



UNIVERSIDADE FEDERAL DA BAHIA
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA - PPGE



DETECÇÃO DE CORROSÃO UTILIZANDO
CORRENTES PARASITAS PULSADAS E
CLASSIFICADORES NEURAIS COM
RESTRIÇÕES DE TREINAMENTO

GILVAN FARIAS DA SILVA

Outubro, 2020

UNIVERSIDADE FEDERAL DA BAHIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA - PPGE

GILVAN FARIAS DA SILVA

**DETECÇÃO DE CORROSÃO UTILIZANDO
CORRENTES PARASITAS PULSADAS E
CLASSIFICADORES NEURAIIS COM
RESTRIÇÕES DE TREINAMENTO**

Dissertação apresentada ao programa de pós-graduação em Engenharia Elétrica - PPGE da Universidade Federal da Bahia como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Paulo César Machado de Abreu Farias - UFBA

Co-orientador: Prof. Dr. Eduardo Furtado de Simas Filho - UFBA

SALVADOR
Outubro, 2020

GILVAN FARIAS DA SILVA

**DETECÇÃO DE CORROSÃO
UTILIZANDO CORRENTES PARASITAS
PULSADAS E CLASSIFICADORES
NEURAIS COM RESTRIÇÕES DE
TREINAMENTO**

Banca Avaliadora:

Paulo César Machado de Abreu Farias
D.Sc. (Orientador - UFBA)

Eduardo Furtado Simas Filho
D.Sc. (Co-orientador - UFBA)

Nadia Nedjah
Ph.D. (UERJ)

Werner Spolidoro Freund
D.Sc. (UFRJ)

Agradecimentos

Agradeço a minha família e amigos pelo apoio e incentivo aos meus estudos. Aos meus orientadores Paulo César Farias e Eduardo Simas e ao doutorando Edmar Egídio por todo o suporte sem o qual este trabalho não seria possível. Ao LABIND/GPEND, mais especificamente aos professores Maria C. S. Albuquerque, Ivan C. da Silva e Cláudia T. T. Farias, por compartilhar os dados de ensaios não destrutivos com correntes parasitas pulsadas. E também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

Resumo da Dissertação apresentada ao PPGEE/UFBA como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

DETECÇÃO DE CORROSÃO UTILIZANDO CORRENTES PARASITAS PULSADAS E CLASSIFICADORES NEURAI COM RESTRIÇÕES DE TREINAMENTO

Gilvan Farias da Silva

Outubro/2020

Orientadores: Paulo César Machado de Abreu Farias

Eduardo Furtado de Simas Filho

Programa: Engenharia Elétrica

Corrosão de tubos metálicos é um problema comum na indústria. Se não for detectada a tempo pode causar rompimento do tubo, resultando em perda na produção e acidentes graves. Em tubos industriais termicamente isolados por material compósito a corrosão pode permanecer escondida sob isolamento térmico, ou ainda na parte interna do tubo. Para detectar a corrosão visualmente é necessário remover temporariamente o isolamento térmico. Entretanto, esta remoção pode ser impraticável com a planta industrial em funcionamento. Portanto, neste trabalho foi utilizado um tipo de ensaio não destrutivo chamado correntes parasitas pulsadas (PEC) para detectar esta corrosão escondida. A interpretação deste tipo de ensaio depende da experiência e habilidade técnica do operador. Em aplicações como essa, cada erro de classificação possui consequências distintas. Classificar um tubo com corrosão como não apresentando corrosão inibe a execução da devida manutenção, aumentando as chances de rompimento do tubo. Nesta dissertação, foi proposto um classificador neural para auxiliar o operador na tomada de decisão. Este classificador foi treinado utilizando otimização por enxame de partículas com restrições, de forma a considerar os diferentes erros de classificação de formas distintas. Foi considerado que os erros que aumentam os riscos de acidentes são mais indesejáveis que os erros que provocam apenas perdas financeiras. Para avaliação do método proposto, foram utilizadas amostras de sinais PEC adquiridas de tubos de uma indústria petroquímica. O classificador treinado pelo método proposto não apresentou erros de classificação que comprometem a segurança da planta industrial. Ao contrário do observado com classificadores treinados pelas diferentes variações do método gradiente descendente analisadas neste trabalho.

Abstract of Dissertation presented to PPGEE/UFBA as a partial fulfillment of the requirements for the degree of Master of Electrical Engineering.

CORROSION DETECTION USING PULSED EDDY CURRENTS AND NEURAL CLASSIFIERS WITH TRAINING CONSTRAINTS

Gilvan Farias da Silva

October/2020

Advisors: Paulo César Machado de Abreu Farias

Eduardo Furtado de Simas Filho

Department: Electrical Engineering

Corrosion of metal pipes is a common problem in the industry. If not detected in time, it can cause pipe disruption, resulting in loss of production and serious accidents. At insulated industrial pipes the corrosion can remain hidden under thermal insulation, or inside the pipe. To detect corrosion visually, it is necessary to temporarily remove the thermal insulation. However, this removal may be impractical with the industrial plant in operation. Therefore, in this work, a type of non-destructive test called pulsed eddy currents (PEC) was used to detect this hidden corrosion. The interpretation of the test depends on the experience and technical skill of the operator. At applications like this, each misclassification has different consequences. Classifying a corroded pipe as not having corrosion inhibits the execution of proper maintenance, increasing the chances of the pipe breaking. In this work, it was proposed a neural classifier to assist the operator in decision making. This classifier was trained with particle swarm optimization with restrictions, in order to consider the different classification errors in different ways. It was considered errors that increase the risk of accidents are more undesirable than errors that cause only financial losses. To evaluate the proposed method, it was used PEC signals acquired from pipes of a petrochemical industry. The classifier trained by the proposed method did not present classification errors that compromise the safety of the industrial plant. Contrary to what was observed with classifiers trained by the different variations of the gradient descent method analyzed in this work.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xiii
Lista de Símbolos	xv
Lista de Siglas	xvii
1 Introdução	1
1.1 Motivação	2
1.2 Objetivo	3
1.3 Metodologia	3
1.4 Organização do documento	4
2 Apresentação do problema	7
2.1 Corrosão sob revestimento	7
2.2 Corrosão Interna de Tubos	10
2.3 Erros de classificação	11
2.4 Procedimentos de Manutenção de Tubos Com Corrosão	12
3 Ensaios Com Correntes Parasitas Pulsadas	17
3.1 Ensaios Não Destrutivos Para Detecção De Corrosão	17
3.2 Correntes Parasitas Pulsadas	18
3.3 Trabalhos Relacionados	21
3.4 Ferramentas Comerciais Para Ensaios Com PEC	21
4 Sistemas De Classificação	23
4.1 Sistemas de classificação	23
4.2 Redes Neurais Artificiais	27
4.2.1 Neurônio Artificial	28
4.2.2 Rede Perceptron Multicamadas	30
4.2.3 Treinamento Baseado no Gradiente do Erro	32
4.2.4 Outros métodos de treinamento	33
4.3 Pré-processamento da informação	34

4.3.1	Transformada Discreta de Fourier	34
4.3.2	Transformada Wavelet Discreta	35
4.3.3	Análise de componentes principais	36
5	Otimização Por Enxame De Partículas	39
5.1	Trabalhos Com Otimização Por Enxame De Partículas	42
5.2	Treinamento De Redes Neurais Artificiais Com PSO	43
6	Método Proposto	47
6.1	Ensaio realizado com PEC	49
6.2	Extração e seleção de Características	52
6.3	Escolha da topologia da rede neural, função <i>wavelet</i> mãe e número de componentes principais	53
6.4	Treinamento da ANN utilizando PSO com restrições	56
6.4.1	Funções objetivo e restrições	58
6.5	Validação Cruzada e <i>Bootstrap</i>	61
7	Resultados	63
7.1	Análise dos sinais típicos para as classes de interesse	63
7.2	Análise de componentes principais	66
7.3	Determinação da topologia das ANN	69
7.4	Resultados dos treinamentos das ANN com o método proposto	74
7.4.1	ANN treinadas com informação no domínio do tempo .	75
7.4.2	ANN treinadas com informação pré-processada por DWT	78
7.4.3	ANN treinadas com informação pré-processada por DFT	79
7.4.4	ANN treinadas com informação pré-processada por DWT+PCA	81
7.4.5	ANN treinadas com informação pré-processada com DFT+PCA	84
7.5	Análise comparativa dos resultados	86
7.6	Resultados de Trabalho Correlato	92
8	Conclusões	95
8.1	Principais Contribuições	96
8.2	Sugestões para trabalhos futuros	97
	Bibliografia	99
A	Trabalhos Publicados	109
A.1	Artigos aprovados em congressos	109
A.2	Artigos aprovados em revistas	110
B	Código utilizado neste trabalho	113

Lista de Figuras

2.1	Exemplos de tubos com corrosão sob isolamento (CUI). Fonte: [1].	9
2.2	Tubo com corrosão interna por CO_2 . Fonte: [14].	11
2.3	Separação de água e petróleo dentro de um tubo na horizontal.	11
2.4	Braçadeira de material composto (em amarelo) instalada em volta de tudo com corrosão externa (em preto). Fonte: [15]	13
2.5	Manga de metal instalada em volta de tudo com corrosão externa. Fonte: [16]	14
2.6	Duas soluções para reparo em tubos com corrosão. Em 2.6a <i>Stop It</i> da fabricante <i>TRI-CHEM</i> . Fonte: [17]. Em 2.6b <i>ARC 5 Emergency Metal To Metal Repair</i> da fabricante <i>Cactus Industrial</i> . Fonte: [18]	14
2.7	Procedimento de aplicação de reparo expandido em um tubo com corrosão interna. Fonte: [15]	15
2.8	Robô utilizado em procedimento de reparo em um tubo com corrosão interna. Fonte: [15]	15
3.1	Ilustração do princípio da técnica PEC [23].	19
3.2	Modo de visualização A-Scan. Fonte: [26]	20
3.3	Modo de visualização B-Scan. Fonte: [26]	20
3.4	Modo de visualização C-Scan. Fonte: [26]	21
3.5	Equipamento para ensaios com PEC da fabricante <i>Maxwell ndt</i> . Fonte: [33].	22
4.1	Modelo de um neurônio. Fonte: Adaptado de [39].	28
4.2	Funções de ativação para neurônios artificiais.	29
4.3	Rede <i>multilayer perceptron forward</i> totalmente conectada. Os valores de <i>bias</i> não estão representados para melhor visualização. Fonte: Adaptado de [39].	30
4.4	Curva hipotética com dois mínimos locais e o mínimo global.	33
4.5	Banco de filtros de nível 2 para transformada discreta wavelet. Fonte: Adaptado de [58].	36
6.1	Diagrama de processamento.	48
6.2	Detalhamento do diagrama de processamento.	49

6.3	Corpo de prova. O comprimento do tubo é 250 mm.	49
6.4	Representação em corte do corpo de prova.	50
6.5	Representação perimétrica do corpo de prova.	50
6.6	Diagrama com detalhe da bobina e sensores GMR.	51
6.7	Validação cruzada <i>K-Fold</i>	61
6.8	Princípio de funcionamento do <i>Bootstrap</i>	62
7.1	Sinais típicos dos ensaios com PEC no domínio do tempo. Fonte: Autor.	64
7.2	Sinais típicos dos ensaios com PEC no domínio do tempo, obtidos da subtração entre os sinais de referência e medido. Fonte: Autor.	65
7.3	Espectros de amplitude das transformadas de Fourier típicas dos sinais de PEC. Fonte: Autor.	65
7.4	Típicos coeficientes de aproximação da DWT aplicada aos si- nais de ensaios com PEC. Foi utilizada o terceiro nível de filtragem da DWT com função <i>wavelet</i> mãe db4. Fonte: Autor.	66
7.5	Curva de carga para a PCA calculada a partir da DFT (a) e DWT (b). Fonte: Autor.	66
7.6	<i>Boxplot</i> do valor do primeiro componente principal calculado a partir da DFT (a) e DWT (b). Fonte: Autor.	68
7.7	<i>Boxplots</i> dos valores dos 7 primeiros componentes principais calculado a partir da DFT. Fonte: Autor.	68
7.8	Gráfico da média da acurácia e desvio padrão em função do número de neurônios na camada intermediária para ANN trei- nadas com sinais no domínio do tempo e pré-processados com DFT	70
7.9	Acurácia calculada para diferentes valores da função wavelet mãe e do número de neurônios na camada intermediária para ANN treinadas com sinais pré-processados com DWT utili- zando funções da classe bior.	70
7.10	Acurácia calculada para diferentes valores da função wavelet mãe e do número de neurônios na camada intermediária para ANN treinadas com sinais pré-processados com DWT utili- zando funções da classe coiflet.	71
7.11	Acurácia calculada para diferentes valores da função wavelet mãe e do número de neurônios na camada intermediária para ANN treinadas com sinais pré-processados com DWT utili- zando funções da classe Daubechies. Os valores máximos con- tém a marcação“M” para facilitar a visualização.	72

7.12	Gráfico de acurácia em função da variância acumulada e do número de neurônios na camada intermediária para ANN treinadas com sinais pré-processados com DFT+PCA. Os valores máximos contém a marcação “M” para facilitar a visualização.	73
7.13	Gráfico de acurácia em função da variância acumulada e do número de neurônios na camada intermediária para ANN treinadas com sinais pré-processados com DWT+PCA. Os valores máximos contém a marcação “M” para facilitar a visualização.	73
7.14	<i>Boxplot</i> do produto das eficiências para ANN treinada por diferentes métodos com sinais sem pré-processamento (informação no domínio do tempo).	76
7.15	<i>Boxplot</i> do produto das eficiências para ANN treinada por diferentes métodos com sinais pré-processados por DWT. . . .	78
7.16	<i>Boxplot</i> do produto das eficiências para ANN treinada por diferentes métodos com sinais pré-processados por DFT. . . .	80
7.17	<i>Boxplot</i> do produto das eficiências para ANN treinada por diferentes métodos com sinais pré-processados por DWT+PCA.	82
7.18	<i>Boxplot</i> do produto das eficiências para ANN treinada por diferentes métodos com sinais pré-processados por DFT+PCA.	85
7.19	Gráfico de TD_{SD} em função de TF_{SD} para diferentes técnicas de pré-processamento e algoritmos de treinamento.	87
7.20	Visão perimétrica do corpo de prova indicando o local de realização de ensaios com PEC e as respectivas predições realizadas pela ANN treinada pelo método ANN-BP Rprop.	89
7.21	Visão perimétrica do corpo de prova indicando o local de realização de ensaios com PEC e as respectivas predições realizadas pela ANN treinada pelo método ANN-PSO $f_1(\mathbf{P})$	90
7.22	Visão perimétrica do corpo de prova indicando o local de realização de ensaios com PEC e as respectivas predições realizadas pela ANN treinada pelo método ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 0,5\%$ e sinais pré-processados por DWT+PCA.	91
7.23	Visão perimétrica do corpo de prova indicando o local de realização de ensaios com PEC e as respectivas predições realizadas pela ANN treinada pelo método ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 1\%$ e sinais pré-processados por DFT+PCA. .	93

Lista de Tabelas

4.1	Matriz de confusão para 3 classes.	25
6.1	Dimensões das regiões com defeito no corpo de prova.	50
6.2	Área da superfície do corpo de prova correspondente as regiões sem defeito, com defeito externo e com defeito interno.	51
6.3	Informações sobre o conjunto de sinais obtidos com ensaio PEC.	52
6.4	Hiper-parâmetros utilizados no algoritmo Adam para testar topologias de redes neurais.	53
6.5	Parâmetros considerados na escolha da topologia da rede neural de acordo com o tipo de pré-processamento utilizado.	54
6.6	Parâmetros utilizado no PSO.	57
7.1	Variância acumulada em função do número de componentes principais utilizados neste trabalho.	67
7.2	Estruturas de ANN utilizadas. Todas ANN são do tipo MLP com função de ativação tangente hiperbólica.	74
7.3	Características do espaço de busca do algoritmo PSO considerando cada tipo de pré-processamento do sinal de entrada.	74
7.4	Matriz de confusão (em %) e respectivo PE (em %) para sinais sem pré-processamento (informação no domínio do tempo). Considerando a ANN com maior PE de cada método de treinamento.	77
7.5	Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) para sinais sem pré-processamento (informação no domínio do tempo). Considerando a ANN com maior PE de cada método de treinamento.	77
7.6	Matriz de confusão (em %) e respectivo PE (em %) para sinais pré-processados por DWT. Considerando a ANN com maior PE de cada método de treinamento.	79
7.7	Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) para sinais pré-processados por DWT. Considerando a ANN com maior PE de cada método de treinamento.	80

7.8	Matriz de confusão (em %) e respectivo PE (em %) para sinais pré-processados por DFT. Considerando a ANN com maior PE de cada método de treinamento.	81
7.9	Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) para sinais pré-processados por DFT. Considerando a ANN com maior PE de cada método de treinamento.	82
7.10	Matriz de confusão (em %) e respectivo PE (em %) para sinais pré-processados por DWT+PCA. Considerando a ANN com maior PE de cada método de treinamento.	83
7.11	Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) para sinais pré-processados por DWT+PCA. Considerando a ANN com maior PE de cada método de treinamento.	84
7.12	Matriz de confusão (em %) e respectivo PE (em %) para sinais pré-processados por DFT+PCA. Considerando a ANN com maior PE de cada método de treinamento.	85
7.13	Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) para sinais pré-processados por DFT+PCA. Considerando a ANN com maior PE de cada método de treinamento.	86
7.14	Matriz de confusão (em %) referente ao teste final da rede treinada com ANN-BP Rprop e sinais pré-processados por DFT.	89
7.15	Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) referente ao teste final da rede treinada com ANN-BP Rprop e sinais pré-processados por DFT.	89
7.16	Matriz de confusão (em %) referente ao teste final da rede treinada com ANN-PSO $f_1(\mathbf{P})$ e sinais pré-processados por DFT.	90
7.17	Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) referente ao teste final da rede treinada com ANN-PSO $f_1(\mathbf{P})$ e sinais pré-processados por DFT.	90
7.18	Matriz de confusão (em %) referente ao teste final da rede treinada com ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 0,5\%$ e sinais pré-processados por DWT+PCA.	92
7.19	Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) referente ao teste final da rede treinada com ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 0,5\%$ e sinais pré-processados por DWT+PCA.	92
7.20	Matriz de confusão (em %) referente ao teste final da rede treinada com ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 1\%$ e sinais pré-processados por DFT+PCA.	92
7.21	Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) referente ao teste final da rede treinada com ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 1\%$ e sinais pré-processados por DFT+PCA.	92

Lista de Símbolos

- δ - profundidade de penetração
 μ - permeabilidade magnética do corpo de prova
 σ - condutividade elétrica do corpo de prova
 f - frequência
 $\mathbb{R}^n$ - conjunto dos números reais de ordem n
 γ_i - classe i
 Γ - conjunto de classes
 G - conjunto de funções discriminantes
 $g_i(\mathbf{x})$ - função discriminante da classe i
 $\mathbf{x}$ - vetor de características que representa um objeto
 $m_{i,j}$ - elemento da matriz de confusão
 $\hat{m}_{i,j}$ - elemento da matriz de confusão normalizada
 b - bias
 w_j - peso sináptico de um neurônio individual
 w_{ij} - peso sináptico de um neurônio que compõe uma rede neural
 $\phi(v)$ - função de ativação dos neurônios
 y - saída de um neurônio
 $\mathbf{B}$ - matriz que contém os bias de determinada camada da rede neural
 $\mathbf{W}$ - matriz que contém os pesos sinápticos entre duas camadas da rede neural
 $\mathbf{Y}$ - matriz com a saída de uma camada da rede neural
 $\bar{x}$ - média da variável x
 dp_x - desvio padrão da variável x
 x' - variável x após a normalização z -score
 t - tempo contínuo
 ω - frequência angular contínua
 $X(\omega)$ - transformada de Fourier
 T - período de amostragem no tempo
 X_r - transformada discreta de Fourier
 N_0 - número de valores amostrados de $x(t)$
 ω_0 - intervalo de amostragem na frequência
 $DWT(p, k)$ - transformada wavelet discreta
 ψ - função wavelet mãe
 $C_A[k]$ - componentes de aproximação

$C_D[k]$ - componentes de detalhe
 $l[k]$ - filtro passa-baixa
 $h[k]$ - filtro passa-alta
 $\mathbf{L}$ - transformação linear associada à PCA
 $\mathbf{z}$ - vetor de componentes principais
 h - iteração do PSO
 $\mathbf{v}_m(h)$ - velocidade da partícula
 c_1 e c_2 - constantes cognitiva e social respectivamente
 r_1 e r_2 - números aleatórios
 $\mathbf{p}_m(h)$ - posição da partícula
 $\mathbf{p}_{\text{best}}(h)$ - posição em que a partícula apresentou melhor *fitness*
 $\mathbf{g}_{\text{best}}(h)$ - posição em que foi obtido o melhor *fitness* de todo o enxame
 $f(X)$ - função a ser otimizada (*fitness*)
 $h_i(X)$ e $g_j(X)$ - funções que definem as restrições do PSO
 $f'(X)$ - função *fitness* expandida
 f_{pior} - *fitness* da pior partícula factível no enxame
 f_1 e f_2 - funções *fitness* 1 e 2 respectivamente

Lista de Siglas

- CUI - *Corrosion Under Insulation* (Corrosão Sob Isolamento Térmico)
END - Ensaio Não Destrutivo
PEC - *Pulsed eddy current* (correntes parasitas pulsadas)
ANN - *Artificial Neural Network* (redes neurais artificiais)
GA - *Genetic Algorithm* (algoritmo genético)
PSO - *Particle Swarm Optimization* (otimização por enxame de partículas)
PCA - *Principal Components Analysis* (análise de componentes principais)
MSE - *Mean Square Error* (erro quadrático médio)
ECT - *Eddy Current Testing* (Teste com Correntes Parasitas)
SVM - *Support Vector Machine* (Máquina de Vetor de Suporte)
 VP_i - Verdadeiros Positivos da classe γ_i
 FP_i - Falsos Positivos da classe γ_i
 FN_i - Falsos Negativos da classe γ_i
 VN_i - Verdadeiros Negativos da classe γ_i
 TD_i - Taxa de detecção da classe γ_i
 TF_i - Taxa de falso alarme da classe γ_i
MLP - *Multi layer perceptron* (perceptron multicamadas)
SGD - *Stochastic Gradient Descent*
Adam - *Adaptive Moment Estimation*
LM - *Levenberg-Marquardt*
Rprop - *Resilient backpropagation*
CS - *Cuckoo Search* (busca cuco)
FA - *Firefly Algorithm* (algoritmo de vagalume)
ABC - *Artificial Bee Colony* (colônia artificial de abelhas)
DFT - *Discret Fourier Transform* (transformada discreta de Fourier)
FFT - *Fast Fourier Transform* (transformada rápida de Fourier)
DWT - *Discret Wavelet Transform* (transformada wavelet discreta)
MSD - *Multiresolution Signal Decomposition* (decomposição multi-resolução do sinal)
QMF - *Quadrature Mirror Filter* (bancos de filtros em quadratura)
ELD - *Economic Load Dispatch* (despacho econômico de carga)
PID - Proporcional Integral Derivativo

IAE - *Integral of Absolute Error* (integral do erro absoluto)
FOPID - *Fractional Order Proportional Integral Derivative* (PID de ordem fracionada)
SA - *Simulated Annealing* (Recozimento Simulado)
RMSE - *Root mean square error* (raiz quadrada do erro quadrático médio)
GMR - *Giant Magneto-Resistance*
SD - Sem defeito
DE - Defeito externo
DI - Defeito interno
GS - *Grid Search* (busca em grade)
PE - Produto das eficiências
ANN-BP - Redes neurais treinadas com *backpropagation* e gradiente descendente
ANN-PSO - Redes neurais treinadas pelo método proposto com PSO

Capítulo 1

Introdução

Corrosão de tubos metálicos é um problema comum em determinadas indústrias e que pode provocar perdas de produção, risco para os operários, danos ambientais e, conseqüentemente, perdas financeiras significativas. A detecção de corrosão nem sempre pode ser realizada por inspeção visual, tornando-se ainda mais difícil quando os tubos possuem algum tipo de revestimento térmico de material composto, ou quando a corrosão localiza-se na parte interna dos tubos [1].

Um tipo de corrosão em tubos bastante relevante é a Corrosão Sob Revestimento (CUI - *Corrosion Under Insulation*), por ser a principal causa de vazamentos em refinarias e indústrias químicas, representando elevado custo financeiro com manutenção [1]. Apenas nos Estados Unidos os gastos relacionados a CUI foram estimados em 1,8 bilhões de dólares em 2014 [2].

Entretanto, com auxílio de um ensaio não destrutivo (END) [1] pode-se detectar a corrosão inacessível à inspeção visual. Este tipo de ensaio, ou inspeção, não altera as propriedades físicas e químicas do corpo de prova. Portanto, podem ser utilizados para inspecionar tubos sem comprometer sua integridade. Posteriormente, o resultado da inspeção é analisado por um operador. Alternativamente, pode-se também utilizar um classificador para fornecer informação de suporte a decisão. Um classificador pode identificar o estado do corpo de prova de maneira rápida e eficiente, auxiliando o operador na tomada de decisão quanto a existência de condições anormais (defeitos).

Existem diferentes formas de criar um classificador utilizando aprendi-

zagem de máquina. Entretanto, os algoritmos de aprendizado tradicionais desconsideram aspectos operacionais relacionados às distintas consequências de possíveis erros de classificação. Para contornar esta limitação, pode-se utilizar métodos heurísticos com inclusão de restrições de treinamento, conforme proposto por [3], que obteve sucesso ao incluir restrições no treinamento de classificadores neurais binários.

1.1 Motivação

Em uma aplicação para detecção de corrosão em tubos industriais, caso uma peça defeituosa seja classificada como não defeituosa, os devidos reparos não são realizados, aumentando o perigo de acidentes. Por outro lado, caso uma peça não defeituosa seja classificada como defeituosa, a manutenção será realizada desnecessariamente. Portanto, nesta aplicação os erros de classificação possuem consequências distintas para a operação da planta industrial. No primeiro caso o erro de classificação aumenta o perigo de acidentes, podendo ocasionar perdas devido à interrupção da produção. No segundo caso, o erro de classificação aumenta desnecessariamente os custos com manutenção.

Portanto, as distintas consequências dos possíveis erros de classificação são aspectos importantes a serem considerados ao interpretar o resultado de ensaios não destrutivos para detecção de corrosão. Assim sendo, é importante que o classificador utilizado pelo operador seja capaz de considerar cada erro de classificação de forma distinta. Entretanto, os métodos tradicionais de concepção de classificadores consideram estes erros de forma equivalente, o que não condiz com a realidade de detecção de corrosão.

Para contornar esta tipo de limitação, foram utilizadas heurísticas bioinspiradas para incluir restrições durante o treinamento de classificadores neurais em [3]. Contudo, o método proposto em [3] é voltado para classificação binária e neste trabalho é considerado que o corpo de prova pode apresentar três condições: (i) sem corrosão, (ii) com corrosão na parte externa ou (iii) com corrosão na parte interna, ou seja, trata-se de um problema de classificação multiclasse. Portanto, para este trabalho, a metodologia pro-

posta em [3] precisa ser ampliada para classificadores multiclasse.

1.2 Objetivo

O objetivo deste trabalho é expandir a metodologia de treinamento de classificadores neurais proposta em [3] para classificação multiclasse, de forma a adequá-lo às aplicações de detecção de corrosão. O método de treinamento proposto neste trabalho possibilita considerar os diferentes erros de classificação de forma distinta por meio da inclusão de restrições de treinamento. O classificador neural concebido com o método proposto permite gerar informação que auxilia o operador na tomada de decisão. Entretanto, cabe ao operador tomar a decisão final quanto ao resultado do ensaio.

O trabalho foi dividido nos seguintes objetivos específicos:

1. Definir o tipo de ensaio não destrutivo apropriado para inspecionar o corpo de prova;
2. Definir técnicas de pré-processamento apropriadas para o sinal obtido no ensaio não destrutivo;
3. Escolher uma heurística adequada para treinar classificadores neurais incluindo restrições durante o treinamento;
4. Considerar restrições de treinamento como minimizar a quantidade de erros de classificação nos quais uma peça defeituosa é classificada como não defeituosa e maximizar a quantidade de acertos de classificação de peças sem defeito;
5. Comparar o método proposto com classificadores neurais treinados por métodos tradicionais.

1.3 Metodologia

A fim de obter um conjunto de dados para treinamento de classificadores neurais, foram realizados ensaios não destrutivos em tubos adquiridos

de uma indústria petroquímica. Considerando os diferentes tipos de ensaio não destrutivo e aspectos como a simplicidade de execução e a segurança para o operador, decidiu-se pela utilização do ensaio com correntes parasitas pulsadas (PEC - *Pulsed eddy current*) [1], [4].

Para facilitar o treinamento do classificador, a informação de interesse foi pré-processada, de forma a evidenciar certas características discriminantes. Para comparação, foram utilizadas três técnicas de processamento: transformada discreta de Fourier (DFT - *Discrete Fourier Transform*), transformada discreta Wavelet (DWT - *Discrete Wavelet Transform*) e análise de componentes principais (PCA - *Principal Components Analysis*).

A DFT e DWT realizam a transformada de um sinal no domínio do tempo para os domínios da frequência e do tempo-frequência, respectivamente [5], [6]. A PCA é uma transformação linear que mapeia um sinal multidimensional para uma nova base de representação na qual os componentes (projeções) são mutuamente não-correlacionados e ordenados, de forma decrescente, pela variância explicada (energia). Essa é uma transformação linear ótima para compactação, considerando o erro quadrático médio de reconstrução da informação [7].

Para encontrar a topologia de rede com melhor desempenho de classificação, foram testados classificadores neurais com diferentes números de neurônios na camada intermediária. Posteriormente, foi utilizado o método proposto neste trabalho para treinar classificadores neurais, cuja forma de tratar restrições de treinamento foi baseada em técnicas de manipulação de restrições voltadas para problemas de otimização. Por fim, os resultados obtidos com o método proposto foram comparados com os métodos tradicionais de treinamento.

1.4 Organização do documento

Esta dissertação está organizada da seguinte forma. No capítulo 2 o problema abordado neste trabalho é descrito. Nele é abordado o problema da corrosão sob isolamento térmico, a forma como classificadores podem ser utilizados na classificação automática dos sinais obtidos nos ensaios não destru-

tivos. Por fim, formas de recuperação do tubo danificado são apresentadas.

No capítulo 3 é apresentado o ensaio com correntes parasitas pulsadas que pode ser utilizado para inspecionar tubos industriais. É feita uma revisão de trabalhos relacionados e em seguida são apresentados equipamentos que são utilizados na indústria para detecção de corrosão sob revestimento.

No capítulo 4 é feita uma revisão bibliográfica sobre sistemas de classificação. É apresentado o conceito de rede neural artificial, formas de treiná-la e como utilizar pré-processamento adequado para melhorar o desempenho de classificação. Também é feita uma revisão sobre transformada de Fourier, transformada wavelet e análise de componentes principais.

No capítulo 5 é apresentada a otimização por enxame de partículas. Inicialmente é definida a forma geral de um problema de otimização. Em seguida são apresentadas as equações utilizada para mover as partículas pelo espaço de busca e abordagens para trabalhar com restrições de treinamento. No final do capítulo é apresentada a forma utilizada na literatura para treinar redes neurais utilizando otimização por enxame de partículas.

No capítulo 6 é apresentada o método proposto neste trabalho. O procedimento utilizado para realizar os ensaios com correntes parasitas pulsadas é detalhado, bem como os procedimentos de processamento da informação dos ensaios. Por fim, o método proposto para treinamento de redes neurais com otimização por enxame de partículas é explicado em detalhes.

No capítulo 7 são detalhados os resultados obtidos em cada etapa dos experimentos. O capítulo se inicia pelos resultados obtidos no ensaio com correntes parasitas pulsadas. Em seguida, os resultados da transformada de Fourier e wavelet e da análise de componentes principais são abordados. Posteriormente, são mostrados os resultados dos testes para determinação da estrutura da rede neural. Para então mostrar os resultados obtidos com as redes neurais.

No capítulo 8 são apresentadas as conclusões obtidas neste trabalho. O método de treinamento de redes neurais proposto mostrou ser adequado para aplicações de classificação onde os erros de cada classe possuem influência distinta no desempenho da aplicação de interesse. O método proposto resultou

em maiores valores de produto das eficiências e menor chance de classificar um tubo com defeito como sendo não defeituoso.

No apêndice A são apresentados os artigos que foram produzidos a partir do trabalho desenvolvido nesta dissertação. Estão separados em artigos aceitos em congresso e em revista. No apêndice B é apresentado o endereço do repositório no qual o código fonte desenvolvido neste trabalho foi compartilhado.

Capítulo 2

Apresentação do problema

Este capítulo apresenta o problema de pesquisa tratado neste trabalho. Inicialmente o problema de corrosão em tubos industriais é abordado, diferenciando-se a corrosão sob revestimento da corrosão interna. Em seguida é apresentada uma discussão sobre os erros de classificação que podem ocorrer ao se trabalhar com ensaios não destrutivos aplicados a detecção de corrosão.

2.1 Corrosão sob revestimento

Em diferentes setores da indústria há instalações complexas de tubos que precisam receber revestimento térmico como medida para diminuir as perdas energéticas [8]. A título de exemplo, na indústria de laticínios 50% da energia consumida é utilizada no aquecimento e resfriamento do leite. A não utilização de revestimento térmico adequado resulta em perdas energéticas consideráveis [9]. A redução de perdas energéticas está relacionada diretamente com a diminuição de emissões de CO_2 . Outra vantagem é que a temperatura tende a manter-se mais uniforme, melhorando o desempenho da planta [10].

Existem diversos materiais que podem ser utilizados para revestimento térmico. A escolha adequada deve levar em consideração a temperatura de operação da planta. Para isolamento em altas temperaturas os seguintes materiais pode ser utilizados: lã de vidro, vidro de celulose, silicato de cálcio

cio, fibra cerâmica, espuma de melamina, perlita expandida e lã de rocha. Para isolamento em baixas temperaturas pode-se utilizar os seguintes materiais: borracha nitrílica expandida, espuma fenólica, espuma de polietileno, polipropileno, poliestireno, espuma de poliuretano e borracha sintética expandida [9].

Existem ainda trabalhos que relatam a utilização de espaçamento entre o tubo e o revestimento, como forma de reduzir a quantidade de material isolante [11]. Existem também pesquisas de novos materiais isolantes, a título de exemplo o trabalho [10] apresenta a caracterização e aplicação de um compósito cerâmico.

Apesar da boa isolamento térmica, muitos materiais não são totalmente impermeáveis a vapores d'água, permitindo a passagem de pequenas quantidades de vapor. A água também pode penetrar no isolamento através de pequenas imperfeições causadas pelo envelhecimento do material, estresse mecânico, má instalação, dentre outros motivos [1].

A penetração de água na isolamento pode causar corrosão de tubos metálicos suscetíveis a oxidação. Este tipo de corrosão é conhecida como corrosão sob revestimento (CUI - *Corrosion Under Insulation*). A Figura 2.1 mostra dois casos de tubos corroídos. Tende a manter-se indetectável por estar escondida, até que a isolamento seja removida para inspeção. É um problema muito comum nas indústrias de petróleo, gás e química que pode causar vazamentos, resultando em acidentes e perda na produção [1].

A corrosão pode ocorrer mesmo em tubos com pintura resistente a água [1]. Uma alternativa é aplicar uma solução *spray* de $Al - Al_2O_3$ na superfície do tubo antes de aplicar o isolamento térmico. Entretanto, esta solução se desgasta rapidamente na presença de altas concentrações de $NaCl$ [12].

Para que a corrosão ocorra são necessários reações de oxidação (perda de elétrons), redução (ganho de elétrons) e um eletrólito (o exemplo mais simples é água com oxigênio dissolvido). Ocorrem as seguintes reações de oxidação no ferro $Fe \rightarrow Fe^{2+} + 2e$ e $Fe \rightarrow Fe^{3+} + 3e$ e a seguinte reação de redução $O_2 + 2H_2O + 4e \rightarrow 4OH^-$.

O OH^- reage com Fe^{2+} e Fe^{3+} formando $Fe(OH)_2$ e $Fe(OH)_3$ res-

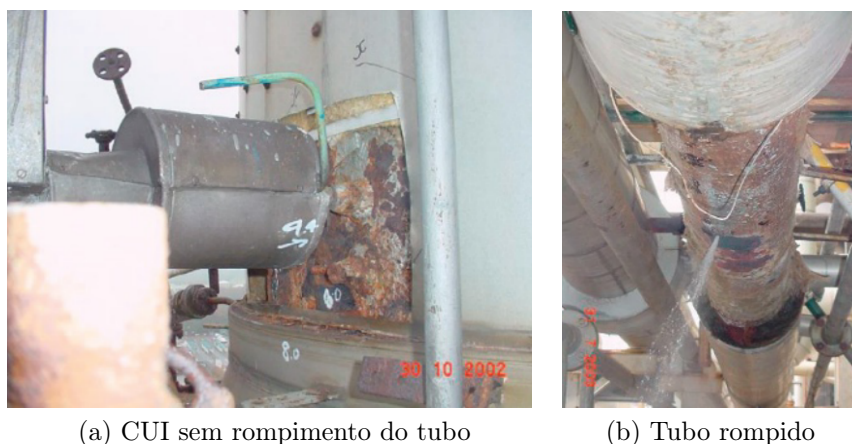


Figura 2.1: Exemplos de tubos com corrosão sob isolamento (CUI). Fonte: [1].

pectivamente. A equação geral da oxirredução, considerando Fe^{3+} , é $4Fe + 6H_2O + 3O_2 \rightarrow 4Fe(OH)_3$. Em seguida ocorre a seguinte reação $4Fe(OH)_3 \rightarrow 6H_2O + 2Fe_2O_3$. Sendo o óxido de ferro (Fe_2O_3) a substância conhecida como ferrugem [1].

As perdas causadas pela CUI apresentam-se em diferentes aspectos. Sejam perdas financeiras, ambientais, de reputação da empresa, de produção ou vidas humanas, elas devem ser levadas em consideração durante o planejamento de programas para combate da CUI [1].

Só nos EUA os gastos relacionados a CUI foram estimados em 1,8 bilhões de dólares no ano de 2014 [2]. Segundo [1], CUI representando 10% dos custos de manutenção em refinarias e indústrias químicas. CUI é a principal causa de vazamentos em refinarias e indústrias químicas. Estes vazamentos podem causar contaminação do solo, ar e cursos d'água, que conseqüentemente tem impactos na fauna e flora local. A segurança dos funcionários também é impactada na ocorrência de vazamentos [1].

A diminuição da reputação da empresa devido à ocorrência de danos ambientais causados por CUI pode influenciar na decisão de órgãos ambientais quanto a liberação ou não de licenças de operação. O valor de mercado da empresa também pode ser afetado, uma vez que, a decisão de investidores em comprar ou vender ações pode ser influenciada pela reputação da empresa [1].

A ocorrência de vazamentos devido à CUI causa perdas na produção.

O custo total da perda depende do volume de material perdido durante o vazamento e da forma como isso influencia na qualidade do produto final [1].

Os vazamentos causados por CUI podem resultar em graves danos físicos aos funcionários. Casos que merecem atenção especial são aqueles em que o tubo opera em pressão elevada e/ou transporta produtos químicos perigosos para a saúde de seres humanos [1].

2.2 Corrosão Interna de Tubos

A corrosão interna de tubos industriais é relacionada à composição química do material transportado. No caso da indústria petrolífera, por exemplo, é comum o fluxo de água misturada ao petróleo cru. Esta água geralmente contém gases dissolvidos como dióxido de carbono (CO_2) e sulfureto de hidrogênio (H_2S). Grande parte da corrosão ocorre devido ao CO_2 que ao reagir com a água forma ácido carbônico (H_2CO_3) [13], [14].

Devido ao meio ácido, ocorrem reações de oxidação do ferro de acordo com a equação $Fe \rightarrow Fe^{2+} + 2e$. Uma possível reação de redução (ganho de elétrons) neste meio é a redução do hidrogênio, representada pela equação $2H^+ + 2e \rightarrow 2H \rightarrow H_2$. Combinando estas duas equações e adicionando água e dióxido de carbono, tem-se a reação global do processo corrosivo do ferro por dióxido de carbono $Fe + CO_2 + H_2O \rightarrow FeCO_3 + H_2$ [14].

Este fenômeno é influenciado pelas características do ambiente interno ao tubo, como potencial hidrogeniônico (PH), pressão parcial de CO_2 , temperatura e o fluxo do fluido. Caso estas características sejam apropriadas, o $FeCO_3$ forma uma camada protetora, impedindo o avanço da corrosão. Entretanto, em outros casos esta camada não se deposita na parede interna do tubo, facilitando o processo corrosivo [14].

A Figura 2.2 mostra parte de um tubo utilizado na indústria petrolífera com corrosão na parte interna por CO_2 . Neste caso, trata-se de uma corrosão localizada do tipo alveolar, que forma sulcos ou escavações no metal. A água e o petróleo formam uma mistura heterogênea. O fluxo de água, em geral ocupa a região inferior do tubo, caso este esteja na horizontal [13], [14],

conforme mostrado na Figura 2.3. Portanto, é provável que a região corroída no tubo da Figura 2.2 seja a parte na qual a água circulava.



Figura 2.2: Tubo com corrosão interna por CO_2 . Fonte: [14].

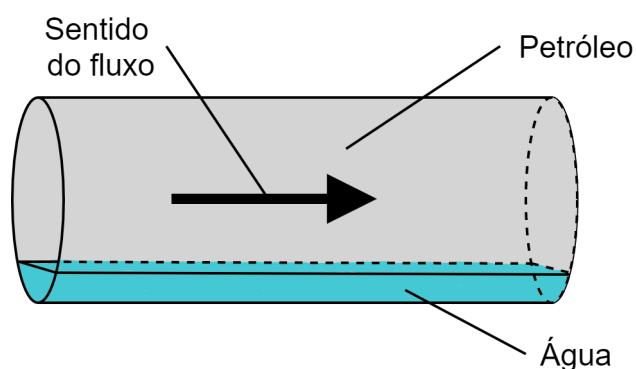


Figura 2.3: Separação de água e petróleo dentro de um tubo na horizontal.

2.3 Erros de classificação

Com auxílio de um ensaio não destrutivo (END) adequado é possível identificar o estado do tubo quanto à existência de corrosão e identificar se é interna ou externa. O resultado do END é, então, interpretado por um operador, que pode utilizar um classificador para auxiliá-lo na tomada de decisão. Caso o tubo apresente algum tipo de corrosão, o procedimento de manutenção adequado deve ser realizado.

O resultado do ensaio depende do tipo de END utilizado. Podendo ser representado por um sinal no domínio do tempo ou até mesmo imagem. Contudo, o resultado pode ser de difícil interpretação para o operador, seja pela grande semelhança do sinal no domínio do tempo para ensaios em regiões com defeito ou sem, ou pelo tamanho reduzido do defeito que pode não ser bem representado em uma imagem. Para auxiliar na interpretação do resultado do ensaio, o operador pode utilizar um classificador que forneça como saída uma indicação quanto ao estado do corpo de prova.

Entretanto, podem ocorrer erros de classificação, que influenciam negativamente na escolha do procedimento de manutenção e ser realizado. A quantidade de erros pode ser minimizada adotando-se um bom algoritmo de treinamento, inicialização dos parâmetros adequada, dados de treinamento de boa qualidade, dentre outros fatores.

Em aplicações de detecção de corrosão cada tipo de erro possui consequências distintas, pois influenciam diretamente na escolha dos procedimentos de manutenção. Porém, os métodos de treinamento de classificadores consideram os diferentes erros de classificação de forma equivalente.

Neste trabalho, a prioridade é evitar erros de classificação que comprometam a segurança dos funcionários, como classificar um tubo com algum tipo de corrosão como sem corrosão. Caso isso aconteça, os funcionários ficam expostos a tubulações que não receberam a devida manutenção, podendo ocasionar acidentes.

Um erro que aumente desnecessariamente os custos de manutenção, apesar de indesejável, é menos crítico que os erros anteriores. Esse tipo de erro ocorre quando um tubo sem corrosão é classificado como apresentando corrosão, ocasionando a mobilização desnecessária da equipe de manutenção.

2.4 Procedimentos de Manutenção de Tubos Com Corrosão

Após detecção e classificação do tipo de corrosão deve-se iniciar os devidos procedimentos de manutenção. Nos casos mais graves pode ser necessário a

substituição completa do tubo. Entretanto, para casos menos graves pode-se aplicar algum tipo de reparo para aumentar a vida útil do tubo. O procedimento adequado depende da extensão da corrosão de sua localização, ou seja, se está na parte externa ou interna do tubo [15].

Para corrosão sob revestimento, ou seja, na parte externa do tubo, os procedimentos de manutenção são mais simples, visto que com a remoção do revestimento a área afetada pela corrosão torna-se acessível aos operários. Os procedimentos mais comuns neste caso são aplicação de braçadeira, soldagem de manga metálica e reforço com material composto [15].

Para aplicação de braçadeira, utiliza-se duas peças em formato de semi-cilíndrico que ao serem unidas envolvem o tubo, fortalecendo a região comprometida pela corrosão. Estas braçadeira podem ser compostas de metal ou de material composto, que possuem a vantagem de serem mais leves e resistentes a corrosão [15]. A Figura 2.4 mostra uma braçadeira de material compósito utilizada para reparar um tubo com corrosão externa.



Figura 2.4: Braçadeira de material composto (em amarelo) instalada em volta de tudo com corrosão externa (em preto). Fonte: [15]

Uma manga metálica tem formato similar ao da braçadeira mostrada acima. As diferenças na utilização são caracterizadas pela ausência de parafusos para fixação das peças em formato semicilíndrico e aplicação de solda para unir os dois semicilindros entre si e ao corpo do tubo que necessita de reparo [15], [16]. A Figura 2.4 mostra o aspecto final do tubo ao receber uma manga metálica.

Outra alternativa é aplicar uma cobertura de material compósito, como

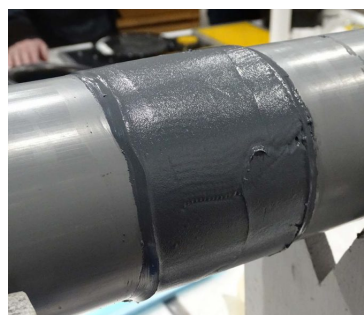


Figura 2.5: Manga de metal instalada em volta de tudo com corrosão externa. Fonte: [16]

resina epoxy ou fibras de poliéster, sobre a região comprometida. Deve-se remover o revestimento, limpar a região que necessita de reparo removendo excesso de oxidação para então fortalecer o tubo aumentando sua espessura com um material compósito adequado [15]. A Figura 2.6 mostra duas opções comerciais de matérias compósitos para reparo de tubos industriais.



(a) *Stop It*



(b) *ARC 5 Emergency Metal To Metal Repair*

Figura 2.6: Duas soluções para reparo em tubos com corrosão. Em 2.6a *Stop It* da fabricante *TRI-CHEM*. Fonte: [17]. Em 2.6b *ARC 5 Emergency Metal To Metal Repair* da fabricante *Cactus Industrial*. Fonte: [18]

Se a corrosão for interna, o acesso a superfície corroída é mais complicado. Ainda pode-se aplicar os procedimentos utilizados para reparar tubos com corrosão externa. Entretanto deve-se considerar que a região corroída continuará exposta e conseqüentemente aumentará com o tempo, tornando a substituição completa do tubo inevitável após certo tempo. Abordagens mais adequadas baseiam-se no acesso ao interior do tubo por meio de ferra-

mental apropriado. Pode-se citar a técnica de reparo de aço expandido e o reparo utilizando um robô [15].

Para aplicar um reparo de aço expandido, uma sonda com uma câmera acessa o interior do tubo para fotografar a região com corrosão. Em seguida outra sonda acessa o interior do tubo levando um reparo metálico fino e maleável. A sonda realiza uma expansão de forma a moldar o reparo à parede interna do tubo. Este processo se repete até que o reparo tenha espessura suficiente para garantir a resistência mecânica do tubo [15]. A Figura 2.7 mostra as etapas de inserção do reparo metálico no interior do tubo com corrosão interna.



Figura 2.7: Procedimento de aplicação de reparo expandido em um tubo com corrosão interna. Fonte: [15]

Pode-se ainda utilizar um robô como mostrado na Figura 2.8, que carrega ferramentas capazes de realizar a limpeza da área corroída e em seguida aplicar *spray* para revestir a superfície interna do tubo [15].



Figura 2.8: Robô utilizado em procedimento de reparo em um tubo com corrosão interna. Fonte: [15]

Diante do apresentado acima conclui-se que, é importante a correta identificação da localização da corrosão, ou seja se é interna ou externa. Tal informação é importante para definir o tipo de procedimento adotado na manutenção, visto que as técnicas adotadas diferem para cada caso. Portanto, não é adequado simplesmente utilizar um classificador binário que apenas identifique o tubo entre sem corrosão ou com corrosão.

Capítulo 3

Ensaaios Com Correntes Parasitas Pulsadas

3.1 Ensaaios Não Destrutivos Para Detecção De Corrosão

Para evitar os males causados pela corrosão, tanto externa quanto interna, técnicas de ensaios não destrutivos (END) como radiografia e ultrassom podem ser utilizadas para detectar CUI antes que a corrosão resulte no rompimento do tubo [2].

Radiografia possui as desvantagens de necessitar de aparato complexo composto por uma fonte de raio x e uma superfície receptora, apenas pequenas áreas podem ser inspecionadas a cada aplicação do raio x e o manuseio de raio x pode trazer riscos a saúde do operador, que necessita de bom conhecimento técnico para realizar os ensaios [2].

Ultrassom proporciona menos riscos para o operador em comparação com a radiografia. Porém, para garantir um bom acoplamento entre o aparato de ultrassom e o corpo de prova é necessário remover parte da isolamento térmica [19]. Entretanto, determinadas plantas não podem operar sem a isolamento térmica, ainda que apenas parcialmente removida, devido à segurança dos funcionários e perdas energéticas [1].

Uma alternativa que é simples, segura para o operador e que não necessita de remoção do isolamento térmico possibilita a realização do ensaio de ma-

neira mais prática e sem a necessidade de interromper a operação da planta. Portanto, a técnica de ensaio não destrutivo escolhida para este trabalho é Correntes Parasitas Pulsadas (PEC - *Pulsed Eddy Current*) [20].

3.2 Correntes Parasitas Pulsadas

A técnica de ensaio com Correntes Parasitas (ECT - *Eddy Current Testing*) é baseada na indução eletromagnética [21]. Uma bobina excitada com corrente alternada gera um campo magnético primário também alternado e com mesma frequência da corrente que o gerou, processo descrito pela lei de Ampère [22].

Segundo a lei de Faraday [22], este campo magnético induz corrente elétrica no corpo de prova. Esta corrente por sua vez gera um campo magnético secundário em oposição ao campo primário. A presença de falhas no corpo de prova altera a corrente, que altera o campo secundário. Portanto, o campo magnético secundário fornece informações sobre a estrutura física do corpo de prova [21], [23].

A escolha das frequências utilizadas no teste deve levar em consideração a profundidade de penetração do campo magnético. A profundidade de penetração (equação 3.1) diminui com o aumento da frequência, sendo δ a profundidade de penetração, μ a permeabilidade magnética do corpo de prova, σ a condutividade elétrica do corpo de prova e f a frequência do sinal elétrico que excita a bobina [21].

$$\delta = \frac{1}{\sqrt{\pi\mu\sigma f}} \quad (3.1)$$

Para aumentar a eficiência do ECT pode-se excitar a bobina com a combinação de correntes de diferentes frequências. Como a profundidade de penetração do campo magnético varia com a frequência da corrente elétrica, a utilização de várias frequências simultaneamente possibilita obter informações sobre diferentes profundidades do corpo de prova de maneira mais rápida [21].

Na técnica de Correntes Parasitas Pulsadas (PEC - *Pulsed Eddy Current*),

a bobina pode ser excitada com uma onda quadrada, triangular ou dente de serra. Estas ondas possuem largo espectro de frequências. Portanto, PEC pode fornecer mais informações que ECT com uma única frequência ou mesmo com a combinação de várias frequências [21], [24]. A Figura 3.1 ilustra o princípio de funcionamento da inspeção por PEC.

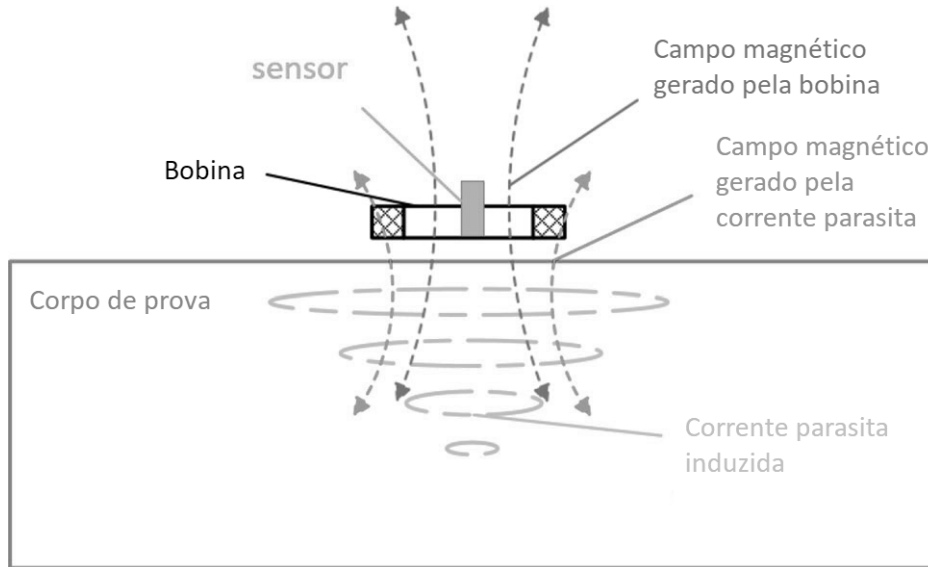


Figura 3.1: Ilustração do princípio da técnica PEC [23].

PEC pode ser utilizada para detectar falhas em tubos metálicos sem o alterar e não necessita de bom acoplamento entre a bobina e o corpo de prova [20]. Isso proporciona uma vantagem em relação a utilização de ultrassom, visto que neste o acoplamento entre a sonda e o corpo de prova é essencial [4]. PEC é mais econômico e menos perigoso que a radiografia, que proporciona riscos a saúde do operador devido a utilização de radiação ionizante (raio X) [25].

Quanto à forma de representação do sinal de PEC existem basicamente três formas, A-scan, B-scan e C-scan [4], [26]. Na representação A-scan, o sinal de amplitude em função do tempo é mostrado na tela do aparelho, Figura 3.2. A representação B-scan mostra a seção transversal de uma peça podendo ser considerada como vários sinais A-scan realizados em diferentes posições (normalmente adjacentes) do corpo a ser inspecionado. A imagem exibida na tela do aparelho propicia ao operador a visão em corte no material inspecionado. A Figura 3.3 ilustra o modo de visualização B-scan, a região

verde representa o material em boas condições, enquanto que as demais cores representam regiões com defeito. Na representação C-scan, a peça é mostrada vista de cima. Detalhes das discontinuidades do material podem ser percebidas a partir das cores azul, amarelo e vermelho na imagem, Figura 3.4. As representações B-scan e C-scan necessitam de aparelhos especiais, portanto não são representações convencionais.

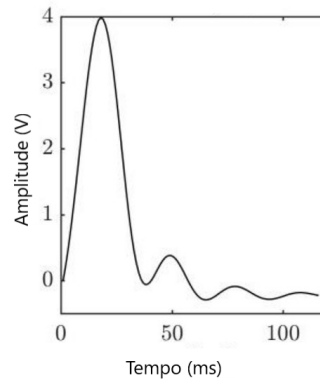


Figura 3.2: Modo de visualização A-Scan. Fonte: [26]

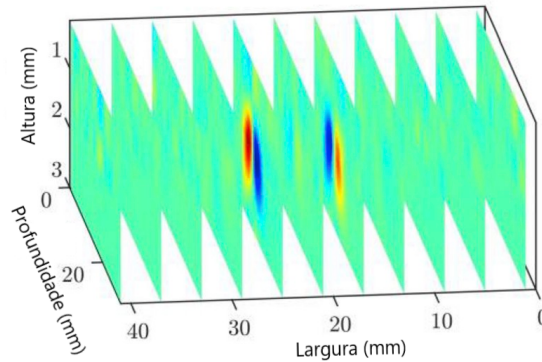


Figura 3.3: Modo de visualização B-Scan. Fonte: [26]

A detecção do defeito pode ser feita por análise temporal ou espectral nas representações A-scan, B-scan ou C-scan [4], [26], [27]. Pode-se ainda utilizar uma ferramenta automatizada de classificação que permita transpor a dificuldade de interpretar os sinais provenientes do ensaio utilizando apenas a experiência do profissional. Classificadores automáticos de ensaios com PEC foram proposto nos trabalhos [28]–[31].

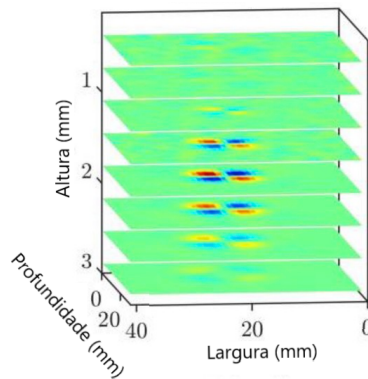


Figura 3.4: Modo de visualização C-Scan. Fonte: [26]

3.3 Trabalhos Relacionados

Em [28], um sensor de campo magnético foi proposto para ensaios com ECT. Após passar por um pré-processamento, o sinal foi classificado com uma árvore de decisão, conseguindo classificar corretamente 100% dos sinais. Em [29], descontinuidades em estruturas metálicas foram detectadas utilizando PEC. Cinco métodos de aprendizado de máquina foram comparados, sendo que a maior quantidade de classificações corretas foram obtidas com o método *Gradient boosting*. Em [30] foi proposta a utilização de uma rede neural convolucional para classificar o tipo e a profundidade de defeito a partir de ensaios com PEC. O modelo proposto proporcionou mais classificações corretas em relação aos métodos utilizados para comparação. Em [31] foi proposta uma rede neural para prever a profundidade e inclinação de rachaduras a partir de ensaios com PEC. As previsões da rede neural foram mais próximas dos valores reais que as previsões realizadas pelos demais métodos testados.

3.4 Ferramentas Comerciais Para Ensaios Com PEC

Alguns fabricantes de ferramentas para ensaios com PEC que podemos citar são a *Eddyfi Technologies* [32], *Maxwell ndt* [33] e *Ether NDE* [34]. Estes fabricantes produzem ferramentas para ensaios com PEC, que são compos-

tas por uma sondas com bobina e sensores de campo magnético e por um equipamento eletrônico adequados para excitação da bobina e visualização do resultado do ensaio.



Figura 3.5: Equipamento para ensaios com PEC da fabricante *Maxwell ndt*. Fonte: [33].

Estes equipamentos, entretanto, são limitado à medição da espessura do corpo de prova. O que dificulta descobrir se a corrosão é interna ou externa. Outra desvantagem destes equipamentos é que eles nem sempre fornecem uma informação clara quanto a existência de corrosão. Cabe ao operador chegar a essa conclusão a partir de informações referentes a espessura em diferentes regiões do corpo de prova

Capítulo 4

Sistemas De Classificação

Este capítulo apresenta as técnicas utilizadas neste trabalho para detectar e classificar a corrosão em tubos industriais. Na escolha do ensaio não destrutivo adequado para esta aplicação foram considerados aspectos relacionados à segurança, praticidade e aplicabilidade em ambiente industrial. Estas considerações levaram a escolha de ensaios com correntes parasitas pulsadas. Em seguida, sistemas de classificação e algumas métricas de desempenho são definidos. Para concepção do classificador, optou-se por utilizar redes neurais artificiais devido à possibilidade de aplicar restrições de treinamento com PSO, conforme proposto em [3]. É mostrado como redes neurais artificiais são estruturadas e como podem ser utilizadas como sistemas de classificação. Alguns métodos de pré-processamento de sinais são apresentados como forma de melhorar o desempenho de classificação da rede. Otimização por enxame de partículas é apresentada e em seguida é mostrado como utilizar este método de otimização no treinamento de redes neurais artificiais. O PSO foi escolhido como método de treinamento de redes neurais devido à sua simplicidade, velocidade de execução e bons resultados previamente obtidos em [3].

4.1 Sistemas de classificação

Classificar consiste em identificar a qual classe determinado objeto pertence, considerando que, classe é um conjunto que contém objetos seme-

lhantes. Cada objeto possui um conjunto de características, que podem ser numéricas (como pressão, velocidade, comprimento e temperatura) ou categóricas (como profissão, tipo sanguíneo e nacionalidade) [35]. Desde que as características do objeto sejam numéricas, um classificador pode ser definido matematicamente como a função:

$$D : \mathbb{R}^n \rightarrow \Gamma \quad (4.1)$$

em que n é o número de características do objeto a ser classificado. Considerando que o objeto pode pertencer a uma de c classes, denominadas de γ_1 a γ_c e ordenadas da seguinte forma $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_c\}$

Em relação ao número de classes do problema, os classificadores podem ser agrupados da seguinte forma:

- Binários - classificam os objetos em duas classes distintas, sendo que o objeto pertence a apenas uma das classes (um novo medicamento pode ser classificado como sendo ou não eficaz);
- Multiclasse - classificam os objetos em três ou mais classes, sendo que o objeto pertence a apenas uma das classes, que são mutuamente excludentes (um ser vivo pode ser classificado como pertencente a um dos 5 reinos dos seres vivos);
- Multilabel - classificam os objetos em duas ou mais classes, sendo que o objeto pode pertencer a mais de uma classe (um fruto pode ser classificado como maduro, cítrico e orgânico, neste caso o objeto pode pertencer a mais de uma classe).

Este trabalho tem foco em classificadores multiclasse, cujo modelo canônico é um conjunto de c funções discriminantes $G = \{g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_c(\mathbf{x})\}$,

$$g_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, 2, \dots, c, \quad (4.2)$$

sendo $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ o vetor de características que representa um objeto. Cada função calcula uma pontuação (*score*) para a classe que representa. Utilizando a regra de associação máxima (*Maximum Membership rule*), o

objeto é identificado como pertencente à classe cuja função discriminante retornou maior pontuação.

A concepção de um classificador passa pelas etapas de escolha do modelo de classificador, treinamento (aprendizagem) e finalmente teste, etapa na qual é averiguada a capacidade do classificador em identificar corretamente novos objetos.

Após a concepção do classificador, é preciso avaliá-lo quanto a sua performance. Utilizando matriz de confusão, é possível analisar como as predições corretas e incorretas estão distribuídas entre as classes. Esta matriz é construída cruzando as predições feitas pelo classificador com as classificações desejadas, ou seja, com as classes que os objetos realmente pertencem. Neste trabalho as colunas da matriz representam as classes reais e as linhas representam as classes preditas pelo classificador. A Tabela 4.1 mostra um exemplo de matriz de confusão para o caso de um classificador de 3 classes.

Tabela 4.1: Matriz de confusão para 3 classes.

		Classe real		
		γ_1	γ_2	γ_3
Classe Predita	γ_1	$m_{1,1}$	$m_{1,2}$	$m_{1,3}$
	γ_2	$m_{2,1}$	$m_{2,2}$	$m_{2,3}$
	γ_3	$m_{3,1}$	$m_{3,2}$	$m_{3,3}$

Para esta matriz, os elementos da diagonal principal representam os objetos corretamente classificados, também chamados de Verdadeiros Positivos da classe γ_i (VP_i). Os elementos de cada linha i que estão fora da diagonal principal representam objetos classificados como pertencentes a classe γ_i , mas que pertencem a outras classes. Somando estes elementos temos Falsos Positivos da classe γ_i (FP_i). Os elementos de cada coluna j que estão fora da diagonal principal representam objetos da classe γ_i que foram erroneamente classificados como pertencentes a outra classe. Somando estes elementos temos Falsos Negativos da classe γ_j (FN_j). Por fim, temos os Verdadeiros Negativos da classe γ_i (VN_i), que são os elementos que restam ao retirar-se a linha e a coluna referente a classe γ_i .

A partir da matriz de confusão pode-se calcular alguns indicadores de desempenho global, como acurácia e confusão para o classificador (D) con-

siderado. A acurácia (equação 4.3) é a proporção de objetos classificados corretamente em relação a todos objetos classificados. A confusão, também chamada de erro, é a proporção de objetos classificados de forma errada em relação ao número total de objetos classificados. Conforme a equação 4.4 a confusão pode ser calculada a partir da acurácia.

$$\text{Acurácia}(D) = \frac{\sum_{i=1}^c m_{i,i}}{\sum_{i=1}^c \sum_{j=1}^c m_{i,j}} \quad (4.3)$$

$$\text{Confusão}(D) = 1 - \text{Acurácia}(D) \quad (4.4)$$

Pode-se ainda calcular indicadores de desempenho para cada classe γ_i , como probabilidade de detecção da classe γ_i (PD_i) e probabilidade de falso alarme, também conhecido como falso positivo, da classe γ_i (PF_i). A probabilidade de detecção da classe γ_i é a probabilidade de um objeto da classe γ_i ser corretamente classificado, é calculada conforme a equação 4.5. A probabilidade de falso alarme (equação 4.6) da classe γ_i é a probabilidade de um objeto ser classificado como pertencente a classe γ_i , sendo que ele pertence a outra classe.

$$PD_i(D) = \frac{VP_i}{VP_i + FN_i} \quad (4.5)$$

$$PF_i(D) = \frac{FP_i}{FP_i + VN_i} \quad (4.6)$$

Na matriz de confusão da Tabela 4.1 as colunas representam a classe real. Portanto, a soma dos elementos de uma coluna é igual ao número de elementos da respectiva classe que foram aplicados ao classificador. Com base nisso, a matriz de confusão pode ser normalizada a partir da divisão dos elementos de uma coluna pela soma dos elementos da respectiva coluna. Ou seja, para cada coluna j da matriz de confusão, o elemento normalizado $\hat{m}_{i,j}$ é calculado de acordo com a equação 4.7.

$$\hat{m}_{i,j} = \frac{m_{i,j}}{\sum_{i=1}^c m_{i,j}} \quad (4.7)$$

em que i e j são os índices que representa as linhas e colunas, respectivamente, da matriz e varia entre 1 e c .

Na matriz de confusão normalizada a soma dos elementos de qualquer coluna j sempre é igual a 1, $\sum_{i=1}^c \hat{m}_{i,j} = 1$. Portanto, o valor máximo que cada elemento da matriz pode assumir é 1. Caso a matriz normalizada seja multiplicada por 100%, os elementos da matriz são representados em porcentagem e temos que $\sum_{i=1}^c \hat{m}_{i,j} = 100\%$.

O produto das eficiências (PE) é a média geométrica dos elementos da diagonal principal da matriz de confusão normalizada (equação 4.8). Caso um dos elementos possua valor demasiadamente baixo, o produto das eficiências tende a ser significativamente reduzido. A ocorrência de um valor nulo zera o produto das eficiências, ainda que os demais elementos da diagonal principal possuam valores altos.

$$PE(D) = \sqrt[c]{\prod_{i=1}^c \hat{m}_{i,i}} \quad (4.8)$$

A concepção de um classificador pode ser feita com diversos métodos como árvore de decisão, máquina de vetor de suporte, redes neurais artificiais, dentre outros [36]. Para este trabalho é importante considerar as distintas consequências de erros de classificação, visto que influenciam nas decisões quanto à procedimentos de manutenção. Portanto, redes neurais artificiais foram utilizadas, visto que com redes neurais é possível aplicar o método proposto em [3] para impor restrições de treinamento, considerando cada tipo de erro de classificação de forma distinta.

4.2 Redes Neurais Artificiais

Uma rede neural artificial (ANN - *Artificial Neural Network*) é um modelo matemático inspirado no funcionamento do sistema nervoso de um organismo inteligente. Trata-se da interligação de neurônios artificiais de forma a obter uma estrutura capaz de resolver problemas sem necessitar de uma solução analítica. Por ser um aproximador universal de funções, pode ser empregada

em regressão, agrupamento e classificação [37].

Apesar das pesquisas iniciais com ANN terem se inspirado na neurociência, métodos mais recentes baseiam-se fortemente em matemática e engenharia. Portanto, uma rede neural não busca imitar perfeitamente o modelo de um cérebro. O objetivo ao se treinar uma ANN é construir uma determinada função baseada na conexão de funções menores, representadas pelos neurônios [38].

4.2.1 Neurônio Artificial

A unidade central de processamento de uma ANN é o neurônio artificial, cujo funcionamento pode ser resumido em três etapas. (i) cada neurônio recebe “ n ” entradas, que são multiplicadas por pesos sinápticos. (ii) os resultados destas multiplicações são somados e (iii) passam por uma função de ativação $\phi(v)$, conforme figura 4.1 [39]. Um neurônio artificial é descrito matematicamente pelas equações 4.9 e 4.10.

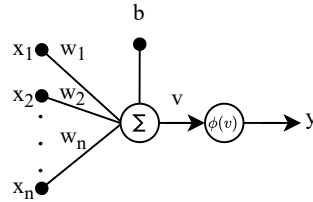


Figura 4.1: Modelo de um neurônio. Fonte: Adaptado de [39].

$$v = \sum_{j=1}^n w_j \times x_j + b \quad (4.9)$$

$$y = \phi(v) \quad (4.10)$$

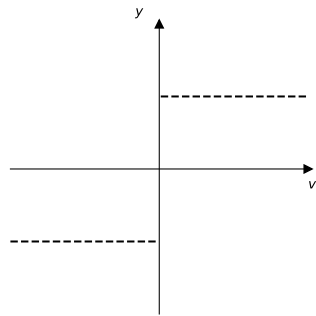
Existem diversas funções de ativação que podem ser aplicadas em neurônios artificiais. Algumas delas são a função degrau (equação 4.11), linear (equação 4.12), logística (equação 4.13) e tangente hiperbólica (equação 4.14). Os gráficos destas funções estão representados na Figura 4.2.

$$\phi(v) = \begin{cases} +1, & \text{se } v \geq 0 \\ -1, & \text{se } v < 0 \end{cases} \quad (4.11)$$

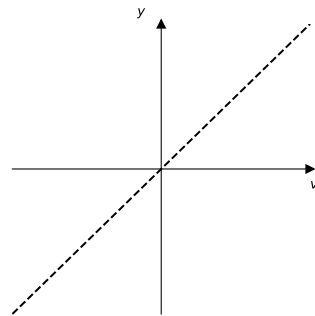
$$\phi(v) = a \times v \quad (4.12)$$

$$\phi(v) = \frac{1}{1 + e^{-v}} \quad (4.13)$$

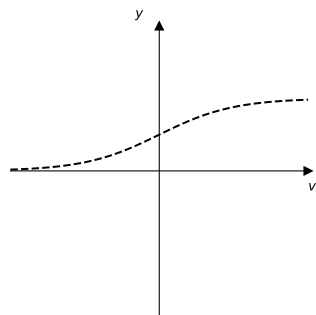
$$\phi(v) = \tanh(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}} \quad (4.14)$$



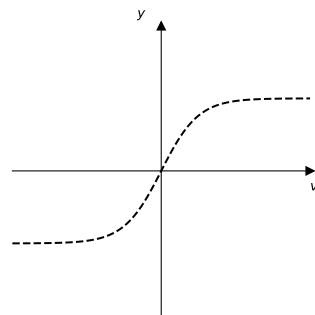
(a) Função degrau



(b) Função linear



(c) Função logística



(d) Função tangente hiperbólica

Figura 4.2: Funções de ativação para neurônios artificiais.

4.2.2 Rede Perceptron Multicamadas

Um único neurônio possui capacidade de processamento apenas para problemas de baixa complexidade. A capacidade de processamento aumenta com a interligação de vários neurônios, que pode ser feita em camadas. A rede perceptron multicamadas (MLP - *multi layer perceptron*) é formada por uma camada de entrada, uma ou mais camadas intermediárias (escondidas) e uma camada de saída [38].

Caso todos os neurônios de uma camada sejam interligados a todos neurônios da camada anterior, a rede é chamada totalmente conectada. Os dados aplicados na camada de entrada são propagados camada a camada até a saída da rede, se não houver realimentação dos neurônios com dados da própria camada ou de camadas posteriores, a rede é chamada *feed-forward*. A Figura 4.3 mostra uma rede MLP *feed-forward* totalmente conectada.

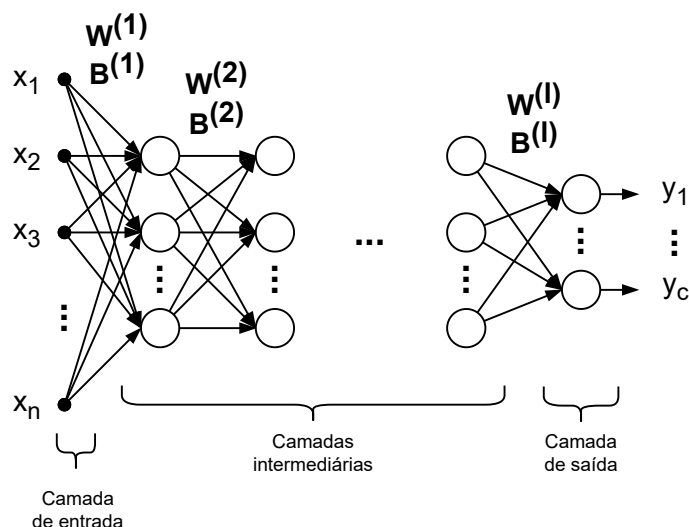


Figura 4.3: Rede *multilayer perceptron forward* totalmente conectada. Os valores de *bias* não estão representados para melhor visualização. Fonte: Adaptado de [39].

Segundo o teorema proposto em [40], dada uma função $f(\cdot)$ existe uma ANN com uma única camada intermediária capaz de gerar uma aproximação de $f(\cdot)$, desde que a ANN possua quantidade suficiente de neurônios na camada intermediária. Porém, o número de neurônios necessários pode ser excessivamente grande, tornando a concepção da ANN impraticável. Uma

possível solução é reorganizar os neurônios em duas ou mais camadas escondidas. Nesta abordagem é possível criar uma ANN capaz de gerar uma aproximação de $f(\cdot)$ utilizando menor número de neurônios em comparação à ANN com uma única camada intermediária [38].

Na concepção da ANN deve-se considerar que redes com grande número de camadas ou de neurônios necessita de um conjunto de treino grande. Utilizar um conjunto de treino pequeno para treinar uma ANN demasiadamente grande pode gerar *overfitting*, comprometendo sua capacidade de generalização. Caso se utilize um algoritmo de treinamento baseado em gradiente descendente, deve-se considerar o efeito de *vanishing*. A medida que o gradiente do erro na camada de saída é propagado em direção da camada de entrada, pode acontecer de o gradiente se tornar cada vez menor, dificultando o processo de aprendizado [38].

Portanto, apesar de redes com mais camadas e neurônios possibilitarem gerar aproximações de funções cada vez mais complexas, deve-se ponderar o tamanho da rede de acordo com a dificuldade de se gerar a aproximação da função desejada e com o tamanho do conjunto de dados.

Uma ANN do tipo MLP pode ser representada matematicamente por um conjunto de matrizes. Para cada camada l tem-se as matrizes $\mathbf{W}^{(l)}$ e $\mathbf{B}^{(l)}$. A matriz $\mathbf{W}^{(l)}$ representa os pesos sinápticos que conectam a camada l com a camada anterior, onde cada linha contém os pesos referentes a um neurônio. A matriz coluna $\mathbf{B}^{(l)}$ representa a polarização da camada, cada linha representa o bias referente a um neurônio. Os dados que fluem pela rede podem ser representados pelas matrizes $\mathbf{X}$, $\mathbf{V}^{(l)}$ e $\mathbf{Y}^{(l)}$. A matriz $\mathbf{X}$ contém os dados que alimentam a respectiva camada da rede. Para simplificar as operações matemáticas, cada instância do conjunto de dados é disposta em uma coluna, portanto, o número de colunas da matriz $\mathbf{X}$ corresponde ao número de instâncias do conjunto de dados. A primeira etapa da propagação dos dados pela camada resulta na matriz $\mathbf{V}^{(l)}$, calculada pela equação 4.15. Em seguida os dados passam pela função de ativação do neurônio, conforme equação 4.16, e são armazenados na matriz $\mathbf{Y}^{(l)}$. Como a saída $\mathbf{Y}^{(l)}$ de uma camada é a entrada $\mathbf{X}$ da camada seguinte, estas equações são repetidas camada a camada até a saída da rede.

$$\mathbf{V}^{(l)} = \mathbf{W}^{(l)} \times \mathbf{X} + \mathbf{B}^{(l)} \quad (4.15)$$

$$\mathbf{Y}^{(l)} = \phi(\mathbf{V}^{(l)}) \quad (4.16)$$

4.2.3 Treinamento Baseado no Gradiente do Erro

Para que a ANN funcione conforme o esperado, é preciso treiná-la, ou seja, alterar seus pesos sinápticos para que os dados de entrada resultem em dados de saída de acordo com a necessidade do problema. Existem duas formas principais de treinamento. O treinamento supervisionado é aquele que busca associar cada entrada a uma saída (alvo) pré-estabelecida automaticamente ou por um humano (supervisor). Neste caso, um algoritmo de treinamento altera os pesos sinápticos para que a entrada corresponda à saída desejada.

O treinamento não supervisionado é aquele em que a saída não é pré-definida por um supervisor. Neste caso, o treinamento da ANN é realizado de forma a evidenciar certas características intrínsecas ao conjunto de dados. Estas definições são informais, visto que a diferença entre treinamento supervisionado e não supervisionado não é formalmente definida. Não existe um teste objetivo que determine se um determinado valor é uma saída pré-estabelecida por um supervisor ou uma característica intrínseca ao conjunto de dados.

Neste trabalho ANN foram treinadas de forma supervisionada e empregadas para classificação. A função de ativação dos neurônios é a tangente hiperbólica, cuja saída varia entre $[-1, 1]$. Portanto, ao aplicar determinado objeto na entrada, o sinal desejado na saída da ANN deve ser 1 no neurônio que correspondente a classe do objeto e -1 nos demais neurônios.

O algoritmo *backpropagation* é utilizado no treinamento supervisionado para obter o gradiente do erro de saída em relação aos parâmetros da rede. Seu princípio baseia-se na retropropagação do erro entre a saída calculada e desejada. Na primeira fase do treinamento, também chamada de fase *forward*, os dados na camada de entrada são propagados camada após camada até a saída. O erro entre a saída desejada e a obtida é então calculado. Na segunda

fase do treinamento, também chamada de fase *backward*, o erro da saída é propagado para as camadas anteriores e os pesos sinápticos são atualizados utilizando o algoritmo gradiente descendente de forma a minimizar o erro na saída [41].

O gradiente descendente não é um algoritmo único, existem diferentes formas de realizar a atualização dos pesos sinápticos. Como SGD - (*Stochastic Gradient Descent*), Adam - (*Adaptive Moment Estimation*) [42], LM - (*Levenberg-Marquardt*) [43] e Rprop - (*Resilient backpropagation*) [44].

4.2.4 Outros métodos de treinamento

Métodos de treinamento baseados no gradiente do erro possuem a desvantagem de realizar uma busca local no espaço de busca, culminando na convergência da solução para um mínimo local. Portanto, a solução fornecida não necessariamente é a melhor possível. Métodos baseados em heurísticas bio-inspiradas tem sido aplicados com sucesso para realizar uma busca global, proporcionando uma solução mais próxima do mínimo global. A Figura 4.4 ilustra uma curva hipotética com a representação de dois mínimos locais e o mínimo global. Alguns exemplos são algoritmo genético (GA - *Genetic Algorithm*) [45], busca cuco (CS - *Cuckoo Search*) [46], algoritmo de vagalume (FA - *Firefly Algorithm*) [47], colônia artificial de abelhas (ABC - *Artificial Bee Colony*) [48] e otimização por enxame de partículas (PSO - *Particle Swarm Optimization*) [49].

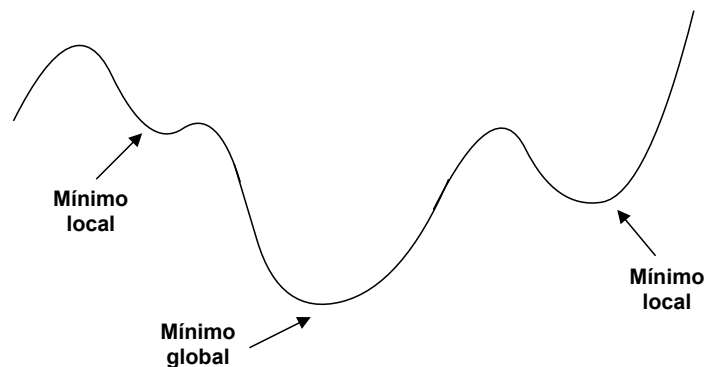


Figura 4.4: Curva hipotética com dois mínimos locais e o mínimo global.

Outra desvantagem de métodos baseados no gradiente do erro é a dificul-

dade de tratar cada classe de forma distinta durante o treinamento da ANN. Em algumas aplicações, a ocorrência de determinados erros de classificação têm impactos diferentes.

Neste trabalho é considerado que os erros de classificação possuem pesos distintos. Mais especificamente, classificar um tubo defeituoso como não defeituoso é uma situação altamente indesejada. Enquanto que, classificar um tubo não defeituoso como defeituoso, apesar de indesejado, é mais tolerável. No trabalho [3], o autor utilizou GA e PSO com sucesso para adicionar determinadas restrições durante o treinamento da ANN em problemas de classificação binária. Portanto, este método é mais apropriado para este trabalho. Entretanto, o método proposto por [3] é voltado para classificação binária. Assim sendo, este trabalho propõe uma adaptação para adicionar restrições no treinamento de ANN para problemas de classificação multiclasse.

4.3 Pré-processamento da informação

A utilização de técnicas de processamento aos sinais antes de aplicá-los a entrada da ANN pode ajudar a realçar características importantes, facilitando o processo de classificação [50]. A técnica empregada deve ser condizente com os tipos de sinais. Neste trabalho foram utilizadas as transformadas de Fourier e wavelet para evidenciar características importantes do sinal proveniente do ensaio PEC. Visto que, o efeito da “maldição da dimensionalidade” tende a dificultar a convergência do algoritmo de treinamento caso a ANN tenha grande número de parâmetros [51], foi utilizada a análise de componentes principais para reduzir a dimensão dos dados.

4.3.1 Transformada Discreta de Fourier

A Transformada Discreta de Fourier (DFT - *Discrete Fourier Transform*) realiza uma transformação de uma série temporal discreta (domínio do tempo) em um conjunto de senoides discretas (domínio da frequência) [52], sua forma é definida pela equação 4.17. Sendo T o período de amostragem de $x(t)$ e $x(iT)$ a i -ésima amostra de $x(t)$, definimos $x_i = x(iT)$. X_r é a r -ésima

componente no domínio da frequência discreta.

$$X_r = \sum_{i=0}^{N_0-1} x_i e^{-jr\Omega_0 i} \quad (4.17)$$

Uma forma de diminuir o esforço computacional utilizado para calcular a DFT é utilizar a Transformada Rápida de Fourier (FFT - *Fast Fourier Transform*), que aumenta a eficiência computacional aproximando o número total de amostras em uma potência de 2 [52].

4.3.2 Transformada Wavelet Discreta

A DFT converte um sinal do domínio do tempo para o domínio da frequência. Por outro lado, a transformada wavelet discreta (DWT - *Discrete Wavelet Transform*) é uma transformada que converte um sinal do domínio do tempo para o domínio tempo-frequência [53].

Esta conversão é feita representando o sinal como a combinação de funções wavelet, que são obtidas por meio do escalonamento e translação da função wavelet mãe $\psi(x)$ [53]. Uma maneira prática de calcular a DWT é por meio do algoritmo de decomposição multi-resolução do sinal (MSD - *Multiresolution Signal Decomposition*). Este algoritmo decompõe o sinal em componentes de aproximação e de detalhe, associados a filtros passa-baixa e passa-alta respectivamente [53]. Os componentes de aproximação C_A e de detalhe C_D são definidos pelas seguintes equações:

$$C_A[k] = \sum_i l[i - 2k]x[i] \quad (4.18)$$

$$C_D[k] = \sum_i h[i - 2k]x[i] \quad (4.19)$$

com $l[i - 2k]$ os coeficientes do filtro passa-baixa e $h[i - 2k]$ os coeficientes do filtro passa-alta.

Estes filtros são inter-relacionados e tem a mesma frequência crítica, que corresponde à metade da frequência máxima do sinal a ser filtrado. A relação

entre os dois filtros é $h[L-1-n] = (-1)^n l[n]$, sendo L o número de coeficientes do filtro e n um número inteiro [54].

A função wavelet mãe é calculada a partir do filtro passa-alta de maneira recursiva, de acordo com a equação $\psi(x) = \sum_n h[n]\psi(2x - n)$. De forma semelhante, a função wavelet pai, $\phi(x)$, é calculada a partir do filtro passa-baixa $\phi(x) = \sum_n l[n]\phi(2x - n)$ [54]. Entretanto, para especificar a DWT, em geral utiliza-se apenas a função wavelet mãe $\psi(x)$.

O MSD pode ser implementado com bancos de filtros em quadratura (QMF - *Quadrature Mirror Filter*) [55]. Os bancos de filtros podem ser colocados em cascata conforme Figura 4.5, neste caso também são chamados de bancos de filtros logarítmicos [56]. Após passar pelo filtro, o sinal passa por um processo de sub-amostragem de forma a manter a quantidade de amostras [57].

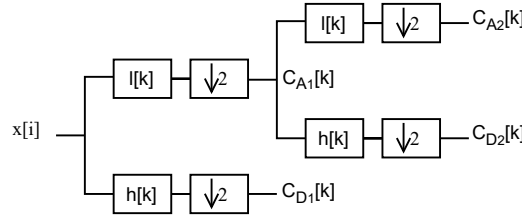


Figura 4.5: Banco de filtros de nível 2 para transformada discreta wavelet. Fonte: Adaptado de [58].

4.3.3 Análise de componentes principais

A transformada de Fourier pode extrair características importantes de um sinal. O sinal é representado como a composição de sinais senoidais. Entretanto, o conjunto de informação resultante pode ser demasiadamente grande, dificultando a análise. Neste trabalho foi utilizada a PCA para selecionar as características mais importantes de um sinal.

A PCA é um método de processamento estatístico de sinais, que aplica uma transformação linear $\mathbf{L}_{n,n}$ em um conjunto de sinais $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ resultando em um conjunto de projeções ortogonais entre si $\mathbf{z} = [z_1, z_2, \dots, z_n]^T$. Neste novo conjunto, um pequeno número de componentes concentra grande

parte da variância (energia). A transformação linear é calculada pela seguinte equação matricial:

$$\mathbf{z} = \mathbf{L}_{n,n} \times \mathbf{x} \quad (4.20)$$

A matriz $\mathbf{L}_{n,n}$ pode ser encontrada através do cálculo de autovalores e autovetores da matriz de covariância dos sinais. Os componentes principais são ordenados de modo decrescente em função da variância explicada por cada um deles [59].

Capítulo 5

Otimização Por Enxame De Partículas

Otimização por enxame de partículas (PSO - *Particle Swarm Optimization*) é uma heurística bio-inspirada proposta em [60]. De forma geral otimização é a busca de uma solução que maximize ou minimize determinada função, respeitando as restrições impostas [51]. Matematicamente, um problema de otimização pode ser representado conforme a equação 5.1.

$$\begin{aligned} & \text{Otimize } f(\mathbf{X}) \\ & \text{Restrito a : } h_i(\mathbf{X}) = 0, i = 1, 2, \dots, m \\ & \quad g_j(\mathbf{X}) \leq 0, j = 1, 2, \dots, p \end{aligned} \tag{5.1}$$

em que $f(\mathbf{X})$ é a função a ser otimizada (função objetivo), ou seja, maximizada ou minimizada, $h_i(\mathbf{X})$ e $g_j(\mathbf{X})$ são funções que definem as restrições e $\mathbf{X}$ é um vetor com os parâmetros que podem ser variados.

PSO baseia - se no comportamento de animais sociais, como pássaros, peixes e o ser humano, em que cada indivíduo possui inteligência própria e a compartilha com os demais indivíduos, gerando assim uma inteligência coletiva. Cada indivíduo corresponde a uma possível solução para um problema de otimização (minimização ou maximização de determinada variável). A cada iteração do algoritmo, as partículas são avaliadas de acordo com o valor da função objetivo, ou seja, sua capacidade de resolver o problema [60].

Na tentativa de encontrar uma solução ótima, as partículas são movidas

pelo espaço de busca. Primeiramente a velocidade de cada partícula é calculada segundo a equação 5.2, em que h é a iteração atual, $\mathbf{v}_m(h)$ é a velocidade da partícula na iteração h , c_1 e c_2 são as constantes cognitiva e social respectivamente, r_1 e r_2 são números aleatórios entre 0 e 1 amostrados de uma distribuição de probabilidade uniforme, $\mathbf{p}_m(h)$ é a posição da partícula na iteração h , $\mathbf{p}_{\text{best}}(h)$ é a posição em que a partícula apresentou melhor valor da função objetivo e $\mathbf{g}_{\text{best}}(h)$ é a posição em que foi obtido o melhor valor da função objetivo de todo o enxame. Em seguida a posição da partícula é modificada segundo a equação 5.3.

$$\mathbf{v}_m(h+1) = w(h) \times \mathbf{v}_m(h) + c_1 \times r_1 \times (\mathbf{p}_{\text{best}}(h) - \mathbf{p}_m(h)) + c_2 \times r_2 \times (\mathbf{g}_{\text{best}}(h) - \mathbf{p}_m(h)) \quad (5.2)$$

$$\mathbf{p}_m(h+1) = \mathbf{p}_m(h) + \mathbf{v}_m(h+1) \quad (5.3)$$

Segundo [61], PSO é um algoritmo simples, rápido e que proporciona bons resultados. Um cuidado que deve-se tomar ao trabalhar com PSO é a escolha dos valores das constantes cognitiva e social. Segundo [62] ocorre convergência prematura se a constante cognitiva for menor que a constante social. Esta convergência prematura impede que determinadas regiões do espaço de busca sejam exploradas pelo algoritmo.

PSO pode ser aplicado em diferentes problemas de otimização. No trabalho [62] utilizou-se PSO para resolver o problema de despacho econômico de carga (ELD - *Economic Load Dispatch*). ELD consiste em suprir um determinado sistema de potência consumindo o mínimo de combustível. Os autores ressaltam a incapacidade de métodos convencionais como gradiente, iteração lambda e Newton de resolver o problema em questão por se tratar de um problema de otimização não-convexa.

Estudos tem sido realizados a fim de criar variações do algoritmo PSO que proporcionem melhor desempenho. A inclusão de interações gravitacionais entre as partículas do enxame foi utilizada em [63] de forma a diminuir o número de coeficientes desconhecidos. A estratégia elite foi utilizada por [64] para evitar que as partículas fiquem presas em mínimos locais, o que carac-

teriza a convergência prematura do algoritmo. Mesmo motivo que levou [65] a modificar o PSO adicionando conceitos de teoria do caos e [66] a criar um algoritmo híbrido com PSO e Recozimento Simulado (SA - *Simulated Annealing*).

O PSO original foi desenvolvido para resolver problemas de otimização sem restrições. Posteriormente foram desenvolvidas abordagens que possibilitam resolver problemas de otimização com restrições utilizando PSO. Alguns mecanismos utilizados para tratar restrições adotados em problemas com PSO são: Aplicação de penalidade, abordagem separatista e transformação das restrições em um problema de otimização multiobjetivo [67].

Na abordagem de penalidade, primeiramente é calculada a função objetivo de cada partícula, em seguida as partículas são testadas em relação às restrições. Caso a partícula satisfaça as restrições (partícula factível) o valor da função objetivo permanece inalterado. Caso a partícula seja infactível, é aplicada uma penalidade de forma a aumentar o valor da função objetivo da partícula, caso trate-se de um problema de minimização, ou diminuir o valor da função objetivo da partícula, caso trate-se de um problema de maximização [68]. Esta abordagem é simples e comumente utilizada, tendo como desvantagem a dificuldade de especificar os valores dos parâmetros de penalidade [67].

A abordagem separatista consiste em tratar as partículas factíveis de forma diferente (ou separada) das partículas não factíveis [67]. De forma a satisfazer as seguintes condições:

1. Uma partícula factível ganha de uma não factível;
2. Entre duas partículas factíveis, aquela com melhor valor da função objetivo ganha;
3. Entre duas partículas não factíveis, ganha aquela com menor montante de restrições violadas.

As partículas infactíveis não são descartadas, elas são mantidas no exame. Entretanto, para satisfazer as condições 1 e 3, a função objetivo das partículas não factíveis é calculado de acordo com a equação 5.4.

$$f'(\mathbf{X}) = f_{pior} \pm \sum_{i=1}^p \text{Max}(0, g_i(\mathbf{X})) \quad (5.4)$$

em que $f'(\mathbf{X})$ é a função objetivo expandida, f_{pior} é o valor da função objetivo da pior partícula factível e $\sum_{i=1}^p \text{Max}(0, g_i(\mathbf{X}))$ representa o montante de restrições violadas. Note que na equação 5.4 o sinal “+” é utilizado para aumentar o valor da função objetivo da partícula infactível, tornando-a menos apta a resolver um problemas de minimização. Caso o objetivo do problema de otimização seja maximizar a função objetivo, o sinal “−” deve ser utilizado para diminuir o valor da função objetivo das partículas infactíveis.

Outra abordagem é converter o problema de otimização com restrições em um problema multi-objetivo. Nesta abordagem, deseja-se otimizar várias funções objetivo, que podem ser concorrentes entre si [69]. Um problema de otimização multi-objetivo pode ser representado conforme a seguir:

$$\begin{aligned} \text{Otimizar } f(\mathbf{X}) &= [f_1(\mathbf{X}), f_2(\mathbf{X}), \dots, f_k(\mathbf{X})] \\ \text{Restrito a : } h_i(\mathbf{X}) &= 0, i = 1, 2, \dots, m \\ g_j(\mathbf{X}) &\leq 0, j = 1, 2, \dots, p \end{aligned} \quad (5.5)$$

Este problema pode ser resolvido utilizando técnicas como dominância de Pareto [70], [71], ou decomposição do enxame em sub-enxames. Esta divisão permite que cada sub-enxame seja especializado na otimização de um objetivo permitindo que o algoritmo convirja rapidamente para a solução ótima de Pareto [72]. Segundo [67] a abordagem multi-objetivo é raramente utilizada e possui a desvantagem de ser um método de difícil resolução que sofre com perda de diversidade das partículas.

5.1 Trabalhos Com Otimização Por Enxame De Partículas

Para estimar a demanda de petróleo do Irã tomando como base dados socioeconômicos, [45] utilizou PSO e algoritmo genético (GA - *genetic algorithm*). Foram utilizados dados socioeconômicos entre os anos 1981 e 1999

para sintonizar os parâmetros dos modelos propostos. Com dados entre os anos 2000 e 2005 os modelos propostos foram testados, O menor erro relativo entre os valores reais e previstos pelo modelo foi de 1,36%.

Controladores Proporcional Integral Derivativos (PID) podem ter seus parâmetros otimizados utilizando PSO, conforme proposto em [73]–[76]. Em [73] o parâmetro integral do erro absoluto (IAE - *Integral of Absolute Error*) foi utilizado como função objetivo durante o processo de sintonia do controlador PID utilizando PSO. Em [74] foi utilizado PSO para sintonizar um controlador PID com dois graus de liberdade. Em [75] um controlador PID de ordem fracionada (FOPID - *Fractional Order Proportional Integral Derivative*) foi projetado utilizando PSO.

5.2 Treinamento De Redes Neurais Artificiais Com PSO

PSO tem sido tema de pesquisas voltadas para treinamento de redes neurais artificiais. Os métodos de treinamento baseados no gradiente do erro são bastante efetivos no treinamento de ANN, porém apresenta certas desvantagens como convergência para mínimos locais [77]. O treinamento de ANN por PSO proporciona melhor busca global [78] e a possibilidade de se adicionar restrições ao treinamento da ANN.

Os resultados destas pesquisas indicam que o PSO é uma boa alternativa ao gradiente descendente, proporcionando maiores valores de acurácia [79]–[81]. No trabalho [82], foi utilizado PSO no treinamento de uma rede neural para determinação do tempo de permanência de pacientes em hospitais. A ANN treinada com PSO obteve menor erro quadrático médio e maior Acurácia em relação a ANN treinada pelo gradiente descendente.

No trabalho [83] foi proposta uma metodologia de projeto de redes neurais baseada em PSO. Nesta metodologia, os pesos sinápticos, as funções de ativação e as devidas ligações entre os neurônios são determinadas utilizando PSO. Como resultado, as redes neurais obtidas apresentaram diferentes funções de ativação na mesma camada. Em alguns casos não houve ligações

entre certos neurônios na camada de entrada e os demais neurônios, indicando que aquela entrada não é relevante para o problema de classificação abordado.

O PSO é executado de forma que os pesos sinápticos da ANN representem uma partícula. A movimentação das partículas pelo espaço de busca equivale a alterar os pesos sinápticos. A forma como os pesos são codificados em uma partícula varia de acordo com o trabalho. Em [84] Os pesos sinápticos e os bias são dispostos em um vetor, que precisa ser decodificado para que as saídas da rede neural sejam calculadas. Em [85] cada partícula é formada por uma lista de quatro matrizes, duas representando os pesos entre as camadas de neurônios e duas representando o bias das camadas escondida e de saída.

Tipicamente, as funções objetivo empregadas no treinamento de redes neurais com PSO baseiam-se nas matrizes $\mathbf{Y}^{(l)}$ e $\mathbf{D}$. Sendo $\mathbf{Y}^{(l)}$ a matriz com as predições realizadas pela rede, com c linhas, correspondendo às c classes consideradas e m colunas, correspondendo aos m sinais de entrada, conforme a equação 5.6. A matriz $\mathbf{D}$ contém as predições desejadas, com c linhas, correspondendo às c classes consideradas e m colunas, correspondendo aos m sinais, conforme equação 5.7.

$$\mathbf{Y}^{(l)} = \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ y_{c,1} & y_{c,2} & \cdots & y_{c,m} \end{bmatrix} \quad (5.6)$$

$$\mathbf{D} = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{c,1} & d_{c,2} & \cdots & d_{c,m} \end{bmatrix} \quad (5.7)$$

A função objetivo mais frequentemente empregada no treinamento de redes neurais com PSO é o erro quadrático médio (MSE - *mean square error*) entre as saídas preditas pela rede $\mathbf{Y}^{(l)}$ e as saídas desejadas $\mathbf{D}$, definida na equação 5.8, como visto em [80], [81]. Também é utilizada a equação 5.9, que define a raiz do erro quadrático médio (RMSE - *root mean square error*) entre as saídas da rede $\mathbf{Y}^{(l)}$ e as saída desejadas $\mathbf{D}$, como visto nos trabalhos [79], [86]. Outras funções objetivo são menos comuns, como em [87], que usa a função objetivo definida pela equação 5.10.

$$MSE(\mathbf{Y}^{(l)}, \mathbf{D}) = \frac{1}{m} \sum_{i=1}^c \sum_{j=1}^m (y_{ij} - d_{ij})^2 \quad (5.8)$$

$$RMSE(\mathbf{Y}^{(l)}, \mathbf{D}) = \sqrt{\frac{1}{m} \sum_{i=1}^c \sum_{j=1}^m (y_{ij} - d_{ij})^2} \quad (5.9)$$

$$f(\mathbf{Y}^{(l)}, \mathbf{D}) = \frac{1}{1 + MSE} \quad (5.10)$$

Existem certas diferenças entre aprendizado e otimização que devem ser esclarecidas. Em aprendizado de máquina, a função P que se deseja otimizar, em relação ao conjunto de teste, pode ser intratável. Assim sendo, utiliza-se a abordagem de minimizar uma função de custo $J(\theta)$, esperando-se assim otimizar a função P indiretamente. Ao se treinar redes neurais, por exemplo, espera-se maximizar os acertos de classificação minimizando o erro quadrático médio entre a saída obtida com os dados de entrada e a saída desejada. Por outro lado, em otimização minimizar a função custo $J(\theta)$ é o objetivo final em si [41].

Outra diferença importante é que em aprendizado, a minimização da função de custo $J(\theta)$ é feita com informação do conjunto de treino. Espera-se que o modelo obtido seja capaz de generalizar sua capacidade de classificação para dados que não foram utilizados para treinar o classificador. Em otimização pura, não é esperado que o resultado encontrado tenha capacidade de generalização. Um exemplo de otimização pura é a busca dos parâmetros de um controlador PID de forma a atender um critério de desempenho previamente estabelecido [41].

Capítulo 6

Método Proposto

Inicialmente, é definida a sequência de processamento adotada neste trabalho. Em seguida é descrita a realização do ensaio com correntes parasitas pulsadas e posterior pré-processamento com DFT, DWT e PCA. Também é comentado o procedimento utilizado para escolher a função *wavelet* mãe mais adequada, o número de componentes principais da PCA e o número de neurônios na camada intermediária da rede neural artificial. O método proposto neste trabalho é descrito detalhadamente, enfatizando como codificar redes neurais como partículas, os parâmetros utilizados no enxame e as funções objetivo utilizadas. Por fim, é explicado o procedimento de validação cruzada adotado.

A sequência de processamento adotada neste trabalho é dividida em três etapas: (i) extração de características, (ii) seleção de características e (iii) classificação, conforme ilustrado na Figura 6.1. Na etapa de extração de características são aplicadas a DFT e DWT ao sinal do ensaio com PEC, de forma a comparar qual proporciona melhor desempenho de classificação. Para seleção de características foi utilizada a PCA. E para classificação dos dados foi utilizada ANN treinada pelo método proposto baseado em PSO com restrições.

Para facilitar o treinamento da ANN, os sinais de entrada passaram por uma normalização. No trabalho [88] foram avaliadas diferentes formas de normalização de dados para problemas de classificação. Os autores concluíram que a normalização *z-score* superou as demais, facilitando o processo de

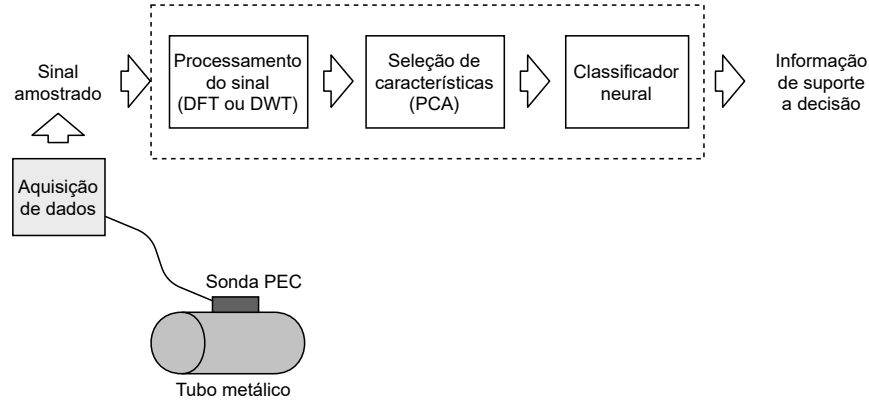


Figura 6.1: Diagrama de processamento.

treinamento da rede neural, que proporcionou melhor desempenho de classificação em comparação às redes treinadas com outros tipos de normalização.

Portanto, neste trabalho foi utilizada a normalização *z-score*, que é calculada conforme equação 6.1. Sendo x a variável a ser normalizada, $\bar{x}$ a média da variável x , dp_x o desvio padrão da variável x e x' a variável normalizada. Normalizar o sinal de entrada ajuda a evitar que o argumento da função de ativação, neste caso a tangente hiperbólica, assuma valores demasiadamente grandes em módulo. Caso no qual ocorreria a saturação da tangente hiperbólica, tornando o processo de aprendizado mais lento [89].

$$x' = \frac{x - \bar{x}}{dp_x} \quad (6.1)$$

Para fins de comparação, também foram utilizados classificadores neurais treinados apenas com informação no domínio do tempo, ou seja, sem as etapas (i) e (ii). E com a DFT ou DWT, porém sem aplicação da PCA, ou seja, com a etapa (i), mas sem a etapa (ii). Portanto foram utilizados ao todo cinco sequencias de processamento, tomando-se o cuidado de aplicar a normalização *z-score* antes de aplicar os dados na entrada da ANN, conforme pode ser visto na Figura 6.2. Sendo que a normalização *z-score* deve ser aplicada antes da PCA quando esta for utilizada.

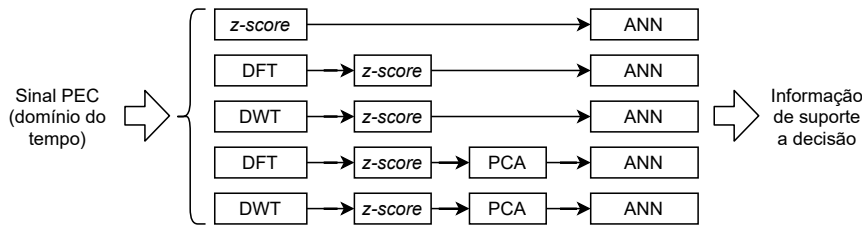


Figura 6.2: Detalhamento do diagrama de processamento.

6.1 Ensaio realizado com PEC

O corpo de prova utilizado neste trabalho é um tubo metálico com 250 mm de comprimento que possui áreas sem corrosão, com corrosão na parede interna e com corrosão na parede externa, Figura 6.3. A Figura 6.4 mostra a representação em corte do corpo de prova com a indicação das medidas em milímetros. A região em cinza corresponde ao isolamento térmico e a parte hachurada representa a parte metálica do tubo. As regiões com defeito interno (DI) e defeito externo (DE) estão localizadas em posições diametralmente opostas.

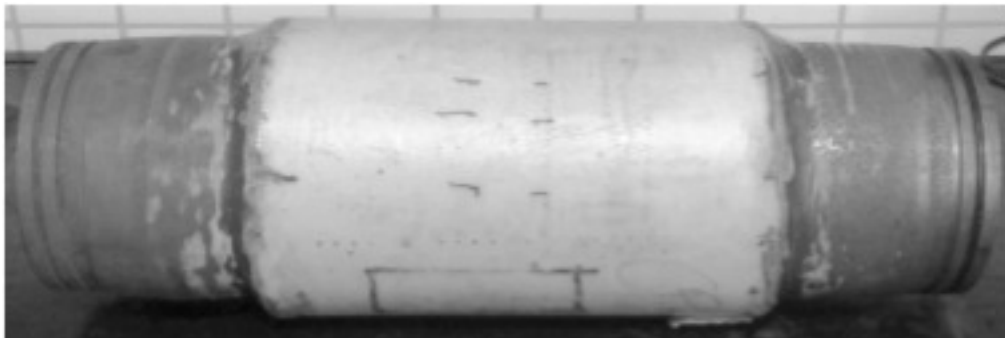


Figura 6.3: Corpo de prova. O comprimento do tubo é 250 mm.

As dimensões das regiões com defeito no corpo de prova estão descritas na Tabela 6.1. A Figura 6.5 mostra uma representação perimétrica em duas dimensões para melhor compreensão das dimensões do corpo de prova. A região com defeito interno está representada em cinza escuro, enquanto que a região com defeito externo aparece em branco. O restante do tubo corresponde à parte sem defeito e é representado em cinza claro. As áreas das regiões com defeito externo e com defeito interno somadas equivalem a menos de 10% da área total do corpo de prova. Informações mais detalhadas sobre

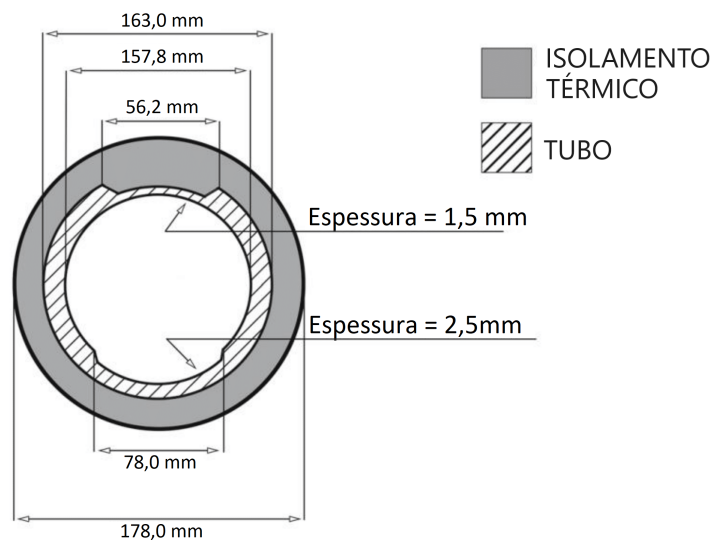


Figura 6.4: Representação em corte do corpo de prova.

as áreas das três regiões consideradas podem ser vistas na Tabela 6.2.

Tabela 6.1: Dimensões das regiões com defeito no corpo de prova.

Defeito	Comprimento (mm)	Largura (mm)	Espessura (mm)
Externo	86,8	56,2	1,5
Interno	96,0	78,0	2,5

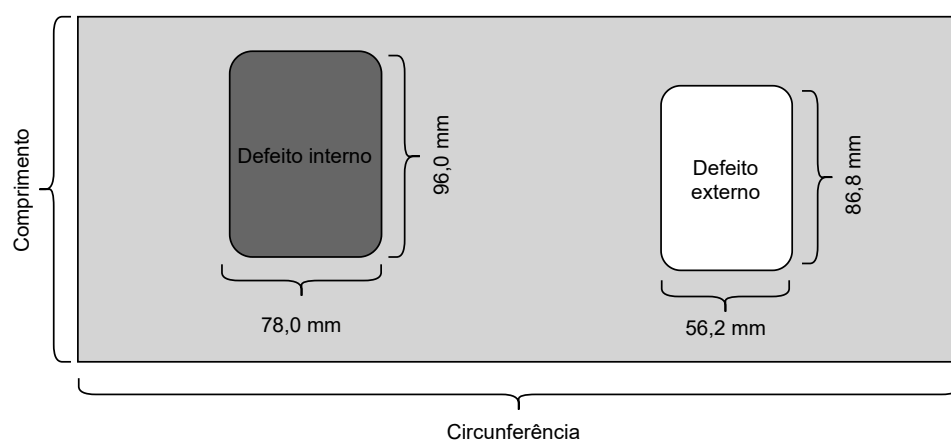


Figura 6.5: Representação perimétrica do corpo de prova.

Foi realizado um ensaio de PEC utilizando uma bobina com as seguintes características: diâmetro externo de 33 mm, diâmetro interno de 17 mm, 500

Tabela 6.2: Área da superfície do corpo de prova correspondente as regiões sem defeito, com defeito externo e com defeito interno.

Região	Área (mm^2)	Fração da área total (%)
Sem defeito	115589	90,3
Defeito externo	4878	3,8
Defeito interno	7488	5,9
Total	127955	100,0

espiras com fio de diâmetro 0,27 mm. Em conjunto com a bobina há dois sensores do tipo GMR (*Giant Magneto-Resistance*), sendo um dos sensores utilizado como referência e outro como medição, conforme pode ser visto na Figura 6.6. Devido a diferença de posição dos dois sensores GMR os sinais de saída fornecidos apresentam leve diferença. Esta diferença ocorre devido à interação dos campos magnéticos produzidos pela bobina e pelas correntes parasitas que geram um campo magnético resultante, cuja intensidade depende do ponto de medição considerado.

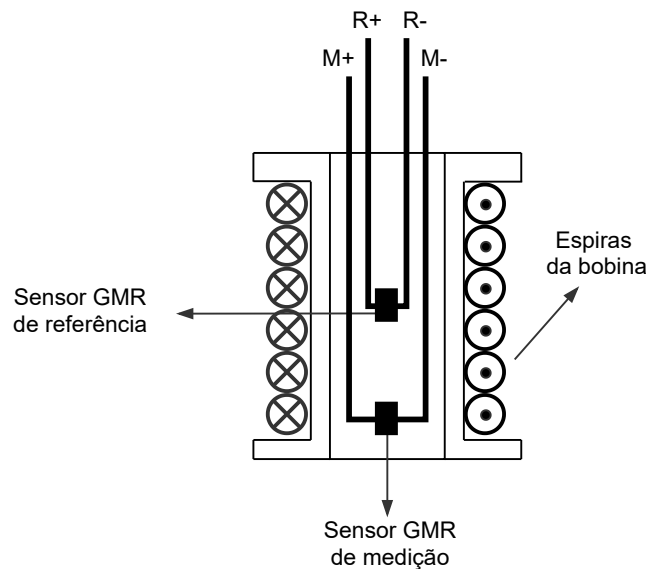


Figura 6.6: Diagrama com detalhe da bobina e sensores GMR.

A bobina foi excitada com uma onda quadrada de amplitude 4 V, frequência de 1 KHz, e *Duty Cycle* de 60%. Estes valores foram escolhidos para maximizar a eficiência dos dispositivos de medição. A bobina foi posicionada nas três regiões de interesse: Sem corrosão (SD - sem defeito), com corrosão

externa (DE - defeito externo) e com corrosão interna (DI - defeito interno). Para cada região, o ensaio foi repetido 100 vezes, realizando uma variação de posição em uma grade imaginária igualmente espaçada. Cada observação possui 200 pontos amostrados do sinal em função do tempo. O intervalo de amostragem utilizado foi 5×10^{-6} segundos. Foi utilizado um conversor analógico para digital (ADC - *Analog to Digital Converter*) de 16 bits embarcado no dispositivo de aquisição de dados NI myDAQ 23GF[®] em conjunto com o *software LabVIEW[®]*, ambos da empresa *National Instruments[®]* para leitura dos sinais dos ensaios. A Tabela 6.3 resume as informações sobre a obtenção dos sinais com ensaio PEC.

Tabela 6.3: Informações sobre o conjunto de sinais obtidos com ensaio PEC.

Dispositivo de aquisição de dados	NI myDAQ 23GF [®]
Resolução do ADC	16 bits
<i>Software</i> de aquisição de dados	<i>LabVIEW[®]</i>
Intervalo de amostragem	5×10^{-6} segundos
Nº de amostras de cada sinal	200
Nº de sinais da classe SD	100
Nº de sinais da classe DE	100
Nº de sinais da classe DI	100

6.2 Extração e seleção de Características

Para extrair informações quanto às frequências contidas no sinal foram utilizadas duas técnicas a DFT e a DWT. No caso da DFT foi utilizada, mais especificamente, a transformada rápida de Fourier (FFT - *Fast Fourier Transform*), por ser um algoritmo eficiente para calcular a DFT [52].

No caso da DWT, foram testadas diferentes funções *wavelet* mãe das classes Biorthogonal (Bior), Coiflets (Coif) e Daubechies (db). A DWT gera componentes de aproximação e de detalhe. Os componentes de aproximação foram escolhidos como entrada da rede neural. Pois, eles conservam grande parte do formato original do sinal. Enquanto que, os coeficientes de detalhes conservam informações de altas frequências mais associadas à ruídos [90]. Utilizou-se três níveis de níveis de filtragem para cada função *wavelet* mãe, a fim de comparar qual proporciona melhor performance de classificação.

A PCA foi aplicada aos sinais provenientes da DFT e da DWT. Foram testados diferentes números de componentes principais a fim de verificar qual proporciona melhores resultados. O número de componentes escolhidos correspondem aos valores de variância entre 80% e 99%.

6.3 Escolha da topologia da rede neural, função *wavelet* mãe e número de componentes principais

Esta seção define as estratégias utilizadas para encontrar a melhor topologia de rede de acordo com o tipo de sinal de entrada, a função *wavelet* mais adequada e o número de componentes principais que proporciona melhor desempenho de classificação. Neste trabalho, topologia é o formato da rede, ou seja, a dimensão de cada camada da rede neural. Diversas ANN foram treinadas utilizando o método *backpropagation* - Adam com os hiper-parâmetros listados na Tabela 6.4. Adam foi escolhido por ser um método atual e que mostrou-se superior a outras variantes do gradiente descendente [42].

Tabela 6.4: Hiper-parâmetros utilizados no algoritmo Adam para testar topologias de redes neurais.

Hiper-parâmetro	Valor adotado
Nº de épocas	2000
Taxa de aprendizagem	0,001
β_1	0,9
β_2	0,999
ϵ	10^{-8}

Para o número de épocas foi escolhido um número grande de forma a garantir a convergência do treinamento. A taxa de aprendizagem foi definida como um valor pequeno para evitar grandes variações de gradiente, o que prejudicaria a convergência do treinamento. Os dois parâmetros de *momentum* (β_1 e β_2) foram definidas conforme recomendado em [42]. O mesmo foi feito com o hiper-parâmetro ϵ , que é utilizado para evitar uma possível divisão por zero.

A acurácia foi a métrica utilizada para avaliar as ANN durante a escolha

da topologia de rede. Ela foi escolhida por ser uma métrica que avalia o desempenho global de classificação. Para, a partir dessa topologia que proporciona melhor classificação global, trabalhar na busca de uma ANN que satisfaça as restrições de treinamento descritas na seção 6.4. Caso fosse utilizada uma métrica que avalia a ANN apenas pelo desempenho de classificação de uma classe, o desempenho de classificação entre as demais classes seria prejudicado.

Para definir a topologia da rede, é importante primeiramente definir quais parâmetros influenciam no desempenho de classificação. A camada de saída possui três neurônios, cada um corresponde a uma classe de interesse. Entretanto, para a camada intermediária é preciso realizar testes para definir o melhor número de neurônios. A camada de entrada tem dimensão coincidente com a dimensão dos sinais de entrada. Sinais no domínio do tempo e sinais provenientes da DFT possuem dimensão fixa. Portanto, para redes treinadas com esses sinais o único parâmetro que precisa ser variado é o número de neurônios da camada intermediária, conforme Tabela 6.5.

Tabela 6.5: Parâmetros considerados na escolha da topologia da rede neural de acordo com o tipo de pré-processamento utilizado.

Pré-processamento	Parâmetros considerados
Domínio do tempo	Nº de neurônios da camada intermediária.
DFT	Nº de neurônios da camada intermediária.
DWT	Nº de neurônios da camada intermediária; Função wavelet mãe; Nível de filtragem.
DFT+PCA	Nº de neurônios da camada intermediária; Nº de componentes principais.
DWT+PCA	Nº de neurônios da camada intermediária; Função wavelet mãe; Nível de filtragem; Nº de componentes principais.

Para as redes treinadas com os demais tipos de pré-processamento, é preciso analisar outros parâmetros além do número de neurônios na camada intermediária. A dimensão de sinais provenientes da DWT depende da função *wavelet* utilizada e do nível de filtragem. Para sinais compactados pela

PCA, é necessário ainda considerar o número de componentes que devem ser utilizados na entrada da rede. A Tabela 6.5 resume os parâmetros que devem ser considerados na escolha da topologia de rede de acordo com o tipo de pré-processamento utilizado no sinal de entrada.

Para definir o número de neurônios das redes neurais treinadas com sinais no domínio do tempo ou sinais provenientes da DFT, foi feito um estudo variando o número de neurônios da camada intermediária. Foram testadas redes com 2 a 20 neurônios na camada intermediária. Para isso realizou-se um procedimento de validação cruzada de acordo com o método *hold-out* em que 50% dos dados foram utilizados para treino e 50% para teste. O procedimento foi executado 10 vezes, calculando-se em seguida a média das acurácias do conjunto de teste obtidas a cada execução da validação cruzada.

Para redes treinada com sinais pré-processados com DWT, é preciso analisar três parâmetros: o número de neurônios na camada intermediária, a função *wavelet* mãe e o nível de filtragem. Portanto, para escolher a combinação que proporciona maior acurácia média foi realizada uma busca em grade (GS - *Grid Search*). Esta técnica consiste em testar exaustivamente as diferentes combinações de parâmetros disponíveis e verificar qual proporciona melhor resultado [38].

Para redes treinada com sinais pré-processados com DFT+PCA, é preciso definir o número de neurônios na camada intermediária e o número de componentes principais extraídos da PCA. Então foi realizada uma busca em grade com estes dois parâmetros.

Considerando redes treinadas com sinais pré-processados com DWT+PCA, é necessário definir o número de neurônios na camada intermediária, a função *wavelet* mãe, o nível de filtragem e o número de componentes principais extraídos da PCA. Foi realizada uma busca em grade com o número de neurônios na camada intermediária e o número de componentes principais. A função *wavelet* mãe e o nível de filtragem foram escolhidos como sendo os mesmos que proporcionaram melhor desempenho para redes treinadas com sinais pré-processados com DWT. Esta escolha ajudou a evitar gasto desnecessário de tempo nesta etapa da pesquisa.

Com este procedimento, foi escolhida a melhor topologia de rede para cada caso de pré-processamento utilizado, a função *wavelet* mãe mais adequada e o melhor número de componentes principais. Em seguida foi realizado o treinamento de redes neurais com as topologias selecionadas utilizando o método proposto baseado em PSO com restrições. Também foi realizado o treinamento utilizando variações do gradiente descendente para posterior comparação do desempenho de classificação.

6.4 Treinamento da ANN utilizando PSO com restrições

As redes neurais utilizadas como classificadores neste trabalho são do tipo perceptron multi-camadas (MLP - *multi-layer-perceptron*) com uma única camada escondida. A dimensão da camada de entrada foi definida como igual a dimensão dos sinais, A camada de saída possui três neurônios, cada um representando uma classe de interesse. A função de ativação escolhida para os neurônios foi a tangente hiperbólica. A tangente hiperbólica foi escolhida por ser uma função contínua, apresentar derivada máxima igual a 1 para valores de entrada próximos de 0, valor superior ao da função sigmoid no mesmo ponto. Ter derivada maior acelera o processo de aprendizado da rede com gradiente descendente [39].

Para calcular as saídas da rede neural basta calcular a equação 6.2. Sendo que $\mathbf{X}$ é a matriz com os dados de entrada, cada coluna correspondente a um ensaio com PEC, $\mathbf{Y}^{(2)}$ é a matriz com as predições feitas pela rede neural, em que o sobrescrito (2) indica que se trata da saída $\mathbf{Y}$ da segunda camada da ANN. Cada coluna de $\mathbf{Y}^{(2)}$ corresponde às predições para um ensaio com PEC. A função de ativação dos neurônios é σ .

$$\mathbf{Y}^{(2)} = \sigma(\mathbf{W}^{(2)} \times (\sigma(\mathbf{W}^{(1)} \times \mathbf{X} + \mathbf{B}^{(1)})) + \mathbf{B}^{(2)}) \quad (6.2)$$

As saídas da rede neural são facilmente obtidas por meio da equação 6.2. Entretanto para movimentar a partícula (ANN) no espaço de busca ao executar o PSO é preciso agrupar os pesos sinápticos da rede em um único vetor.

O procedimento de conversão das matrizes que representam a ANN no vetor utilizado no PSO é mostrado no Algoritmo 1.

Algoritmo 1: Procedimento de conversão das matrizes $\mathbf{W}^{(1)}$, $\mathbf{B}^{(1)}$, $\mathbf{W}^{(2)}$ e $\mathbf{B}^{(2)}$ no vetor $\mathbf{P}$.

```

início
    Inicializa o vetor  $\mathbf{P}$  sem pesos;
    for Cada linha da matriz  $\mathbf{W}^{(1)}$  do
        | Concatene os pesos da linha da matriz  $\mathbf{W}^{(1)}$  no vetor  $\mathbf{P}$ ;
    end
    Transponha a matriz  $\mathbf{B}^{(1)}$ , gerando a matriz  $\mathbf{B}^{(1)\mathbf{T}}$ ;
    Concatene os pesos da matriz  $\mathbf{B}^{(1)\mathbf{T}}$  no vetor  $\mathbf{P}$ ;
    for Cada linha da matriz  $\mathbf{W}^{(2)}$  do
        | Concatene os pesos da linha da matriz  $\mathbf{W}^{(2)}$  no vetor  $\mathbf{P}$ ;
    end
    Transponha a matriz  $\mathbf{B}^{(2)}$ , gerando a matriz  $\mathbf{B}^{(2)\mathbf{T}}$ ;
    Concatene os pesos da matriz  $\mathbf{B}^{(2)\mathbf{T}}$  no vetor  $\mathbf{P}$ ;
fim

```

No início da execução do PSO, cada partícula é inicializada. Neste trabalho foi utilizado o método de Nguyen-Widrow para inicializar os pesos sinápticos e bias das redes que representam a posição da partícula. As matrizes de velocidade foram inicializadas com zeros. O enxame foi inicializado com os parâmetros da Tabela 6.6.

Tabela 6.6: Parâmetros utilizado no PSO.

Topologia do enxame	Global
Nº de partículas	200
Constante cognitiva	2
Constante social	1
Limites de inércia (w)	[0,4; 0,9]
Limites de posição	[-1,5; 1,5]
Limites de velocidade	[-0,18; 0,18]
Número de iterações	100

A topologia de enxame global foi escolhida devido à sua simplicidade de implementação. Trata-se de um tipo de PSO em que todas as partículas interagem entre si. Alguns parâmetros do enxame foram definidos baseados no trabalho [3], que utilizou PSO para treinar redes neurais para diagnóstico médico. Estes parâmetros são: número de partículas, constantes cognitiva

e social e limites de inércia. O valor da constante cognitiva é maior que a constante social, evitando a convergência prematura, conforme descrito em [62]. A constante cognitiva foi definida como o dobro da constante social para garantir uma boa pesquisa global no espaço de busca. A inércia (w) é iniciada em 0,9 e decai linearmente até 0,4 no final da execução do PSO. A diminuição do valor de w no fim da execução do PSO evita que ocorra grande oscilação em torno do ponto que proporciona melhor valor da função objetivo.

Os limites de posição foram definidos entre -1,5 e 1,5 para evitar que os pesos sinápticos assumam valores demasiadamente grandes em módulo, dificultando a convergência do treinamento. As velocidades das partículas foram limitadas entre -0,18 e 0,18 para evitar que se movam para longe de uma possível solução ótima. Estes limites de posição e velocidade foram baseados em recomendações propostas em [77], [91]. O número máximo de iterações foi definido igual a 100, por ser grande o suficiente para garantir a convergência do treinamento da rede sem perda desnecessária de tempo e processamento.

6.4.1 Funções objetivo e restrições

Várias funções objetivo foram testadas a fim de selecionar as mais adequadas ao problema em questão. Duas se destacaram pelo resultados obtidos. A primeira função, chamada $f_1(\mathbf{P})$, avalia a partícula pelo somatório das taxas de falso alarme das três classes do problema (equação 6.3). A segunda função, chamada $f_2(\mathbf{P})$, avalia a partícula pelo produto das eficiências (equação 6.4).

$$f_1(\mathbf{P}) = \sum_{j=1}^3 PF_j(y) \quad (6.3)$$

$$f_2(\mathbf{P}) = PE \quad (6.4)$$

Conforme discutido na seção 4.2.4, neste trabalho é considerado que os erros de classificação possuem consequências distintas para a aplicação. Mais especificamente, classificar um tubo defeituoso como não defeituoso é uma

situação altamente indesejada. Enquanto que, classificar um tubo não defeituoso como defeituoso, apesar de indesejado, é mais tolerável. Portanto, as restrições aplicadas focam especificamente na taxa de detecção da classe sem defeito TD_{SD} e de falso alarme TF_{SD} .

Minimização das taxas de falso alarme

A função $f_1(\mathbf{P})$ avalia as partículas pelo somatório das taxas de falso alarme (PF) das três classes, considerando quanto menor o valor, melhor a classificação. Portanto, trata-se de um problema de minimização. A restrição adotada para este caso foi um valor mínimo para a taxa de detecção da classe SD ($TD_{SD} \geq 90\%$). Este problema de otimização é representado pela equação 6.5.

$$\min f_1(\mathbf{P}), \quad \text{restrito a } TD_{SD} \geq 90\%. \quad (6.5)$$

Inicialmente as saídas da ANN são calculadas. Em seguida monta-se a matriz de confusão, que é necessária para calcular as taxas de detecção e falso alarme para cada classe. A partir destes valores, a função objetivo é calculada e a restrição verificada de acordo com as equações 6.3 e 6.5, respectivamente. Caso a partícula não satisfaça a restrição, uma penalidade é aplicada. Por se tratar de um problema de minimização, a penalidade é aplicada de forma a aumentar o valor da função objetivo da partícula. Os passos de execução desta função para cada partícula são descritos no Algoritmo 2.

Algoritmo 2: Função objetivo $f_1(\mathbf{P})$.

início

 Calcular as saídas ($\mathbf{Y}^{(2)}$) da ANN;

 Criar a matriz de confusão;

 Calcular a função objetivo da partícula (ANN) de acordo com a equação 6.3;

 Verificar se a ANN satisfaz as restrições com a equação 6.5;

if *partícula (ANN) é ineficaz* **then**

 Aplicar penalidade que aumente o valor da função objetivo da partícula (somar 10^8 ao valor da função objetivo);

end

fim

Maximização do produto das eficiências

A função $f_2(\mathbf{P})$ avalia a partícula pelo produto das eficiências (PE). Portanto, para este caso é desejável maximizar a função objetivo. A restrição adotada com a função $f_2(\mathbf{P})$ é um valor máximo λ de taxa de falso alarme da classe SD (TF_{SD}). Sendo que foram testados dois valores de λ 0,5% e 1%. Este problema de otimização é representado pela equação 6.6.

$$\max f_2(\mathbf{P}), \quad \text{restrito a } TF_{SD} \leq \lambda. \quad (6.6)$$

Inicialmente as saídas da ANN são calculadas. Em seguida monta-se as matrizes de confusão e de confusão normalizada, que são necessárias para calcular as taxas de detecção para cada classe e o produto das eficiências. A partir destes valores, a função objetivo é calculada de acordo com a equação 6.4 e a restrição é verificada de acordo com a equação 6.6. Caso a partícula não satisfaça a restrição, uma penalidade é aplicada. Por se tratar de um problema de maximização, a penalidade é aplicada de forma a diminuir o valor da função objetivo da partícula. Os passos de execução desta função para cada partícula são descritos no Algoritmo 3.

Algoritmo 3: Função objetivo $f_2(\mathbf{P})$.

início

 Calcular as saídas ($\mathbf{Y}^{(2)}$) da ANN;

 Criar a matriz de confusão;

 Criar a matriz de confusão normalizada;

 Calcular a função objetivo da partícula (ANN) de acordo com a equação 6.4;

 Verificar se a ANN satisfaz as restrições com a equação 6.6;

if *partícula (ANN) é infactível* **then**

 Aplicar penalidade que diminua o valor da função objetivo da partícula (subtrair $\sum_{j=1}^3 PF_j(y)$ do valor da função objetivo);

end

fim

6.5 Validação Cruzada e *Bootstrap*

Validação cruzada é um método utilizado para estimar a capacidade de generalização e o verdadeiro erro de predição de um classificador neural. O conjunto de dados disponível deve ser dividido em dois subconjuntos, um conjunto de treino e um de teste. Com o conjunto de treino o classificador neural é treinado. Entretanto, o erro de classificação do conjunto de treino é subestimado, uma vez que o classificador já conhece os dados apresentados. Portanto, para avaliar o classificador deve ser utilizado o conjunto de teste [92].

Dentre as diferentes formas de validação cruzada, a mais conhecida e utilizada é *k-fold*. Nesta forma de validação cruzada, o conjunto de dados é dividido em subconjuntos denominados *folds*. O número de *folds*, em geral, varia entre 5 e 10. Utiliza-se $k-1$ *folds* durante o treinamento do classificador e o *fold* restante para teste. O processo de treinamento se repete alterando o *fold* de teste até que todos *folds* tenham sido utilizados como conjunto de teste [92]. A Figura 6.7 ilustra o procedimento de validação cruzada *k-fold*.

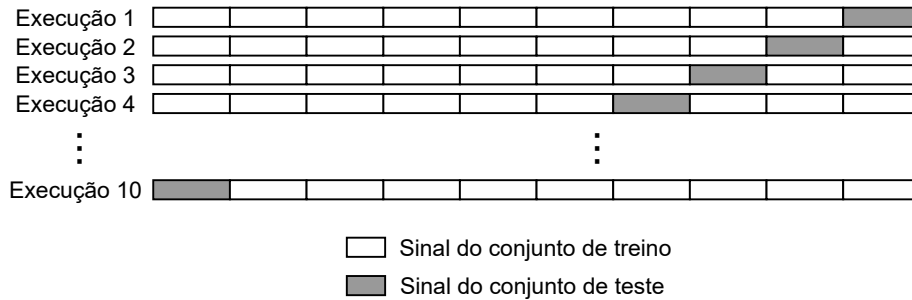


Figura 6.7: Validação cruzada *K-Fold*.

Frequentemente, os conjuntos de treino e teste são divididos nas proporções 70% e 30%, respectivamente, ou proporções similares [93]. Como este trabalho propõe um novo método de treinamento de classificadores neurais, é importante que o conjunto de teste seja grande o suficiente para permitir a comparação entre o método proposto e os métodos tradicionais. Por outro lado, para treinar um classificador neural de maneira satisfatória é preciso ter informação suficiente no conjunto de treino. Deve-se ainda considerar o tamanho do conjunto de dados disponível, que neste trabalho possui 100 si-

nais para cada classe de interesse. A partir destas considerações, foi definido utilizar a proporção 50% para treino, 50% para teste e não utilizar conjunto de validação, pois diminuiria ainda mais os conjuntos de treino e de teste.

Como foi definido utilizar a proporção 50% do conjunto de dados para treino e 50% para teste, o *k-fold* é executado apenas duas vezes. Portanto, como alternativa ao *k-fold*, optou-se pela utilização de *Bootstrap*. Este método divide o conjunto de dados de forma aleatória em treino e teste. O processo pode ser usado repetidas vezes amostrando diferentes sinais para os conjuntos de treino e teste de forma independente da execução anterior [94]. A Figura 6.8 ilustra o funcionamento do método *Bootstrap*.

Na Figura 6.8, cada linha representa uma execução do *Bootstrap*, que neste trabalho foi executado 10 vezes. Os retângulos brancos representam sinais utilizados no conjunto de treino, enquanto que os retângulos cinzas representam sinais amostrados para o conjunto de teste. Para cada seleção realizada pelo *Bootstrap*, o classificador foi treinado 10 vezes. Portanto, cada modelo de classificador foi treinado 100 vezes.

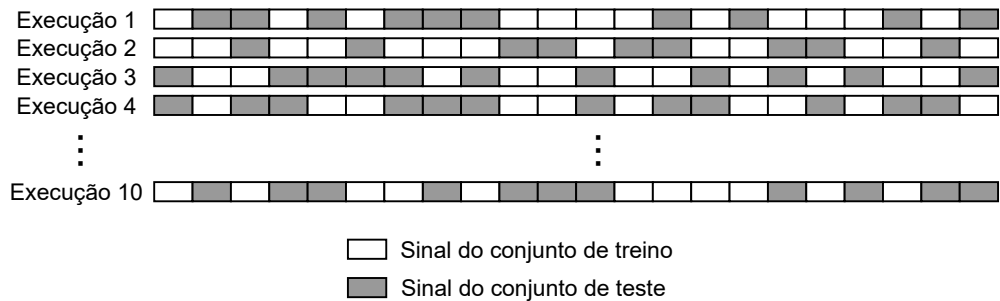


Figura 6.8: Princípio de funcionamento do *Bootstrap*.

Capítulo 7

Resultados

Este capítulo inicia-se com uma análise dos sinais no domínio do tempo obtidos nos ensaios com PEC. São feitas considerações sobre os componentes obtidos com a DFT e com a DWT. A seção sobre PCA descreve as diferenças entre os componentes obtidos com DFT+PCA e com DWT+PCA. Em seguida são apresentados os resultados da busca da melhor topologia de rede de acordo com o tipo de sinal de entrada, da função *wavelet* mãe mais adequada e do número de componentes principais que proporciona melhor desempenho de classificação. Na seção seguinte, os resultados obtidos com o método de treinamento proposto são apresentados. Por fim é feita uma análise comparativa dos resultados obtidos.

7.1 Análise dos sinais típicos para as classes de interesse

A amostragem dos sinais foi realizada a cada $5\ \mu s$, durante $1\ ms$, totalizando 200 amostras. Este intervalo de $1\ ms$ está relacionado com a frequência da onda que excita a bobina (1 kHz). Caso o sinal fosse amostrado por mais de $1\ ms$, haveria informação repetida devido à periodicidade da onda. Por outro lado, caso a amostragem durasse menos de $1\ ms$ poderia haver perda de informação útil.

Para visualização dos sinais obtidos dos ensaios com PEC foram criados gráficos de amplitude em função do tempo dos sensores de referência e me-

dição. A Figura 7.1 mostra exemplos destes sinais para as três classes de interesse deste trabalho. Os sensores fornecem sinais de tensão semelhantes em todos os casos. Entretanto, gráficos relativamente distintos são obtidos ao analisar-se a subtração entre o sinal dos sensores de leitura e referência, conforme pode ser visto na Figura 7.2. Enquanto os sinais dos sensores tem amplitude da ordem de volt (V), a diferença entre os sinais é da ordem de dezenas de milivolt (mV).

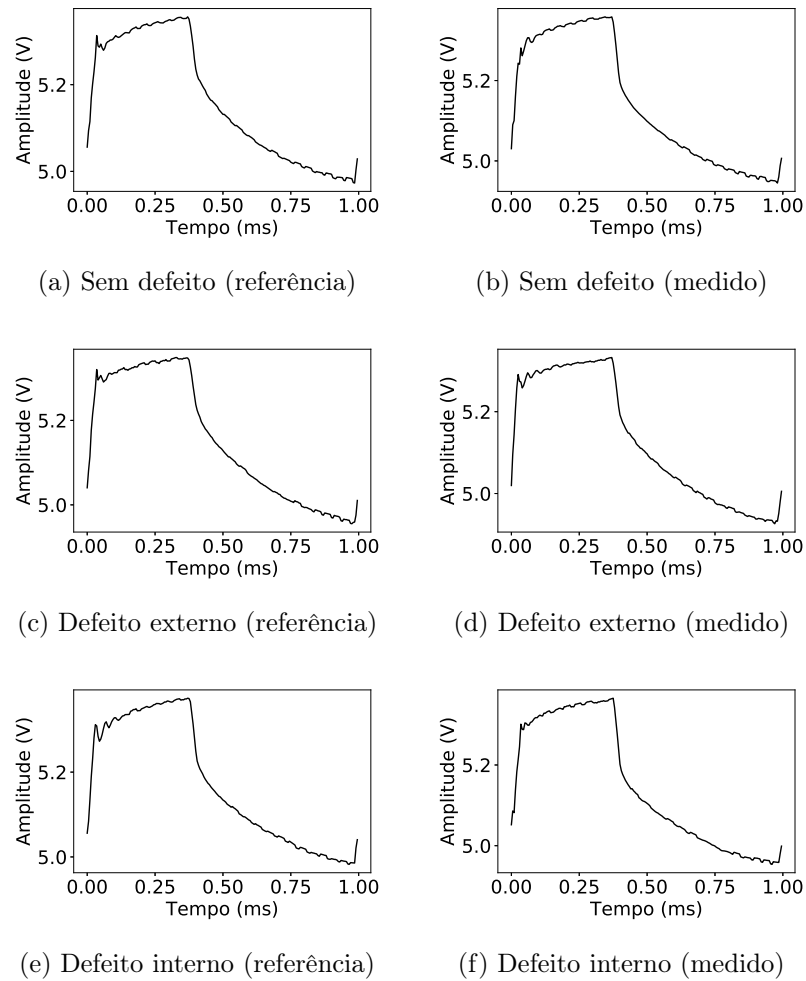


Figura 7.1: Sinais típicos dos ensaios com PEC no domínio do tempo. Fonte: Autor.

A transformada discreta de Fourier teve como resultado um sinal de 200 componentes. Foi obtido este número de componentes devido à utilização da transformada rápida de Fourier como forma de calcular a DFT, que tem a

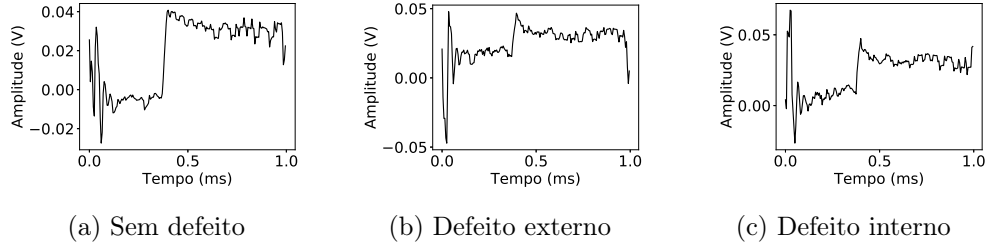


Figura 7.2: Sinais típicos dos ensaios com PEC no domínio do tempo, obtidos da subtração entre os sinais de referência e medido. Fonte: Autor.

característica de fornecer número de componentes no domínio da frequência igual ao número de componentes do sinal no domínio do tempo utilizado como entrada [52]. Entretanto, o sinal é simétrico em relação ao eixo vertical. Esta simetria ocorre devido à ausência de parte imaginária no sinal no domínio do tempo, ou seja, o sinal é puramente real. Devido à simetria da DFT, apenas metade das componentes foram consideradas na segunda etapa de processamento, reduzindo a dimensão do sinal para 100 pontos. Foi considerada apenas a informação do espectro de amplitude, por conter a maior parte da informação relevante. A Figura 7.3 mostra as 100 componentes consideradas de um sinal.

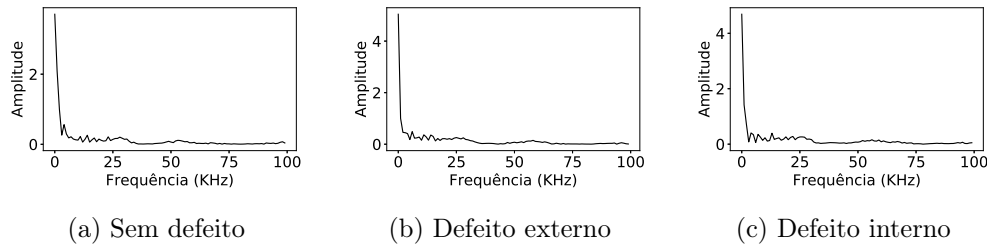


Figura 7.3: Espectros de amplitude das transformadas de Fourier típicas dos sinais de PEC. Fonte: Autor.

Considerando a DWT, a função *wavelet* mãe foi escolhida através de uma busca em grade descrita na seção 6.3. Os resultados desta busca estão detalhados na seção 7.3, que expõe as razões pelas quais a função *wavelet* mãe Daubechies 4 com três níveis de filtragem (db4_3) foi escolhida para este trabalho. A Figura 7.4 mostra gráficos dos 31 coeficientes de aproximação do terceiro nível de filtragem obtidos com esta função *wavelet* mãe.

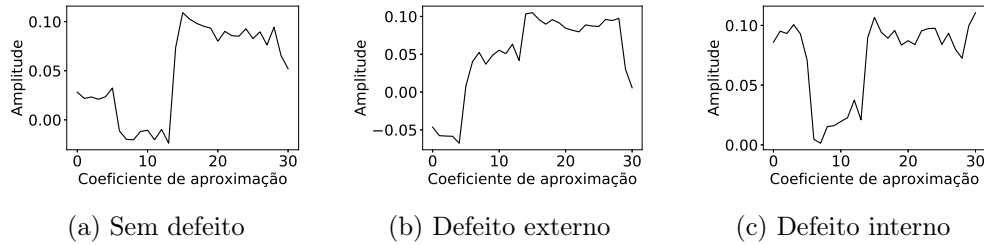


Figura 7.4: Típicos coeficientes de aproximação da DWT aplicada aos sinais de ensaios com PEC. Foi utilizada o terceiro nível de filtragem da DWT com função *wavelet* mãe db4. Fonte: Autor.

7.2 Análise de componentes principais

A PCA calculada a partir da DFT utilizou todos os componentes obtidos para a parte positiva do espectro de de módulo (100 componentes). Após o cálculo da PCA foi feito o gráfico de variância acumulada (carga). A Figura 7.5a mostra o gráfico de variância acumulada, em que cada ponto é a soma dos autovalores de ordem menor ou igual. Para calcular a PCA a partir da DWT, utilizou-se os coeficientes de aproximação de terceiro nível (31 coeficientes) obtidos com a função *wavelet* mãe db4. Neste caso, o gráfico da variância acumulada é mostrado na Figura 7.5b.

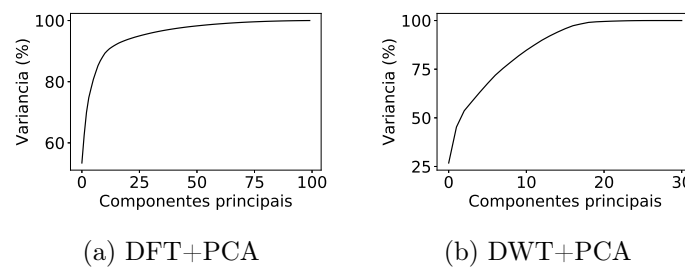


Figura 7.5: Curva de carga para a PCA calculada a partir da DFT (a) e DWT (b). Fonte: Autor.

Comparando os dois gráficos da Figura 7.5, percebe-se que a curva de carga da PCA aplicada à DFT é mais inclinada próximo à origem do sistema de coordenadas. Para chegar a 90% de variância a PCA aplicada a DFT usou 12 dos 100 componentes, correspondendo a 12% do número total de componentes. Para a PCA aplicada a DWT, utilizou-se 14 dos 31 componentes,

ou seja, 45% do número total de componentes. Portanto, a PCA aplicada à DFT é mais eficiente na compactação se comparada à PCA aplicada à DWT.

Um dos métodos utilizados para selecionar o número de componentes principais para análise é a porcentagem acumulada da variância total, que seleciona determinado número de componentes, tal que sua soma represente uma porcentagem desejada da variância total [7]. Para este trabalho foram testados diferentes valores de porcentagem de variância, sendo que os melhores resultados de classificação foram obtidos para variância acumulada de 90%. Este valor equivale a 12 componentes principais para os sinais pré-processados com DFT+PCA e 14 componentes para sinais pré-processados com DWT+PCA. A porcentagem de variância acumulada em função do número de componentes pode ser vista na Tabela 7.1. Nesta tabela são mostrados apenas os números de componentes principais que foram utilizados como entradas das redes neurais.

Tabela 7.1: Variância acumulada em função do número de componentes principais utilizados neste trabalho.

Variância acumulada (%)	Nº de componentes	
	DFT+PCA	DWT+PCA
80	6	10
83	7	11
85	8	12
87	10	-
89	11	13
90	12	14
92	16	-
94	22	15
96	32	17
98	48	18
99	63	19

A Figura 7.6 mostra *boxplots* com o valor do primeiro componente principal para as diferentes classes de interesse. Para a PCA calculada a partir da DFT, Figura 7.6a, os valores do primeiro componente principal apresentam baixa distinção entre classes. Enquanto que, para a PCA calculada a partir da DWT, Figura 7.6b, os valores do primeiro componente principal apresentam maior distinção entre as classe, facilitando a classificação. Isso indica

que a combinação de DWT e PCA parece produzir boa distinção entre as classes de interesse.

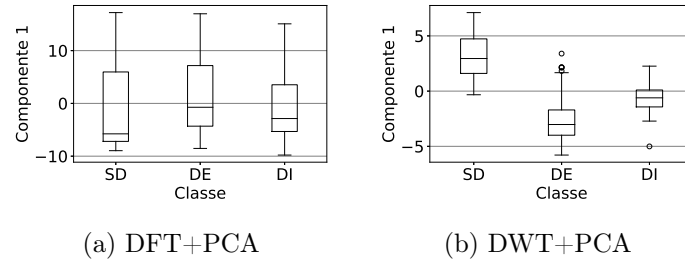


Figura 7.6: *Boxplot* do valor do primeiro componente principal calculado a partir da DFT (a) e DWT (b). Fonte: Autor.

A Figura 7.7 estende a análise feita anteriormente para um número maior de componentes principais calculados a partir da DFT. Mesmo considerando-se maior número de componentes, neste caso 7, não há distinção clara entre classes conforme observado com os componentes calculados a partir da DWT.

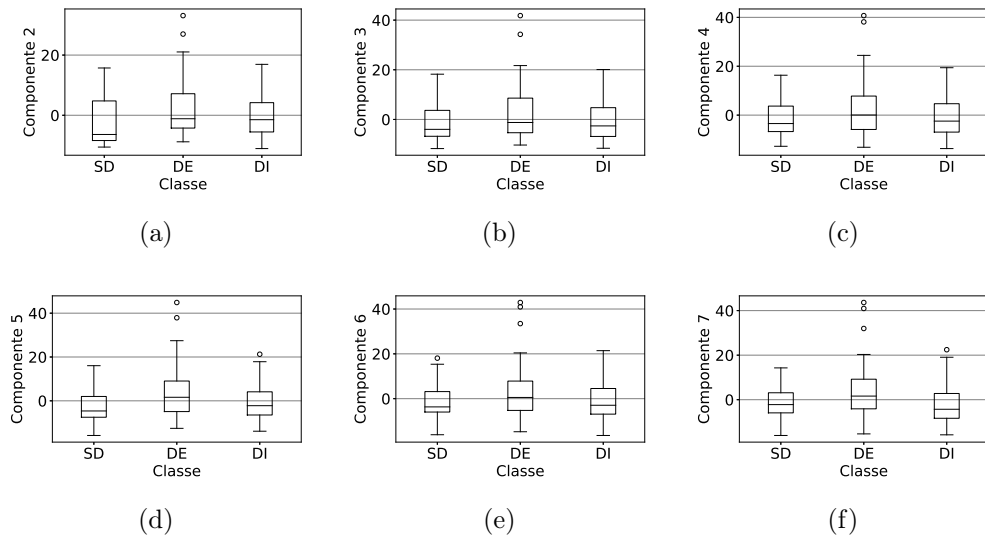


Figura 7.7: *Boxplots* dos valores dos 7 primeiros componentes principais calculado a partir da DFT. Fonte: Autor.

7.3 Determinação da topologia das ANN

Conforme apresentado na seção 6.3, para determinar a melhor topologia das redes neurais foi necessário realizar uma busca variando-se alguns parâmetros. A quantidade de parâmetros considerados depende do tipo de pré-processamento utilizado nos sinais de ensaios com PEC. Cada topologia de rede foi avaliada de acordo com sua acurácia média, por ser uma métrica que representa o desempenho global de classificação. A partir da topologia escolhida, as redes foram treinadas com imposição de restrições utilizando PSO.

Tanto as ANN treinadas com sinais no domínio do tempo quanto as ANN treinadas com sinais pré-processados com DFT forneceram maior acurácia média com a camada intermediária composta por dez neurônios, como pode ser visto na Figura 7.8. Note que, para ANN treinadas com sinais no domínio do tempo 10, 14 e 16 neurônios na camada intermediária forneceram o mesmo valor de acurácia média. Neste caso, a rede com menor número de parâmetros, consequentemente menor número de neurônios, foi escolhida. Tendo como pré-suposto que menor número de parâmetros diminui o esforço computacional durante o treinamento da ANN e também atende ao princípio da parcimônia, evitando problemas de *overtraining*. Vale ressaltar que na Figura 7.8 é apresentado um gráfico de médias (representadas pelos marcadores centrais) e desvios padrões (representados pelas barras).

Os resultados da busca em grade realizada para sinais pré-processados com DWT foram divididos de acordo com a função *wavelet* mãe para melhor visualização. Foram testadas as funções da classe Biorthogonal (Figura 7.9), Coiflet (Figura 7.10) e Daubechies (Figura 7.11). Os gráficos nas Figuras 7.9, 7.10 e 7.11 representam a acurácia média em função do número de neurônios na camada intermediária, da função *wavelet* mãe e do nível de filtragem. No eixo vertical as funções wavelet estão representadas com a seguinte nomenclatura: `func_n`, em que `func` é o nome da função wavelet mãe avaliada e `n` é o nível utilizado na filtragem, podendo assumir os valores 1, 2 e 3. A título de exemplo a nomenclatura `bior1.3_1` refere-se à função `bior1.3` com o nível de filtragem 1.

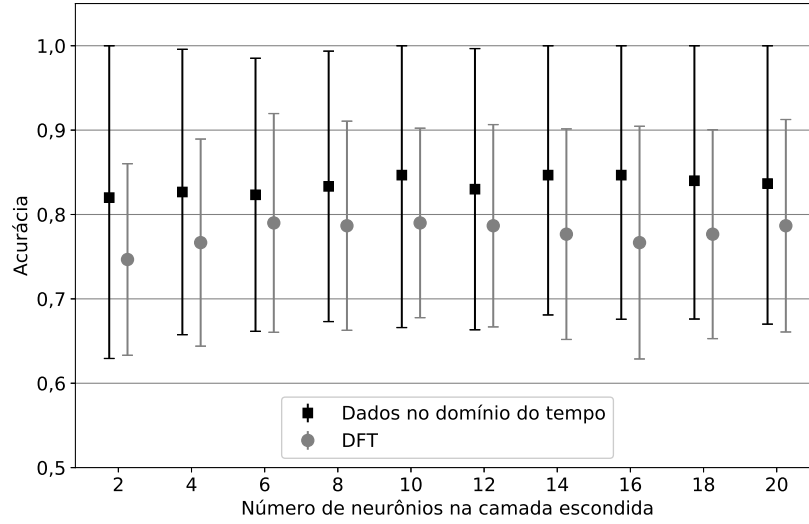


Figura 7.8: Gráfico da média da acurácia e desvio padrão em função do número de neurônios na camada intermediária para ANN treinadas com sinais no domínio do tempo e pré-processados com DFT .

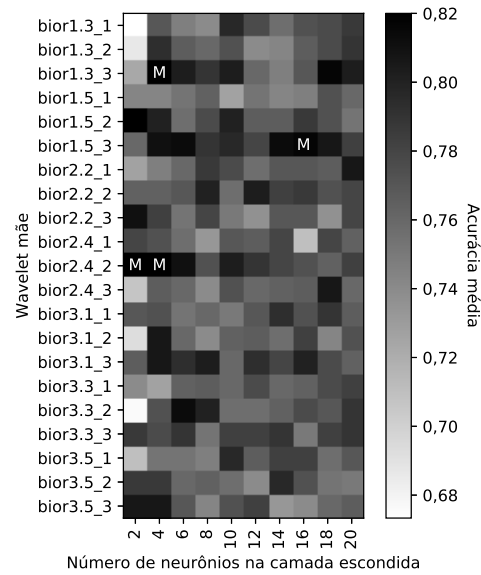


Figura 7.9: Acurácia calculada para diferentes valores da função wavelet mãe e do número de neurônios na camada intermediária para ANN treinadas com sinais pré-processados com DWT utilizando funções da classe bior.

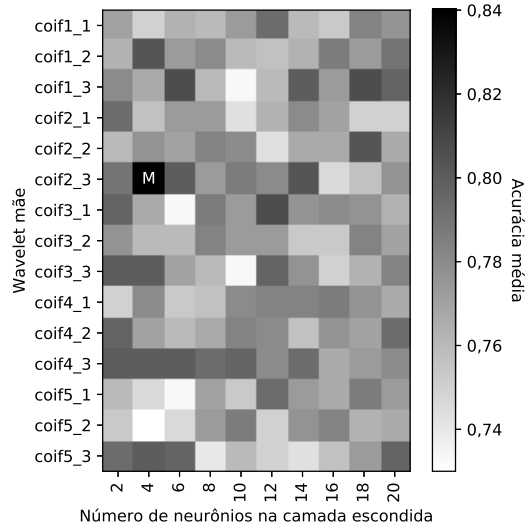


Figura 7.10: Acurácia calculada para diferentes valores da função wavelet mãe e do número de neurônios na camada intermediária para ANN treinadas com sinais pré-processados com DWT utilizando funções da classe coiflet.

Os melhores resultados foram obtidos com as funções Daubechies. No gráfico da Figura 7.11 aparecem quatro valores máximos demarcados com a letra “M”. A ANN treinada com (db4) com três níveis de filtragem (db4_3) e 10 neurônios na camada intermediária possui menor número de parâmetros comparada com as demais que apresentaram mesmo valor de acurácia média. Portanto, optou-se por utilizar os coeficientes de aproximação do terceiro nível obtidos com a função Daubechies 4 (db4_3).

A busca em grade realizada com sinais pré-processados por DFT+PCA proporcionou cinco combinações de variância acumulada e número de neurônios na camada intermediária com mesmo valor máximo de acurácia média = 0,79. A Figura 7.12 Mostra um gráfico em duas dimensões da acurácia em função da porcentagem de variância acumulada (DFT+PCA) e do número de neurônios na camada intermediária da rede neural. Tons de cinza mais escuros denotam maiores valores de acurácia. Os valores máximos contém a marcação “M” para facilitar a visualização. O número de pesos da rede é calculado somando as quantidades de elementos das matrizes $\mathbf{W}^{(1)}$, $\mathbf{B}^{(1)}$, $\mathbf{W}^{(2)}$ e $\mathbf{B}^{(2)}$. Dentre as cinco combinações com mesmo valor máximo de acu-

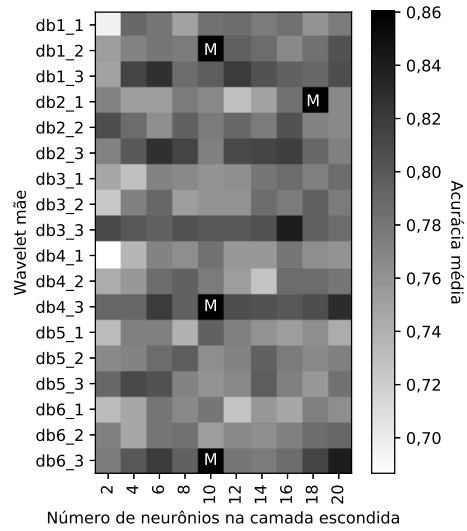


Figura 7.11: Acurácia calculada para diferentes valores da função wavelet mãe e do número de neurônios na camada intermediária para ANN treinadas com sinais pré-processados com DWT utilizando funções da classe Daubechies. Os valores máximos contêm a marcação “M” para facilitar a visualização.

rácia, a rede com menor número de parâmetros é treinada utilizando como entradas os componentes principais que representam 90% de variância e tem 10 neurônios na camada intermediária, apresentando 163 pesos. Portanto, este modelo foi utilizado no treinamento da rede neural.

Desempenho similar foi obtido ao realizar a busca em grade para redes treinadas com sinais pré-processados por DWT+PCA. Neste caso seis combinações apresentaram mesmo valor máximo, conforme pode ser visto na Figura 7.13. A rede com menor número de parâmetros continuou sendo aquela treinada com 90% de variância e 10 neurônios na camada intermediária. Porém, neste caso a rede neural possui 183 pesos. A Tabela 7.2 resume as estruturas de redes utilizadas para cada tipo de sinal de entrada.

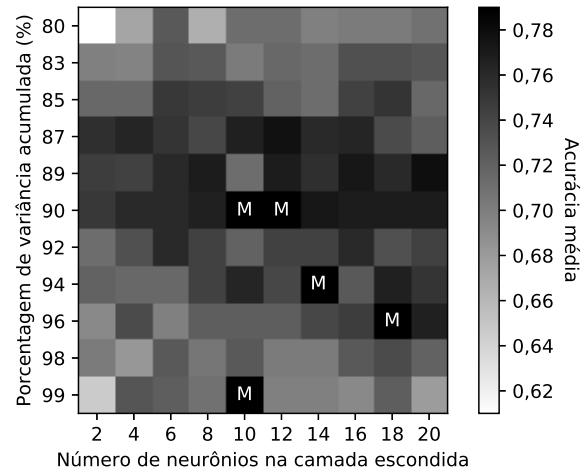


Figura 7.12: Gráfico de acurácia em função da variância acumulada e do número de neurônios na camada intermediária para ANN treinadas com sinais pré-processados com DFT+PCA. Os valores máximos contêm a marcação “M” para facilitar a visualização.

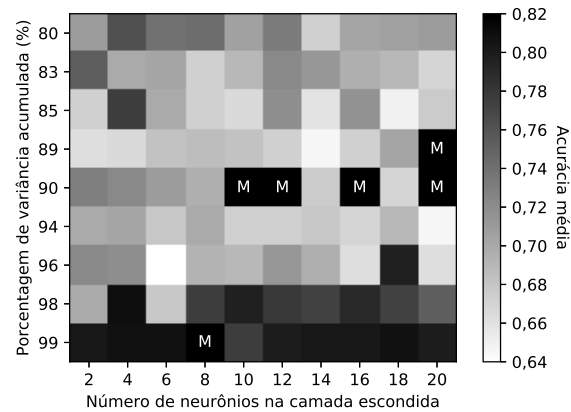


Figura 7.13: Gráfico de acurácia em função da variância acumulada e do número de neurônios na camada intermediária para ANN treinadas com sinais pré-processados com DWT+PCA. Os valores máximos contêm a marcação “M” para facilitar a visualização.

Tabela 7.2: Estruturas de ANN utilizadas. Todas ANN são do tipo MLP com função de ativação tangente hiperbólica.

Pré-processamento	Dimensão da camada			Nº pesos
	de entrada	intermediária	de saída	
Domínio do tempo	200	10	3	2043
DWT	31	10	3	353
DFT	100	10	3	1043
DWT+PCA	14	10	3	183
DFT+PCA	12	10	3	163

7.4 Resultados dos treinamentos das ANN com o método proposto

Para cada sequencia de pré-processamento treinou-se ANN com diferentes variações do gradiente descendente e com o método proposto. As topologias de redes foram descritas na seção 7.3 e resumidas na Tabela 7.2. A dimensão do espaço de busca do PSO é igual ao número de pesos sinápticos contidos na ANN. A Tabela 7.3 indica a dimensão do espaço de busca e seus limites de acordo com o tipo de pré-processamento aplicado aos sinais de entrada.

Tabela 7.3: Características do espaço de busca do algoritmo PSO considerando cada tipo de pré-processamento do sinal de entrada.

Pré-processamento	Dimensão do espaço de busca (Nº de pesos)	Limites do espaço de busca	Função de ativação
Domínio do tempo	2043	$[-1,5; 1,5]$	$\tanh(v)$
DWT	353	$[-1,5; 1,5]$	$\tanh(v)$
DFT	1043	$[-1,5; 1,5]$	$\tanh(v)$
DWT+PCA	183	$[-1,5; 1,5]$	$\tanh(v)$
DFT+PCA	163	$[-1,5; 1,5]$	$\tanh(v)$

As siglas utilizadas para redes treinadas com gradiente descendente são representadas pela sigla ANN-BP (BP denota *backpropagation*, visto que este é o algoritmo utilizado para calcular o gradiente) seguidas de SGD (para *Stochastic Gradient Descent*), Adam (para *Adaptive Moment Estimation*), LM (para *Levenberg-Marquardt*) ou Rprop (para *Resilient backpropagation*). As siglas utilizadas para redes treinadas pelo método proposto são representadas

por ANN-PSO seguidas de $f_1(\mathbf{P})$ em casos onde se utilizou a função objetivo $f_1(\mathbf{P})$, ou $f_2(\mathbf{P})$ para casos onde se utilizou a função objetivo $f_2(\mathbf{P})$. Sendo que para a função objetivo $f_2(\mathbf{P})$ foram considerados dois casos distintos. No primeiro caso, foi considerada a restrição de $TF_{SD} \leq 0,5\%$, no segundo, foi considerada a restrição de $TF_{SD} \leq 1\%$.

Portanto, foram utilizados quatro formas diferentes de gradiente descendente e duas funções objetivo para o método proposto com PSO. Após o treinamento, foi feito o *boxplot* com o PE obtido com o conjunto de teste de todas as ANN treinadas, totalizando 100 ANN para cada método de treinamento. Em seguida selecionou-se a rede com maior PE de cada método de treinamento considerando-se o desempenho obtido com o conjunto de teste. O desempenho de cada rede está sintetizado em uma matriz de confusão normalizada, com seus respectivos valores de PE (calculado com a equação 4.8, a partir da matriz de confusão normalizada), taxa de falso alarme e de detecção da classe SD (calculados com as equações 4.6 e 4.5, respectivamente, a partir da matriz de confusão não normalizada).

7.4.1 ANN treinadas com informação no domínio do tempo

A Figura 7.14 mostra o *boxplot* do PE para ANN treinadas por diferentes métodos com sinais sem pré-processamento (informação no domínio do tempo). Pode-se observar que o método PSO $f_1(\mathbf{P})$ proporcionou o maior valor de PE. Entretanto, em algumas inicializações o algoritmo diverge, resultando em baixa eficiência. Entre os métodos gradiente descendente, o SGD proporcionou os menores valores de PE, enquanto que Adam proporcionou maiores valores de PE, entretanto, sem superar o valor máximo obtido com o método PSO $f_1(\mathbf{P})$. As redes treinadas com ANN-PSO $f_2(\mathbf{P})$ proporcionaram valores menores de PE devido às restrições de valor máximo de TF_{SD} .

As Tabelas 7.4 e 7.5 resumizam o melhor resultado (maior PE) obtido com cada método de treinamento. A Tabela 7.4 mostra as matrizes de confusão normalizadas. Pode-se observar que a rede ANN-PSO $f_1(\mathbf{P})$ obteve maior

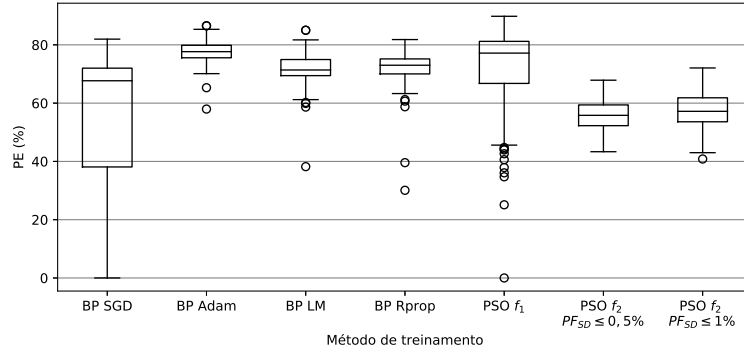


Figura 7.14: *Boxplot* do produto das eficiências para ANN treinada por diferentes métodos com sinais sem pré-processamento (informação no domínio do tempo).

PE, seguida da rede ANN-BP Adam. Para o gradiente descendente, os maiores valores de PE foram obtidos por ANN-BP Adam e ANN-BP LM. Apesar de proporcionar menor PE, as ANN treinadas com ANN-PSO $f_2(\mathbf{P})$ apresentaram o menor erro de classificação de sinais com defeito sendo classificados como sem defeito.

Da Tabela 7.5 observa-se que o maior valor da taxa de detecção da classe SD (TD_{SD}) foi obtido com a rede ANN-PSO $f_1(\mathbf{P})$ (98%), enquanto que os menores valores da taxa de falso alarme da classe SD (TF_{SD}) foram obtidos com as redes ANN-PSO $f_2(\mathbf{P})$. Nota-se que, a rede treinada com ANN-PSO $f_2(\mathbf{P})$ e restrição de $TF_{SD} \leq 1\%$ apresentou $TD_{SD} = 64\%$, valor significativamente maior que o obtido com a restrição de $TF_{SD} \leq 0,5\%$, que proporcionou $TD_{SD} = 36\%$. O segundo maior valor de TD_{SD} foi obtido com ANN-BP LM (94%), a desvantagem desta rede é apresentar o maior valor de $TF_{SD} = 8$. Para o gradiente descendente, os maiores valores de TD_{SD} foram obtidos por ANN-BP LM e ANN-BP SGD. Entretanto, estas redes apresentaram altos valores de TF_{SD} , o que é indesejável para a aplicação considerada neste trabalho.

Tabela 7.4: Matriz de confusão (em %) e respectivo PE (em %) para sinais sem pré-processamento (informação no domínio do tempo). Considerando a ANN com maior PE de cada método de treinamento.

		SD	DE	DI
ANN-BP SGD	SD	92	4	4
	DE	2	88	28
	DI	6	8	68
$PE = 82,0$				
ANN-BP Adam	SD	90	12	2
	DE	6	80	8
	DI	4	8	90
$PE = 86,5$				
ANN-BP LM	SD	94	10	6
	DE	2	86	18
	DI	4	4	76
$PE = 85,0$				
ANN-BP Rprop	SD	90	2	2
	DE	4	66	18
	DI	6	32	80
$PE = 78,0$				
ANN-PSO $f_1(\mathbf{P})$	SD	98	4	8
	DE	2	84	4
	DI	0	12	88
$PE = 89,8$				
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 0,5\%$	SD	36	0	0
	DE	0	50	20
	DI	64	50	80
$PE = 52,4$				
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 1\%$	SD	64	2	0
	DE	20	60	28
	DI	16	38	72
$PE = 65,2$				

Tabela 7.5: Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) para sinais sem pré-processamento (informação no domínio do tempo). Considerando a ANN com maior PE de cada método de treinamento.

	TF_{SD}	TD_{SD}
ANN-BP SGD	4	92
ANN-BP Adam	7	90
ANN-BP LM	8	94
ANN-BP Rprop	2	90
ANN-PSO $f_1(\mathbf{P})$	6	98
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 0,5\%$	0	36
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 1\%$	1	64

7.4.2 ANN treinadas com informação pré-processada por DWT

A Figura 7.15 mostra que o pré-processamento de sinais com DWT resulta na melhora da mediana do PE se comparado ao caso anterior (exceto para o método SGD). Os maiores valores de PE foram obtidos com ANN-PSO $f_1(\mathbf{P})$. Entre os métodos gradiente descendente, o ANN-BP Adam proporcionou os melhores resultados, seguido por ANN-BP Rprop e ANN-BP LM. O ANN-BP SGD foi o único método que mostrou resultados piores em relação ao caso anterior. O método ANN-PSO $f_2(\mathbf{P})$ proporcionou valores de PE similares ao caso anterior, que foram menores em relação aos demais métodos devido à imposição de valor máximo de TF_{SD} .

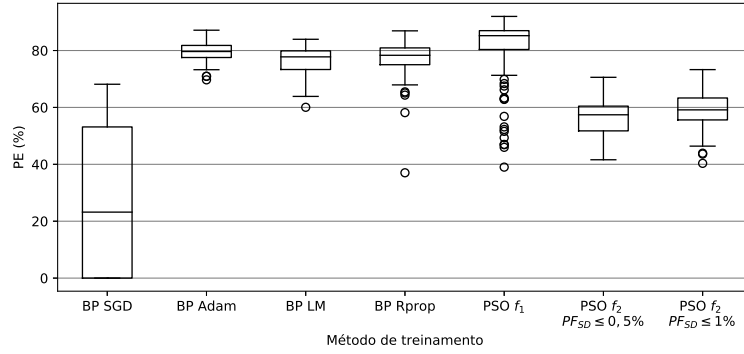


Figura 7.15: *Boxplot* do produto das eficiências para ANN treinada por diferentes métodos com sinais pré-processados por DWT.

Da Tabela 7.6 observa-se que o método ANN-PSO $f_1(\mathbf{P})$ apresentou o maior PE entre as redes analisadas. Entre as redes treinadas com gradiente descendente, o método ANN-BP Adam novamente proporcionou o maior PE, seguido do ANN-BP Rprop. O PE da rede treinada com ANN-PSO $f_2(\mathbf{P})$ e restrição de $TF_{SD} \leq 0,5\%$ aumentou de 52,4% do caso anterior para 64,9%. Considerando a restrição de $TF_{SD} \leq 1\%$, o PE aumentou de 65,2% para 65,9%.

A Tabela 7.7 mostra que o único método que proporcionou $TF_{SD} = 0\%$ foi ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 0,5\%$. Os outros menores valores de TF_{SD} foram obtidos com $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 1\%$ (1%), ANN-PSO $f_1(\mathbf{P})$ (4%) e ANN-BP Adam (4%). Os maiores valores de

Tabela 7.6: Matriz de confusão (em %) e respectivo PE (em %) para sinais pré-processados por DWT. Considerando a ANN com maior PE de cada método de treinamento.

		SD	DE	DI
ANN-BP SGD	SD	88	14	12
	DE	2	62	30
	DI	10	24	58
$PE = 68,1$				
ANN-BP Adam	SD	88	8	0
	DE	4	80	6
	DI	8	12	94
$PE = 87,1$				
ANN-BP LM	SD	86	14	6
	DE	4	80	8
	DI	10	6	86
$PE = 84,0$				
ANN-BP Rprop	SD	96	12	2
	DE	2	76	8
	DI	2	12	90
$PE = 86,9$				
ANN-PSO $f_1(\mathbf{P})$	SD	92	4	4
	DE	6	88	6
	DI	2	8	90
$PE = \mathbf{90,0}$				
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 0,5\%$	SD	56	0	0
	DE	0	66	26
	DI	44	34	74
$PE = 64,9$				
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 1\%$	SD	64	2	0
	DE	6	62	28
	DI	30	36	72
$PE = 65,9$				

TD_{SD} foram obtidos com ANN-BP Rprop (96%) e ANN-PSO $f_1(\mathbf{P})$ (92%). Sendo que, o ANN-BP Rprop proporcionou $TF_{SD} = 7\%$. Nota-se que a ANN treinada com $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 1\%$ proporcionou maior TD_{SD} em comparação à ANN treinada com $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 0,5\%$.

7.4.3 ANN treinadas com informação pré-processada por DFT

A Figura 7.16 mostra que o pré-processamento por DFT proporcionou efeito similar ao observado para sinais pré-processados com DWT. Houve melhora da mediana do PE se comparado ao caso em que as ANN foram treinadas com sinais no domínio do tempo (exceto para redes treinadas com ANN-BP SGD). Novamente, os maiores valores de PE foram obtidos

Tabela 7.7: Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) para sinais pré-processados por DWT. Considerando a ANN com maior PE de cada método de treinamento.

	TF_{SD}	TD_{SD}
ANN-BP SGD	13	88
ANN-BP Adam	4	88
ANN-BP LM	10	86
ANN-BP Rprop	7	96
ANN-PSO $f_1(\mathbf{P})$	4	92
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 0,5\%$	0	56
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 1\%$	1	64

com ANN-PSO $f_1(\mathbf{P})$. Entre os métodos gradiente descendente, o ANN-BP Rprop proporcionou melhores resultados, seguido do ANN-BP Adam e ANN-BP LM. Tanto a mediana, quanto o valor máximo de PE proporcionados pelo método ANN-PSO $f_2(\mathbf{P})$ diminuíram em comparação com as ANN treinadas com informação no domínio do tempo. Comportamento contrário ao observado com ANN treinadas com sinais pré-processados com DWT com o método ANN-PSO $f_2(\mathbf{P})$.

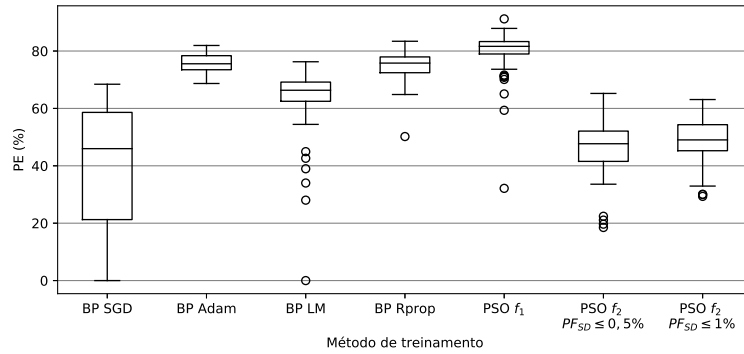


Figura 7.16: *Boxplot* do produto das eficiências para ANN treinada por diferentes métodos com sinais pré-processados por DFT.

A Tabela 7.8 mostra que o método ANN-PSO $f_1(\mathbf{P})$ apresentou PE 7,8 pontos percentuais maior que o melhor caso com o algoritmo gradiente descendente (ANN-BP Rprop). Da Tabela 7.9 observa-se que a maior taxa de detecção da classe SD (TD_{SD}) foi obtida com ANN-BP Rprop (98%), seguida da rede treinada com ANN-PSO $f_1(\mathbf{P})$ (96%). Apesar da diferença de 2 pontos percentuais de TD_{SD} , o método ANN-PSO $f_1(\mathbf{P})$ proporcionou

$TF_{SD} = 4\%$, 8 pontos percentuais a menos que o TF_{SD} obtido com ANN-BP Rprop. A rede treinada com o método $f_2(\mathbf{P})$ e restrição de $TF_{SD} \leq 0,5\%$ obteve melhora significativa no valor de TD_{SD} em relação à rede treinada com sinais no domínio do tempo. O contrário foi observado ao se aplicar a restrição de $TF_{SD} \leq 1\%$.

Tabela 7.8: Matriz de confusão (em %) e respectivo PE (em %) para sinais pré-processados por DFT. Considerando a ANN com maior PE de cada método de treinamento.

		SD	DE	DI
ANN-BP SGD	SD	76	4	12
	DE	14	62	20
	DI	10	34	68
$PE = 68,4$				
ANN-BP Adam	SD	86	6	6
	DE	2	78	12
	DI	12	16	82
$PE = 81,9$				
ANN-BP LM	SD	80	6	0
	DE	6	66	16
	DI	14	28	84
$PE = 76,3$				
ANN-BP Rprop	SD	98	8	16
	DE	0	74	4
	DI	2	18	80
$PE = 83,4$				
ANN-PSO $f_1(\mathbf{P})$	SD	96	2	6
	DE	0	92	8
	DI	4	6	86
$PE = \mathbf{91,2}$				
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 0,5\%$	SD	52	0	0
	DE	38	86	78
	DI	10	14	22
$PE = 46,2$				
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 1\%$	SD	54	0	2
	DE	24	74	52
	DI	22	26	46
$PE = 56,9$				

7.4.4 ANN treinadas com informação pré-processada por DWT+PCA

Após pré-processamento por DWT e posterior compactação de sinais por PCA, o método proposto baseado em PSO proporcionou resultados mais estáveis e menor número de *outliers* conforme pode ser observado na Figura 7.17. Novamente, o método ANN-PSO $f_1(\mathbf{P})$ proporcionou o maior valor de PE.

Tabela 7.9: Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) para sinais pré-processados por DFT. Considerando a ANN com maior PE de cada método de treinamento.

	TF_{SD}	TD_{SD}
ANN-BP SGD	8	76
ANN-BP Adam	6	86
ANN-BP LM	3	80
ANN-BP Rprop	12	98
ANN-PSO $f_1(\mathbf{P})$	4	96
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 0,5\%$	0	52
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 1\%$	1	54

Seguido por ANN-BP Adam, ANN-BP Rprop e ANN-BP LM. De forma similar aos casos anteriores, o método ANN-BP SGD proporcionou resultados menos satisfatórios se comparado ao caso de treinamento com informação no domínio do tempo.

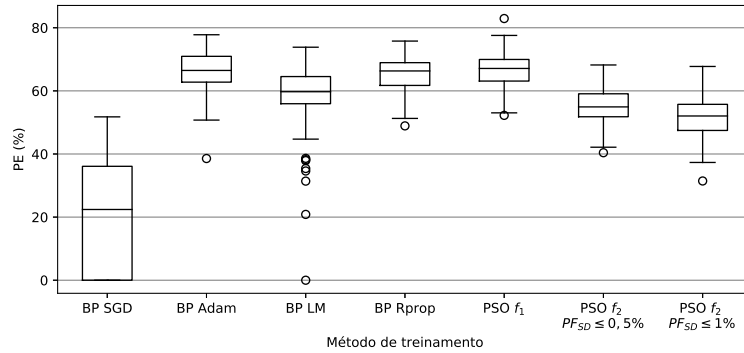


Figura 7.17: *Boxplot* do produto das eficiências para ANN treinada por diferentes métodos com sinais pré-processados por DWT+PCA.

Da Tabela 7.10 observa-se que o maior valor de PE foi obtido com o método ANN-PSO $f_1(\mathbf{P})$ (82,9%). Entre os métodos gradiente descendente, o maior PE foi obtido com ANN-BP Adam (77,8%). Nota-se que a compactação com PCA reduziu os valores de PE para todos métodos de treinamento ao se comparar com a Tabela 7.6. Entretanto, o método ANN-PSO f_2 apresentou aumento no número de verdadeiros positivos da classe sem defeito, impactando positivamente em TD_{SD} .

A Tabela 7.11 mostra que o método ANN-PSO $f_1(\mathbf{P})$ proporcionou $TD_{SD} =$

Tabela 7.10: Matriz de confusão (em %) e respectivo PE (em %) para sinais pré-processados por DWT+PCA. Considerando a ANN com maior PE de cada método de treinamento.

		SD	DE	DI
ANN-BP SGD	SD	60	8	6
	DE	12	34	26
	DI	28	58	68
$PE = 51,8$				
ANN-BP Adam	SD	82	4	4
	DE	0	70	14
	DI	18	26	82
$PE = 77,8$				
ANN-BP LM	SD	80	8	4
	DE	6	74	28
	DI	14	18	68
$PE = 73,8$				
ANN-BP Rprop	SD	80	6	0
	DE	6	68	20
	DI	14	26	80
$PE = 75,8$				
ANN-PSO $f_1(\mathbf{P})$	SD	88	4	8
	DE	2	90	20
	DI	10	6	72
$PE = \mathbf{82,9}$				
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 0,5\%$	SD	70	0	0
	DE	2	64	50
	DI	28	36	50
$PE = 60,7$				
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 1\%$	SD	66	2	0
	DE	4	62	48
	DI	26	36	52
$PE = 59,7$				

88%, sendo 6 pontos percentuais maior que o segundo maior valor, $TD_{SD} = 82\%$, obtido com ANN-BP Adam. Entretanto, ANN-PSO $f_1(\mathbf{P})$ apresentou alto valor de $TF_{SD} = 6\%$. O método ANN-PSO $f_2(\mathbf{P})$ apresentou $TF_{SD} = 0\%$, como nos demais casos, contudo o pré-processamento com DWT+PCA auxiliou no aumento do TD_{SD} , que obteve melhora significativa, alcançando $TD_{SD} = 70\%$ para a restrição de $TF_{SD} \leq 0,5\%$. No caso da ANN treinada com ANN-PSO f_2 com restrição de $TF_{SD} \leq 1\%$ foi obtido o valor de $TD_{SD} = 66\%$. Este foi o único caso em que o método ANN-PSO f_2 com restrição de $TF_{SD} \leq 1\%$ proporcionou menor TD_{SD} que o método ANN-PSO f_2 com restrição de $TF_{SD} \leq 0,5\%$. Vale ressaltar que os métodos de treinamento de redes neurais, tanto os baseados em gradiente descendente, quanto o método proposto baseado em PSO, não são exatos e os resultados podem variar de acordo com diversos fatores como a inicialização e escolha

dos hiper-parâmetros.

Tabela 7.11: Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) para sinais pré-processados por DWT+PCA. Considerando a ANN com maior PE de cada método de treinamento.

	TF_{SD}	TD_{SD}
ANN-BP SGD	7	60
ANN-BP Adam	4	82
ANN-BP LM	6	80
ANN-BP Rprop	3	80
ANN-PSO $f_1(\mathbf{P})$	6	88
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 0,5\%$	0	70
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 1\%$	1	66

7.4.5 ANN treinadas com informação pré-processada com DFT+PCA

A Figura 7.18 mostra o PE das redes treinadas com informação pré-processada com DFT e posteriormente compactada com PCA. O efeito da compactação com PCA foi similar ao observado nas redes treinadas com sinais pré-processados com DWT+PCA. Os *boxplots* apresentaram menor número de *outliers*, com exceção para o método ANN-PSO f_2 . O método ANN-PSO $f_1(\mathbf{P})$ apresentou maiores valores de PE, seguido de ANN-BP Rprop, ANN-BP Adam e ANN-BP LM. De forma similar aos casos anteriores, o ANN-BP SGD apresentou piora em comparação ao caso de treinamento com informação no domínio do tempo. O ANN-PSO f_2 apresentou valores menores de PE se comparado aos demais métodos devido às restrições impostas.

Comparando as Tabelas 7.8 e 7.12, nota-se que após a compactação com PCA os métodos ANN-BP SGD e ANN-PSO $f_1(\mathbf{P})$ obtiveram menores valores de PE. Enquanto que os demais métodos de treinamento obtiveram aumento de PE. Contudo, o maior valor de PE novamente foi obtido com ANN-PSO $f_1(\mathbf{P})$ (88,7%). O segundo maior valor foi obtido com ANN-BP Rprop (84,7%), quatro pontos percentuais a menos que o maior valor. A rede treinada com o método ANN-PSO f_2 com restrição de $TF_{SD} \leq 0,5\%$ não classificou nenhum sinal de uma região com defeito como sendo sem defeito.

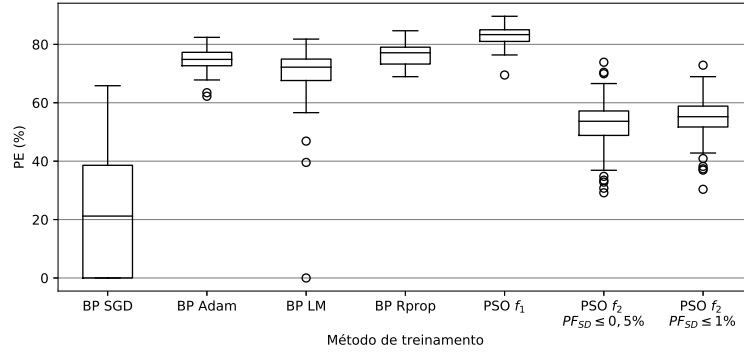


Figura 7.18: *Boxplot* do produto das eficiências para ANN treinada por diferentes métodos com sinais pré-processados por DFT+PCA.

Para a rede treinada com restrição de $TF_{SD} \leq 1\%$ houve apenas 2 erros de classificação deste tipo.

Tabela 7.12: Matriz de confusão (em %) e respectivo PE (em %) para sinais pré-processados por DFT+PCA. Considerando a ANN com maior PE de cada método de treinamento.

		SD	DE	DI
ANN-BP SGD	SD	82	20	26
	DE	14	60	16
	DI	4	20	58
	$PE = 65,8$			
ANN-BP Adam	SD	90	6	6
	DE	4	74	10
	DI	6	20	84
	$PE = 82,4$			
ANN-BP LM	SD	86	10	8
	DE	4	74	6
	DI	10	16	86
	$PE = 81,8$			
ANN-BP Rprop	SD	84	2	2
	DE	8	86	14
	DI	8	12	84
	$PE = 84,7$			
ANN-PSO $f_1(\mathbf{P})$	SD	90	6	6
	DE	4	88	6
	DI	6	6	88
	$PE = 88,7$			
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 0,5\%$	SD	62	0	0
	DE	18	32	38
	DI	20	68	62
	$PE = 49,7$			
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 1\%$	SD	74	2	0
	DE	14	68	52
	DI	12	30	48
	$PE = 62,3$			

Segundo a Tabela 7.13, os métodos ANN-BP Adam e ANN-PSO $f_1(\mathbf{P})$ coincidiram no valor de $TD_{SD} = 90\%$. Contudo, da Tabela 7.12, percebe-se que o valor de PE foi maior para ANN-PSO $f_1(\mathbf{P})$ (88,7%) contra 82,4% para ANN-BP Adam. Dentre os métodos gradiente descendente, o menor valor de TF_{SD} foi obtido com ANN-BP Rprop (2%). O valor TD_{SD} correspondente foi 84%. O único método a obter $TF_{SD} = 0\%$ foi o ANN-PSO $f_2(\mathbf{P})$ com restrição de $TD_{SD} \leq 0,5\%$, cujo TD_{SD} correspondente foi 62%. O método ANN-PSO $f_2(\mathbf{P})$ com restrição de $TD_{SD} \leq 1\%$ proporcionou uma ANN com $TD_{SD} = 74\%$. Sendo este o maior valor de TD_{SD} obtido com a função objetivo $f_2(\mathbf{P})$.

Tabela 7.13: Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) para sinais pré-processados por DFT+PCA. Considerando a ANN com maior PE de cada método de treinamento.

	TF_{SD}	TD_{SD}
ANN-BP SGD	23	82
ANN-BP Adam	6	90
ANN-BP LM	9	86
ANN-BP Rprop	2	84
ANN-PSO $f_1(\mathbf{P})$	6	90
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 0,5\%$	0	62
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 1\%$	1	74

7.5 Análise comparativa dos resultados

Foi observado que o método proposto com função objetivo $f_1(\mathbf{P})$ proporcionou maiores valores de PE em comparação aos demais métodos de treinamento. A maior diferença foi encontrada para ANN treinada com sinais pré-processados com DWT+PCA (Tabela 7.10), em que ANN treinada com o método proposto proporcionou PE com 31,1 pontos percentuais maior que ANN treinada com ANN-BP SGD. A menor diferença ocorreu para ANN treinadas com sinais pré-processados por DWT (Tabela 7.6), em que o método proposto forneceu PE 2,9 pontos percentuais maior que ANN treinada com ANN-BP Adam.

A ANN que apresentou maior PE foi treinada pelo método ANN-PSO

$f_1(\mathbf{P})$ e sinais pré-processados por DFT (Tabela 7.8), resultando em $PE = 91,2\%$ e $TD_{SD} = 96\%$. Neste caso, a taxa de falso alarme da classe SD (TF_{SD}) foi de 4%. Para um classificador treinado com *backpropagation* com igual TD_{SD} , foi observado $TF_{SD} = 7\%$, correspondendo ao método ANN-BP Rprop para sinais pré-processados com DWT (Tabela 7.7).

A Figura 7.19 mostra o gráfico de $TD_{SD} \times PF_{SD}$ para redes treinadas com ANN-PSO $f_1(\mathbf{P})$, ANN-PSO $f_2(\mathbf{P})$ e a rede para treinamento por gradiente descendente com menor TF_{SD} para cada caso de pré-processamento da informação. Os menores valores de TF_{SD} foram obtidos com o método proposto e função objetivo $f_2(\mathbf{P})$, enquanto que os maiores valores de TD_{SD} foram obtidos com o método proposto e função objetivo $f_1(\mathbf{P})$. Em todos os casos de pré-processamento foi observado que o método proposto com função objetivo $f_1(\mathbf{P})$ proporcionou valores maiores de TD_{SD} em comparação ao gradiente descendente, por outro lado o método proposto com função objetivo $f_2(\mathbf{P})$ proporcionou valores menores de TF_{SD} .

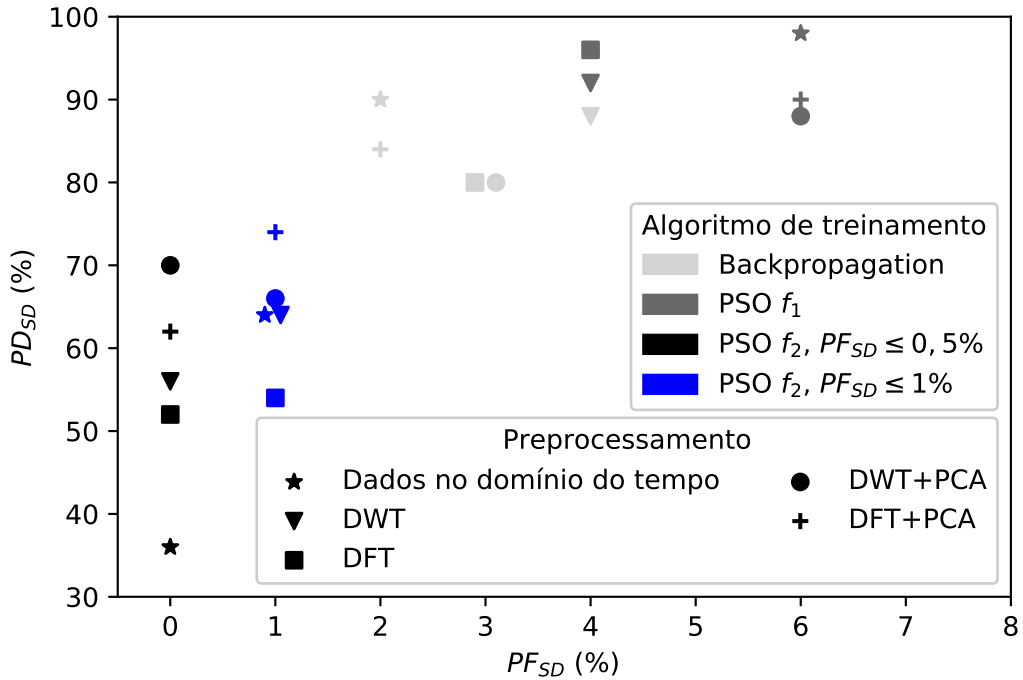


Figura 7.19: Gráfico de TD_{SD} em função de TF_{SD} para diferentes técnicas de pré-processamento e algoritmos de treinamento.

Para a aplicação de interesse neste trabalho, é considerado que classificar incorretamente um tubo defeituoso como não defeituoso é mais grave que cometer o erro oposto (classificar um tubo não defeituoso como defeituoso). Portanto, é desejável obter baixo valor de TF_{SD} . O método proposto com função objetivo $f_2(\mathbf{P})$ e restrição de $TF_{SD} \leq 0,5\%$ proporcionou $TF_{SD} = 0\%$ para todos os casos de pré-processamento. Com este método, o maior valor de TD_{SD} foi obtido com sinais pré-processados com DWT+PCA ($TD_{SD} = 70\%$, conforme ilustrado na Tabela 7.11). Foi observado que ao aliviar a restrição de TF_{SD} aumentando de 0,5% para 1% houve aumento do valor de TD_{SD} em todos os casos analisados com exceção da rede treinada com sinais pré-processados com DWT+PCA. Com a restrição de $TF_{SD} \leq 1\%$ o maior valor de TD_{SD} foi obtido com a rede treinada com sinais pré-processados com DFT+PCA, alcançando o valor $TD_{SD} = 74\%$.

Após treinamento das ANN e avaliação com o conjunto de teste, o sistema de classificação foi testado com novas leituras. Tomando-se o cuidado de analisar os acertos e erros de classificação de acordo com a posição em que o ensaio foi realizado. Esta análise foi realizada com três ANN, sendo treinadas por ANN-BP Rprop (com sinais pré-processados por DFT), ANN-PSO $f_1(\mathbf{P})$ (com sinais pré-processados por DFT+PCA) e ANN-PSO $f_2(\mathbf{P})$ (com sinais pré-processados por DWT+PCA). Para cada região de interesse foram realizados dez ensaios, totalizando trinta.

A Figura 7.20 mostra uma visão perimétrica do corpo de prova indicando o local e resultado da análise realizada com a rede treinada por ANN-BP Rprop. As regiões de interesse estão representadas em cores distintas. As predições realizadas pela ANN estão representadas por símbolos, triângulo para classe sem defeito, círculo para defeito externo e quadrado para defeito interno. Esta ANN classificou 10% dos ensaios na região de defeito interno como sendo sem defeito e 10% como defeito externo. Na região de defeito externo houve 30% dos ensaios classificados como sem defeito, os demais ensaios foram classificados corretamente. Na região sem defeito, 80% dos ensaios foram classificados corretamente, os demais foram classificados como defeito interno. A Tabela 7.14 mostra a matriz de confusão desta análise e a Tabela 7.15 mostra os valores de TF_{SD} e TD_{SD} .

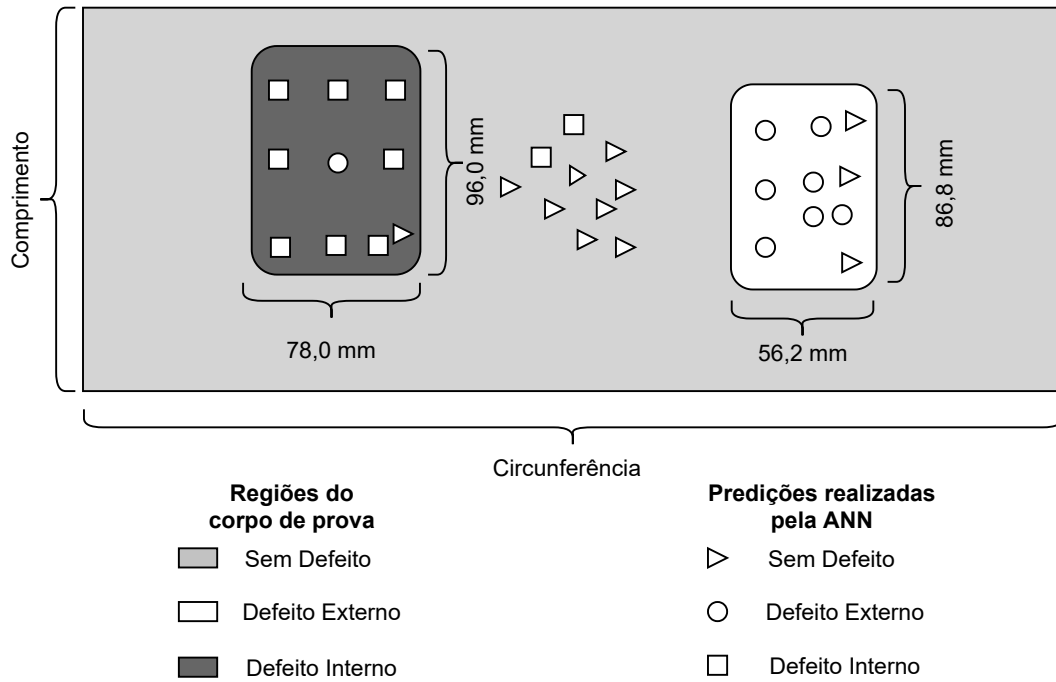


Figura 7.20: Visão perimétrica do corpo de prova indicando o local de realização de ensaios com PEC e as respectivas predições realizadas pela ANN treinada pelo método ANN-BP Rprop.

Tabela 7.14: Matriz de confusão (em %) referente ao teste final da rede treinada com ANN-BP Rprop e sinais pré-processados por DFT.

		SD	DE	DI
ANN-BP Rprop	SD	80	30	10
	DE	0	70	10
	DI	20	0	80
$PE = 76,5$				

Tabela 7.15: Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) referente ao teste final da rede treinada com ANN-BP Rprop e sinais pré-processados por DFT.

	TF_{SD}	TD_{SD}
ANN-BP Rprop	20	80

A ANN treinada com ANN-PSO $f_1(\mathbf{P})$ (Figura 7.21) classificou 10% ensaios da região de defeito interno como sem defeito, os demais ensaios foram classificados corretamente. Na região de defeito externo 80% dos ensaios foram classificados corretamente, 10% foram classificados como defeito externo e 10% como sem defeito. Na região sem defeito, 90% dos ensaios foram

classificados corretamente e 10% foram classificados como defeito externo. Estes resultados estão resumidos na matriz de confusão da Tabela 7.16. A Tabela 7.17 mostra os valores de TF_{SD} e TD_{SD} desta análise.

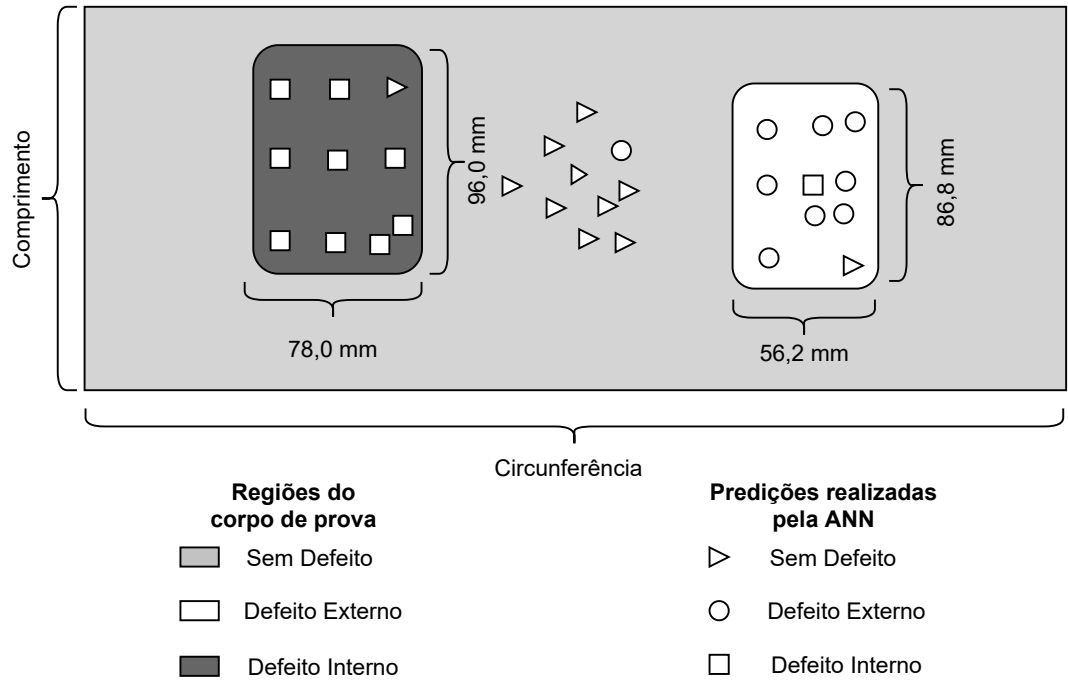


Figura 7.21: Visão perimétrica do corpo de prova indicando o local de realização de ensaios com PEC e as respectivas predições realizadas pela ANN treinada pelo método ANN-PSO $f_1(\mathbf{P})$.

Tabela 7.16: Matriz de confusão (em %) referente ao teste final da rede treinada com ANN-PSO $f_1(\mathbf{P})$ e sinais pré-processados por DFT.

		SD	DE	DI
ANN-PSO $f_1(\mathbf{P})$	SD	90	10	10
	DE	10	80	0
	DI	0	10	90
$PE = 86,5$				

Tabela 7.17: Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) referente ao teste final da rede treinada com ANN-PSO $f_1(\mathbf{P})$ e sinais pré-processados por DFT.

	TF_{SD}	TD_{SD}
ANN-PSO $f_1(\mathbf{P})$	10	80

Para a ANN treinada com ANN-PSO $f_2(\mathbf{P})$ (Figura 7.22), 60% dos ensaios na região com defeito interno foram classificados corretamente e 40% foram classificados como defeito externo, entretanto não houve classificações como sem defeito. Mesmo fato ocorreu na região de defeito externo. 60% dos ensaios foram classificados corretamente e 40% como defeito interno, porém sem classificação como sem defeito. Na região sem defeito, 70% dos ensaios foram classificados corretamente, 10% como defeito externo e 20% como defeito interno. Na Tabela 7.18 é mostrada a matriz de confusão para este teste, cujos valores de TD_{SD} e TF_{SD} são mostrados na Tabela 7.19.

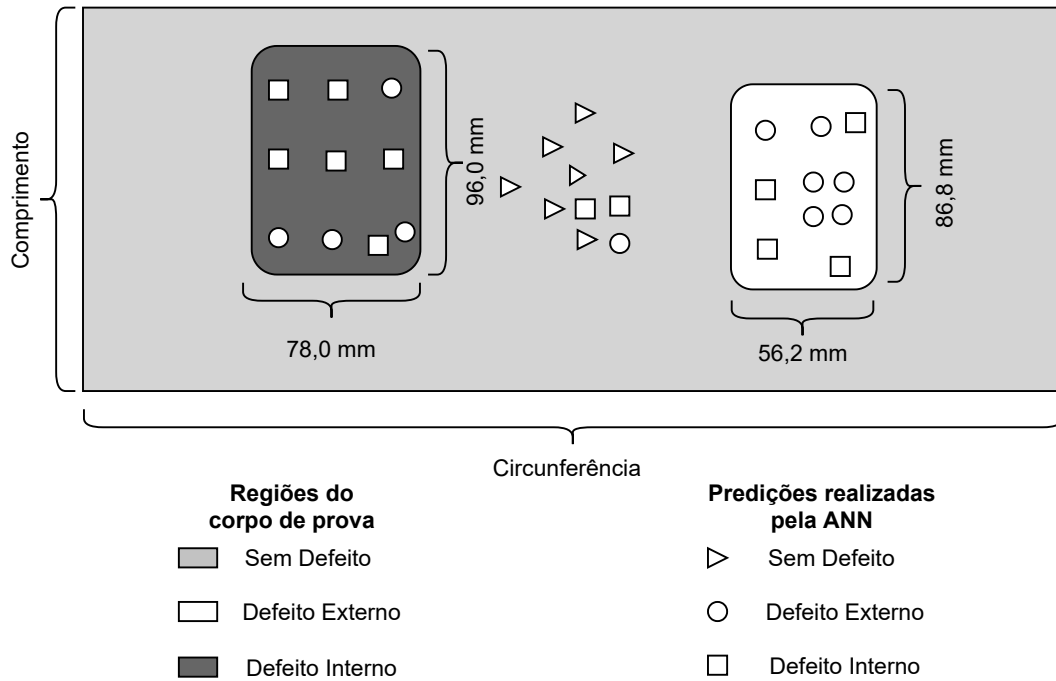


Figura 7.22: Visão perimétrica do corpo de prova indicando o local de realização de ensaios com PEC e as respectivas predições realizadas pela ANN treinada pelo método ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 0,5\%$ e sinais pré-processados por DWT+PCA.

Para a ANN treinada com ANN-PSO $f_2(\mathbf{P})$ e $TF_{SD} \leq 1\%$ (Figura 7.23), 50% dos ensaios na região com defeito interno foram classificados corretamente e 50% foram classificados como defeito externo, entretanto não houve classificações como sem defeito. Mesmo fato ocorreu na região de defeito externo, Onde 70% dos ensaios foram classificados corretamente e 30% como defeito interno, porém sem classificação como sem defeito. Na região sem

Tabela 7.18: Matriz de confusão (em %) referente ao teste final da rede treinada com ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 0,5\%$ e sinais pré-processados por DWT+PCA.

		SD	DE	DI
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 0,5\%$ $PE = 63,2$	SD	70	0	0
	DE	10	60	40
	DI	20	40	60

Tabela 7.19: Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) referente ao teste final da rede treinada com ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 0,5\%$ e sinais pré-processados por DWT+PCA.

	TF_{SD}	TD_{SD}
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 0,5\%$	0	70

defeito, 80% dos ensaios foram classificados corretamente, 10% como defeito externo e 10% como defeito interno. Na Tabela 7.20 é mostrada a matriz de confusão para este teste, cujos valores de TD_{SD} e TF_{SD} são mostrados na Tabela 7.21.

Tabela 7.20: Matriz de confusão (em %) referente ao teste final da rede treinada com ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 1\%$ e sinais pré-processados por DFT+PCA.

		SD	DE	DI
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 1\%$ $PE = 65,4$	SD	80	0	0
	DE	10	70	50
	DI	10	30	50

Tabela 7.21: Taxa de falso alarme TF_{SD} e detecção TD_{SD} da classe SD (em %) referente ao teste final da rede treinada com ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 1\%$ e sinais pré-processados por DFT+PCA.

	TF_{SD}	TD_{SD}
ANN-PSO $f_2(\mathbf{P})$, $TF_{SD} \leq 1\%$	0	80

7.6 Resultados de Trabalho Correlato

O conjunto de dados utilizado neste trabalho também foi utilizado no trabalho [95], que utilizou como pré-processamento as transformadas DFT,

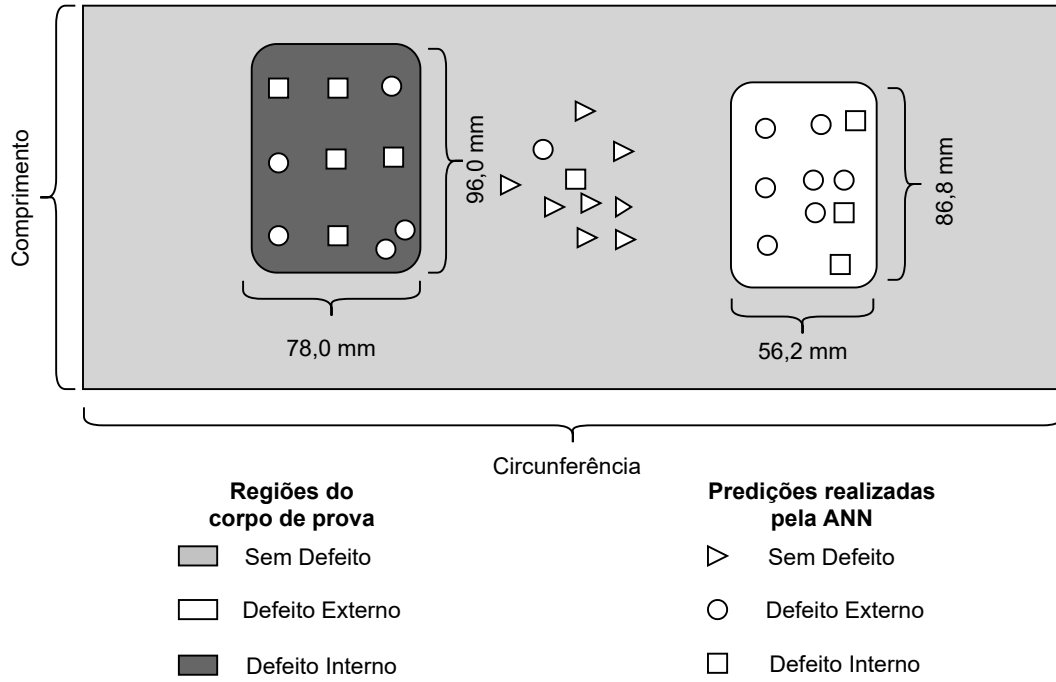


Figura 7.23: Visão perimétrica do corpo de prova indicando o local de realização de ensaios com PEC e as respectivas predições realizadas pela ANN treinada pelo método ANN-PSO $f_2(\mathbf{P})$ com restrição de $TF_{SD} \leq 1\%$ e sinais pré-processados por DFT+PCA.

DWT, a transformada discreta de cossenos (DCT - *Discrete Cosine Transform*), em seguida a PCA, para então aplicar os sinais na entrada de uma rede neural treinada pelo método *Lavenberg Marquart* (LM).

Os autores analisaram o desempenho dos classificadores a partir do PE, sendo que o classificador que proporcionou o maior PE foi treinado com informação pré-processada por DWT+PCA, com $PE = 96,50\%$. Dentre as técnicas de pré-processamento analisadas, a DCT proporcionou os piores resultados, com $PE = 72,60\%$.

Apesar dos autores não analisarem os classificadores quanto a TF_{SD} e TD_{SD} , as matrizes de confusão de cada classificador constam no artigo. A partir delas é possível calcular que o classificador treinado com informação pré-processada por DWT+PCA proporcionou $TF_{SD} = 0\%$ e $TD_{SD} = 100\%$. Entretanto, o conjunto de dados foi dividido em 80% para treino e 20% para teste. Visto que para cada classe existem 100 sinais, utilizar apenas

20% desses sinais no conjunto de teste dificulta uma comparação junta entre classificadores.

Capítulo 8

Conclusões

Neste trabalho é proposta uma metodologia para detecção automática de corrosão em tubos industriais sem a necessidade de remoção do isolamento térmico. Trata-se de um ensaio não destrutivo com correntes parasitas pulsadas em conjunto com um classificador neural treinado por um método proposto baseado em PSO com restrições. O método de correntes parasitas pulsadas é adequado para detecção de corrosão, pois é possível inspecionar tubos sem retirar o isolamento térmico. Portanto, é uma opção viável para indústrias que não podem operar caso os tubos estejam sem isolamento, seja por questões de eficiência energética ou de segurança.

Para pré-processamento da informação utilizou-se transformada discreta de Fourier, transformada discreta wavelet e análise de componentes principais. Cada tipo de pré-processamento influenciou de forma distinta no desempenho dos classificadores neurais. Caso o objetivo principal seja obter maior valor de produto das eficiências, a melhor opção é utilizar transformada discreta de Fourier e classificador neural treinado pelo método proposto com função objetivo f_1 . Caso no qual foi obtido o maior valor de $PE = 91.2$.

A opção que proporciona o maior valor da taxa de detecção da classe SD (TD_{SD}) é utilizar um classificador treinado pelo método proposto e função f_1 com informação no domínio do tempo. Neste caso o classificador proporcionou $TD_{SD} = 98\%$. Há outro classificador que proporcionou $TD_{SD} = 98\%$. É o caso do classificador treinado pelo método *resilient back-propagation* (Rprop) com sinal pré-processado por DFT. Entretanto, este

proporcionou maior valor de TF_{SD} e menor valor de PE se comparado ao classificador treinado pelo método proposto e função f_1 com informação no domínio do tempo.

Entretanto, se a necessidade de valores baixos de taxa de falso positivo da classe SD (TF_{SD}) for mais importante, então a melhor opção é utilizar transformada discreta wavelet e análise de componentes principais juntamente com um classificador neural treinado pelo método proposto e função objetivo f_2 . Neste caso foi possível obter $TF_{SD} = 0$ e $TD_{SD} = 70.0\%$. Com este valor de TF_{SD} nenhum tubo defeituoso é incorretamente classificado como não defeituoso e devido a $TD_{SD} = 70.0\%$ apenas 30% dos tubos sem defeito são incorretamente classificados como com algum tipo de defeito.

8.1 Principais Contribuições

A principal contribuição deste trabalho é o método proposto para treinamento de classificadores neurais utilizando PSO com restrições para considerar cada tipo de erro de classificação de forma distinta. O método proposto possibilita maior liberdade no treinamento de classificadores neurais se comparado ao método gradiente descendente. Pois, o gradiente descendente desconsidera que cada erro de classificação pode provocar diferentes consequências, enquanto o PSO pode considerar cada erro de classificação de maneira distinta. Essa distinção é feita aplicando-se restrições durante o treinamento do classificador, o que não é facilmente realizável com o gradiente descendente.

As funções objetivo e restrições propostas também são contribuições importante, visto que estes são fatores que influenciam no desempenho de classificadores treinados com o método proposto. Neste trabalho foram testadas diversas funções objetivo e restrições, sendo que duas funções proporcionaram melhores resultados, que foram discutidos nos capítulos anteriores.

Existes diversas formas de treinar um classificador neural, seja com algoritmos baseados em gradiente descendente ou em heurísticas bio-inspiradas. Entretanto, pouco se desenvolveu até o momento em relação à inclusão de restrições de treinamento para considerar os erros de classificação de maneira

distinta. A proposta deste trabalho, de um novo método para treinamento de classificadores neurais, pode auxiliar outros engenheiros e profissionais afins a treinar classificadores neurais com melhor desempenho de classificação tanto em aplicações acadêmicas, quanto empresariais cujos erros de classificação possuem consequências distintas.

Os impactos deste trabalho não se restringem às aplicações industriais. Pois, o método proposto não se restringe à classificadores neurais para sinais de ensaios não destrutivos. Pode também ser aplicado em outros problemas que envolvem diferentes consequências dos diferentes erros de classificação, como classificação de informação para auxílio ao diagnóstico médico ou avaliação de conteúdo multimídia quanto à classificação indicativa. Para cada aplicação é interessante avaliar novas funções objetivo e restrições de forma a encontrar a que sejam mais adequadas ao problema em questão.

8.2 Sugestões para trabalhos futuros

Os próximos trabalhos serão focados em inspecionar tubos de diferentes ligas metálicas e espessuras distintas. De forma a abranger uma certa variedade de tubos utilizados atualmente na indústria. Espera-se que, a inclusão de mais sinais de inspeção com PEC possibilite treinar um classificador neural com maior capacidade de generalização e detecção das classes desejadas.

Neste trabalho foi inspecionado um corpo de prova com regiões sem defeito, com defeito externo ou com defeito interno. Em aplicações mais gerais o corpo de prova pode apresentar defeito interno o externo simultaneamente na mesma região. Neste caso, é mais apropriado treinar um classificador multietiqueta em vez de multiclasse. Entretanto, para treinar tal classificador é necessário a inclusão de sinais de inspeção de outros corpos de prova.

Outro ponto a ser investigado é a possibilidade de criar um *ensemble* [38], que consiste em treinar vários classificadores neurais e combiná-los de alguma forma a criar um novo classificador com maior desempenho. Esta combinação pode ser feita, por exemplo, utilizando-se média ponderada das predições de cada classificador. No experimento pretendido, os pesos serão determinados com PSO. Sendo que, por meio de função objetivo e das restrições aplicadas

ao PSO pode-se criar um classificador que considere cada classe de maneira distinta.

Neste trabalho foram treinadas redes do tipo MLP. Trabalhos futuros podem testar outros tipos de redes neurais, como redes neurais convolucionais (CNN - *convolutional neural network*) e redes neurais recorrentes (RNN - *Recurrent Neural Network*) [38]. Pode-se também utilizar um classificador hierárquico, que é composto de dois subclassificadores. O primeiro classifica se o sinal corresponde a uma região sem defeito ou com defeito, caso haja defeito, o segundo classifica o sinal quanto ao tipo de defeito [96].

Por fim, pode-se embarcar o sistema de classificação num dispositivo portátil capaz de realizar as medições e o diagnóstico em campo. Este dispositivo deve ter capacidade de processamento suficiente realizar o pré-processamento do sinal e posterior propagação *forward* pelo classificador neural.

Bibliografia

- [1] S. Winnik, ed., *Corrosion-Under-Insulation (CUI) Guidelines*. Cambridge: Elsevier, 2016, p. 162, ISBN: 9780081007396. DOI: 10.1016/C2015-0-01794-4.
- [2] J. Bailey, N. Long e A. Hunze, “Eddy Current Testing with Giant Magnetoresistance (GMR) Sensors and a Pipe-Encircling Excitation for Evaluation of Corrosion under Insulation,” *Sensors*, v. 17, n. 10, p. 2229, set. de 2017. DOI: 10.3390/s17102229.
- [3] E. E. P. d. Souza, E. F. Simas Filho e P. C. M. A. Farias, “Método De Treinamento De Redes Neurais Através De Otimização Evolutiva Com Restrições Para Auxílio Ao Diagnóstico Médico,” *Anais SBrT*, 2018.
- [4] K. Zhang, Y. He e Z. Dong, “Pulsed Eddy Current Nondestructive Testing for Defect Evaluation and Imaging of Automotive Lightweight Alloy Materials,” *Journal of Sensors*, v. 2018, pp. 1–11, mar. de 2018, ISSN: 1687-725X. DOI: 10.1155/2018/1639387.
- [5] L. F. Rodrigues, F. C. Cruz, M. A. Oliveira, E. F. Simas Filho, M. C. Albuquerque, I. C. Silva e C. T. Farias, “Carburization level identification in industrial HP pipes using ultrasonic evaluation and machine learning,” *Ultrasonics*, v. 94, pp. 145–151, April 2019. DOI: 10.1016/j.ultras.2018.10.005.
- [6] E. L. Chuma, L. G. Meloni, Y. Iano e L. L. B. Roger, “FPGA implementation of a de-noising using Haar level 5 wavelet transform,” *the Proceedings of XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, SBrT2017*, pp. 3–6, 2017.
- [7] I. T. Jolliffe, *Principal Component Analysis*, 2^a ed. Springer, 2002.
- [8] M. Ertürk, “Optimum insulation thicknesses of pipes with respect to different insulation materials, fuels and climate zones in Turkey,” *Energy*, v. 113, pp. 991–1003, 2016, ISSN: 03605442. DOI: 10.1016/j.energy.2016.07.115.

- [9] G. Deshmukh, P. Birwal, R. Datir e S. Patel, “Thermal Insulation Materials: A Tool for Energy Conservation,” *Journal of Food Processing & Technology*, v. 08, n. 04, pp. 8–11, 2017. DOI: 10.4172/2157-7110.1000670.
- [10] S. Xu, L. Chen, M. Gong, X. Hu, X. Zhang e Z. Zhou, “Characterization and engineering application of a novel ceramic composite insulation material,” *Composites Part B: Engineering*, v. 111, pp. 143–147, 2017, ISSN: 13598368. DOI: 10.1016/j.compositesb.2016.12.010.
- [11] T. Ural, A. Daşdemir e A. Keçebaş, “Sensitivity Analysis of Optimum Insulation Thickness for Pipe Diameters in Pipe Insulation with Air Gap for City Pipelines,” *Environmental Progress and Sustainable Energy*, v. 38, n. 5, pp. 1–11, 2019, ISSN: 19447450. DOI: 10.1002/ep.13155.
- [12] X. Bai, J. Tang, J. Gong e X. Lü, “Corrosion performance of Al–Al₂O₃ cold sprayed coatings on mild carbon steel pipe under thermal insulation,” *Chinese Journal of Chemical Engineering*, v. 25, n. 4, pp. 533–539, 2017, ISSN: 10049541. DOI: 10.1016/j.cjche.2016.10.004.
- [13] L. D. Paolinelli e S. Nesic, “Hydrodynamic and phase wetting criteria to assess corrosion risk in two-phase oil-water pipe flow,” em *NACE International Corrosion Conference Proceedings*, NACE International, 2016, p. 19.
- [14] J. Oliveira, “O Pré-Sal Brasileiro e o Problema da Corrosão por CO₂,” *Revista da Pós-Graduação da Faculdade do Centro Leste*, abr. de 2015.
- [15] A. Eidaninezhad, P. Ziyaei e A. Zare, “An overview of marine pipeline repair methods,” jun. de 2019, pp. 1–11.
- [16] M. Shamsuddoha, M. Islam, T. Aravinthan, A. Manalo e K. t. Lau, “Effectiveness of using fibre-reinforced polymer composites for underwater steel pipeline repairs,” *Composite Structures*, v. 100, pp. 40–54, jun. de 2013. DOI: 10.1016/j.compstruct.2012.12.019.
- [17] *Stop It: Pipe Repair*, Acessado: 14 dezembro 2020. endereço: <https://www.tri-chem.com/>.
- [18] *ARC 5 Emergency Metal To Metal Repair Epoxy*, Acessado: 14 dezembro 2020. endereço: <https://www.cactusindustrial.com/>.
- [19] S. Fateri, P. S. Lowe, B. Engineer e N. V. Boulgouris, “Investigation of ultrasonic guided waves interacting with piezoelectric transducers,” *IEEE Sensors Journal*, v. 15, n. 8, pp. 4319–4328, 2015, ISSN: 1530437X. DOI: 10.1109/JSEN.2015.2414874.

- [20] K. Sambasiva Rao, S. Mahadevan, B. Purna Chandra Rao e S. Thirunavukkarasu, “A new approach to increase the subsurface flaw detection capability of pulsed eddy current technique,” *Measurement*, v. 128, pp. 516–526, nov. de 2018, ISSN: 0263-2241. DOI: 10.1016/J.MEASUREMENT.2018.06.065.
- [21] J. García-Martín, J. Gómez-Gil e E. Vázquez-Sánchez, “Non-Destructive Techniques Based on Eddy Current Testing,” *Sensors*, v. 11, n. 3, pp. 2525–2565, fev. de 2011, ISSN: 1424-8220. DOI: 10.3390/s110302525.
- [22] F. T. Ulaby, *Eletromagnetismo para Engenheiros*. Bookman Editora, 2007, 382p, ISBN: 978-85-600-3119-1.
- [23] A. Sophian, G. Tian e M. Fan, “Pulsed Eddy Current Non-destructive Testing and Evaluation: A Review,” *Chinese Journal of Mechanical Engineering*, v. 30, n. 3, pp. 500–514, mai. de 2017, ISSN: 1000-9345. DOI: 10.1007/s10033-017-0122-4.
- [24] G. Y. Tian, Y. He, I. Adewale e A. Simm, “Research on spectral response of pulsed eddy current and NDE applications,” *Sensors and Actuators A: Physical*, v. 189, pp. 313–320, jan. de 2013, ISSN: 09244247. DOI: 10.1016/j.sna.2012.10.011.
- [25] I. Z. Abidin, C. Mandache, G. Y. Tian e M. Morozov, “Pulsed eddy current testing with variable duty cycle on rivet joints,” *NDT & E International*, v. 42, n. 7, pp. 599–605, out. de 2009, ISSN: 0963-8695. DOI: 10.1016/J.NDTEINT.2009.04.001.
- [26] L. Ferrigno, M. Laracca, H. Malekmohammadi, G. Y. Tian e M. Ricci, “Comparison of time and frequency domain features’ immunity against lift-off in pulse-compression eddy current imaging,” *NDT & E International*, v. 107, pp. 1–12, out. de 2019, ISSN: 09638695. DOI: 10.1016/j.ndteint.2019.102152.
- [27] X. Qiu, P. Zhang, J. Wei, X. Cui, C. Wei e L. Liu, “Defect classification by pulsed eddy current technique in con-casting slabs based on spectrum analysis and wavelet decomposition,” *Sensors and Actuators A: Physical*, v. 203, pp. 272–281, dez. de 2013, ISSN: 0924-4247. DOI: 10.1016/J.SNA.2013.09.004.
- [28] Y. Xu, Y. Yang e Y. Wu, “Eddy Current Testing of Metal Cracks Using Spin Hall Magnetoresistance Sensor and Machine Learning,” *IEEE Sensors Journal*, v. 20, n. 18, pp. 10 502–10 510, 2020.
- [29] Y. Liu, S. Liu, H. Liu, C. Mandache e Z. Liu, “Pulsed Eddy Current Data Analysis for the Characterization of the Second-Layer Discontinuities,” *Journal of Nondestructive Evaluation*, v. 38, 1 2019, ISSN: 1573-4862. DOI: 10.1007/s10921-018-0545-6.

- [30] X. Fu, C. Zhang, X. Peng, L. Jian e Z. Liu, “Towards end-to-end pulsed eddy current classification and regression with CNN,” em *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2019, pp. 1–5.
- [31] F. Nafiah, A. Sophian, M. R. Khan e I. M. Zainal Abidin, “Quantitative evaluation of crack depths and angles for pulsed eddy current non-destructive testing,” *NDT & E International*, v. 102, pp. 180–188, mar. de 2019, ISSN: 09638695. DOI: 10.1016/j.ndteint.2018.11.019.
- [32] *Lift, Corrosion assessment redefined*, Acessado: 14 dezembro 2020. endereço: <https://www.eddyfi.com/>.
- [33] *PECT instrument*, Acessado: 14 dezembro 2020. endereço: <https://www.maxwellndt.com/>.
- [34] *Eddy Current Solutions*, Acessado: 14 dezembro 2020. endereço: <https://ethernde.com/>.
- [35] D. Silva-Palacios, C. Ferri e M. J. Ramírez-Quintana, “Improving Performance of Multiclass Classification by Inducing Class Hierarchies,” *Procedia Computer Science*, v. 108, pp. 1692–1701, 2017, ISSN: 18770509. DOI: 10.1016/j.procs.2017.05.218.
- [36] L. I. Kuncheva e E. Alpaydin, *Combining Pattern Classifiers: Methods and Algorithms*, 3. 2007, vol. 18, pp. 964–964, ISBN: 9786468600. DOI: 10.1109/tnn.2007.897478.
- [37] S. Haykin, *Neural Networks and Learning Machines*. 2009, pp. 1–46, ISBN: 9780131471399.
- [38] I. Goodfellow, Y. Bengio e A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [39] S. Haykin, *Neural Network and Machine Learning*, 3^a ed. Pearson, 2009.
- [40] K. Hornik, M. Stinchcombe, H. White et al., “Multilayer feedforward networks are universal approximators,” *Neural networks*, v. 2, n. 5, pp. 359–366, 1989.
- [41] I. Goodfellow, Y. Bengio, A. Courville e Y. Bengio, *Deep learning*, 2. MIT press Cambridge, 2016, vol. 1, p. 800, ISBN: 978-0262035613.
- [42] D. P. Kingma e J. L. Ba, “Adam: A method for stochastic optimization,” em *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Y. Bengio e Y. LeCun, ed., San Diego, CA, USA, 2015, pp. 1–15. arXiv: 1412.6980.

- [43] A. Thakkar, D. Mungra e A. Agrawal, “Sentiment analysis: an empirical comparison between various training algorithms for artificial neural network,” *International Journal of Innovative Computing and Applications*, v. 11, n. 1, p. 9, 2020, ISSN: 1751-648X. DOI: 10.1504/IJICA.2020.105315.
- [44] M. Riedmiller e H. Braun, “A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm,” *International Conference on Neural Networks*, pp. 586–591, 1993.
- [45] E. Assareh, M. A. Behrang, M. R. Assari e A. Ghanbarzadeh, “Application of PSO (particle swarm optimization) and GA (genetic algorithm) techniques on demand estimation of oil in Iran,” *Energy*, v. 35, n. 12, pp. 5223–5229, 2010, ISSN: 03605442. DOI: 10.1016/j.energy.2010.07.043.
- [46] L. Wang, Y. Zhuang, Z. Chen e Y. Zhong, “An enhanced cuckoo search using dimension selection,” *International Journal of Innovative Computing and Applications*, v. 10, n. 1, pp. 3–11, 2019, ISSN: 1751-6498. DOI: 10.1504/IJICA.2019.100512.
- [47] J. Wang e G. Liu, “A novel firefly algorithm with self-adaptive step strategy,” *International Journal of Innovative Computing and Applications*, v. 10, n. 1, pp. 18–26, 2019, ISSN: 1751-6498. DOI: 10.1504/IJICA.2019.100525.
- [48] X. Xiong e J. Tang, “selection,” *International Journal of Innovative Computing and Applications*, v. 10, n. 1, pp. 12–17, 2019, ISSN: 1751-6498. DOI: 10.1504/IJICA.2019.100526.
- [49] J. Kennedy e R. Eberhart, “Particle Swarm Optimization,” em *Proceedings of ICNN’95 - International Conference on Neural Networks*, IEEE, vol. 4, ago. de 1995, pp. 1942–1948, ISBN: 0-7803-2768-3. DOI: 10.1109/ICNN.1995.488968.
- [50] E. E. P. Souza, E. F. Simas Filho, P. C. M. A. Farias e J. M. Seixas, “Otimização do Sistema Neural de Seleção Online de Eventos num Detector de Partículas através do Processamento Estatístico de Sinais,” em *Anais do 11 Congresso Brasileiro de Inteligência Computacional*, A. d. P. Braga e C. J. A. Bastos Filho, ed., Porto de Galinhas, PE: SBIC, 2013, pp. 1–6. DOI: 10.21528/CBIC2013-133.
- [51] A. Gaspar-Cunha, R. Takahashi e C. Antunes, *Manual de computação evolutiva e metaheurística*. Imprensa da Universidade de Coimbra / Coimbra University Press, 2012, ISBN: 9789892601502.
- [52] B. P. Lathi, *Linear systems and signals*, 2^a ed. Oxford University Press New York, 2005.

- [53] A. Ukil e A. Barlocher, “Implementation of discrete wavelet transform for embedded applications using tms320vc5510,” em *2007 International Symposium on Industrial Embedded Systems*, IEEE, 2007, pp. 357–360.
- [54] R. C. Guido, “A note on a practical relationship between filter coefficients and scaling and wavelet functions of Discrete Wavelet Transforms,” *Applied Mathematics Letters*, v. 24, n. 7, pp. 1257–1259, 2011, ISSN: 0893-9659. DOI: <https://doi.org/10.1016/j.aml.2011.02.018>.
- [55] G. Strang e T. Nguyen, *Wavelets and filter banks*. SIAM, 1996.
- [56] V. Martin e K. Jelena, “Wavelets and Subband Coding,” *Ed. Prentice Hall, Englewood Cliffs, New Jersey-1995*, 1995.
- [57] M. Messias Silva Júnior, “Sistema de Auxílio ao Diagnóstico para Inspeções por ultrassom Implementado num Processador Digital de Sinais,” Universidade Federal da Bahia (UFBA), 2015.
- [58] D. Sundararajan, *Discrete wavelet transform: a signal processing approach*. John Wiley & Sons, 2016.
- [59] A. Tharwat, “Principal component analysis-a tutorial,” *International Journal of Applied Pattern Recognition*, v. 3, n. 3, pp. 197–240, 2016.
- [60] J. Kennedy e R. Eberhart, “Particle Swarm Optimization,” em *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 1995, pp. 1942–1948, ISBN: 9781424441211. DOI: 10.1109/IEMBS.2011.6091617.
- [61] J.-F. Chen, Q. H. Do e H.-N. Hsieh, “Training artificial neural networks by a hybrid PSO-CS Algorithm,” *Algorithms*, v. 8, n. 2, pp. 292–308, 2015, ISSN: 19994893. DOI: 10.3390/a8020292.
- [62] M. Abdullah, A. Bakar, N. Rahim, H. Mokhlis, H. Illias e J. Jamian, “Modified Particle Swarm Optimization with Time Varying Acceleration Coefficients for Economic Load Dispatch with Generator Constraints,” *Journal of Electrical Engineering and Technology*, v. 9, n. 1, pp. 15–26, jan. de 2014, ISSN: 1975-0102. DOI: 10.5370/JEET.2014.9.1.015.
- [63] M. Spichakova, “Gravitatsioonilist vastasmõju arvestav osakeste parvega optimeerimise meetod,” *Proceedings of the Estonian Academy of Sciences*, v. 65, n. 1, pp. 15–27, 2016, ISSN: 17367530. DOI: 10.3176/proc.2016.1.01.
- [64] P. H. Chen, “Two-level hierarchical approach to unit commitment using expert system and elite PSO,” *IEEE Transactions on Power Systems*, v. 27, n. 2, pp. 780–789, 2012, ISSN: 08858950. DOI: 10.1109/TPWRS.2011.2171197.

- [65] M. Li, X. Huang, H. Liu, B. Liu e Y. Wu, "Prediction of the gas solubility in polymers by a radial basis function neural network based on chaotic self-adaptive particle swarm optimization and a clustering method," *Journal of Applied Polymer Science*, v. 130, n. 5, pp. 3825–3832, 2013, ISSN: 00218995. DOI: 10.1002/app.39525.
- [66] F. Z. Azayite e S. Achchab, "A hybrid neural network model based on improved PSO and SA for bankruptcy prediction," *International Journal of Computer Science Issues*, v. 16, n. 1, pp. 1–13, jul. de 2019. arXiv: 1907.12179. endereço: <http://arxiv.org/abs/1907.12179>.
- [67] A. R. Jordehi, "A review on constraint handling strategies in particle swarm optimisation," *Neural Computing and Applications*, v. 26, n. 6, pp. 1265–1275, 2015, ISSN: 09410643. DOI: 10.1007/s00521-014-1808-5.
- [68] C. Pornsing, K. Saichareon e T. Karot, "A Modified Particle Swarm Optimization For EPornsing, C., Saichareon, K., & Karot, T. (2015). A Modified Particle Swarm Optimization For Engineering Constrained Optimization Problems, 3(1), 1–5. engineering Constrained Optimization Problems," v. 3, n. 1, pp. 1–5, 2015.
- [69] Y. Sun, Y. Gao e X. Shi, "Chaotic multi-objective particle swarm optimization algorithm incorporating clone immunity," *Mathematics*, v. 7, n. 2, pp. 1–16, 2019, ISSN: 22277390. DOI: 10.3390/math7020146.
- [70] V. Palakonda e R. Mallipeddi, "Pareto Dominance-Based Algorithms with Ranking Methods for Many-Objective Optimization," *IEEE Access*, v. 5, n. October, pp. 11 043–11 053, 2017, ISSN: 21693536. DOI: 10.1109/ACCESS.2017.2716779.
- [71] J. Aghaei, K. M. Muttaqi, A. Azizivahed e M. Gitizadeh, "Distribution expansion planning considering reliability and security of energy using modified PSO (Particle Swarm Optimization) algorithm," *Energy*, v. 65, pp. 398–411, 2014, ISSN: 03605442. DOI: 10.1016/j.energy.2013.10.082.
- [72] P. Hu, L. I. Rong, C. A. O. Liang-lin e L. I. Li-xian, "Procedia Engineering Multiple Swarms Multi-objective Particle Swarm Optimization Based on Decomposition," v. 15, pp. 3371–3375, 2011, ISSN: 1877-7058. DOI: 10.1016/j.proeng.2011.08.632.
- [73] M. Bošković, M. Rapaić e Z. Jeličić, "Particle Swarm Optimization of PID Controller under Constraints on Performance and Robustness," *Ijeec - International Journal of Electrical Engineering and Computing*, v. 2, n. 1, 2018, ISSN: 2566-3682. DOI: 10.7251/ijeec1801001b.

- [74] M. H. Hmidi, I. Ben Salem e L. El Amraoui, "Design and simulation of a 2DOF PID controller based on particle swarm optimization algorithms for a thermal phase of hybrid vehicle," *International Journal of Engineering & Technology*, v. 7, n. 4, p. 4644, 2018. DOI: 10.14419/ijet.v7i2.8428.
- [75] E. S. Alaviyan Shahri, A. Alfi e J. A. Tenreiro Machado, "Fractional fixed-structure H_∞ controller design using Augmented Lagrangian Particle Swarm Optimization with Fractional Order Velocity," *Applied Soft Computing Journal*, v. 77, pp. 688–695, 2019, ISSN: 15684946. DOI: 10.1016/j.asoc.2019.01.037.
- [76] S. Bouallegue, J. Haggege e M. Benrejeb, "A New Method for Tuning PID-Type Fuzzy Controllers Using Particle Swarm Optimization," *Fuzzy Controllers- Recent Advances in Theory and Applications*, 2012. DOI: 10.5772/48305.
- [77] N. Mohammadi e S. J. Mirabedini, "Comparison Of Particle Swarm Optimization And Backpropagation Algorithms For Training Feedforward Neural Network," *Journal of Mathematics and Computer Science*, v. 12, n. 02, pp. 113–123, 2014, ISSN: 2008949X. DOI: 10.22436/jmcs.012.02.03.
- [78] S. Toghyani, M. H. Ahmadi, A. Kasaeian e A. H. Mohammadi, "Artificial neural network, ANN-PSO and ANN-ICA for modelling the Stirling engine," *International Journal of Ambient Energy*, v. 37, n. 5, pp. 456–468, 2016, ISSN: 21628246. DOI: 10.1080/01430750.2014.986289.
- [79] S. Chatterjee, S. Sarkar, S. Hore, N. Dey, A. S. Ashour e V. E. Balas, "Particle swarm optimization trained neural network for structural failure prediction of multistoried RC buildings," *Neural Computing and Applications*, v. 28, n. 8, pp. 2005–2016, 2017, ISSN: 14333058. DOI: 10.1007/s00521-016-2190-2.
- [80] K. JP e S. SS, "Training of Multilayer Perceptrons with Improved Particle Swarm Optimization for the Heart Diseases Prediction," *International Journal of Swarm Intelligence and Evolutionary Computation*, v. 06, n. 02, pp. 1–8, 2017, ISSN: 20904908. DOI: 10.4172/2090-4908.1000156.
- [81] A. Nandi e N. D. Jana, "Accuracy Improvement of Neural Network Training using Particle Swarm Optimization and its Stability Analysis for Classification," mai. de 2019. arXiv: 1905.04522. endereço: <http://arxiv.org/abs/1905.04522>.

- [82] A. Suresh, K. V. Harish e N. Radhika, “Particle Swarm Optimization over back propagation neural network for length of stay prediction,” *Procedia Computer Science*, v. 46, n. Ict 2014, pp. 268–275, 2015, ISSN: 18770509. DOI: 10.1016/j.procs.2015.02.020.
- [83] B. A. Garro e R. A. Vázquez, “Designing Artificial Neural Networks Using Particle Swarm Optimization Algorithms,” *Computational Intelligence and Neuroscience*, v. 2015, pp. 1–20, jun. de 2015, ISSN: 16875273. DOI: 10.1155/2015/369298.
- [84] E. Bas, V. R. Uslu e E. Egrioglu, “Robust learning algorithm for multiplicative neuron model artificial neural networks,” *Expert Systems with Applications*, v. 56, n. 1, pp. 80–88, 2016, ISSN: 09574174. DOI: 10.1016/j.eswa.2016.02.051.
- [85] S. Mirjalili, S. Z. Mohd Hashim e H. Moradian Sardroudi, “Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm,” *Applied Mathematics and Computation*, v. 218, n. 22, pp. 11 125–11 137, 2012, ISSN: 00963003. DOI: 10.1016/j.amc.2012.04.069.
- [86] P. Barmapalexis, A. Karagianni, G. Karasavvaides e K. Kachrimanis, “Comparison of multi-linear regression, particle swarm optimization artificial neural networks and genetic programming in the development of mini-tablets,” *International Journal of Pharmaceutics*, v. 551, n. 1-2, pp. 166–176, 2018, ISSN: 18733476. DOI: 10.1016/j.ijpharm.2018.09.026.
- [87] R. Chai, S. H. Ling, G. P. Hunter, Y. Tran e H. T. Nguyen, “Brain–Computer Interface Classifier for Wheelchair Commands Using Neural Network With Fuzzy Particle Swarm Optimization,” *IEEE Journal of Biomedical and Health Informatics*, v. 18, n. 5, pp. 1614–1624, set. de 2014, ISSN: 2168-2194. DOI: 10.1109/JBHI.2013.2295006.
- [88] N. M. Nawi, A. S. Hussein, N. A. Samsudin, N. A. Hamid, M. A. M. Yunus e M. F. A. Aziz, “The effect of pre-processing techniques and optimal parameters selection on back propagation neural networks,” *International Journal on Advanced Science, Engineering and Information Technology*, v. 7, n. 3, pp. 770–777, 2017, ISSN: 24606952. DOI: 10.18517/ijaseit.7.3.2074.
- [89] A. Bilge e A. Yargıç, “Improving Accuracy of Multi-Criteria Collaborative Filtering by Normalizing User Ratings,” *Anadolu University Journal of Science and Technology A - Applied Sciences and Engineering*, pp. 225–237, 2017, ISSN: 1302-3160. DOI: 10.18038/aubtda.273802.
- [90] H. Oliveira, “Análise de fourier e wavelets: sinais estacionários e não estacionários,” *Editora Universitária da UFPE*, 2007.

- [91] M. Carvalho e T. B. Ludermir, “Particle swarm optimization of neural network architectures and weights,” *Proceedings - 7th International Conference on Hybrid Intelligent Systems, HIS 2007*, pp. 336–339, 2007. DOI: 10.1109/ICHIS.2007.4344074.
- [92] P. Dangeti, *Statistics for Machine Learning*. Birmingham: Packt Publishing, 2017, p. 425, ISBN: 978-1-78829-575-8.
- [93] D. Berrar, “Cross-Validation,” em *Encyclopedia of Bioinformatics and Computational Biology*, Elsevier, 2019, pp. 542–545. DOI: 10.1016/b978-0-12-809633-8.20349-x.
- [94] R. O. Duda, P. E. Hart e D. G. Stork, *Pattern classification*, 2nd. Wiley, 2001, p. 738, ISBN: 9780471056690.
- [95] M. Messias Silva Júnior, E. F. S. Filho, P. C. M. A. Farias, M. C. S. Albuquerque, I. Silva e C. T. T. Farias, “SISTEMA DE CLASSIFICAÇÃO NEURAL UTILIZANDO A TÉCNICA DE PEC NA DETECÇÃO DE DEFEITOS EM MATERIAIS REVESTIDOS COM COM-PÓSITOS,” ago. de 2018, pp. 1–10.
- [96] D. Li, A. Cong e S. Guo, “Sewer damage detection from imbalanced CCTV inspection data using deep convolutional neural networks with hierarchical classification,” *Automation in Construction*, v. 101, pp. 199–208, 2019, ISSN: 0926-5805. DOI: <https://doi.org/10.1016/j.autcon.2019.01.017>. endereço: <http://www.sciencedirect.com/science/article/pii/S0926580518306174>.

Apêndice A

Trabalhos Publicados

Os artigos científicos relacionados a esta dissertação estão listados abaixo em ordem cronológica. Os artigos foram separados pela forma de publicação (congresso ou revista). O título e resumo foram escritos no idioma de publicação.

A.1 Artigos aprovados em congressos

1 - Silva, G. F, et al. **A Constrained Neural Classifier for Pulsed Eddy Current based Flaw Detection in Industrial Pipes**. 26th IEEE International Conference on Electronics Circuits and Systems (ICECS). Genova - Italy, November, 27 - 29, 2019.

Observação: Este artigo, apesar de aceito, não foi apresentado por falta de recursos financeiros.

Abstract:

Decision support systems are important to improve the efficiency of non-destructive evaluation, specially for industrial equipment. Pulsed eddy-current is a magnetic method used for evaluation of metallic equipment. In this paper, is proposed the combination of pulsed eddy current evaluation, digital signal processing, and neural networks to detect flaws in industrial pipes. A novel method using particle swarm optimization is proposed for imposing performance constraints during neural classifier training process. Experimental results are used to validate the proposed method by comparing it to

neural networks trained from the traditional back-propagation algorithm.

2 - Silva, G. F, et al. **A Constrained Neural Classifier for Pulsed Eddy Current based Flaw Detection in Industrial Pipes.** XIV Brazilian Congress of Computational Intelligence (CBIC). Belém - Brazil, November, 3-6, 2019.

Abstract:

Decision support systems are important to improve the efficiency of non-destructive evaluation, specially for industrial equipment. Pulsed eddy-current is a magnetic method used for evaluation of metallic equipment. In this paper, is proposed the combination of pulsed eddy current evaluation, digital signal processing, and neural networks to detect flaws in industrial pipes. A novel method using particle swarm optimization is proposed for imposing performance constraints during neural classifier training process. Results obtained for experimental signals acquired from composite-insulated metallic industrial pipes presenting internal and external corrosion areas are used to validate the proposed method. A comparison to neural networks trained from the traditional back-propagation algorithm was presented.

A.2 Artigos aprovados em revistas

1 - Silva, G. F, et al. **Constrained Neural Classifier Training Method for Flaw Detection in Industrial Pipes Using Particle Swarm Optimization.** International Journal of Innovative Computing and Applications. Até a finalização desta dissertação o artigo não foi publicado.

Abstract:

A novel method for constrained training of multi-class artificial neural network classifiers is proposed in this work. The traditional training procedure is usually based on mean square error minimization and thus, all classes of interest are considered as having the same relevance for system performance. This is not always the case for real-world applications in which the class relevance may be unbalanced. In this paper, cost functions designed to introduce classification performance constraints for specific classes

are presented and particle swarm optimization is used as global optimization method. The proposed method is applied to a non-destructive evaluation decision support problem using pulsed eddy currents signals. Experimental results obtained from thermally insulated industrial pipes indicate the efficiency of the proposed method in comparison to neural networks trained from the traditional back-propagation algorithm.

Apêndice B

Código utilizado neste trabalho

O método proposto neste trabalho foi desenvolvido com a linguagem de programação Python 3. Os arquivos com código fonte podem ser consultados no repositório público <https://gitlab.com/gilvan.farias/redes_neurais>.