4	5	6	4	5	5	<b>4</b>
<b>Baixa imp</b>	5	3	3	4	9	3	5	9	<b>3</b>

Fonte: o autor.

A partir destas tabelas, verifica-se que, entre os 8 critérios de validação considerados neste trabalho, apenas os índices de validação relativos Dunn e Silhuetas acertaram a quantidade de protótipos por classe que geraram as maiores taxas de acertos nas metodologias de treinamento propostas.

Aferiu-se, também, as relevâncias encontradas para os atributos durante a etapa de treinamento do GRLVQ para as 3 metodologias propostas. Os vetores de ponderação  $\lambda$  obtidos na metodologia 1 são mostrados na Tabela 39. Para a metodologia 2, os vetores de ponderação são mostrados na Tabela 40. Já para a metodologia 3, estes vetores são mostrados na Tabela 41.

A partir destas 3 tabelas, verifica-se, em todas estas as metodologias, que o atributo  $3f_s$  fora considerado como o de maior relevância e que o atributo  $1,5f_s$  fora considerado como o de menor relevância.

Assim, encerra-se a análise dos resultados obtidos no banco de dados 1 e estes serão utilizados na comparação do banco de dados (BD2) que possui classes balanceadas para tarefa de classificação ternária.

Finalmente, os resultados obtidos através do segundo banco de dados serão apresentados e discutidos na próxima seção.

Tabela 39 – Valores encontrados para os elementos do vetor de ponderação para o banco de dados 1 e metodologia de treinamento 1

	$0,5f_s$	$1,5f_s$	$2,5f_s$	$3f_s$	$5f_s$	$7f_s$
<b>Máxima</b>	0,1373	0,0952	0,1757	<b>0,2277</b>	0,1689	0,1952
<b>Mediana</b>	0,1490	0,1308	0,1703	<b>0,2735</b>	0,1234	0,1529
<b>Mínima</b>	0,1266	0,0838	0,1900	<b>0,2460</b>	0,1637	0,1899

Fonte: o autor.

Tabela 40 – Valores encontrados para os elementos do vetor de ponderação para o banco de dados 1 e metodologia de treinamento 2

	$0,5f_s$	$1,5f_s$	$2,5f_s$	$3f_s$	$5f_s$	$7f_s$
<b>Máxima</b>	0,1939	0,0794	0,1700	<b>0,2148</b>	0,1337	0,2083
<b>Mediana</b>	0,1170	0,1008	0,1799	<b>0,2672</b>	0,1150	0,2200
<b>Mínima</b>	0,1471	0,1196	0,1576	<b>0,2623</b>	0,1504	0,1631

Fonte: o autor.

Tabela 41 – Valores encontrados para os elementos do vetor de ponderação para o banco de dados 1 e metodologia de treinamento 3

	$0,5f_s$	$1,5f_s$	$2,5f_s$	$3f_s$	$5f_s$	$7f_s$
<b>Máxima</b>	0,1465	0,1283	0,1861	<b>0,2466</b>	0,1056	0,1869
<b>Mediana</b>	0,1443	0,1223	0,1788	<b>0,2102</b>	0,1479	0,1964
<b>Mínima</b>	0,1410	0,1232	0,1885	<b>0,2126</b>	0,1402	0,1945

Fonte: o autor.

## 7.2 Banco de dados 2

Os resultados obtidos dos classificadores LVQ utilizados neste trabalho para o banco de dados 2 encontram-se divididos em quatro seções. A Seção 7.2.1 apresenta, em diagramas de caixa (*boxplots*), e discute sobre as taxas de acertos encontradas por cada algoritmo LVQ em classificação ternária e a representação destas em classificação binária. As matrizes de confusão e suas estatísticas associadas são apresentadas na Seção 7.2.2 para os classificadores de melhores rendimentos na classificação. Os resultados específicos sobre o classificador de melhor aptidão são apresentados na Seção 7.2.3. Para evitar sobrecarga de informação, as estatísticas encontradas, em formato de tabelas, sobre as taxas de acertos em classificação ternária e binária, sensibilidade e especificidade são mostradas apenas no Apêndice A.

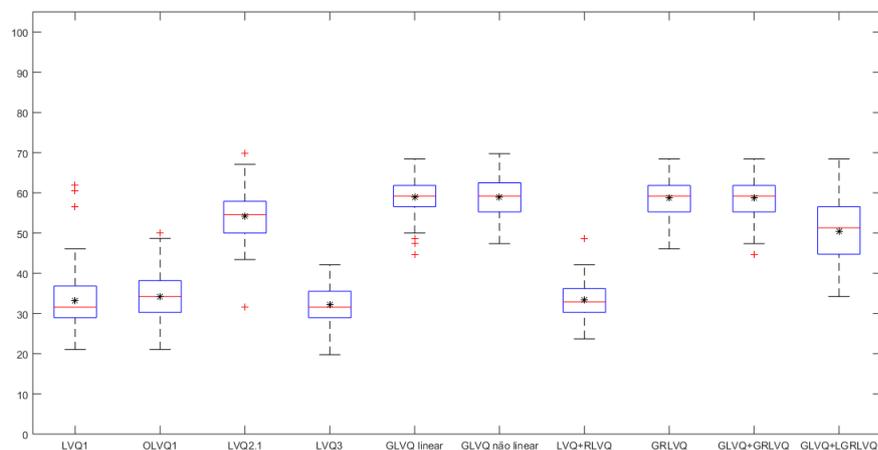
### 7.2.1 Taxas de acertos

Inicialmente, verificou-se o desempenho dos classificadores em função das taxas de acertos encontradas por cada classificador LVQ em cada metodologia de treinamento proposta.

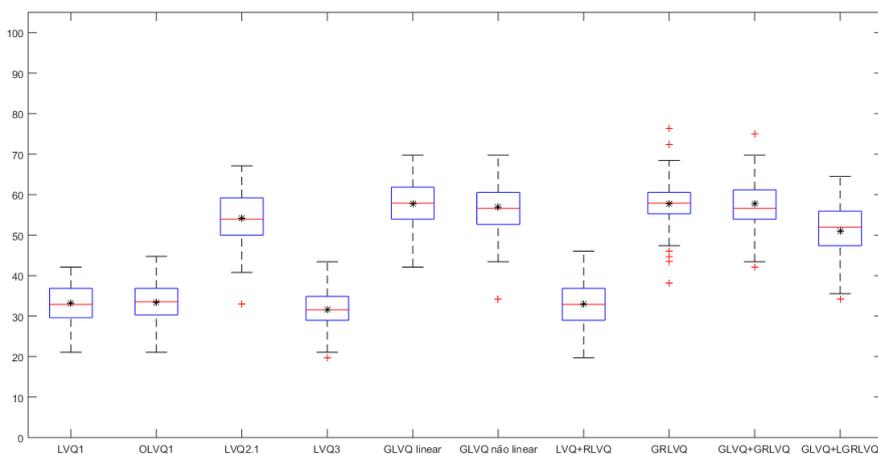
Os resultados encontrados na classificação ternária são ilustrados na Figura 30. Já o mapeamento destes resultados para classificação binária são ilustrados na Figura 31.

Figura 30 – Taxas de acertos encontradas para classificação ternária do banco de dados 2

(a) Metodologia de Treinamento I



(b) Metodologia de Treinamento II



(c) Metodologia de Treinamento III

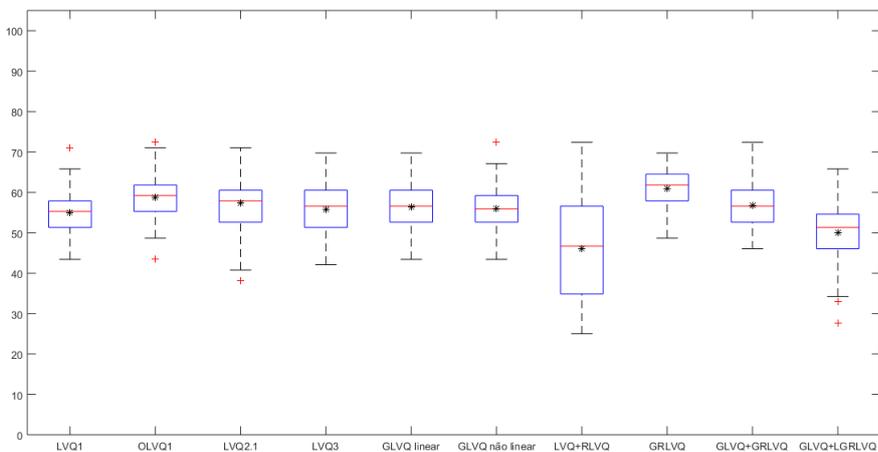
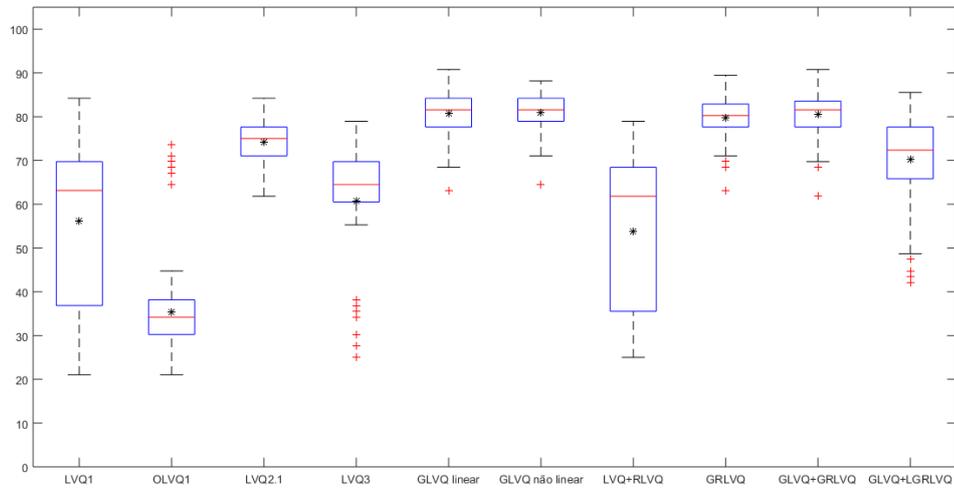
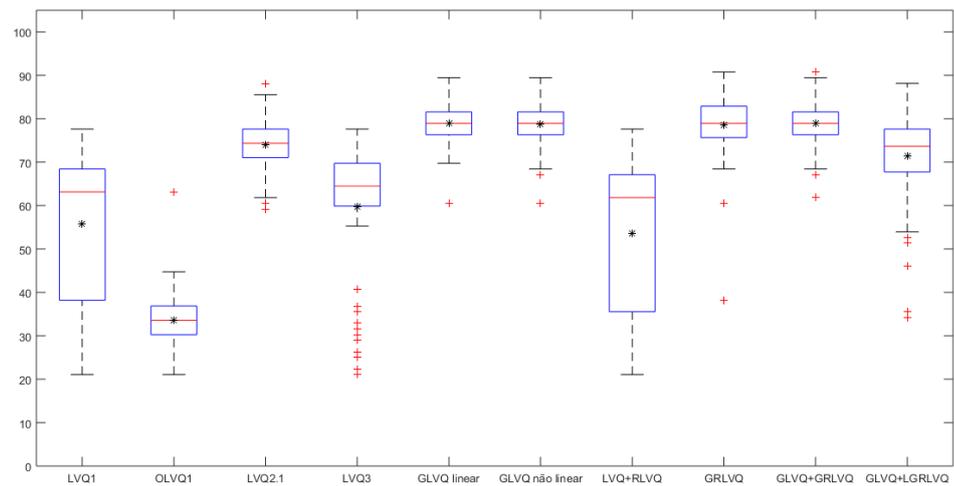


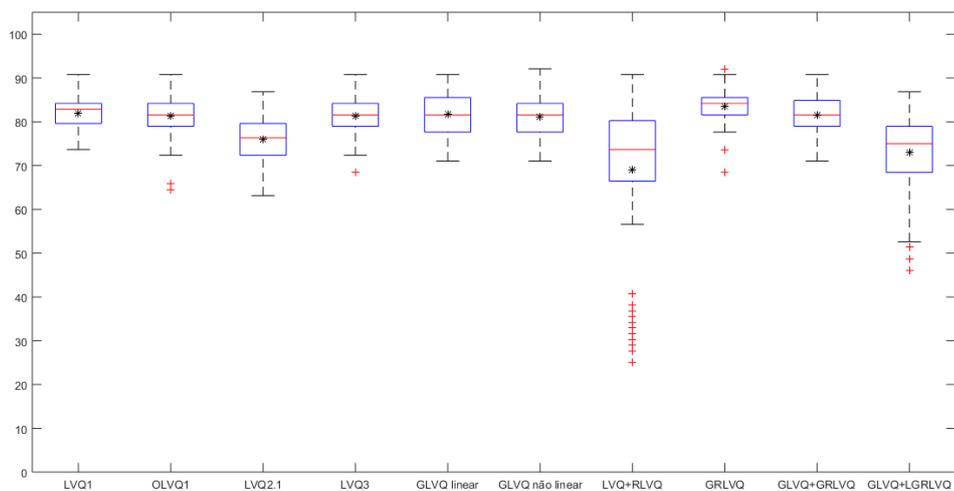
Figura 31 – Taxas de acertos mapeadas para classificação binária do banco de dados 2  
 (a) Metodologia de Treinamento I



(b) Metodologia de Treinamento II



(c) Metodologia de Treinamento III



Fonte: o autor.

As estatísticas encontradas a partir da classificação ternária ao utilizar a metodologia de treinamento 1 são mostradas na Tabela 85 e estes resultados mapeados em classificação binária são mostradas na Tabela 88. Da mesma forma, as estatísticas das taxas de acertos para a metodologia de treinamento 2 são mostradas na Tabela 86 para classificação ternária e na Tabela 89 para classificação binária. Enquanto para a metodologia de treinamento 3, as respectivas estatísticas são mostradas nas Tabelas 87 e 90.

Através da comparação geral entre os classificadores para as três metodologias, verifica-se a variação do OLVQ1 é menor neste banco de dados e, assim como o banco de dados 1, este algoritmo apresenta baixas taxas de acertos (em comparação aos demais) nas metodologias 1 e 2, porém apresenta taxas próximas às taxas dos classificadores com maiores taxas obtidas na metodologia 3.

Neste banco de dados, as taxas máximas são atenuadas. Já as taxas mínimas são atenuadas apenas para os classificadores que são baseados na atualização de dois protótipos por iteração. Além disso, verifica-se as taxas mínimas dos classificadores LVQ1, OLVQ1 e LVQ+RLVQ (que são baseados em apenas um protótipo por iteração) aumentaram consideravelmente.

Ao verificar as taxas de acerto binárias, verifica-se que os algoritmos GLVQ linear e GRLVQ são os algoritmos que apresentaram as melhores estatísticas entre os algoritmos nas metodologias 1 e 2. Já na metodologia 3, o algoritmo LVQ1 (que é o algoritmo mais simples) também é um forte candidato a melhor classificador.

Logo, como visto no banco de dados 1 que a especificidade é a métrica de maior variabilidade nos resultados obtidos, os algoritmos GLVQ linear, GRLVQ e LVQ1 serão avaliados quanto à especificidade como critério definitivo na definição do classificador de melhor aptidão.

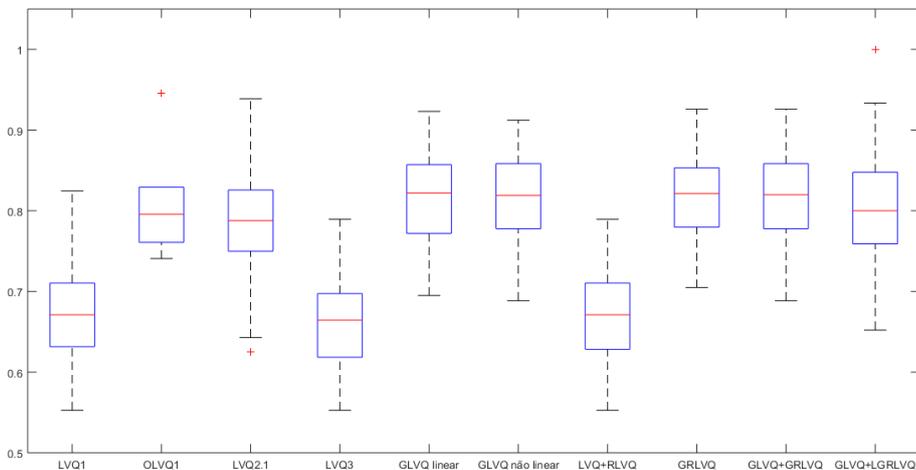
### ***7.2.2 Matrizes de confusão e estatísticas associadas***

Após a seleção dos algoritmos GLVQ linear e GRLVQ como classificadores mais adequados para as metodologias no BD1, verificou-se que estes continuaram como classificadores mais relevantes no BD2 para as metodologias 1 e 2. Além disso, ao verificar as especificidades obtidas pelos classificadores para o segundo banco de dados que estão ilustradas na Figura 33, verificou-se que o algoritmo LVQ1 apresentou os melhores resultados desta métrica na metodologia 3. Logo, a metodologia 3 será analisada pelos algoritmos LVQ1 e GRLVQ.

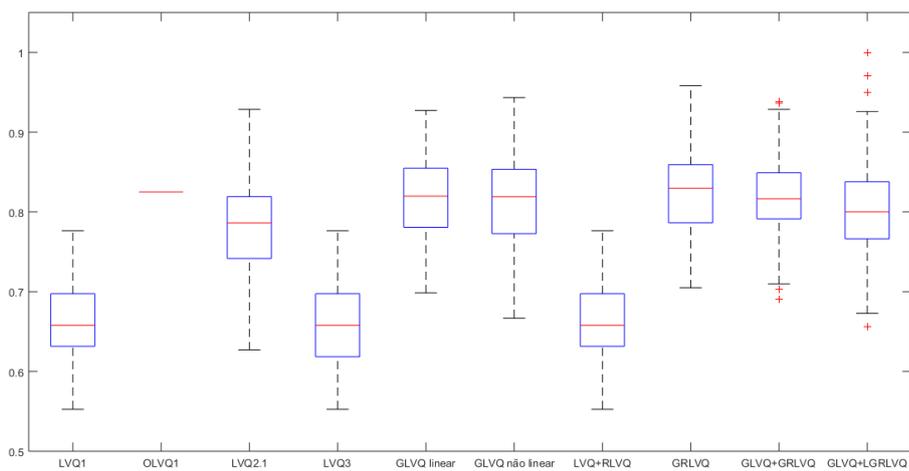
Primeiramente, analisou-se o algoritmo GLVQ linear para as metodologias 1 e 2.

Figura 32 – Taxas de sensibilidade encontradas para o banco de dados 2.

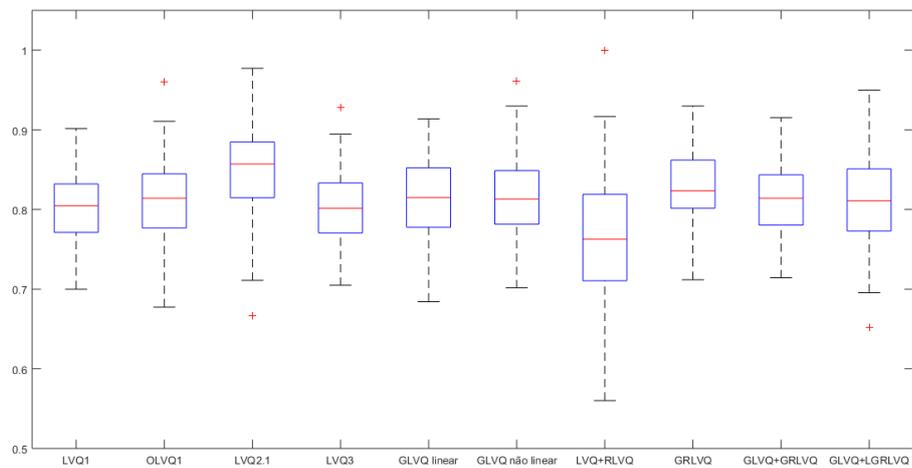
(a) Treinamento I



(b) Treinamento II



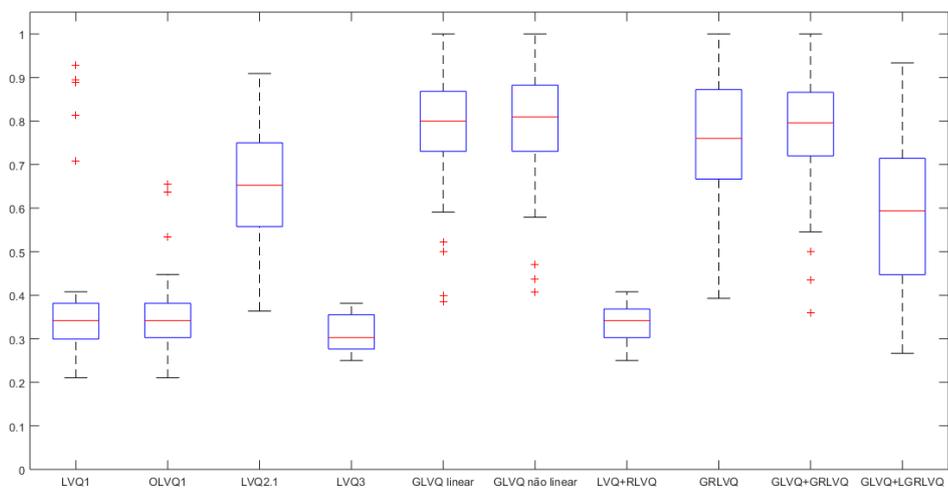
(c) Treinamento III



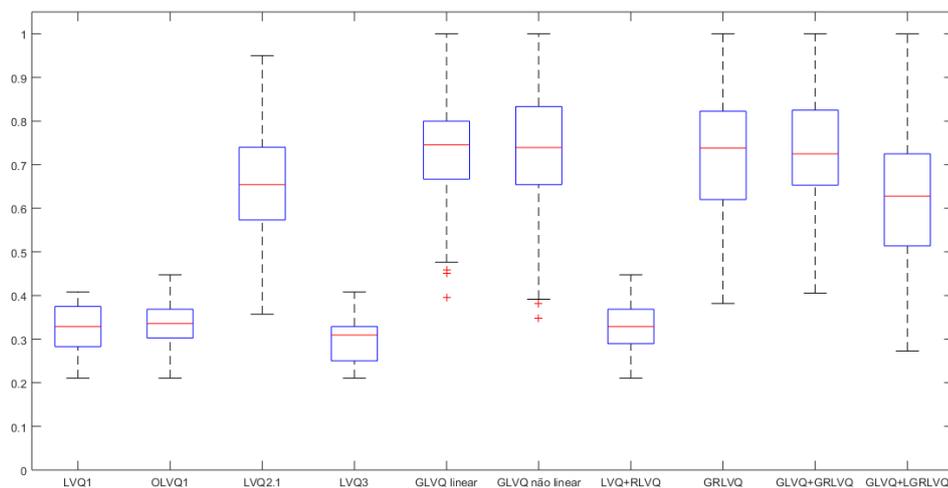
Fonte: o autor.

Figura 33 – Taxas de especificidade encontradas para o banco de dados 2.

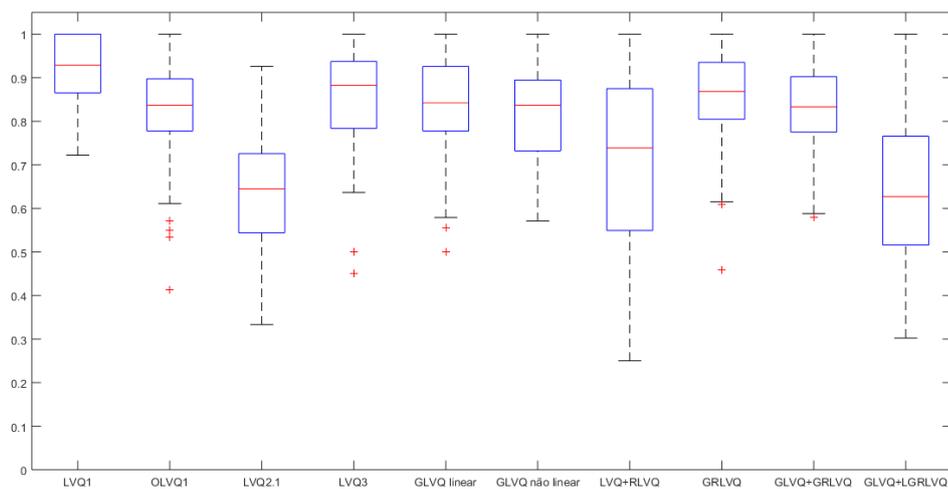
(a) Treinamento I



(b) Treinamento II



(c) Treinamento III



Fonte: o autor.

Tem-se, abaixo, a análise do GLVQ linear:

As matrizes de confusão obtidas a partir da maior taxa de acerto binária na metodologia 1 são mostradas nas Tabelas 42 e 43. Já as matrizes de confusão obtidas da menor taxa de acerto são mostradas nas Tabelas 44 e 45.

A partir das Tabelas 42 e 43, verifica-se que ocorrem 2 falsos positivos onde 2 exemplos da classe normal são preditos como falha de alta impedância. Verifica-se, também, que há 5 ocorrências de falsos negativos onde estes erros ocorrem em exemplos de falha de AI.

Tabela 42 – Melhor matriz de confusão do GLVQ linear em classificação binária do banco de dados 2 para metodologia de treinamento 1

GLVQ linear		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	21	5
	Falha	2	48

Fonte: o autor.

Tabela 43 – Melhor matriz de confusão do GLVQ linear em classificação ternária do banco de dados 2 para metodologia de treinamento 1

GLVQ linear		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	21	5	0
	AI	2	17	2
	BI	0	18	11

Fonte: o autor.

A partir das Tabelas 44 e 45, verifica-se a ocorrência de 16 falsos positivos onde 10 destes exemplos da classe normal são preditos como falha de baixa impedância e outros 6 como falha de baixa impedância. Verifica-se, também, que há ocorrência de 12 falsos negativos onde 8 destes erros ocorrem em exemplos de falha de alta impedância e em 4 da classe falha de baixa impedância. Além disso, observa-se o surgimento de falsos positivos em predições de falhas de BI ao comparar as taxas de máximo e de mínimo.

Tabela 44 – Pior matriz de confusão do GLVQ linear em classificação binária do banco de dados 2 para metodologia de treinamento 1

GLVQ linear		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	10	12
	Falha	16	38

Fonte: o autor.

Tabela 45 – Pior matriz de confusão do GLVQ linear em classificação ternária do banco de dados 2 para metodologia de treinamento 1

GLVQ linear		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	10	8	4
	AI	10	13	6
	BI	6	5	14

Fonte: o autor.

Já na metodologia 2, as matrizes de confusão obtidas a partir da maior taxa de acerto binária são mostradas nas Tabelas 46 e 47. Já as matrizes de confusão obtidas da menor taxa de acerto são mostradas nas Tabelas 48 e 49.

A partir das Tabelas 46 e 47, verifica-se que ocorreram 4 falsos positivos onde duas amostras normais foram classificadas como falha de AI e duas foram classificadas como falha de BI. Verifica-se, também, que ocorreram 4 falsos negativos onde 3 destes erros ocorreram para amostras da classe falha de AI e um erro para uma amostra classe falha de BI.

Tabela 46 – Melhor matriz de confusão do GLVQ linear em classificação binária do banco de dados 2 para metodologia de treinamento 2

GLVQ linear		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	17	4
	Falha	4	51

Fonte: o autor.

Tabela 47 – Melhor matriz de confusão do GLVQ linear em classificação ternária do banco de dados 2 para metodologia de treinamento 2

GLVQ linear		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	17	3	1
	AI	2	16	5
	BI	2	14	16

Fonte: o autor.

A partir das Tabelas 48 e 49, verifica-se que ocorreram 23 falsos positivos onde 16 amostras normais foram classificadas como falha de AI e 7 foram classificadas como falha de BI. Verifica-se, também, que ocorreram 7 falsos negativos onde 2 destes erros ocorreram para amostras da classe falha de AI e 5 erros para amostras da classe falha de BI.

Observa-se, ao comparar as piores matrizes de confusão das metodologias 1 e 2, que falsos positivos foram os erros que mais ocorreram. Já entre as melhores matrizes de confusão

Tabela 48 – Pior matriz de confusão do GLVQ linear em classificação binária do banco de dados 2 para metodologia de treinamento 2

GLVQ linear		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	15	7
	Falha	23	31

Fonte: o autor.

Tabela 49 – Pior matriz de confusão do GLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 2

GLVQ linear		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	15	2	5
	AI	16	6	7
	BI	7	3	15

Fonte: o autor.

entre estas metodologias, verifica-se que falsos negativos foram os erros que mais ocorreram.

Analisou-se, posteriormente, o algoritmo GRLVQ para as metodologias 1 e 2:

As matrizes de confusão obtidas a partir maior taxa de acerto binária na metodologia 1 são mostradas nas Tabelas 50 e 51. Já as matrizes de confusão obtidas da menor taxa de acerto são mostradas nas Tabelas 52 e 53.

A partir das Tabelas 50 e 51, verifica-se que ocorrem 2 falsos positivos onde 1 exemplo da classe normal fora predito como falha de alta impedância e outro como falha de baixa impedância. Verifica-se, também, que há 6 ocorrências de falsos negativos onde estes erros ocorrem em 3 exemplos da classe falha de alta impedância e em 3 exemplos da classe falha de baixa impedância.

A partir das Tabelas 52 e 53, verifica-se a ocorrência de 17 falsos positivos onde 11 destes exemplos da classe normal são preditos como falha de baixa impedância e outros 6 como falha de baixa impedância. Verifica-se, também, que há ocorrência de 11 falsos negativos onde 6 destes erros ocorrem em exemplos de falha de alta impedância e em 5 da classe falha de baixa impedância.

Tabela 50 – Melhor matriz de confusão do GRLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 1

GRLVQ		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	17	6
	Falha	2	51

Fonte: o autor.

Tabela 51 – Melhor matriz de confusão do GRLVQ em classificação binária do banco de dados 2 para metodologia de treinamento 1

GRLVQ		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	17	3	3
	AI	1	17	7
	BI	1	11	16

Fonte: o autor.

Tabela 52 – Pior matriz de confusão do GRLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 1

GRLVQ		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	11	11
	Falha	17	37

Fonte: o autor.

Tabela 53 – Pior matriz de confusão do GRLVQ em classificação binária do banco de dados 2 para metodologia de treinamento 1

GRLVQ		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	11	6	5
	AI	11	13	5
	BI	6	5	14

Fonte: o autor.

Observa-se, também, ao comparar a melhor e a pior matriz de confusão, que há elevação considerável de falsos positivos na pior matriz de confusão.

Já na metodologia 2, as matrizes de confusão obtidas a partir maior taxa de acerto binária são mostradas nas Tabelas 54 e 55. Já as matrizes de confusão obtidas da menor taxa de acerto são mostradas nas Tabelas 56 e 57.

A partir das Tabelas 54 e 55, verifica-se que ocorreram 5 falsos positivos onde duas amostras normais foram classificadas como falha de AI e três foram classificadas como falha de BI. Verifica-se, também, que ocorreram 2 falsos negativos onde 1 destes erros ocorreu para uma amostra da classe falha de AI e o outro para uma amostra classe falha de BI.

Tabela 54 – Melhor matriz de confusão do GRLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 2

GRLVQ		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	23	2
	Falha	5	46

Fonte: o autor.

Tabela 55 – Melhor matriz de confusão do GRLVQ em classificação binária do banco de dados 2 para metodologia de treinamento 2

GRLVQ		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	23	1	1
	AI	2	20	9
	BI	3	5	12

Fonte: o autor.

Tabela 56 – Pior matriz de confusão do GRLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 2

GRLVQ		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	29	0
	Falha	47	0

Fonte: o autor.

Tabela 57 – Pior matriz de confusão do GRLVQ em classificação binária do banco de dados 2 para metodologia de treinamento 2

GRLVQ		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	29	0	0
	AI	24	0	0
	BI	23	0	0

Fonte: o autor.

A partir das Tabelas 56 e 57, verifica-se que ocorreram 47 falsos positivos onde 24 amostras normais foram classificadas como falha de AI e 23 foram classificadas como falha de BI em um cenário onde apenas amostras verdadeiramente normais foram testadas.

Logo, ao comparar os algoritmo GLVQ linear e GRLVQ para as metodologias 1 e 2, verificou-se que o GLVQ apresentou menos falsos positivos em predições para classe de falha de baixa impedância e, também, menos falsos negativos para exemplos verdadeiramente da classe falha de baixa impedância.

Consequentemente, o algoritmo GLVQ linear é o classificador de melhor aptidão para representar o conjunto de dados 2 nas metodologias 1 e 2.

Posteriormente, analisou-se os algoritmos LVQ1 e GRLVQ para a metodologia 3. Tem-se, abaixo, a análise do LVQ1:

As matrizes de confusão obtidas a partir maior taxa de acerto binária na metodologia 3 são mostradas nas Tabelas 58 e 59. Já as matrizes de confusão obtidas da menor taxa de acerto são mostradas nas Tabelas 60 e 61.

A partir das Tabelas 58 e 59, verifica-se que ocorreu apenas um falso positivo onde

este exemplo da classe normal fora predito como falha de alta impedância. Verifica-se, também, que há 6 ocorrências de falsos negativos onde 5 destes erros ocorrem em exemplos de falha de alta impedância e um erro em falha de baixa impedância.

A partir das Tabelas 60 e 61, verifica-se a ocorrência de 2 falsos positivos onde 1 ocorreu na predição de classe de falha de alta impedância e o outro ocorreu na predição de classe de falha de baixa impedância. Verifica-se, também, que há ocorrência de 18 falsos negativos onde 10 destes erros ocorrem em exemplos de falha de alta impedância e em 8 da classe falha de baixa impedância.

Tabela 58 – Melhor matriz de confusão do LVQ1 em classificação ternária do banco de dados 2 para metodologia de treinamento 3

LVQ1		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	14	6
	Falha	1	55

Fonte: o autor.

Tabela 59 – Melhor matriz de confusão do LVQ1 em classificação binária do banco de dados 2 para metodologia de treinamento 3

LVQ1		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	14	5	1
	AI	1	24	7
	BI	0	8	16

Fonte: o autor.

Tabela 60 – Pior matriz de confusão do LVQ1 em classificação ternária do banco de dados 2 para metodologia de treinamento 3

LVQ1		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	11	18
	Falha	2	45

Fonte: o autor.

Observa-se, também, ao comparar a melhor e a pior matriz de confusão, que a elevação de ocorrência de erros se dá no sentido de falsos negativos.

Finalmente, analisou-se o algoritmo GRLVQ para a metodologia 3:

As matrizes de confusão obtidas a partir da maior taxa de acerto binária na metodologia 3 são mostradas nas Tabelas 62 e 63. Já as matrizes de confusão obtidas da menor taxa

Tabela 61 – Pior matriz de confusão do LVQ1 em classificação binária do banco de dados 2 para metodologia de treinamento 3

LVQ1		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	11	10	8
	AI	1	20	4
	BI	1	15	6

Fonte: o autor.

de acerto são mostradas nas Tabelas 64 e 65.

A partir das Tabelas 62 e 63, verifica-se que ocorreram dois falsos positivos onde estes exemplos da classe normal foram preditos como falha de baixa impedância. Verifica-se, também, que há 4 ocorrências de falsos negativos onde estes erros ocorreram em exemplos de falha de alta impedância.

A partir das Tabelas 64 e 65, verifica-se a ocorrência de 13 falsos positivos onde 6 ocorreram nas predições de classe de falha de alta impedância e 7 ocorreram nas predições de classe de falha de baixa impedância. Verifica-se, também, que há ocorrência de 11 falsos negativos onde 5 destes erros ocorrem em exemplos de falha de alta impedância e em 6 da classe falha de baixa impedância.

Finalmente, ao comparar as matrizes de confusão dos algoritmos LVQ1 e GRLVQ para a metodologia 3, verificou-se que o GRLVQ possui a melhor matriz de confusão para a máxima taxa de acertos encontrada e que o LVQ1 apresentou a melhor matriz de confusão para a taxa mínima encontrada.

Logo, após a definição do GLVQ linear como o melhor algoritmo para o BD2 nas metodologias de treinamento 1 e 2, os valores dos critérios de validação de agrupamentos que foram obtidos a partir da etapa de treinamento não supervisionada serão avaliados na próxima seção.

Já para a metodologia de treinamento 3, após a definição do GRLVQ como o melhor algoritmo para este banco de dados, os vetores de relevância utilizados na métrica de dissimilaridade deste algoritmo serão avaliados em sequência.

Tabela 62 – Melhor matriz de confusão do GRLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 3

GRLVQ		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	17	4
	Falha	2	53

Fonte: o autor.

Tabela 63 – Melhor matriz de confusão do GRLVQ em classificação binária do banco de dados 2 para metodologia de treinamento 3

LVQ1		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	17	4	0
	AI	0	18	5
	BI	2	14	16

Fonte: o autor.

Tabela 64 – Pior matriz de confusão do GRLVQ em classificação ternária do banco de dados 2 para metodologia de treinamento 3

GRLVQ		Classe Verdadeira	
		Normal	Falha
Classe Predita	Normal	11	11
	Falha	13	41

Fonte: o autor.

Tabela 65 – Pior matriz de confusão do GRLVQ em classificação binária do banco de dados 2 para metodologia de treinamento 3

GRLVQ		Classe Verdadeira		
		NO	AI	BI
Classe Predita	NO	11	5	6
	AI	6	15	8
	BI	7	4	14

Fonte: o autor.

### 7.2.3 Resultados secundários

Nesta seção, verificou-se os valores de parâmetros que foram obtidos a partir dos resultados encontrados das taxas de acertos binárias máxima, mediana e mínima do GLVQ linear para as metodologias 1 e 2, e do GRLVQ para a metodologia 3.

Analisou-se, primeiramente, o algoritmo GLVQ linear para as metodologias citadas onde aferiu-se as distribuições de protótipos por classe utilizada em cada metodologia que influenciaram as taxas de acertos citadas. Estas distribuições são mostradas na Tabela 66.

A partir desta tabela, verifica-se que a distribuição por classe que gerou as taxas mínimas e medianas, entre estas duas metodologias, são idênticas. Porém, observa-se que a

Tabela 66 – Sugestões de distribuição de protótipos para as taxas de acerto do banco de dados 2

Met. 1	NO	AI	BI	Met. 2	NO	AI	BI
<b>Máxima</b>	7	3	4	<b>Máxima</b>	5	3	4
<b>Mediana</b>	4	3	4	<b>Mediana</b>	4	3	4
<b>Mínima</b>	4	3	4	<b>Mínima</b>	4	3	4

Fonte: o autor.

metodologia 1 (que apresentou melhores taxas de acertos) utilizou a mesma quantidade de protótipos para as classes de falha de AI e de BI e, para a classe normal, utilizou dois protótipos a mais.

Além disso, verificou-se a sugestão reportada por cada critério de validação com o objetivo de aferir quais sugeriram a quantidade de protótipos que gerou o melhor resultado de classificação. Estas sugestões são mostradas nas Tabelas 67 e 68 para, respectivamente, as metodologias 1 e 2.

Tabela 67 – Sugestões e moda de distribuição de protótipos para a maior taxa de acerto do banco de dados 2 para metodologia de treinamento 1

	<b>DB</b>	<b>Dunn</b>	<b>S</b>	<b>CH</b>	<b>AIC</b>	<b>BIC</b>	<b>FPE</b>	<b>MDL</b>	<b>MODA</b>
<b>Normal</b>	7	7	7	2	7	1	5	5	<b>7</b>
<b>Alta imp</b>	3	3	3	3	8	3	8	8	<b>3</b>
<b>Baixa imp</b>	5	2	4	4	8	4	7	7	<b>4</b>

Fonte: o autor.

Tabela 68 – Sugestões e moda de distribuição de protótipos para a maior taxa de acerto do banco de dados 2 para metodologia de treinamento 2

	<b>DB</b>	<b>Dunn</b>	<b>S</b>	<b>CH</b>	<b>AIC</b>	<b>BIC</b>	<b>FPE</b>	<b>MDL</b>	<b>MODA</b>
<b>Normal</b>	7	5	5	3	5	2	5	5	<b>5</b>
<b>Alta imp</b>	3	3	3	3	8	4	5	5	<b>3</b>
<b>Baixa imp</b>	4	4	4	4	5	4	5	5	<b>4</b>

Fonte: o autor.

A partir destas tabelas, verifica-se que, entre os 8 critérios de validação considerados neste trabalho, apenas o índice de validação relativo Silhuetas acertou a quantidade de protótipos por classe que gerou a maior taxa de acertos na metodologia de treinamento 1. Já na metodologia de treinamento 2, assim como no banco de dados 1, os índices de validação relativos Dunn e Silhuetas acertaram a quantidade de protótipos ótima.

Posteriormente, aferiu-se as relevâncias encontradas para os atributos durante a etapa de treinamento do GRLVQ para a terceira metodologia proposta. Os vetores de ponderação  $\lambda$  obtidos na metodologia 3 são mostrados na Tabela 69.

A partir desta tabela, verifica-se que os atributos de relevância se aproximaram em relação aos vetores de ponderação apresentados para o banco de dados 1. Pode-se concluir que a inserção de amostras ruidosas tornou esta métrica de dissimilaridade mais próxima de ponderações igualitárias a todos protótipos.

Tabela 69 – Valores encontrados para os elementos do vetor de ponderação para o banco de dados 2 e metodologia de treinamento 3

	$0,5f_s$	$1,5f_s$	$2,5f_s$	$3f_s$	$5f_s$	$7f_s$
<b>Máxima</b>	0,1640	0,1301	0,1865	0,1869	0,1484	0,1841
<b>Mediana</b>	0,1766	0,1626	0,1557	0,1867	0,1903	0,1281
<b>Mínima</b>	0,1735	0,1551	0,1535	0,1800	0,1794	0,1586

Fonte: o autor.

Assim, encerra-se a análise dos resultados obtidos no BD2.

### 7.3 Conclusão

Neste capítulo, foram comparados os diferentes classificadores LVQ descritos nesta dissertação a partir das diferentes metodologias de treinamento consideradas. A Tabela 70 mostra um resumo dos resultados encontrados para o BD1. Nesta tabela há a definição dos dois melhores classificadores LVQ e suas respectivas taxas máximas de acertos e de sensibilidade em classificação binária. Já a Tabela 71 mostra um resumo dos resultados encontrados para o BD2.

Sobre os resultados encontrados, vale ressaltar que a definição dos melhores algoritmos fora baseada, além das taxas de acertos, nas métricas sensibilidade e especificidade e na relevância dos diferentes erros de classificação.

Tabela 70 – Tabela resumo de resultados encontrados para o BD1

	<b>Melhor</b>			<b>2o Melhor</b>		
	<b>Algoritmo</b>	<b>Tx. máx</b>	<b>Sens. máx</b>	<b>Algoritmo</b>	<b>Tx. máx</b>	<b>Sens. máx</b>
<b>Met. 1</b>	GRLVQ	98,3%	98,3%	GLVQ linear	96,6%	98,2%
<b>Met. 2</b>	GLVQ linear	94,9%	98,2%	GRLVQ	96,6%	96,5%
<b>Met. 3</b>	GRLVQ	98,3%	98,2%	GLVQ linear	93,2%	98,1%

Fonte: o autor.

Tabela 71 – Tabela resumo de resultados encontrados para o BD2

	<b>Melhor</b>			<b>2o Melhor</b>		
	<b>Algoritmo</b>	<b>Tx. máx</b>	<b>Sens. máx</b>	<b>Algoritmo</b>	<b>Tx. máx</b>	<b>Sens. máx</b>
<b>Met. 1</b>	GLVQ linear	90,8%	92,3%	GRLVQ	89,5%	92,6%
<b>Met. 2</b>	GLVQ linear	89,5%	92,7%	GRLVQ	90,8%	95,8%
<b>Met. 3</b>	GRLVQ	92,1%	93,0%	LVQ1	90,8%	90,2%

Fonte: o autor.

Além disso, destacam-se os resultados obtidos para o vetor de ponderação do GRLVQ

para o banco de dados original. Neste vetor, foram encontrados resultados de taxas máximas de acertos onde o atributo  $3f_s$  é o de maior relevância o atributo de  $1,5f_s$  é o de menor relevância.

Por fim, entre os resultados mostrados neste capítulo, destacam-se também os valores encontrados de  $K$  para os dois bancos de dados ao utilizar a metodologia de treinamento 1 que geraram as taxas máximas de acertos. Logo, estes são mostrados na Tabela 72 assim como os critérios de validação de agrupamentos que acertaram a quantidade necessária em todas as classes.

Tabela 72 – Tabela resumo dos valores ótimos de  $K$  para metodologia 1

<b>Banco de Dados</b>	<b>Melhor Algoritmo</b>	<b>NO</b>	<b>AI</b>	<b>BI</b>	<b>Critério de Validação</b>
BD1	GRLVQ	3	4	3	Dunn e Silhuetas
BD2	GLVQ	7	3	4	Silhuetas

Fonte: o autor.

## 8 CONCLUSÕES E PERSPECTIVAS

### 8.1 Objetivo Geral

Nesta dissertação, buscou-se detectar falhas incipientes por curto-circuito entre espiras de um motor de indução trifásico através de técnicas de aprendizado de máquina baseado em protótipos. Considerando este propósito, técnicas de aprendizado supervisionado e de aprendizado não supervisionado foram utilizadas.

A partir do banco de dados reais previamente gerado e seus atributos extraídos através do método análise de corrente do motor, este trabalho buscou diagnosticar falhas de curto-circuito de diferentes severidades por meio de classificação multiclases ao utilizar classificadores da família *Learning Vector Quantization*.

A partir da aplicação de um algoritmo de *clustering* e de diversos critérios de validação de agrupamentos, propôs-se uma etapa de pré-processamento dos dados de treinamento dos classificadores baseados em LVQ em condições de poucos protótipos disponíveis e em um cenário com protótipos em abundância.

A partir da aplicação da metodologia proposta, verificou-se que excelentes taxas de acertos binárias foram obtidas entre a maioria dos algoritmos LVQ utilizados onde pode-se citar a taxa máxima de acerto de 98,3% obtida pelo GRLVQ e de 96,6% obtida pelo LVQ1. Porém, o desbalanceamento existente entre as classes ocasionou taxas de especificidades indesejadas ao mapear os resultados obtidos da classificação ternária para binária.

Para solucionar o problema citado acima, a probabilidade *a priori* de seleção das amostras para treinamento fora equilibrada ao inserir amostras artificiais de amostras do motor em funcionamento em condições de normalidade. Verificou-se que a métrica especificidade melhorou significativamente.

### 8.2 Objetivos Específicos

Todos os objetivos específicos foram atingidos, e serão comentados abaixo:

- O banco de dados fora avaliado estatisticamente através das técnicas de validação de agrupamentos e verificou-se, que ao utilizar a metodologia proposta neste trabalho, que este é eficientemente representado por 10 protótipos (onde 3 representam a classe normal, 4 representam a classe falha de alta impedância e 3 representam a classe falha de baixa

impedância) ao considerar o BD1 ou 14 protótipos (onde são acrescentados 4 protótipos à classe normal) ao considerar o BD2;

- A quantidade ótima de protótipos fora sugerida por classe através da moda de todos o critérios de validação de agrupamentos listados neste trabalho, porém apenas os índices de validação relativos Dunn e Silhuetas foram capazes de acertar a moda sugerida que geraram as taxas máximas de acertos;
- Ao considerar o banco de dados desbalanceado, verificou-se, através do algoritmo GRLVQ, que a melhor solução encontrada utilizou maior relevância para o atributo  $3f_5$  e menor relevância para o atributo  $1,5f_5$ ;
- Para o BD1, os melhores algoritmos foram o GLVQ linear e GRLVQ. Entre estes, o algoritmo GRLVQ obtivera melhor desempenho que o GLVQ linear. Já para o BD2, isto se reverteu e o GRLVQ apresentou valores de relevâncias com diferenças não significativas;
- Os algoritmos da família maximização da margem de separação apresentaram melhores convergência de resultados em cenários com poucos protótipos (metodologias 1 e 2) do que os algoritmos LVQ heurísticos. Porém, no cenário com protótipos em abundância (metodologia 3), os resultados obtidos pelos algoritmos heurísticos melhoraram consideravelmente.

### 8.3 Trabalhos Futuros

Mesmo com os diversos estudos realizados nesta dissertação, o problema de detecção de falhas por curto-circuito ainda está longe de ser esgotado. Então, algumas sugestões para estudos futuros são listadas abaixo:

- Primeiramente, um único método de extração de atributos fora realizado neste trabalho e, conseqüentemente, diversos outros podem ser utilizados para futuras análises. Por exemplo, durante a fase final desta dissertação, estudos iniciais com método *Mel Frequency Cepstral Coefficients* (MFCC) apresentaram resultados interessantes e, em consequência disto, resultados obtidos a partir deste método serão reportados em um futuro próximo;
- Este trabalho utilizou o algoritmo  $K$ -médias na etapa de aprendizado não supervisionado com o objetivo de inferência, em caso de sucesso desta heurística, que técnicas que utilizem redes neurais competitivas, entre elas as redes SOM e NG, como procedimento de iniciação dos protótipos no treinamento dos algoritmos LVQ. para o tratamento do problema exposto neste trabalho;

- Sugere-se a seleção de amostras da classe normal para a elaboração de amostras artificiais para balancear o banco de dados a partir do método proposto que gerou a Tabela 3, pois esta ilustra as sobreposições encontradas ao realizar aprendizado não supervisionado em todo o conjunto de dados;
- Adaptação da metodologia proposta neste trabalho para a possibilidade de inserção ou remoção de protótipos durante a etapa de treinamento supervisionado;
- Verificar, seguindo a metodologia proposta neste trabalho, o rendimento dos algoritmos LVQ da família maximização da verossimilhança.

## REFERÊNCIAS

- AHN, K. K.; NGUYEN, H. T. C. Intelligent switching control of a pneumatic muscle robot arm using learning vector quantization neural network. **Mechatronics**, Elsevier, v. 17, n. 4-5, p. 255–262, 2007.
- AIMER, F.; BOUDINAR, A.; BENOZZA, N.; AZEDDINE, B. Use of the root-ar method in the diagnosis of induction motor's mechanical faults. **Revue Roumaine des Sciences Techniques Serie Electrotechnique et Energetique**, v. 62, p. 134–141, 04 2017.
- AKAIKE, H. Fitting autoregressive models for prediction. **Annals of the institute of Statistical Mathematics**, Springer, v. 21, n. 1, p. 243–247, 1969.
- AKAIKE, H. A new look at the statistical model identification. **IEEE Transactions on automatic control**, IEEE, v. 19, n. 6, p. 716–723, 1974.
- BASHYAL, S.; VENAYAGAMOORTHY, G. K. Recognition of facial expressions using gabor wavelets and learning vector quantization. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 21, n. 7, p. 1056–1064, 2008.
- BELKA, A.; FISCHER, M.; POHLMANN, A.; BEER, M.; HÖPER, D. LVQ-KNN: Composition-based DNA/RNA binning of short nucleotide sequences utilizing a prototype-based k-nearest neighbor approach. **Virus research**, Elsevier, v. 258, p. 55–63, 2018.
- BIEHL, M.; GHOSH, A.; HAMMER, B. Dynamics and generalization ability of lvq algorithms. **Journal of Machine Learning Research**, v. 8, n. Feb, p. 323–360, 2007.
- BIEHL, M.; HAMMER, B.; SCHNEIDER, P. Matrix learning in learning vector quantization. **Institute of Informatics, Clausthal University of Technology**, p. 06–14, 2006.
- BIEHL, M.; HAMMER, B.; VILLMANN, T. Prototype-based models in machine learning. **Wiley Interdisciplinary Reviews: Cognitive Science**, Wiley Online Library, v. 7, n. 2, p. 92–111, 2016.
- BLUME, M.; BALLARD, D. R. Image annotation based on learning vector quantization and localized haar wavelet transform features. In: **Applications and Science of Artificial Neural Networks III**. [S.l.]: SPIE, 1997. p. 181–191.
- BOJER, T.; HAMMER, B.; SCHUNK, D.; TOSCHANOWITZ, K. T. V. Relevance determination in learning vector quantization. In: EUROPEAN SYMPOSIUM ON ARTIFICIAL NEURAL NETWORKS. **Proceedings**. [S.l.], 2001. p. 271–276.
- BONNETT, A. H. Root cause failure analysis for ac induction motors in the petroleum and chemical industry. In: IEEE PETROLEUM AND CHEMICAL INDUSTRY CONFERENCE (PCIC). **Record of Conference Papers Industry Applications Society 57th Annual**. [S.l.], 2010. p. 1–13.
- CALIŃSKI, T.; HARABASZ, J. A dendrite method for cluster analysis. **Communications in Statistics-theory and Methods**, Taylor & Francis, v. 3, n. 1, p. 1–27, 1974.
- COELHO, D. N. **Sobre o desempenho de algoritmos de aprendizado de máquinas na detecção de falhas em motores de indução trifásicos: um estudo comparativo**. Dissertação (Mestrado em Engenharia de Teleinformática) — Universidade Federal do Ceará, 2015.

- COELHO, D. N.; BARRETO, G.; MEDEIROS, C. M.; SANTOS, J. D. A. Performance comparison of classifiers in the detection of short circuit incipient fault in a three-phase induction motor. In: **IEEE SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE FOR ENGINEERING SOLUTIONS (CIES)**. **Proceedings**. [S.l.], 2014. p. 42–48.
- COELHO, D. N.; BARRETO, G. A.; MEDEIROS, C. M. Detection of short circuit faults in 3-phase converter-fed induction motors using kernel SOMs. In: **IEEE 12TH INTERNATIONAL WORKSHOP ON SELF-ORGANIZING MAPS AND LEARNING VECTOR QUANTIZATION, CLUSTERING AND DATA VISUALIZATION (WSOM)**. **Proceedings**. [S.l.], 2017. p. 1–7.
- CRAMMER, K.; GILAD-BACHRACH, R.; NAVOT, A.; TISHBY, N. Margin analysis of the LVQ algorithm. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2003. p. 479–486.
- DAVIES, D. L.; BOULDIN, D. W. A cluster separation measure. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, n. 2, p. 224–227, 1979.
- DUNN, J. C. **A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters**. [S.l.]: Taylor & Francis, 1973.
- EVERITT, B. S. **Cluster analysis**. 5. ed. [S.l.]: Wiley, 2011. ISBN 0470749911.
- FRANCHI, C. M. **Acionamentos elétricos**. [S.l.]: Editora Saraiva, 2018.
- 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.
- GLOWACZ, A. Fault diagnosis of single-phase induction motor based on acoustic signals. **Mechanical Systems and Signal Processing**, Elsevier, v. 117, p. 65–80, 2019.
- GROSSBERG, S. Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors. **Biological cybernetics**, Springer, v. 23, n. 3, p. 121–134, 1976.
- GUEDES, M. V. **O Motor de Indução Trifásico - selecção e aplicação**. 1a. ed. [S.l.]: Faculdade de Engenharia da Faculdade do Porto, 1994.
- HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. Clustering validity checking methods: part II. **ACM Sigmod Record**, ACM, v. 31, n. 3, p. 19–27, 2002.
- HAMMER, B.; SCHLEIF, F.-M.; ZHU, X. Relational extensions of learning vector quantization. In: **Proceedings**. [S.l.: s.n.], 2011. p. 481–489.
- HAMMER, B.; STRICKERT, M.; VILLMANN, T. Supervised neural gas with general similarity measure. **Neural Processing Letters**, Springer, v. 21, n. 1, p. 21–44, 2005.
- HAMMER, B.; VILLMANN, T. Generalized relevance learning vector quantization. **Neural Networks**, Elsevier, v. 15, n. 8-9, p. 1059–1068, 2002.
- HOFMANN, D.; GISBRECHT, A.; HAMMER, B. Efficient approximations of Kernel Robust Soft LVQ. In: **Advances in Self-Organizing Maps**. [S.l.]: Springer, 2013. p. 183–192.

- JAIN, A. K.; DUBES, R. C. **Algorithms for clustering data**. [S.l.]: Prentice-Hall, Inc., 1988.
- JIRAYUSAKUL, A.; AUWATANAMONGKOL, S. A supervised growing neural gas algorithm for cluster analysis. **International Journal of Hybrid Intelligent Systems**, IOS Press, v. 4, n. 4, p. 217–229, 2007.
- KASHYAP, R. Inconsistency of the aic rule for estimating the order of autoregressive models. **IEEE Transactions on Automatic Control**, IEEE, v. 25, n. 5, p. 996–998, 1980.
- KLIMAN, G.; PREMERLANI, W.; KOEGL, R.; HOEWELER, D. A new approach to on-line turn fault detection in ac motors. In: IEEE INDUSTRY APPLICATIONS CONFERENCE. **Proceedings**. [S.l.], 1996. v. 1, p. 687–693.
- KOHONEN, T. An introduction to neural computing. **Neural networks**, Elsevier, v. 1, n. 1, p. 3–16, 1988.
- KOHONEN, T. Improved versions of learning vector quantization. In: IEEE. **Neural Networks, 1990., 1990 IJCNN International Joint Conference on**. [S.l.], 1990. p. 545–550.
- KOHONEN, T.; SCHROEDER, M.; HUANG, T. The self-organizing maps. **Inc., Secaucus, NJ**, Springer-Verlag New York, v. 4, n. 2, 2001.
- LIEBERMAN, M. A.; PATIL, R. B. Evaluation of learning vector quantization to classify cotton trash. **Optical Engineering**, International Society for Optics and Photonics, v. 36, n. 3, p. 914–922, 1997.
- LIU, R.; LI, B.; ZHANG, L.; JIAO, L. A new two-step learning vector quantization algorithm for image compression. **Transactions of the Institute of Measurement and Control**, SAGE Publications Sage UK: London, England, v. 37, n. 1, p. 3–14, 2015.
- MACQUEEN, J. *et al.* Some methods for classification and analysis of multivariate observations. In: 5TH BERKELEY SYMPOSIUM ON MATHEMATICAL STATISTICS AND PROBABILITY. **Proceedings**. [S.l.], 1967. v. 1, n. 14, p. 281–297.
- MATSUOKA, K. Noise injection into inputs in back-propagation learning. **IEEE Transactions on Systems, Man, and Cybernetics**, IEEE, v. 22, n. 3, p. 436–440, 1992.
- MENDONÇA, P.; BONALDI, E.; OLIVEIRA, L. de; LAMBERT-TORRES, G.; SILVA, J. B. da; SILVA, L. B. da; SALOMON, C.; SANTANA, W.; SHINOHARA, A. Detection and modelling of incipient failures in internal combustion engine driven generators using electrical signature analysis. **Electric Power Systems Research**, Elsevier, v. 149, p. 30–45, 2017.
- METZ, J. **Interpretação de clusters gerados por algoritmos de clustering hierárquico**. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação - USP, 2006.
- NANDI, S.; BHARADWAJ, R. M.; TOLIYAT, H. A. Performance analysis of a three-phase induction motor under mixed eccentricity condition. **IEEE Transactions on Energy Conversion**, IEEE, v. 17, n. 3, p. 392–399, 2002.
- NANDI, S.; TOLIYAT, H. A.; LI, X. Condition monitoring and fault diagnosis of electrical motors—a review. **IEEE Transactions on Energy Conversion**, IEEE, v. 20, n. 4, p. 719–729, 2005.

NASCIMENTO, N. d. M. M. e. Mestrado em Energias Renováveis, **Detecção e classificação de falhas de curto-circuito em geradores de indução aplicados em geração eólica utilizando aprendizado de máquina**. 2018.

NATARAJAN, R. Failure identification of induction motors by sensing unbalanced stator currents. **IEEE Transactions on Energy Conversion**, IEEE, v. 4, n. 4, p. 585–590, 1989.

NOVA, D.; ESTÉVEZ, P. A. A review of learning vector quantization classifiers. **Neural Computing and Applications**, Springer, v. 25, n. 3-4, p. 511–524, 2014.

OLIVEIRA, A.; MEDEIROS, C.; PONTES, R. 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 (EEMODS) CONFERENCE. **Proceedings**. [S.l.], 2013.

OLIVEIRA, Á. G. de; PONTES, R. S. T.; MEDEIROS, C. M. de S. Neural network used to stator winding interturn short-circuit fault detection in an induction motor driven by frequency converter. In: 2013 BRICS CONGRESS ON COMPUTATIONAL INTELLIGENCE AND 11TH BRAZILIAN CONGRESS ON COMPUTATIONAL INTELLIGENCE. **Proceedings**. [S.l.], 2013. p. 459–464.

PAKHIRA, M. K.; BANDYOPADHYAY, S.; MAULIK, U. Validity index for crisp and fuzzy clusters. **Pattern recognition**, Elsevier, v. 37, n. 3, p. 487–501, 2004.

PANG, X. Y.; YIN, S. N.; LI, H. Z.; ZHU, J. M.; CHEN, Z. C. Research of Classification Methods of EEG Signal Based on Wavelet Packet Transform and LVQ Neural. In: TRANS TECH PUBL. **Advanced Materials Research**. [S.l.], 2014. v. 1049, p. 1626–1630.

PENMAN, J.; SEDDING, H.; LLOYD, B.; FINK, W. Detection and location of interturn short circuits in the stator windings of operating motors. **IEEE transactions on Energy conversion**, IEEE, v. 9, n. 4, p. 652–658, 1994.

PRADHAN, N.; SADASIVAN, P.; ARUNODAYA, G. Detection of seizure activity in EEG by an artificial neural network: A preliminary study. **Computers and Biomedical Research**, Elsevier, v. 29, n. 4, p. 303–313, 1996.

QIN, A. K.; SUGANTHAN, P. N. A novel kernel prototype-based learning algorithm. In: IEEE 17TH INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION (ICPR). **Proceedings**. [S.l.], 2004. v. 4, p. 621–624.

QIN, A. K.; SUGANTHAN, P. N.; LIANG, J. J. A new generalized LVQ algorithm via harmonic to minimum distance measure transition. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS. **Proceedings**. [S.l.], 2004. v. 5, p. 4821–4825.

RISSANEN, J. Modeling by shortest data description. **Automatica**, Elsevier, v. 14, n. 5, p. 465–471, 1978.

ROBBINS, H.; MONRO, S. A stochastic approximation method. In: **Herbert Robbins Selected Papers**. [S.l.]: Springer, 1985. p. 102–109.

ROCHA NETO, A. R. Doutorado em Engenharia de Teleinformática, **Sinpatco II: Novas estratégias de aprendizado de máquina para classificação de patologias da coluna vertebral**. 2011.

ROUSSEEUW, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. **Journal of computational and applied mathematics**, Elsevier, v. 20, p. 53–65, 1987.

RUMELHART, D. E.; ZIPSER, D. Feature discovery by competitive learning. **Cognitive science**, Wiley Online Library, v. 9, n. 1, p. 75–112, 1985.

SAKHARA, S.; SAAD, S.; NACIB, L. Diagnosis and detection of short circuit in asynchronous motor using three-phase model. **International Journal of System Assurance Engineering and Management**, Springer, v. 8, n. 2, p. 308–317, 2017.

SATO, A.; YAMADA, K. Generalized learning vector quantization. In: NEURAL INFORMATION PROCESSING SYSTEMS. **Proceedings**. [S.l.], 1996. p. 423–429.

SATO, A.; YAMADA, K. An analysis of convergence in generalized LVQ. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS. **Proceedings**. [S.l.], 1998. p. 171–176.

SCHLEIF, F.-M.; VILLMANN, T.; HAMMER, B.; SCHNEIDER, P. Efficient kernelized prototype based classification. **International journal of neural systems**, World Scientific, v. 21, n. 06, p. 443–457, 2011.

SCHNEIDER, P.; BIEHL, M.; HAMMER, B. Distance learning in discriminative vector quantization. **Neural computation**, MIT Press, v. 21, n. 10, p. 2942–2969, 2009.

SCHWARZ, G. *et al.* Estimating the dimension of a model. **The annals of statistics**, Institute of Mathematical Statistics, v. 6, n. 2, p. 461–464, 1978.

SEO, S.; BODE, M.; OBERMAYER, K. Soft nearest prototype classification. **IEEE Transactions on Neural Networks**, Citeseer, v. 14, n. 2, p. 390–398, 2003.

SEO, S.; OBERMAYER, K. Soft learning vector quantization. **Neural computation**, MIT Press, v. 15, n. 7, p. 1589–1604, 2003.

SESHADRINATH, J.; SINGH, B.; PANIGRAHI, B. K. Incipient interturn fault diagnosis in induction machines using an analytic wavelet-based optimized bayesian inference. **IEEE transactions on neural networks and learning systems**, IEEE, v. 25, n. 5, p. 990–1001, 2014.

SOUSA, D. P.; BARRETO, G. A.; MEDEIROS, C. M. Efficient selection of data samples for fault classification by the clustering of the som. In: CONGRESSO BRASILEIRO DE INTELIGÊNCIA COMPUTACIONAL. **Anais**. [S.l.], 2017.

STRICKERT, M.; BOJER, T.; HAMMER, B. Generalized relevance LVQ for time series. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS. **Proceedings**. [S.l.], 2001. p. 677–683.

TALLAM, R. M.; LEE, S. B.; STONE, G. C.; KLIMAN, G. B.; YOO, J.; HABETLER, T. G.; HARLEY, R. G. A survey of methods for detection of stator-related faults in induction machines. **IEEE Transactions on Industry Applications**, IEEE, v. 43, n. 4, p. 920–933, 2007.

THOMSON, W. T.; FENGER, M. Current signature analysis to detect induction motor faults. **IEEE Industry Applications Magazine**, IEEE, v. 7, n. 4, p. 26–34, 2001.

THORSEN, O.; DALVA, M. Condition monitoring methods, failure identification and analysis for high voltage motors in petrochemical industry. In: INTERNATIONAL CONFERENCE ON ELECTRICAL MACHINES AND DRIVES. **Proceedings**. [S.l.], 1997.

TORKKOLA, K.; CAMPBELL, W. M. Mutual information in learning feature transformations. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING. **Proceedings**. [S.l.], 2000. p. 1015–1022.

VASUKI, A.; VANATHI, P. A review of vector quantization techniques. **IEEE Potentials**, IEEE, v. 25, n. 4, p. 39–47, 2006.

VESANTO, J.; ALHONIEMI, E. Clustering of the self-organizing map. **IEEE Transactions on neural networks**, Ieee, v. 11, n. 3, p. 586–600, 2000.

VILLMANN, T.; HAASE, S. Divergence-based vector quantization. **Neural Computation**, MIT Press, v. 23, n. 5, p. 1343–1392, 2011.

VILLMANN, T.; HAMMER, B.; SCHLEIF, F.-M.; HERMANN, W.; COTTRELL, M. Fuzzy classification using information theoretic learning vector quantization. **Neurocomputing**, Elsevier, v. 71, n. 16-18, p. 3070–3076, 2008.

WEBB, A. **Statistical Pattern Recognition**. 2nd. ed. [S.l.]: John Wiley & Sons, 2002.

WEBB, A. R. **Statistical pattern recognition**. [S.l.]: John Wiley & Sons, 2011.

WITOELAR, A.; BIEHL, M.; HAMMER, B. Learning vector quantization: generalization ability and dynamics of competing prototypes. In: SCHLOSS DAGSTUHL-LEIBNIZ-ZENTRUM FÜR INFORMATIK. **Proceedings**. [S.l.], 2007.

YU, D.; ZHANG, Y.; BIAN, H.; WANG, X.; QI, W. A new diagnostic method for identifying working conditions of submersible reciprocating pumping systems. **Petroleum Science**, Springer, v. 10, n. 1, p. 81–90, 2013.

## APÊNDICE A – TABELAS DE RESULTADOS

Tabela 73 – Estatísticas de taxas de acerto encontradas em classificação ternária do banco de dados 1 para metodologia de treinamento 1

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	11,8644	3,3898	28,8136	28,8136	44,0678	44,0678	10,1695	45,7627	44,0678	30,5085
<b>Máxima</b>	62,7119	62,7119	69,4915	57,6271	72,8814	71,1864	54,2373	69,4915	69,4915	67,7966
<b>Média</b>	48,5763	23,8475	48,8136	43,339	56,7458	56,5085	41,3898	56,0339	55,4915	51,0678
<b>Mediana</b>	50	15,2542	48,3051	42,3729	55,9322	55,9322	42,3729	55,9322	55,9322	50,8475
<b>Desvio padrão</b>	8,7731	16,3731	7,9499	5,5787	5,5746	5,7375	8,8553	5,6024	5,374	6,2247

Fonte: o autor.

Tabela 74 – Estatísticas de taxas de acerto encontradas em classificação ternária do banco de dados 1 para metodologia de treinamento 2

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	15,2542	3,3898	28,8136	28,8136	37,2881	38,9831	10,1695	42,3729	37,2881	35,5932
<b>Máxima</b>	57,6271	50,8475	71,1864	57,6271	71,1864	72,8814	57,6271	67,7966	74,5763	66,1017
<b>Média</b>	45,0678	17,8814	47,1864	43,2881	54,1356	54,0169	40,4576	54,9492	54,678	51,9153
<b>Mediana</b>	44,9153	15,2542	45,7627	42,3729	54,2373	54,2373	42,3729	54,2373	54,2373	50,8475
<b>Desvio padrão</b>	6,3285	10,5569	8,4788	5,2858	6,2441	6,1355	8,9998	5,6094	6,6875	6,1403

Fonte: o autor.

Tabela 75 – Estatísticas de taxas de acerto encontradas em classificação ternária do banco de dados 1 para metodologia de treinamento 3

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	32,2034	40,678	37,2881	37,2881	30,5085	38,9831	33,8983	40,678	37,2881	33,8983
<b>Máxima</b>	67,7966	71,1864	66,1017	69,4915	62,7119	67,7966	67,7966	66,1017	66,1017	64,4068
<b>Média</b>	49,3559	53,4576	52,4746	51,5254	50,1525	49,1695	50,5254	54,5424	50,2203	50,8644
<b>Mediana</b>	49,1525	52,5424	52,5424	52,5424	50,8475	49,1525	50,8475	54,2373	50,8475	50,8475
<b>Desvio padrão</b>	6,2972	6,4074	6,8305	6,5044	6,0949	5,4802	6,3312	5,4424	5,8548	5,923

Tabela 76 – Estatísticas de taxas de acerto encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 1

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	11,8644	3,3898	74,5763	74,5763	76,2712	76,2712	10,1695	77,9661	77,9661	69,4915
<b>Máxima</b>	96,6102	91,5254	96,6102	96,6102	94,9153	96,6102	96,6102	98,3051	96,6102	94,9153
<b>Média</b>	83,8814	33,7119	85,6441	85,8305	87,8136	87,8475	81	87,5763	87,4068	84,0678
<b>Mediana</b>	86,4407	15,2542	86,4407	86,4407	88,1356	88,1356	84,7458	88,1356	88,1356	83,8983
<b>Desvio padrão</b>	12,9002	31,7345	3,6966	3,7283	3,3786	3,606	17,7328	3,5736	3,7496	4,7453

Fonte: o autor.

Tabela 77 – Estatísticas de taxas de acerto encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 2

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	15,2542	3,3898	74,5763	76,2712	76,2712	74,5763	10,1695	76,2712	76,2712	59,322
<b>Máxima</b>	96,6102	93,2203	96,6102	96,6102	96,6102	96,6102	96,6102	94,9153	98,3051	96,6102
<b>Média</b>	84,7458	22,1356	85,4915	85,5932	87,4407	87,0847	79,8475	87,2373	87,0508	85,1525
<b>Mediana</b>	84,7458	15,2542	84,7458	84,7458	88,1356	86,4407	84,7458	88,1356	86,4407	84,7458
<b>Desvio padrão</b>	7,8802	22,3806	3,7886	3,6575	3,7172	3,8453	19,6288	3,5279	3,7977	5,1506

Fonte: o autor.

Tabela 78 – Estatísticas de taxas de acerto encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 3

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	69,4915	76,2712	74,5763	72,8814	71,1864	72,8814	67,7966	74,5763	66,1017	72,8814
<b>Máxima</b>	91,5254	98,3051	96,6102	94,9153	93,2203	94,9153	94,9153	98,3051	91,5254	93,2203
<b>Média</b>	82,5424	86,2373	86,2712	83,9831	84,5424	84,0678	85,9153	86,1695	83,661	85,0169
<b>Mediana</b>	83,0508	86,4407	86,4407	84,7458	84,7458	84,7458	86,4407	86,4407	84,7458	85,5932
<b>Desvio padrão</b>	5,2255	4,1536	3,7592	4,8324	4,7531	4,6214	4,6132	4,381	4,8272	4,1845

Fonte: o autor.

Tabela 79 – Estatísticas de taxas de sensibilidade encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 1

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	0,7719	0,7966	0,7458	0,7458	0,7679	0,7679	0,7458	0,7719	0,7719	0,7586
<b>Máxima</b>	0,9661	0,9600	0,9661	0,9661	0,9818	0,9825	0,9661	0,9828	0,9821	0,9636
<b>Média</b>	0,8750	0,8559	0,8600	0,8598	0,8896	0,8902	0,8536	0,8854	0,8849	0,8710
<b>Mediana</b>	0,8750	0,8571	0,8621	0,8644	0,8909	0,8909	0,8475	0,8841	0,8868	0,8704
<b>Desvio padrão</b>	0,0384	0,0407	0,0387	0,0367	0,0363	0,0373	0,0378	0,0375	0,0369	0,0407

Fonte: o autor.

Tabela 80 – Estatísticas de taxas de sensibilidade encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 2

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	0,7627	0,7797	0,7458	0,7627	0,7679	0,7544	0,7458	0,7679	0,7679	0,7458
<b>Máxima</b>	0,9661	0,9322	0,9661	0,9661	0,9825	0,9821	0,9661	0,9655	0,9825	0,9808
<b>Média</b>	0,8604	0,8459	0,8579	0,8569	0,8866	0,8838	0,8550	0,8809	0,8816	0,8766
<b>Mediana</b>	0,8644	0,8305	0,8609	0,8583	0,8830	0,8841	0,8475	0,8793	0,8783	0,8772
<b>Desvio padrão</b>	0,0379	0,0411	0,0390	0,0362	0,0377	0,0392	0,0384	0,0370	0,0376	0,0419

Fonte: o autor.

Tabela 81 – Estatísticas de taxas de sensibilidade encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 3

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	0,7586	0,7719	0,7458	0,7679	0,7636	0,7593	0,7778	0,7544	0,7544	0,7586
<b>Máxima</b>	0,9815	0,9825	0,9661	0,9818	0,9815	0,9821	0,9649	0,9825	0,9808	0,9636
<b>Média</b>	0,8813	0,8817	0,8646	0,8825	0,8856	0,8844	0,8770	0,8824	0,8798	0,8807
<b>Mediana</b>	0,8807	0,8793	0,8644	0,8846	0,8889	0,8784	0,8772	0,8761	0,8788	0,8857
<b>Desvio padrão</b>	0,0423	0,0376	0,0389	0,0403	0,0396	0,0394	0,0388	0,0374	0,0392	0,0404

Fonte: o autor.

Tabela 82 – Estatísticas de taxas de especificidade encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 1

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+GRLVQ	GLVQ+LGRLVQ
<b>Mínima</b>	0	0	0	0	0	0	0,1017	0	0	0
<b>Máxima</b>	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,1525	1,0000	1,0000	1,0000
<b>Média</b>	0,5954	0,2109	0,6437	0,7619	0,7284	0,7317	0,1271	0,7553	0,7338	0,4298
<b>Mediana</b>	0,6000	0,1525	1,0000	1,0000	0,7500	0,7500	0,1271	0,8000	0,7500	0,4143
<b>Desvio padrão</b>	0,3080	0,2177	0,3978	0,3149	0,2529	0,2405	0,0279	0,2640	0,2756	0,3531

Fonte: o autor.

Tabela 83 – Estatísticas de taxas de especificidade encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 2

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+GRLVQ	GLVQ+LGRLVQ
<b>Mínima</b>	0	0,0339	0	0	0	0	0,1017	0	0	0
<b>Máxima</b>	1,0000	0,2542	1,0000	1,0000	1,0000	1,0000	0,1864	1,0000	1,0000	1,0000
<b>Média</b>	0,5707	0,1442	0,6299	0,8250	0,7046	0,6874	0,1483	0,7309	0,6977	0,5254
<b>Mediana</b>	0,5000	0,1525	0,7500	1,0000	0,6667	0,6667	0,1525	0,7500	0,6667	0,5000
<b>Desvio padrão</b>	0,3335	0,0370	0,4066	0,3344	0,2592	0,2668	0,0252	0,2719	0,3000	0,3360

Fonte: o autor.

Tabela 84 – Estatísticas de taxas de especificidade encontradas em classificação binária do banco de dados 1 para metodologia de treinamento 3

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+GRLVQ	GLVQ+LGRLVQ
<b>Mínima</b>	0	0	0	0	0	0	0	0	0	0
<b>Máxima</b>	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
<b>Média</b>	0,3797	0,6344	0,7864	0,4553	0,4707	0,4533	0,5778	0,6296	0,4246	0,5097
<b>Mediana</b>	0,4000	0,6667	1,0000	0,4286	0,5000	0,4143	0,5000	0,6667	0,4000	0,5000
<b>Desvio padrão</b>	0,2044	0,3148	0,3753	0,2734	0,2553	0,2439	0,3176	0,3130	0,2588	0,2978

Fonte: o autor.

Tabela 85 – Estatísticas de taxas de acerto encontradas em classificação ternária do banco de dados 2 para metodologia de treinamento 1

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+GRLVQ	GLVQ+LGRLVQ
<b>Mínima</b>	21,0526	21,0526	31,5789	19,7368	44,7368	47,3684	23,6842	46,0526	44,7368	34,2105
<b>Máxima</b>	61,8421	50	69,7368	42,1053	68,4211	69,7368	48,6842	68,4211	68,4211	68,4211
<b>Média</b>	33,2237	34,1053	54,2368	32,1316	58,8553	59,0263	33,3553	58,6579	58,6842	50,4211
<b>Mediana</b>	31,5789	34,2105	54,6053	31,5789	59,2105	59,2105	32,8947	59,2105	59,2105	51,3158
<b>Desvio padrão</b>	7,2586	5,8596	5,9629	4,4774	4,8978	4,9868	4,5171	4,6348	4,9197	7,2501

Fonte: o autor.

Tabela 86 – Estatísticas de taxas de acerto encontradas em classificação ternária do banco de dados 2 para metodologia de treinamento 2

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+GRLVQ	GLVQ+LGRLVQ
<b>Mínima</b>	21,0526	21,0526	32,8947	19,7368	42,1053	34,2105	19,7368	38,1579	42,1053	34,2105
<b>Máxima</b>	42,1053	44,7368	67,1053	43,4211	69,7368	69,7368	46,0526	76,3158	75	64,4737
<b>Média</b>	33,1842	33,4211	54,1316	31,5132	57,6974	57	32,9868	57,7105	57,75	50,9605
<b>Mediana</b>	32,8947	33,5526	53,9474	31,5789	57,8947	56,5789	32,8947	57,8947	56,5789	51,9737
<b>Desvio padrão</b>	4,7848	5,0943	6,3007	4,8023	5,4021	6,0279	5,274	5,6479	5,4361	6,6249

Fonte: o autor.

Tabela 87 – Estatísticas de taxas de acerto encontradas em classificação ternária do banco de dados 2 para metodologia de treinamento 3

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	43,4211	43,4211	38,1579	42,1053	43,4211	43,4211	25	48,6842	46,0526	27,6316
<b>Máxima</b>	71,0526	72,3684	71,0526	69,7368	69,7368	72,3684	72,3684	69,7368	72,3684	65,7895
<b>Média</b>	54,9605	58,6842	57,3947	55,75	56,3289	55,8947	46,0789	60,9737	56,8026	50,1053
<b>Mediana</b>	55,2632	59,2105	57,8947	56,5789	56,5789	55,9211	46,7105	61,8421	56,5789	51,3158
<b>Desvio padrão</b>	4,8835	5,3161	6,0537	5,8056	5,4771	5,0477	12,4209	4,6688	5,2032	7,5312

Fonte: o autor.

Tabela 88 – Estatísticas de taxas de acerto encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 1

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	21,0526	21,0526	61,8421	25	63,1579	64,4737	25	63,1579	61,8421	42,1053
<b>Máxima</b>	84,2105	73,6842	84,2105	78,9474	90,7895	88,1579	78,9474	89,4737	90,7895	85,5263
<b>Média</b>	56,1842	35,4474	74,2237	60,8026	80,7368	80,9737	53,75	79,7368	80,6316	70,2895
<b>Mediana</b>	63,1579	34,2105	75	64,4737	81,5789	81,5789	61,8421	80,2632	81,5789	72,3684
<b>Desvio padrão</b>	17,4564	9,8621	4,8772	13,7751	4,3801	4,1836	17,1297	4,6304	4,4101	9,8219

Fonte: o autor.

Tabela 89 – Estatísticas de taxas de acerto encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 2

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	21,0526	21,0526	59,2105	21,0526	60,5263	60,5263	21,0526	38,1579	61,8421	34,2105
<b>Máxima</b>	77,6316	63,1579	88,1579	77,6316	89,4737	89,4737	77,6316	90,7895	90,7895	88,1579
<b>Média</b>	55,75	33,6184	74,0395	59,6447	79,0395	78,8289	53,6184	78,5658	78,9079	71,3684
<b>Mediana</b>	63,1579	33,5526	74,3421	64,4737	78,9474	78,9474	61,8421	78,9474	78,9474	73,6842
<b>Desvio padrão</b>	16,56	5,8168	5,2892	14,6174	4,6727	5,3098	17,1582	6,2148	4,8208	9,3436

Fonte: o autor.

Tabela 90 – Estatísticas de taxas de acerto encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 3

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	73,6842	64,4737	63,1579	68,4211	71,0526	71,0526	25	68,4211	71,0526	46,0526
<b>Máxima</b>	90,7895	90,7895	86,8421	90,7895	90,7895	92,1053	90,7895	92,1053	90,7895	86,8421
<b>Média</b>	82,0132	81,3289	75,9342	81,3158	81,7368	81,1974	68,9737	83,4079	81,6184	72,9079
<b>Mediana</b>	82,8947	81,5789	76,3158	81,5789	81,5789	81,5789	73,6842	84,2105	81,5789	75
<b>Desvio padrão</b>	3,7546	4,6367	5,5739	4,1735	4,4934	4,5284	17,2607	3,6017	4,1712	8,9576

Fonte: o autor.

Tabela 91 – Estatísticas de taxas de sensibilidade encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 1

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	0,5526	0,7407	0,6250	0,5526	0,6949	0,6885	0,5526	0,7049	0,6885	0,6522
<b>Máxima</b>	0,8246	0,9459	0,9388	0,7895	0,9231	0,9123	0,7895	0,9259	0,9259	1,0000
<b>Média</b>	0,6736	0,8114	0,7884	0,6638	0,8173	0,8169	0,6678	0,8182	0,8184	0,8066
<b>Mediana</b>	0,6711	0,7958	0,7878	0,6645	0,8220	0,8189	0,6711	0,8214	0,8198	0,8000
<b>Desvio padrão</b>	0,0599	0,0729	0,0565	0,0510	0,0513	0,0500	0,0535	0,0487	0,0520	0,0681

Fonte: o autor.

Tabela 92 – Estatísticas de taxas de sensibilidade encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 2

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	0,5526	0,8250	0,6271	0,5526	0,6984	0,6667	0,5526	0,7049	0,6909	0,6563
<b>Máxima</b>	0,7763	0,8250	0,9286	0,7763	0,9273	0,9434	0,7763	0,9583	0,9388	1,0000
<b>Média</b>	0,6652	0,8250	0,7854	0,6608	0,8197	0,8130	0,6640	0,8251	0,8213	0,8020
<b>Mediana</b>	0,6579	0,8250	0,7863	0,6579	0,8198	0,8189	0,6579	0,8298	0,8165	0,8000
<b>Desvio padrão</b>	0,0494	0	0,0570	0,0471	0,0510	0,0549	0,0475	0,0523	0,0526	0,0608

Fonte: o autor.

Tabela 93 – Estatísticas de taxas de sensibilidade encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 3

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	0,7000	0,6774	0,6667	0,7049	0,6842	0,7018	0,5600	0,7119	0,7143	0,6515
<b>Máxima</b>	0,9016	0,9600	0,9773	0,9286	0,9138	0,9608	1,0000	0,9298	0,9153	0,9500
<b>Média</b>	0,7994	0,8129	0,8507	0,8029	0,8130	0,8140	0,7653	0,8290	0,8117	0,8113
<b>Mediana</b>	0,8049	0,8142	0,8571	0,8015	0,8151	0,8130	0,7626	0,8236	0,8142	0,8111
<b>Desvio padrão</b>	0,0444	0,0525	0,0583	0,0468	0,0492	0,0520	0,0777	0,0444	0,0478	0,0613

Fonte: o autor.

Tabela 94 – Estatísticas de taxas de especificidade encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 1

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	0,2105	0,2105	0,3636	0,2500	0,3846	0,4074	0,2500	0,3929	0,3600	0,2667
<b>Máxima</b>	0,9286	0,6552	0,9091	0,3816	1,0000	1,0000	0,4079	1,0000	1,0000	0,9333
<b>Média</b>	0,3936	0,3453	0,6499	0,3150	0,7897	0,7983	0,3337	0,7586	0,7854	0,5913
<b>Mediana</b>	0,3421	0,3421	0,6522	0,3026	0,8000	0,8095	0,3421	0,7600	0,7958	0,5938
<b>Desvio padrão</b>	0,1897	0,0722	0,1355	0,0426	0,1231	0,1227	0,0450	0,1402	0,1302	0,1553

Fonte: o autor.

Tabela 95 – Estatísticas de taxas de especificidade encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 2

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	0,2105	0,2105	0,3571	0,2105	0,3947	0,3478	0,2105	0,3816	0,4054	0,2727
<b>Máxima</b>	0,4079	0,4474	0,9500	0,4079	1,0000	1,0000	0,4474	1,0000	1,0000	1,0000
<b>Média</b>	0,3285	0,3340	0,6474	0,3034	0,7294	0,7399	0,3276	0,7207	0,7269	0,6110
<b>Mediana</b>	0,3289	0,3355	0,6542	0,3092	0,7454	0,7391	0,3289	0,7380	0,7247	0,6277
<b>Desvio padrão</b>	0,0522	0,0506	0,1314	0,0544	0,1244	0,1388	0,0544	0,1384	0,1304	0,1489

Fonte: o autor.

Tabela 96 – Estatísticas de taxas de especificidade encontradas em classificação binária do banco de dados 2 para metodologia de treinamento 3

	LVQ1	OLVQ1	LVQ2.1	LVQ3	GLVQ linear	GLVQ não linear	LVQ+RLVQ	GRLVQ	GLVQ+ GRLVQ	GLVQ+ LGRLVQ
<b>Mínima</b>	0,7222	0,4138	0,3333	0,4500	0,5000	0,5714	0,2500	0,4583	0,5789	0,3023
<b>Máxima</b>	1,0000	1,0000	0,9259	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
<b>Média</b>	0,9165	0,8255	0,6308	0,8574	0,8365	0,8102	0,6870	0,8581	0,8336	0,6313
<b>Mediana</b>	0,9286	0,8367	0,6452	0,8824	0,8421	0,8367	0,7386	0,8681	0,8333	0,6273
<b>Desvio padrão</b>	0,0801	0,1176	0,1185	0,1146	0,1093	0,1070	0,2318	0,1009	0,0979	0,1667

Fonte: o autor.