

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Programa de Pós-Graduação em Engenharia Mecânica

Laís Marino Brandão Pinto

**REDES NEURAS ARTIFICIAIS PARA O MONITORAMENTO DO ACÚMULO DE
DANOS EM ATIVOS**

Belo Horizonte

2019

Laís Marino Brandão Pinto

**REDES NEURAIS ARTIFICIAIS PARA O MONITORAMENTO DO ACÚMULO DE
DANOS EM ATIVOS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Mecânica da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Mestre em Engenharia Mecânica.

Orientador: Prof. Dr. Jánes Landre Jr.

Área de concentração: Projeto e Fabricação

Belo Horizonte

2019

FICHA CATALOGRÁFICA

Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

P659r	<p>Pinto, Laís Marino Brandão Redes neurais artificiais para o monitoramento do acúmulo de danos em ativos / Laís Marino Brandão Pinto. Belo Horizonte, 2019. 114 f. : il.</p> <p>Orientador: Jánes Landre Júnior Dissertação (Mestrado) – Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Engenharia Mecânica</p> <p>1. Redes neurais (Computação). 2. Inteligência artificial. 3. Banco de dados. 4. Máquinas de mineração - Manutenção. 5. Localização de falhas (Engenharia). 6. Confiabilidade (Engenharia). I. Landre Junior, Jánes. II. Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Engenharia Mecânica. III. Título.</p>
-------	--

SIB PUC MINAS

CDU: 681.3.091

Ficha catalográfica elaborada por Fabiana Marques de Souza e Silva - CRB 6/2086

Laís Marino Brandão Pinto

**REDES NEURAIS ARTIFICIAIS PARA O MONITORAMENTO DO ACÚMULO DE
DANOS EM ATIVOS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Mecânica da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Mestre em Engenharia Mecânica.

Área de concentração: Projeto e Fabricação

Prof. Dr. Jánes Landre Júnior – PUC Minas (Orientador)

Prof. Dr. Claysson Bruno Santos Vimieiro – PUC Minas (Banca Examinadora)

Prof. Dr. Luis Henrique Andrade Maia – PUC Minas (Banca Examinadora)

Prof. Dr. Fernando Helder Teixeira - VALE (Banca Examinadora)

Belo Horizonte, 28 de Fevereiro de 2019

AGRADECIMENTO

Tantas pessoas fizeram parte dessa jornada!

Em primeiro lugar agradeço à professora Rose pelo suporte e pela recomendação, que me abriram este caminho. Agradeço ao professor Jánes pela oportunidade, pela orientação e por ensinar tanto, não apenas sobre a Engenharia Mecânica, mas também sobre a vida profissional e o equilíbrio com a vida pessoal.

Obrigada a todos do grupo PUC e VALE, sobretudo ao Lucas e ao Ronei, pela parceria e pelo trabalho em conjunto, ao fundir academia e indústria, possibilitando novos olhares de diferentes perspectivas sobre cada aspecto estudado.

Agradeço à minha família – em especial ao meu irmão Sólon, à minha mãe Gislene e à Fefê – e também ao Guilherme e à Lena pelo amparo nos momentos de estresse e correção do texto nos mínimos detalhes.

Obrigada aos amigos, à professora Renata e a todas as pessoas que contribuíram de alguma forma ao longo desta etapa.

Agradeço a todos pelo carinho!

“The greatest enemy of knowledge is not ignorance, it is the illusion of knowledge.”

(Stephen Hawking)

RESUMO

Esta pesquisa surgiu da necessidade de se entender os problemas estruturais de equipamentos portuários voltados para a mineração, dada a importância desta atividade, pois os minérios são elementos essenciais para a produção de inúmeros bens de consumo e a indústria da mineração é responsável por 4% do PIB no Brasil. Os equipamentos são robustos e de custo elevado para manutenção. Portanto, faz-se necessário o acompanhamento do desgaste dos ativos a partir do sensoriamento, inspeções e monitoramento de danos, tendo em vista o aumento da confiabilidade e previsões confiáveis, já que atualmente há a tendência de grandes empresas no Brasil começarem a implementação da Indústria 4.0, que pode contar com a utilização de técnicas de inteligência artificial para diagnosticar e prever falhas. A metodologia compreendeu a definição do objeto de estudo, entendimento do funcionamento do ativo e o desenvolvimento dos modelos de Redes Neurais Artificiais para classificação de falhas, obtenção do nível de criticidade de falhas e previsão temporal de parada do ativo. Foi utilizado o banco de dados real de rolamentos e também do motor do sistema Alimentador de um Virador de Vagões, a fim de alimentar para treinamento e teste os modelos de rede propostos. A modelagem da rede abarca o treinamento de diversas topologias (*Cross-validation*) com a finalidade de compará-las quanto à capacidade de generalização e previsão e, em seguida foi calculado o intervalo de confiança da rede pelo método da Máxima Verossimilhança. Foi encontrada dificuldade na obtenção de um banco de dados confiável e completo para aplicação nas redes com todos os objetivos propostos. Entretanto, as redes Classificadora e Aproximadora apresentaram bons resultados com o banco de dados de rolamentos, com 90% e 81,42% de acerto, respectivamente, bem como a rede para previsão temporal, cuja aproximação entre resultado desejado e obtido chegou a mais de 90%. Este trabalho contribuiu com uma metodologia para monitoramento de danos em ativos por Redes Neurais Artificiais, passível de replicação para os diversos equipamentos, podendo auxiliar o departamento de manutenção, confiabilidade e operação no planejamento das ações.

Palavras-chave: Redes Neurais Artificiais. Inteligência Artificial. Máquinas de mineração. Manutenção. Confiabilidade.

ABSTRACT

This research arose from the need to understand the structural problems of port equipment aimed at mining, given the importance of this activity, since ores are essential elements for the production of many consumer goods and the mining industry accounts for 4% of the GDP in Brazil. The equipment is robust and costly for maintenance. Therefore, it is necessary to monitor the wear and tear of assets from the sensing, inspections and damage monitoring, in view of the increase of reliability and reliable forecasts, since currently there is the tendency of large companies in Brazil to begin the implementation of Industry 4.0, which can rely on the use of artificial intelligence techniques to diagnose and predict failures. The methodology included the definition of the object of study, understanding of the functioning of the asset and the development of the Artificial Neural Networks models to classify failures, obtain fault criticality level and temporal prediction of asset stop. We used the actual bearing database as well as the engine of the Wagon Turner Feeder system in order to feed training and test the proposed network models. The modeling of the network involves the training of several topologies (Cross-validation) in order to compare them with the generalization and predictability capacity, and then the network confidence interval was calculated by the Maximum Likelihood method. It was found difficulty in obtaining a reliable and complete database for application in the networks with all the proposed objectives. However, the Classifier and Approach networks presented good results with the bearing database, with 90% and 81.42% accuracy, respectively, as well as the network for temporal prediction, whose approximation between desired and obtained results reached more than 90%. This work contributed with a methodology for monitoring damage to assets by Artificial Neural Networks, which can be replicated for the various assets and can help the maintenance department, reliability moreover operation in the planning of actions.

Key words: Artificial Neural Networks. Artificial intelligence. Mining machines. Maintenance. Reliability.

LISTA DE FIGURAS

Figura 1 - Tipos de manutenção.....	41
Figura 2 - Mudança das técnicas de manutenção	43
Figura 3 - Padrões de falha em cada geração	43
Figura 4 - Diagrama em blocos do sistema nervoso	47
Figura 5 - Neurônio biológico	48
Figura 6 - Modelo não linear de um neurônio	48
Figura 7 - Principais funções de ativação de um neurônio	50
Figura 8 - Rede com um nó a) linear e b) não linear	52
Figura 9 - Rede de a) camada única e b) de múltiplas camadas	55
Figura 10 - Arquitetura de uma rede recorrente	56
Figura 11 - Arquitetura de uma rede PMC.....	56
Figura 12 - Domínios de definição e de operação de PMC para a) aproximação de funções e b) sistemas dinâmicos	57
Figura 13 - PMC com entradas atrasadas no tempo	58
Figura 14 - Sinais funcionais e de erro de uma rede PMC	60
Figura 15 - Curva do erro de treinamento a) com sobre-ajuste e b) normal	63
Figura 16 - a) Ajuste em excesso e b) boa generalização.....	63
Figura 17 - Fluxograma do curso de trabalho.....	69
Figura 18 - Fluxograma para modelagem das RNAs.....	73
Figura 19 - Operação das RNAs	74
Figura 20 - Evolução do MSE de treinamento e teste para rede Classificadora	80
Figura 21 - Evolução da taxa de acerto para rede Classificadora	80
Figura 22 - Evolução do MSE de treinamento e teste para rede Classificadora com Toolbox do Matlab®.....	81
Figura 23 - Evolução do MSE para rede Aproximadora	84
Figura 24 - Evolução da taxa de erro para rede Aproximadora	85
Figura 25 - Evolução da taxa de acerto para rede Aproximadora com Toolbox do Matlab®	86
Figura 26 - Virador de Vagões	87
Figura 27 - Evolução do erro relativo médio para diferentes tamanhos de janela deslizante para (a) AL1 e (b) AL2.....	95

Figura 28 - Evolução do MSE de teste para diferentes tamanhos de janela deslizante para (a) AL1 e (b) AL2.....	96
Figura 29 - Variáveis de rede com uma camada escondida	111

LISTA DE QUADROS

Quadro 1 - Evolução da manutenção ao longo das gerações	44
Quadro 2 - Esforços gastos no processo de exploração de dados.....	46
Quadro 3 - Comparação de desempenho entre neurônio artificial e biológico.....	49
Quadro 4 - Comparação entre modelo Perceptron e Adaline	54
Quadro 5 - Operação janela deslizante	59
Quadro 6 - Variáveis do motor dos Alimentadores	70
Quadro 7 - Correspondência entre variáveis de treinamento e resultados	88
Quadro 8 - Variáveis utilizadas no treinamento da RNA com 8 entradas variantes no tempo.....	89
Quadro 9 - Variáveis utilizadas no treinamento da RNA com 5 entradas variantes no tempo.....	90
Quadro 10 - Variáveis normalizadas utilizadas no treinamento da RNA com 5 entradas variantes no tempo.....	90
Quadro 11 - Modelos A e B de banco de dados para treinamento da RNA	98

LISTA DE TABELAS

Tabela 1 - Exemplos de amostras para rede Classificadora de Padrões.....	78
Tabela 2 - Melhores topologias da rede Classificadora de Padrões com uma camada intermediária	79
Tabela 3 - Melhores topologias da rede Classificadora de Padrões com duas camadas intermediárias	79
Tabela 4 - Exemplos de amostras para rede Aproximadora de Funções	82
Tabela 5 - Melhores topologias da rede Aproximadora de Funções com uma camada intermediária	83
Tabela 6 - Melhores topologias da rede Aproximadora de Funções com duas camadas intermediárias	83
Tabela 7 - Resultado 1 - Alimentador 1 para RNA com 8 entradas e $np=3$	89
Tabela 8 - Resultado 2 - Alimentador 1 para RNA com 5 entradas e $np=3$	90
Tabela 9 - Resultado 3 - Alimentador 1 para RNA com 5 entradas normalizadas e $np=3$	91
Tabela 10 - Resultado 4 - Alimentador 2 para RNA com 5 entradas 3 $np=3$	91
Tabela 11 - Resultado 5 - Alimentador 2 para RNA com 5 entradas normalizadas e $np=3$	92
Tabela 12 - Tabela de t-student	92
Tabela 13 - Comparação entre os Resultados 1 a 5 das RNAs com entradas variantes no tempo com $np=3$	93
Tabela 14 - Resultado 6 - Alimentador 1 para RNA com 5 entradas normalizadas e $np=2$	93
Tabela 15 - Resultado 7 - Alimentador 1 para RNA com 5 entradas normalizadas e $np=4$	94
Tabela 16 - Resultado 8 - Alimentador 2 para RNA com 5 entradas normalizadas e $np=2$	94
Tabela 17 - Resultado 9 - Alimentador 2 para RNA com 5 entradas normalizadas e $np=4$	94
Tabela 18 - Comparação entre os Resultados 3, 5, 6, 7, 8 e 9 das RNAs com entradas variantes no tempo.....	95

LISTA DE ABREVIATURAS E SIGLAS

AL1	Alimentador 1
AL2	Alimentador 2
ALT	Teste de Vida Acelerado
ANFIS	Sistema de Inferência Adaptativo Neuro-fuzzy
CBM	<i>Condition Based Maintenance</i> (Manutenção Baseada em Condição)
CSALT	Teste de Vida Acelerado Constante
ETA	<i>Event Tree Analysis</i> (Análise da Árvore de Eventos)
F-FMEA	<i>Fuzzy Mode and Effects Analysis</i> (Análise de Modos de Falhas e Efeitos Fuzzy)
FFT	Transformada Rápida de Fourier
FMEA	<i>Failure Mode and Effect Analysis</i> (Análise de Modos de Falhas e Efeitos)
FNN	Rede Neuro-fuzzy
FTA	<i>Fault Tree Analysis</i> (Análise da Árvore de Falhas)
IA	Inteligência Artificial
IoE	<i>Internet of Everything</i> (Internet de Tudo)
IoP	<i>Internet of People</i> (Internet das Pessoas)
IoT	<i>Internet of Things</i> (Internet das Coisas)
LVQ	Rede de quantização vetorial por aprendizagem
MSE	<i>Mean squared error</i> (Erro quadrático médio)
NSA	Algoritmo de Seleção Negativa (<i>Negative Selection Algorithm</i>)
PCBM	<i>Probabilistic Condition Based Maintenance</i>
PLC	Controlador Lógico Programável (<i>Programmable Logic Controller</i>)
PMC	<i>Perceptron</i> de múltiplas camadas
RBF	Redes de funções de base radial
RCM	<i>Reliability-centered Maintenance</i> (Manutenção Centrada na confiabilidade)
RNA	Rede Neural Artificial
SMC	Simulação de Monte Carlo
VV	Virador de Vagão

LISTA DE SÍMBOLOS

x	Entrada da RNA
k	Neurônio da RNA
w_{kj}	Peso sináptico entre entrada j e neurônio k da RNA
y	Saída (resultado) da RNA
β	Constante referente à inclinação das funções de ativação do neurônio
c	Centro da função gaussiana
σ	Desvio padrão
d	Saída (valor desejado) da RNA
θ	Limiar do neurônio
η	Taxa de aprendizagem da RNA
E	Erro entre resultado e valor desejado da RNA
δ	Taxa de momentum da RNA
n	Número de entradas da RNA
n_1	Número de neurônios na camada intermediária da RNA
n_2	Número de neurônios na camada de saída da RNA

SUMÁRIO

1	INTRODUÇÃO	27
1.1	Objetivos	28
1.2	Justificativa	29
1.3	Escopo da dissertação	30
2	REVISÃO BIBLIOGRÁFICA	31
2.1	Estado da arte	31
2.1.1	<i>Síntese do estado da arte</i>	39
2.2	Manutenção	40
2.2.1	<i>Tipos de manutenção</i>	41
2.2.2	<i>História da manutenção</i>	42
2.3	Indústria 4.0	44
2.4	Processo de exploração de dados	45
2.5	Redes neurais artificiais	47
2.5.1	<i>Neurônio</i>	47
2.5.2	<i>Tipos de função de ativação</i>	49
2.5.3	<i>Primeiras modelagens neurais e treinamentos</i>	52
2.5.3.1	Perceptron	52
2.5.3.2	Adaline	53
2.5.3.3	Comparação entre modelos neurais Perceptron e Adaline	54
2.5.4	<i>Arquiteturas de redes neurais artificiais</i>	54
2.5.5	<i>Perceptron de múltiplas camadas (PMC)</i>	56
2.5.6	<i>Aplicação da rede PMC em sistemas dinâmicos</i>	57
2.5.6.1	PMC com entradas atrasadas no tempo	58
2.5.7	<i>Técnicas de treinamento</i>	59
2.5.7.1	Backpropagation	60
2.5.7.2	Momentum	62
2.5.8	<i>Desempenho das RNAs</i>	62
2.5.9	<i>Divisão dos dados para treinamento</i>	64
2.5.10	<i>Escolha da rede neural</i>	65
2.5.11	<i>Intervalo de confiança de uma Rede Neural Artificial</i>	67
3	METODOLOGIA	69
3.1	Etapas do trabalho	69
3.2	Banco de dados do motor do Alimentador do Virador de Vagões	70

3.3	Modelos das RNAs	71
3.4	Metodologia de implementação das RNAs.....	72
3.5	Metodologia de implementação do intervalo de confiança.....	75
4	RESULTADOS	77
4.1	Rede neural artificial PMC programada	77
<i>4.1.1</i>	<i>Rede Classificadora de Padrões.....</i>	<i>77</i>
<i>4.1.2</i>	<i>Rede Aproximadora de Funções</i>	<i>82</i>
4.2	Definição do objeto de estudo e condições de operação	86
4.3	Banco de dados do objeto de estudo	87
4.4	Banco de dados aplicado na rede com entradas variantes no tempo..	88
4.5	Banco de dados aplicado nas redes Classificadora e Aproximadora ..	96
	CONCLUSÃO	101
	REFERÊNCIAS.....	105
	APÊNDICE I.....	111

1 INTRODUÇÃO

Os minérios são elementos essenciais para a produção de inúmeros bens de consumo, como celulares, carros, eletrodomésticos, e na construção civil. Fazem parte dessa gama de minérios: carvão, cobalto, cobre, ferroligas, minério de ferro e pelotas. Nesta indústria de mineração, é crucial que uma grande mineradora mantenha a distribuição da produção a preços competitivos em vários continentes. Presente em todo o território brasileiro, a mineração faz parte da atividade da indústria extrativa e é responsável por 4% do PIB no Brasil (DNPM, 2016). Esta atividade é responsável pela atração de investidores, por elevar o PIB das cidades que fazem parte do processo produtivo, com minas de extração, usinas de beneficiamento, pelotizadoras e portos.

Na atividade de mineração, são necessários ativos robustos e caros, como Recuperadoras, Empilhadeiras, Viradores de Vagão (VVs), e Carregadores de Navio, responsáveis pelo processo de produção desde a extração e estocagem até a sua mistura e distribuição. Para que o processo de produção ocorra de forma ideal, os ativos devem estar disponíveis para uso sempre que houver demanda. Entretanto, à medida que o tempo passa e que os ciclos de operação vão sendo realizados, os ativos sofrem deteriorações, que podem interferir no seu bom funcionamento e na sua disponibilidade. O desgaste dos ativos é influenciado pelo tipo de carregamento, como variações no tamanho do grão, grau de compactação e umidade do minério. Por esta razão, além dos sensores já instalados, responsáveis pela medição de variáveis relevantes para o bom funcionamento do ativo, são feitas inspeções para detecção de danos nos componentes do ativo, a fim de que a manutenção possa ser feita corretamente, uma vez que essas avarias podem comprometer sistemas e, conseqüentemente, o funcionamento de todo o ativo.

Os equipamentos utilizados na mineração são compostos de vários sistemas e subsistemas que operam em conjunto. Por isso, a falha em um componente pode afetar o funcionamento adequado do equipamento. A inspeção cotidiana dos ativos é feita predominantemente de forma visual pelos inspetores, sendo o dano detectado em estado mais avançado. Dessa maneira, o tempo hábil para a realização da manutenção preventiva é reduzido, considerando-se ainda a necessidade de manter o ativo operando, o que minimiza o tempo disponível para manutenção.

Atualmente, grandes empresas têm feito o monitoramento e o diagnóstico de falhas para redução de custos e aumento da confiabilidade, segurança e melhoria na manutenção. Este monitoramento é necessário para que se possa aproximar de previsões mais exatas e confiáveis, prolongar a atividade do ativo, e para que a manutenção possa ser planejada e efetuada antes que uma falha catastrófica ocorra.

A tendência do controle da manutenção por monitoramento e diagnóstico de falhas das empresas pode ser inserida no conceito da Indústria 4.0, definida por Schmidtke et al. (2018) como a implementação de diversas inovações tecnológicas e constante desenvolvimento de redes inteligentes para atender às novas demandas do mercado. O desenvolvimento no sentido dessa nova revolução é apontado pelos autores como diretamente dependente da digitalização de toda a informação (*big data*), que deve ser gerada de forma correta e criteriosa para que possa garantir a eficiência da análise e controle que será feito pelas novas redes inteligentes.

Neste contexto, as técnicas de Inteligência Artificial (IA) em sistemas automáticos apresentam-se apropriadas para cumprir com essas finalidades: classificações das condições de operação do ativo podem ser realizadas, com o diagnóstico dependendo da interpretação dos dados por um especialista ou *software*. Além disso, essas técnicas podem ser facilmente ampliadas e modificadas para aplicação em diversos ativos, tendo como base dados do histórico de falhas e suspensão do sensoriamento referente ao ativo.

1.1 Objetivos

Este trabalho tem como objetivo principal elaborar uma metodologia para o monitoramento do acúmulo de danos em equipamentos. Os objetivos específicos são:

- Identificar os modos de falha passíveis de classificação com base no banco de dados coletado e disponibilizado;
- Avaliar a aplicabilidade de redes neurais artificiais para monitoramento e estimativa da evolução dos danos nos ativos;
- Classificar a falha da máquina de acordo com dados que indicam o acúmulo de danos;
- Prever o dano acumulado no ativo utilizando redes neurais artificiais;

1.2 Justificativa

Hoje em dia há a propensão de implementação da Indústria 4.0 no Brasil, principalmente pelas grandes empresas, para atender às demandas de mercado, como o aumento da segurança dos processos e a obtenção de previsões assertivas que auxiliem o planejamento de tarefas. Um fator importante das fábricas inteligentes, segundo Man, Zhou (2018), é a capacidade de estimar a vida remanescente e fazer previsões das condições de vida de um equipamento.

Essa busca é explicitada pela gama de artigos publicados recentemente que tratam da necessidade da previsão de tempo de vida remanescente e diagnóstico de falhas em equipamentos. Diferentes metodologias de previsão foram aplicadas em sistema de softwares, sistemas eletrônicos, motores, dentre outros. Porém, na pesquisa não consta aplicação na indústria da mineração. A literatura sobre RNAs e suas variações para predição de vida remanescente é muito limitada e ainda menor para aplicações em cenários reais de teste e que realmente relatem o desempenho do modelo (Wu et al., 2017 e Sikorska et al., 2011 apud ELSHEIKH, YACOUT, OUALI, 2019).

Os equipamentos portuários utilizados na mineração são robustos e compostos de vários sistemas e subsistemas que operam em conjunto. Por isso, a falha em um componente pode afetar direta ou indiretamente o bom funcionamento do equipamento. A inspeção cotidiana dos ativos é feita predominantemente de forma visual pelos inspetores, sendo o dano detectado em estado mais avançado. Dessa forma, o tempo hábil para a manutenção preventiva é reduzido, considerando-se ainda a necessidade de manter o ativo operando, o que minimiza o tempo disponível para manutenção.

A partir de dados de monitoramento de componentes e subsistemas de ativos, pode ser feita a detecção e avaliação dos danos, desde o seu surgimento e durante a evolução de sua criticidade, como o aumento de sinais vibracionais e do tamanho de trincas, por exemplo, a fim de aprimorar a manutenção com antecedência e assertividade. As técnicas de Inteligência Artificial (IA), especificamente as Redes Neurais Artificiais (RNAs) podem ser poderosas para a classificação de padrões no monitoramento de dados em tarefas de diagnósticos e, conseqüentemente, uma ótima ferramenta para análise das respostas e previsão em sistemas não lineares, cujo

equacionamento do sistema mostra-se complexo, como uma função de, no mínimo, segunda ordem.

1.3 Escopo da dissertação

Este trabalho é composto de quatro capítulos. O Capítulo 2 contém, primeiramente, uma revisão bibliográfica referente ao monitoramento e tendência de falhas, e previsão de vida útil de ativos, incluindo abordagens de inteligência artificial, estatística, probabilidade e integridade estrutural. Em seguida, é apresentado o referencial teórico utilizado como base para este estudo, composto por conceitos relativos à manutenção – histórico e tipos –, noções da Indústria 4.0, características das redes neurais artificiais, no que diz respeito a arquitetura, seus tipos e treinamentos, e avaliação de seu desempenho. O Capítulo 3 contém a metodologia do estudo, com fluxogramas e descrição das etapas de trabalho, treinamento e operação das RNAs. Os resultados preliminares e discussões encontram-se no Capítulo 4. Por fim, é feita a conclusão deste estudo.

2 REVISÃO BIBLIOGRÁFICA

Nesta seção são apresentados estudos e trabalhos atuais que tratam de variados métodos para detecção e diagnóstico de modos de falha em ativos, bem como teóricos das metodologias e ferramentas a serem utilizadas.

2.1 Estado da arte

O método Análise de Modos de Falhas e Efeitos (FMEA, do inglês *Failure Mode and Effect Analysis*), é utilizado para análise de riscos, em que se relaciona um falha em determinado equipamento ou componente ao sistema. Esta técnica pode ser aplicada na detecção de falhas e seus potenciais efeitos e riscos associados, tendo como base o conhecimento na confiabilidade do equipamento, visando obter o domínio das falhas de componentes e seus efeitos no sistema. O FMEA pode ser muito eficaz para sistemas e modos de falha mais simples (KAERCHER; LUZ, 2016, MCDERMOTT, MIKULAK, BEAUREGARD, 2009).

Do mesmo modo, a técnica dos observadores de estado, da teoria de controle moderno, pode ser aplicada com o objetivo de detectar e diagnosticar falhas. De acordo com Souza et al. (2014), esta técnica representa um sistema por espaço de estados em forma matricial, cujas variáveis de entrada e saída do processo podem ser estimadas. Almeja-se que o erro de estimação em relação ao sistema real seja zero. Com isso, é possível comparar o sistema estimado e o real, de forma a localizar a existência de falhas no sistema real. A desvantagem deste método é a demanda de uma boa compreensão da modelagem do processo que se deseja controlar.

Ainda visando a detecção de falhas, as técnicas de IA podem ser uma boa alternativa aos métodos mais tradicionais, seja como complemento a eles ou utilizadas individualmente.

O Algoritmo de Seleção Negativa (NSA, do inglês *Negative Selection Algorithm*) é uma técnica de Inteligência Artificial baseada no sistema imunológico para gerar um conjunto de detectores que pode perceber anomalias em um sistema, a partir do reconhecimento de dados que são considerados próprios ou não-próprios (AMARAL, 2006).

A lógica *fuzzy* ou lógica difusa foi criada para resolver os problemas em que a natureza de cada conjunto modelado não é bem definida. Ao contrário da lógica

clássica – ou *crisp* –, cujos elementos são pertencentes ou não pertencentes a um dado conjunto, a lógica *fuzzy* considera que todos os elementos possuem um grau de pertinência (temperatura, vibração, emissão acústica etc.) que os classificam em um dado universo. Entretanto, a construção de um conjunto *fuzzy* depende da identificação do universo de discurso e da especificação de uma função de pertinência apropriada, que varia de pessoa para pessoa. Portanto, o conjunto *fuzzy* é subjetivo e não aleatório. Um modelo *fuzzy* possui variáveis de entrada e de saída – cujas funções de pertinência devem ser determinadas – e um conjunto de regras do tipo SE-ENTÃO, que relaciona, de forma linguística, o resultado de saída dadas as entradas (JANG, SUN, MIZUTANI, 1997).

As Redes Neurais Artificiais (RNAs) são, segundo Silva, Spatti e Flauzino (2010), um conjunto de unidades de processamento, com camadas de neurônios artificiais que conectam as entradas direta ou indiretamente até as saídas, e podem ser utilizadas como uma técnica para classificação de padrões. Para isto, é preciso fazer o treinamento da RNA a partir de um banco de dados, durante o qual os pesos das ligações entre os neurônios são atualizados. Após atualizados os parâmetros da rede, novos dados podem ser processados e as saídas estimadas. Dessa forma, as RNAs podem ter boa capacidade de aprendizado e habilidade de generalização.

O objetivo do estudo de Mohammadi e Tavakolan (2013) foi identificar modos de falha e seus impactos para o planejamento de ações corretivas. Em razão disto, os autores propuseram uma lógica *fuzzy* para determinar o peso dos riscos de impactos no tempo, custos, qualidade e segurança. A metodologia foi aplicada em um estudo de caso, no qual foram feitas 343 regras *fuzzy*, que englobaram todas as combinações de entradas, cujas funções de pertinência utilizadas foram a trapezoidal e a triangular – pela facilidade de manipulação algébrica – e a defuzzificação foi feita pelo método do centro de área. Eles concluíram que a lógica *fuzzy* atendeu ao objetivo de suprir as limitações do FMEA e apresentou resultados mais realistas, além de possibilitar a combinação de probabilidade, impacto e detecção para cada risco.

Os autores Pokoradi e Szamosi (2015) utilizaram o método baseado em regras de estimativa de confiabilidade Análise de Modos de Falhas e Efeitos *Fuzzy* (F-FMEA, do inglês *Fuzzy Mode and Effects Analysis*) para identificar e estimar falhas, visto que o método *fuzzy*, baseado em regras, investiga a intersubjetividade e as incertezas. O FMEA, método multicritério de estimativa de risco, utiliza cenários futuros para prever possíveis consequências de falhas. Entretanto, possui como desvantagem a

necessidade de dados estatísticos suficientes, o que nem sempre é viável. Do contrário, são usados parâmetros estimados, que trazem incertezas consigo. A proposta foi suprir esta limitação do FMEA e apresentou resultados satisfatórios para estimar o nível de criticidade de um sistema.

Utilizando o método de Monte Carlo, Gao et al. (2014) fizeram uma comparação com uma metodologia *fuzzy* dependente do tempo para obter um modelo matemático de confiabilidade para mecanismos baseados na degradação por força. Eles concluíram que o método de análise *fuzzy* pode efetivamente resolver problemas de confiabilidade de variáveis transientes, inclusive de forma melhor que pela teoria da confiabilidade clássica, ao considerar a dependência do tempo e a lógica *fuzzy* no cálculo da probabilidade de falha. Ao considerar a dependência do tempo e ao lidar com o erro cinemático e o desgaste através da lógica *fuzzy*, os resultados mostraram-se mais aproximados das condições reais de trabalho.

Para a detecção e localização de falhas em um modelo de plataforma sobre uma suspensão, Monte Alegre (2009) utilizou a metodologia dos observadores de estado, analisando o valor RMS dos erros entre os sinais dos sistemas sem falhas e com pequenas falhas. Concluiu-se que esta metodologia foi satisfatória para detectar falhas de ao menos 10% da ordem de grandeza com diferentes percentuais de falha e pode ser usada como uma forma de manutenção preditiva, visto que permite o acompanhamento do avanço da falha.

Lima et al. (2014) propuseram um sistema de monitoramento da integridade estrutural de estruturas aeronáuticas a partir da aplicação do Algoritmo de Seleção Negativa (NSA), baseado em sistemas imunológicos artificiais. Para validar a metodologia, utilizaram 2.900 dados retirados de simulações em elementos finitos de uma viga de alumínio com desgastes e falhas, variando em nível e localização. Cada teste, composto por 3 conjuntos de detectores, foi executado 20 vezes, utilizando diferentes percentuais das amostras de cada padrão. Com 30% de amostras como base de dados para o detector, o acerto foi de 100%. A geração de detectores foi feita *off-line* e o monitoramento *on-line* requereu apenas 170ms, podendo ser usada em tempo real. Os autores concluíram que o sistema com base em sistemas imunológicos artificiais é confiável, eficaz e robusto para monitoramento da integridade de estruturas aeronáuticas.

O estudo de Pereira et al. (2016) propõe o monitoramento de motores de indução pela lógica *fuzzy* a partir de dados vibracionais, permitindo identificar quatro

tipos de falhas mecânicas – desbalanceamento, desalinhamento, folgas e falhas em rolamentos – em três estágios – sem defeito, intermediário e crítico. Para o processamento de sinais vibracionais, utilizaram a transformada rápida de Fourier (FFT) e envelope. As entradas do modelo são as frequências detectadas e as saídas são os modos de falha e a intensidade da falha. Os resultados apontaram que o modelo proposto, foi eficiente para o diagnóstico de falhas na carcaça de um motor, entretanto foi menos eficaz para diagnóstico de rolamentos. Essa proposta pode ser estendida a outras máquinas e falhas, desde que adequadas as funções de pertinência e regras.

Foi proposta de Wood et al. (2012) uma Manutenção Baseada na Criticidade de equipamentos industriais a partir da lógica *fuzzy* dos tipos Mamdani e Sugeno desenvolvida no *Fuzzy Toolbox* do *Matlab*, visando melhorar a capacidade e confiabilidade de um sistema. Foram levados em conta três critérios – ambiental, segurança e saúde; impacto no negócio; e custo anual com manutenção. Em seguida, formou-se um time de experts para avaliar os equipamentos segundo estes critérios, a fim de implementar um sistema de avaliação de criticidade *crisp*. Por lidar bem com incertezas, conceitos vagos e variáveis qualitativas e quantitativas, foi feito um sistema baseado na teoria *fuzzy*. Os critérios ambiental, segurança e saúde e impacto no negócio foram as duas entradas do modelo, com cinco funções de pertinência triangulares em cada uma. Como no modelo *crisp*, o primeiro critério na abordagem *fuzzy* teve maior relevância na saída “nível de criticidade”. Os dois métodos foram testados em 25 casos e concluiu-se que o *fuzzy* oferece uma resposta mais detalhada que a *crisp*, podendo avaliar equipamentos individualmente e também classificá-los em um mesmo nível de criticidade com maior confiabilidade. O tipo Mamdani e o Sugeno apresentaram resultados semelhantes, sendo a diferença apenas no tempo de processamento menor do tipo Sugeno.

Cheng, Qu e Qiao (2017) propuseram um prognóstico de vida útil remanescente para a caixa de câmbios de uma turbina eólica a partir de um Sistema de Inferência Adaptativo Neuro-fuzzy (ANFIS) em comparação com RNA. Dentro das classes de prognósticos, os métodos orientados por dados foram utilizados neste trabalho. O valor PSD obtido através de sinais vibracionais foi a entrada do sistema de predição. O ANFIS e a RNA possuem a habilidade de aprender e atualizar os parâmetros, permitindo melhorar a predição com o passar do tempo. O ANFIS apresentou maior acurácia que a RNA, prevendo com precisão o tempo de vida

restante de uma caixa de engrenagens com 32,7% de vida. A escolha da quantidade de amostras levou em conta o custo computacional e a melhora na precisão da predição, relacionando-as pela equação da raiz quadrada do erro (*root-mean-square error*), para a obtenção o melhor custo benefício.

Nos estudos de Tian (2009), Tian, Worg e Safaei (2010) e Wu, Tian e Chen (2013), foram feitas propostas para predição de vida remanescente por RNAs. O modelo de RNA que apresentou melhores resultados foi uma rede neural *feedforward* com uma camada de entrada, duas camadas intermediárias e uma camada de saída, cujas entradas foram o tempo e as condições de monitoramento e a saída foi a porcentagem da vida do ativo. Para n medições de condição significantes e usando dois pontos temporais, o número total de nós é dado por $(2+2n)$. Segundo Tian, Worg e Safaei (2010), estas entradas refletem nas condições de saúde do equipamento e dão uma generalização melhor à rede. A função de transferência escolhida para as camadas intermediárias foi a tangente hiperbólica. Para a avaliação da performance da rede durante o treinamento, utilizou-se o erro quadrático médio (MSE, do inglês *mean squared error*) e, para otimizar o treinamento da RNA, empregou-se, por Tian, Worg e Safaei (2010), o *Resilient propagation*, e, por Tian (2009), o *Levenberg-Marquardt*. Ao usarem não apenas o histórico de falhas, como também o histórico de suspensão, os autores puderam ter uma ideia da progressão da degradação do equipamento. Então, para construir o histórico para o treinamento, juntou-se o histórico de suspensão e o de falhas.

Durante o treinamento ocorreu o *overfitting*, apresentando um problema na generalização da rede, e, como consequência, um erro pequeno na fase de treinamento e um erro grande na fase de teste. Para evitá-lo, os autores usaram o método da validação, na qual um terço das amostras foram destinadas à validação e o restante, ao treinamento. Quanto menor é o valor MSE na validação, melhor é o treinamento da rede. Logo, utilizaram uma equação polinomial de terceira ordem, na qual $y=MSE$ e $x=tempo\ de\ falha$. A rede foi treinada cinco vezes e foi escolhida a rede com o menor MSE. Com esta rede, os resultados foram satisfatórios na predição de vida remanescente do ativo através da relação dos dados da suspensão e das falhas.

Para estimar a incerteza, após treinada, a RNA foi testada com dados ainda não utilizados e foi obtida a diferença entre o tempo de falha previsto pela RNA e o tempo de falha dos dados do histórico. Este erro apresentou distribuição normal e pôde-se estimar sua média e desvio padrão. A probabilidade de falha até a próxima

inspeção foi calculada pelo resultado da RNA e, a partir da otimização do CBM, pôde ser encontrado o limiar do valor de probabilidade de falha. Dessa maneira, foi possível prever se a probabilidade de falha será superior ao valor limite, obtido pela distribuição de Weibull, até a próxima inspeção. Este método foi aplicado na predição de vida remanescente de equipamentos e apresentou resultados mais precisos que os métodos geralmente empregados.

De acordo com Santos, Silva e Suetake (2012), dentre os sistemas inteligentes para diagnósticos de falhas com foco em máquinas de indução, as RNAs foram o método mais utilizado em meio aos estudos pesquisados pelos autores. Isto porque são pouco complexas para implementar e podem ter diversas topologias e boa eficácia na organização de dados.

Com a finalidade de se obter a probabilidade de um determinado componente de um ativo falhar, vários estudos utilizam métodos como Análise da Árvore de Falhas (FTA, do inglês *Fault Tree Analysis*), Análise da Árvore de Eventos (ETA, *Event Tree Analysis*), e, fundamentado na Manutenção Baseada na Condição (CBM, *Condition Based Maintenance*), o método de Manutenção Baseada na Condição Probabilística (PCBM, *Probabilistic Condition Based Maintenance*), podendo incorporar a eles técnicas de inteligência artificial ou não.

A FTA é utilizada para análise de riscos a partir da probabilidade de um evento ou combinação de eventos inferiores ocorrer e provocar o evento do topo. Este valor de probabilidade é obtido por meio da obtenção de um diagrama com as falhas que levam à ocorrência do evento do topo. Podem ser feitas simplificações booleanas para expressar matematicamente a combinação dos eventos. Este método pode ser usado para sistemas simples e complexos, desde que este possua um banco de dados confiável (KAERCHER; LUZ, 2016).

A manutenção baseada na condição (CBM) é um método usado para definir, a partir da avaliação da condição real de um ativo, quando serão feitas as manutenções. Então, os dados do ativo são coletados, de forma periódica ou contínua, e analisados a fim de obter-se o tempo esperado até a falha, o modo de falha e seu desenvolvimento (MECHEFSKE, 2005).

Cai, Hu e Zhang (2016) utilizaram o método de conjunto de corte *fuzzy* (*fuzzy cut set method*) para obter a probabilidade de falha de um equipamento. A análise *bow-tie*, técnica probabilística integrada para prevenir, controlar e atenuar eventos indesejados, foi usada para incorporar a FTA e a análise da árvore de eventos (*Event*

Tree Analysis). Para a obtenção da probabilidade do evento de risco, ou seja, o evento do topo da árvore, a probabilidade de eventos primários deve ser conhecida. Entretanto, como esta probabilidade não é quantificável, o método de conjunto de corte *fuzzy* foi usado para obtê-la. Foi feita uma lógica *fuzzy* com duas entradas – probabilidade de falha e consequência – e uma saída de risco integrado. Por fim, os autores calcularam a severidade da consequência da falha nos âmbitos pessoal, perda econômica e dano ambiental. Os resultados mostram que este método contribuiu na identificação de fatores de risco e de riscos de impactos, visando uma gestão de segurança mais eficaz.

A proposta do artigo de Wang et al. (2016) foi apresentar um método para o diagnóstico de falhas e predição do estado de um circuito de via, que identifica trens nos trilhos. Os dados são monitorados por uma central com funções de aquisição, detecção, alarme e tem como saída um sinal que varia na frequência. Foi feita uma análise de árvore de falhas (FTA) do equipamento e selecionadas as variáveis dos modos de falha e, em seguida, uma rede neuro-*fuzzy* (FNN) para diagnosticar as falhas, cujas regras foram definidas a partir da FTA. A FNN possui a primeira camada ligada às entradas, a segunda camada convertendo as entradas para conjuntos *fuzzy* a partir de funções de pertinência, a terceira camada para calcular o produto dos sinais segundo as regras *fuzzy*, a quarta camada para normalizar as regras e a quinta camada para a defuzzificação. Para treinar a FNN foi usado o algoritmo de otimização *Momentum* gradiente, que ajusta os pesos de acordo com uma função de erro e um fator *Momentum*. Ao combinar FTA e experiência de experts, foi possível tomar decisões sobre a manutenção de falhas reais. Concluiu-se que o modelo de FNN foi verossímil e válido para diagnóstico e predição de falhas.

Zheng et al. (2016) propõem utilizar a teoria de conjuntos *fuzzy* para obter a probabilidade da ocorrência do evento do topo da árvore de falhas e da significância de probabilidade de evento básico a partir do teorema da decomposição combinado à teoria *Cut Set* e intervalo aritmético, formando a análise de árvore de falhas *fuzzy* (FFTA). Esta é análise de falha hierárquica, em que um evento inferior contribui para a ocorrência de um superior. Essa análise de causa e efeito é obtida pelo *Minimum Cut Sets*, que determina a menor combinação de falha. Foram usadas funções de pertinência triangulares, devido à simplicidade para operações algébricas OU e E na FTA. A análise quantitativa englobou a probabilidade com números *fuzzy* e a qualitativa foi feita pelo método da significância da probabilidade do evento básico

para encontrar os componentes que afetam mais a confiabilidade do sistema. Os resultados mostram que a FFTA pode fornecer uma referência para a manutenção do sistema de acionamento.

O método de Monte Carlo foi utilizado por Defiltro e Gomes (2016) para a análise de confiabilidade de duas vigas, em que eles consideraram falha o critério de sair do regime linear-elástico, com base na Mecânica dos Sólidos. Tomando como base a teoria da confiabilidade estrutural, eles fizeram dois estudos: o primeiro para averiguar a correlação de variáveis aleatórias na probabilidade de falha estimada de uma estrutura; o segundo pauta a relação entre o critério de falha de diferentes materiais e a probabilidade de falha. Os autores concluíram que o critério de falha altera significativamente a probabilidade de falha, assim como as relações entre as variáveis aleatórias.

Teixeira (2016) propõe uma metodologia para avaliação da confiabilidade estrutural, incorporando ao conceito de CBM (*Condition Based Maintenance*) conhecimentos probabilísticos, formando o PCBM (*Probabilistic Condition Based Maintenance*), cujo objetivo é obter a probabilidade de falha ao longo de sua evolução e seu índice de segurança. Esta metodologia foi aplicada, para análise de falha por fadiga, em uma bancada experimental composta por três eixos de diferentes diâmetros acionados por um motor elétrico, controlado por um inversor de frequência. O ponto de falha é estabelecido pela linha de Goodman e, quanto mais próximo desta linha, maior foi a probabilidade de falha do ponto.

Pitakrat et al. (2018) fizeram a previsão de falhas em sistemas de softwares complexos através de uma abordagem hierárquica de previsão *online* modelada por redes Bayesianas, que melhorou o desempenho do sistema em 9,9%. Apresenta a vantagem de cada preditor poder ser retreinado individualmente, entretanto, tem como desvantagem a necessidade de manter o modelo de propagação de falhas atualizado, uma vez que mudanças arquitetônicas podem alterar a forma de propagação de falhas.

A proposta de Man, Zhou (2018) para prever a vida útil restante de sistemas sujeitos a falhas graves, probabilisticamente relacionados a sinais de degradação do sistema, é uma modelagem composta pelo processo de Wiener – utilizado para modelar sinais de degradação – e pelo modelo PH para dados no tempo até um evento, que pode ser substituído pelo método de tempo de falha acelerado. O resultado apresentou bom desempenho de previsão quando o último sinal de

degradação foi consistente com a tendência de degradação, porém, por depender apenas do último sinal, o método é muito influenciado por ele e pode se distanciar da previsão correta. Em trabalhos futuros, os autores sugerem a utilização de métodos de mineração de dados para extrair conhecimento de grandes conjuntos de dados, e incorporar dados operacionais na modelagem de deterioração do sistema.

2.1.1 Síntese do estado da arte

A previsão de tempo de vida remanescente, tempo até a falha ou diagnóstico de falhas em equipamentos é uma necessidade abordada nos artigos pesquisados publicados nos últimos anos. Muitos deles apresentam métodos mais consolidados na literatura, como FTA e FMEA, e outros utilizam ferramentas de Inteligência Artificial simples ou híbridas.

Os artigos que apresentaram utilização de métodos mais tradicionais, sem o uso da IA, careceram de um modelo que possa se atualizar com o passar do tempo, ou que possa considerar o histórico de falhas ou condições de operação na modelagem de previsão, para a obtenção de modelos mais precisos. Além disto, os modelos que utilizaram abordagens que necessitam da modelagem física do sistema em que se deseja fazer previsões, como elementos finitos, não se mostraram metodologias de replicação simples – sem a necessidade de uma nova modelagem física para cada componente – como as técnicas de IA se mostraram replicáveis sem grandes mudanças estruturais do modelo.

A inteligência artificial, seja com métodos fuzzy, redes neurais, híbridos ou outras técnicas inteligentes, mostrou-se presente na maior parte dos trabalhos pesquisados, tanto para detecção, como para obtenção da probabilidade de ocorrência de um modo de falha. Mesmo para alguns estudos que utilizaram métodos tradicionais com esta finalidade, a IA foi incorporada a eles e agregou bons resultados, permitindo a redução das limitações neles identificadas e a obtenção de soluções mais próximas da realidade de trabalho. Além disso, as técnicas de IA apontaram como vantagem a possibilidade de aprenderem ao longo do tempo, reajustando os pesos, para a obtenção de resultados mais pertinentes e verossímeis.

Dentre as metodologias de IA, foram destaque os métodos *fuzzy* e as RNAs. O primeiro apresentou a vantagem de lidar bem com incerteza e gerar resultados de fácil interpretação, entretanto, sua principal desvantagem foi a subjetividade para definição

das funções de pertinência e parametrização do modelo *fuzzy*, dependendo do conhecimento de especialistas e de sua interpretação das variáveis no contexto de operação de um ativo. As RNAs apresentaram a desvantagem de ser preciso treinar diferentes topologias de redes para que se obtenha a melhor opção a ser utilizada, mas, por outro lado, retratou-se a vantagem de sua implementação ser pouco complexa.

2.2 Manutenção

Esta seção introduz conceitos e teorias relevantes que permeiam este estudo, referentes ao funcionamento do ativo, e à evolução da manutenção ao longo do tempo.

A perda da capacidade de um equipamento desempenhar uma função é denominada falha, que pode ser crítica, quando coloca pessoas em situação de risco ou causa expressivos danos materiais. As falhas podem ocorrer por diversos motivos, dentre eles devido à fragilidade, projeto inadequado, deterioração ou de forma aleatória. Porém o dano ocorre se um item sofre modificação em alguma característica, independentemente da capacidade em cumprir sua função (ABNT, 1994).

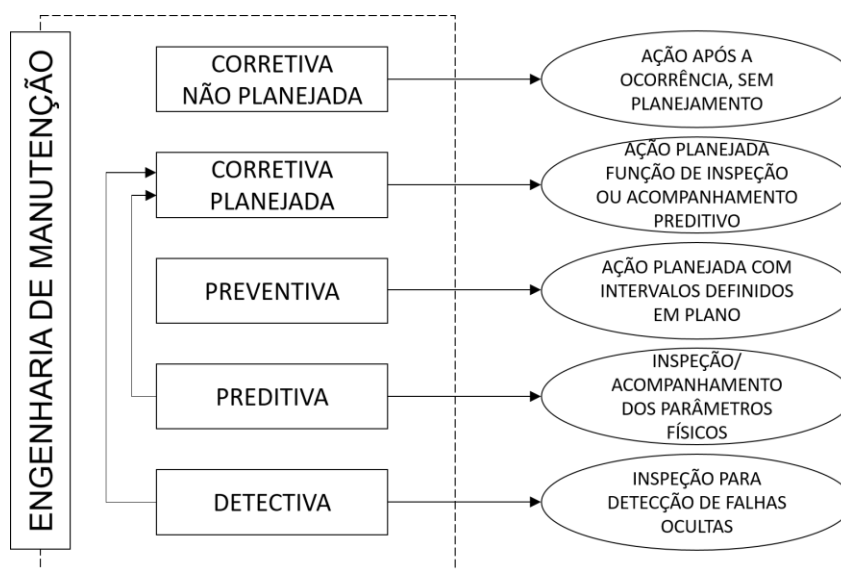
A falha funcional é definida por Souza e Lima (2003) como a incapacidade da realização de uma função por um ativo, podendo ser parcial, em que o ativo deteriorado pode estar em funcionamento, porém fora dos limites aceitáveis de operação, ou total. Segundo Moubray (1997), cada equipamento possui variadas funções que podem falhar de diferentes maneiras, sendo a falha funcional a inabilidade de um ativo exercer uma função com desempenho padrão aceitável para o usuário. As falhas podem ocorrer na produção de produtos que extrapolem as especificações dos limites máximo e mínimo de determinada grandeza ou não atendam à especificação de qualidade.

Para que um equipamento cumpra a função a ele requerida, é preciso que ele tenha disponibilidade para operar. Entretanto, as horas disponíveis em determinado período de tempo não são completamente utilizadas para operação, visto que é preciso descontar as horas não operadas, seja devido a paradas para inspeção, esperas e interrupções ou paradas para manutenção.

2.2.1 Tipos de manutenção

Manutenção é definida como a “combinação de todas as ações técnicas e administrativas, incluindo as de supervisão, destinadas a manter ou recolocar um item em um estado no qual possa desempenhar uma função requerida” (ABNT, 1994). A manutenção pode ser subdividida em diferentes tipos, como mostra a Figura 1.

Figura 1 - Tipos de manutenção



Fonte: Kardec e Nascif (2010).

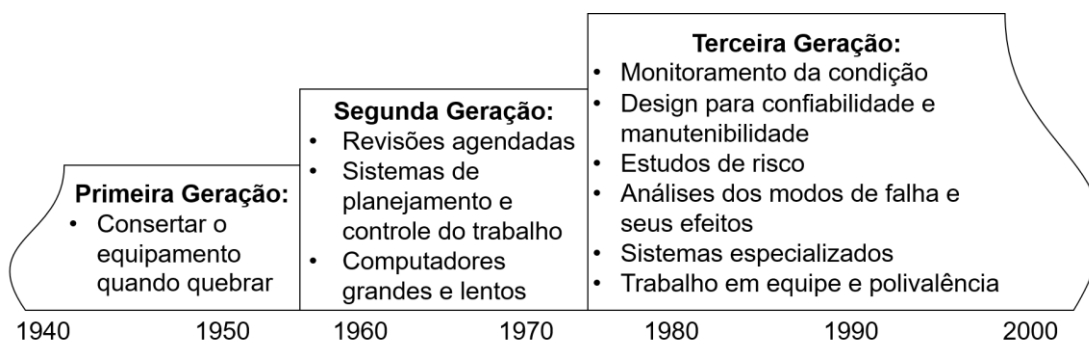
Manutenção corretiva é a atuação para corrigir uma falha ou o desempenho inferior ao desejado de um equipamento, visando recolocá-lo em condições de executar sua função. Quando é não planejada ou emergencial caracteriza-se pela atuação após ocorrência de uma falha aleatória inesperada, sem haver preparação para execução do serviço, ocasionando custos elevados. Podem causar mais danos ao equipamento e também comprometer a produção de toda a planta. Quando planejada é a atuação no equipamento por decisão gerencial, tomando como base condições advertidas pela manutenção preditiva. Mesmo que a escolha seja permanecer em operação até a falha, é possível fazer um planejamento até o momento em que a quebra ocorrer. Dessa forma, podem ser compatibilizados interesses da manutenção e da produção (KARDEC E NASCIF, 2010, p.37-42; ABNT, 1994).

A manutenção preventiva é programada para ser realizada em determinados intervalos de tempo, segundo critérios que visam reduzir a probabilidade de falha ou queda no desempenho de um equipamento, para evitar que a falha ocorra. É apropriada para falhas que acarretem altos custos e grande prejuízo de produção, visto que, para ser realizada, promove a parada do equipamento. Já a manutenção preditiva ou permite prever o tempo de vida útil de um componente a partir de dados sobre suas condições reais de operação, a fim de determinar antecipadamente a manutenção e reduzir a manutenção corretiva não planejada, visando permitir que o equipamento permaneça em funcionamento pelo maior tempo possível. Para ser viável, é necessário que seja possível monitorar as causas e acompanhar sistematicamente a progressão das falhas. A manutenção detectiva é a busca por falhas imperceptíveis pelo departamento de manutenção em sistemas de proteção, comando e controle, e a manutenção de oportunidade ocorre quando um equipamento não está operando por algum motivo e este período de tempo é aproveitado para inspeções ou melhorias (KARDEC E NASCIF, 2010, p.42-49; ABNT, 1994).

2.2.2 História da manutenção

A evolução da manutenção ao longo da história pode ser dividida em gerações. A primeira geração ocorreu até a Segunda Guerra Mundial, quando a indústria não dirigia esforços para a prevenção de falhas no equipamento. Nessa época, os ativos eram mais simples e, conseqüentemente o reparo era mais fácil. A Segunda Guerra Mundial trouxe uma maior demanda de bens de consumo e as máquinas se tornaram mais complexas e numerosas, quando começou a segunda geração e surgiu o conceito de manutenção preventiva. A partir disso, a manutenção foi se tornando mais complexa e com custos maiores e mais significativos para a produção. Após os anos 70, durante a terceira geração, as técnicas e os processos se aperfeiçoaram com o crescimento da mecanização e automação de tal maneira, que a produtividade passou a ser bem mais comprometida devido ao tempo de inatividade dos equipamentos, além do risco de ocorrerem graves conseqüências ambientais e de segurança, por exemplo, com a falha dos ativos. Nesta geração foram introduzidos os conceitos de Manutenção Centrada na Confiabilidade (RCM, do inglês *Reliability-centered Maintenance*) e disponibilidade do equipamento (MOUBRAY, 1997, p.2-5). A Figura 2 mostra a mudança das técnicas no decorrer das gerações.

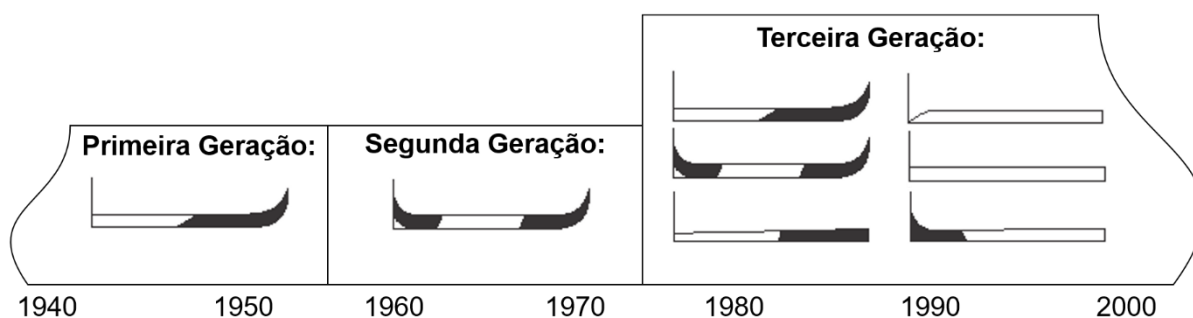
Figura 2 - Mudança das técnicas de manutenção



Fonte: Tradução livre de Moubray (1997, p.5).

Assim como as técnicas foram se desenvolvendo, os estudos também foram ampliando o conhecimento sobre as falhas. Na primeira geração considerava-se que quanto mais velho um equipamento fosse, maior seria a probabilidade de uma falha ocorrer. Na segunda geração houve a conscientização sobre a “mortalidade infantil”, levando ao surgimento da curva da banheira. Foi na terceira geração que se revelou a existência de seis padrões de falha (Figura 3).

Figura 3 - Padrões de falha em cada geração



Fonte: Adaptado de Moubray (1997, p.4)

Segundo Kardec e Nascif (2010, p.4), após os anos 2000 começou a quarta geração da manutenção, que, assim como a terceira, tem como objetivo a confiabilidade dos equipamentos para a maximização de sua disponibilidade. Esta geração é pautada em disponibilidade, confiabilidade e manutenibilidade. O Quadro 1 mostra a evolução da manutenção ao longo das quatro gerações em relação às técnicas, à falha e demais expectativas.

Quadro 1 - Evolução da manutenção ao longo das gerações

Ano	Primeira Geração	Segunda Geração	Terceira Geração	Quarta Geração
	1940 1950	1960 1970	1980 1990	2000 2010
Expectativas em relação à manutenção	<ul style="list-style-type: none"> ▪ Conserto após falha 	<ul style="list-style-type: none"> ▪ Maior vida útil ▪ Disponibilidade crescente 	<ul style="list-style-type: none"> ▪ Maior disponibilidade e confiabilidade ▪ Preservação do ambiente 	<ul style="list-style-type: none"> ▪ Maior disponibilidade e confiabilidade ▪ Preservação do ambiente ▪ Segurança ▪ Gerenciar os ativos ▪ Influir nos resultados do negócio
Visão quanto à falha do ativo	<ul style="list-style-type: none"> ▪ Equipamentos falham devido ao desgaste pela idade 	<ul style="list-style-type: none"> ▪ Falha conforme curva da banheira 	<ul style="list-style-type: none"> ▪ Seis padrões de falha 	<ul style="list-style-type: none"> ▪ Reduzir drasticamente falhas prematuras
Mudança nas técnicas de manutenção	<ul style="list-style-type: none"> ▪ Técnicas visando reparo 	<ul style="list-style-type: none"> ▪ Planejamento manual ▪ Computadores lentos ▪ Manutenção preventiva por tempo 	<ul style="list-style-type: none"> ▪ Monitoramento da condição ▪ Manutenção preditiva ▪ Análise de risco ▪ Computadores rápidos ▪ Softwares potentes ▪ Projetos voltados para confiabilidade 	<ul style="list-style-type: none"> ▪ Aumento da Manutenção preditiva e Monitoramento da Condição ▪ Diminuição das Manutenções Preventiva e Corretiva não Planejada ▪ Técnicas de confiabilidade ▪ Projetos voltados para confiabilidade, manutenibilidade e custo do ciclo de vida

Fonte: Adaptado de Kardec e Nascif (2010).

2.3 Indústria 4.0

As três primeiras Revoluções Industriais trouxeram expressivas mudanças na indústria: na primeira, houve um aumento da produção com a utilização de teares mecânicos a vapor; a segunda foi marcada pela divisão do trabalho em linhas de produção contínuas; e, a terceira, pelo controlador lógico programável (PLC), ainda presente nos sistemas de automação. No contexto da Quarta Revolução Industrial, surgiu na Alemanha o termo Indústria 4.0 (do alemão *Industrie 4.0*), que se refere à inclusão de tecnologias da internet na indústria. Entretanto, ao contrário das outras três Revoluções Industriais, que foram observadas após seu acontecimento, este é um fenômeno que está sendo previsto para o futuro de forma inevitável (DRATH, HORCH, 2014).

Hermann, Pentek e Otto (2016) identificaram quatro princípios de *design* da Indústria 4.0, elucidados a partir dos estudos de Schuh et al. (2013), Gorecky et al. (2014), Bauernhansl (2014), Malone (1999) e Lee (2008). São eles:

- Interconexão de máquinas, sensores e pessoas através da Internet de Tudo (IoE, do inglês *Internet of Everything*) – formada pela Internet das Coisas (IoT, *Internet of Things*) e pela Internet das Pessoas (IoP, *Internet of People*) –, permitindo uma base de colaboração entre homens, entre máquinas e entre homem-máquina.
- Transparência da informação, em que são agregadas informações e interpretações contextualizadas dos mundos físico e virtual às medições brutas dos sensores, e os resultados ficam acessíveis a todos os participantes da IoE. É importante que informações críticas sejam fornecidas em tempo real.
- Decisões descentralizadas combinadas à transparência da informação permitem melhor tomada de decisão através da combinação de informações locais e globais, além do controle do mundo físico de forma autônoma.
- Assistência técnica de sistemas que agregam e visualizam as informações para ajudar os humanos na tomada de decisões em curto prazo.

A rede de conexões entre humanos, máquinas e objetos têm ficado mais complexas no âmbito da Indústria 4.0. O seu processo de implementação permeia a necessidade, a partir de interpretações contextualizadas, da transformação de dados e informações em conhecimento, com a finalidade de auxiliar a tomada de decisões. Em vista disso, o processo de exploração de dados abrange os tópicos para se alcançar o conhecimento de um sistema ou processo através da análise de dados. Alguns riscos para a implementação da Indústria 4.0 foram identificados por Tupa, Simota e Steiner (2017) como perda de dados, problemas com disponibilidade e integridade dos dados.

2.4 Processo de exploração de dados

Segundo Pyle (1999), as etapas processo de exploração de dados são:

- Explorar o espaço problema;
- Explorar o espaço solução;

- Especificar o método de implementação;
- Minerar os dados (preparação, levantamento e modelagem dos dados).

Quadro 2 - Esforços gastos no processo de exploração de dados

	Tempo para completar (% do total)	Importância para sucesso (% do total)	
Explorar o espaço problema	10	15	80
Explorar o espaço solução	9	14	
Especificar o método de implementação	1	51	
Preparação dos dados	60	15	20
Levantamento dos dados	15	3	
Modelagem dos dados	5	2	

Fonte: Traduzido de Pyle (1999).

O Quadro 2 mostra o esforço gasto em cada etapa do processo de exploração de dados. O primeiro momento – de exploração do espaço problema – consiste em identificar os problemas certos a se resolver e isto é mais difícil do que aparenta, entretanto é importante determinar os problemas de forma precisa. Normalmente a definição é feita em forma de declarações, que devem ser analisadas em componentes menores, para que possam, a princípio, serem respondidas com a análise dos dados. Os problemas podem ser identificados por meio de métodos como mapas cognitivos e Pairwise Ranking.

Na exploração do espaço solução, a solução almejada deve ser passível de implementação no mundo real e específica o suficiente para que possa ser praticamente implementada ao final desta etapa. A técnica de resolução de ambiguidades se mostra importante para estas duas etapas, visto que busca identificar falhas de comunicação e assegurar que os pontos essenciais tenham sido compreendidos por todos.

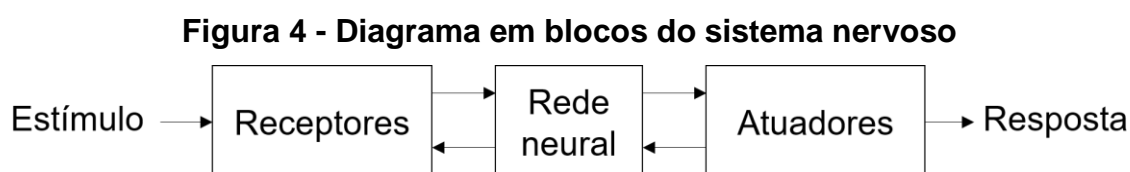
O terceiro passo é a especificação do método de implementação, que objetiva a aplicação dos resultados para aumentar a lucratividade, melhorar segurança ou atender a alguma meta específica. Neste momento é detalhada a forma como a solução será aplicada na prática.

Assim como na mineração geológica é preciso fazer todo um preparo das condições e do local descoberto, a mineração de dados também envolve várias etapas:

- Preparação dos dados e do minerador, mantendo sempre interação entre ambos, mesmo que a preparação dos dados seja feita de forma automática. Assim, os modelos produzidos serão melhores e mais rápidos.
- Levantamento dos dados para estimar quais informações estão contidas nos dados.
- Modelagem dos dados, levando em conta conhecimento dos dados e do problema.

2.5 Redes neurais artificiais

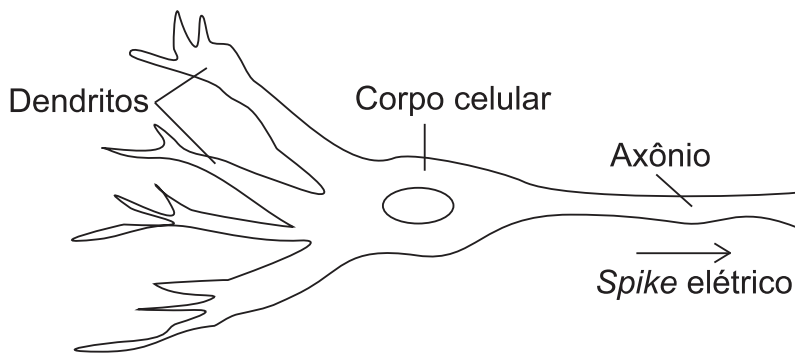
O sistema nervoso humano transmite sinais portadores de informações, em que receptores convertem estímulos corporais ou do ambiente em impulsos elétricos e os impulsos gerados pela rede neural são convertidos em respostas discerníveis, como saída do sistema (Figura 4). No cérebro existem níveis de organizações anatômicas para diferentes funções, sendo as sinapses, onde ocorre transmissão de impulsos elétricos, o nível mais fundamental (HAYKIN, 2007).



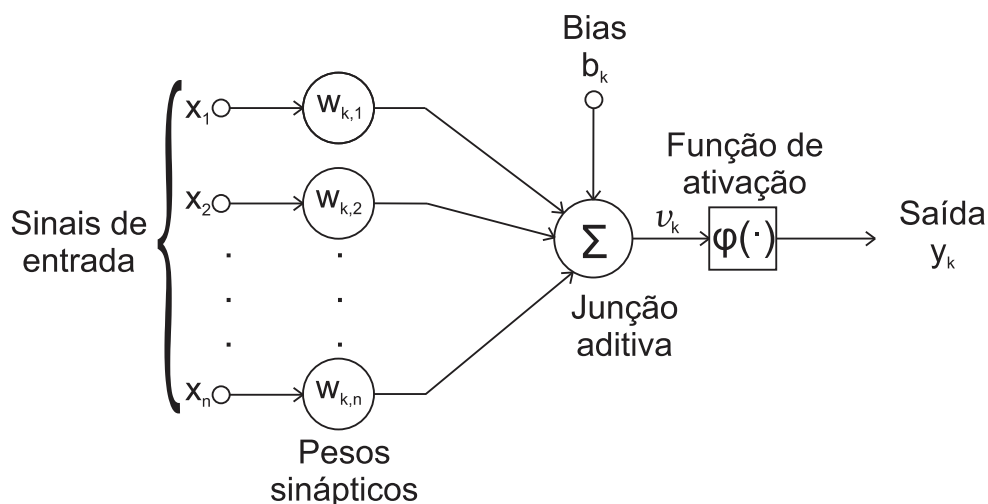
Fonte: Haykin (2007).

2.5.1 Neurônio

O neurônio é a célula elementar da rede neural e têm como função a condução de impulsos. O neurônio biológico recebe informações de outros neurônios pelos dendritos (conhecidos como soma) e propaga a informação processada por um axônio através de um potencial de ação (*spike*), como mostra a Figura 5. A sinapse no cérebro humano é um processo químico de liberação de uma substância no terminal pré-sináptico, que podem ser excitatórias ou inibitórias da transmissão (ROSA, 2011; SILVA, SPATTI, FLAUZINO, 2010).

Figura 5 - Neurônio biológico

Fonte: Rich e Knight (1994 apud ROSA, 2011).

Figura 6 - Modelo não linear de um neurônio

Fonte: Haykin (2007).

O neurônio artificial se baseia no biológico em sua construção. O modelo não linear de um neurônio artificial pode ser representado de forma genérica pela Figura 6. Os sinais de entrada x_1 a x_n são adquiridos externamente e normalizados para melhor eficiência computacional. As sinapses são a conexão entre uma entrada x_j ao neurônio k e possuem uma força de ligação, que pode ser positiva ou negativa, chamada peso sináptico w_{kj} – sendo o primeiro índice (k) referente ao neurônio e o segundo (j) à entrada. O combinador linear (Σ) faz a soma ponderada dos sinais de entrada com o respectivo peso sináptico, resultando em um potencial de ativação. No entanto, para haver ativação da saída do neurônio, é preciso que este resultado esteja em um patamar apropriado, o qual é definido pelo limiar de ativação ou bias. A diferença entre o combinador linear e o bias é denominada potencial de ativação, ou seja, se essa diferença for positiva, o potencial é excitatório, se for negativa, será

inibitório. Por fim, a função de ativação limita a amplitude do sinal de saída do neurônio (y_k), que normalmente é um intervalo unitário fechado, entre 0 e 1, ou alternado, entre -1 e 1 (HAYKIN, 2007; SILVA, SPATTI, FLAUZINO, 2010).

Quadro 3 - Comparação de desempenho entre neurônio artificial e biológico

PARÂMETRO	NEURÔNIO ARTIFICIAL	NEURÔNIO BIOLÓGICO
Eficiência energética (operação/segundo)	10^{-6} J	10^{-16} J
Tempo de processamento (operação/neurônio)	10^{-9} s	10^{-3} s
Mecanismo de processamento	Tipicamente sequencial	Tipicamente paralelo

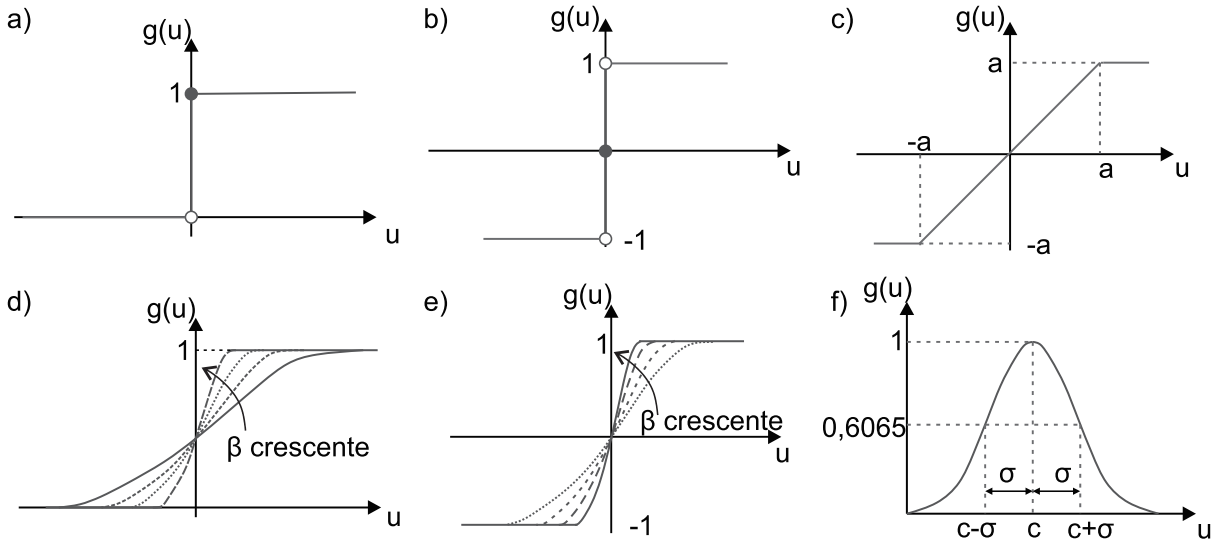
Fonte: Silva, Spatti e Flauzino (2010).

As diferenças dos parâmetros de desempenho entre o neurônio biológico e o artificial encontram-se no Quadro 3. O processamento do neurônio artificial é consideravelmente superior ao biológico, entretanto, ao contrário deste, o biológico aplica ao processo um elevado nível de paralelismo.

2.5.2 Tipos de função de ativação

As funções de ativação de um neurônio podem ser divididas em: parcialmente diferenciáveis, isto é, em alguns pontos as derivadas de primeira ordem são inexistentes, como as funções degrau, degrau bipolar e rampa simétrica; e em totalmente diferenciáveis, em que as derivadas de primeira ordem existem em todos os pontos de seu domínio, por exemplo as funções logísticas, tangente hiperbólica, gaussiana e linear. Estas funções de ativação são mostradas na Figura 7.

Figura 7 - Principais funções de ativação de um neurônio



Fonte: Adaptada de Silva, Spatti e Flauzino (2010, p.36-41).

De acordo com Haykin (2007) e Silva, Spatti e Flauzino (2010), as principais funções de ativação são elucidadas a seguir, sendo a , b e c parcialmente diferenciáveis e as demais totalmente diferenciáveis:

a) Função degrau

Também denominada *Heaviside*, a função degrau é representada pela Figura 7a e dada pela Eq.1.

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0 \end{cases} \quad (1)$$

b) Função degrau bipolar

Esta função assume valores unitários positivos se o potencial de ativação for positivo, nulo para potencial igual a zero e negativo quando potencial é negativo (Figura 7b), como na Eq. 2. Para classificações de padrões, esta função pode ser aproximada pela Eq. 3:

$$g(u) = \begin{cases} 1, & \text{se } u > 0 \\ 0, & \text{se } u = 0 \\ -1, & \text{se } u < 0 \end{cases} \quad (2)$$

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ -1, & \text{se } u < 0 \end{cases} \quad (3)$$

c) Função rampa simétrica

Retorna os valores do potencial de ativação em determinado intervalo, como mostra a Figura 7c, de acordo com a Eq.4.

$$g(u) = \begin{cases} a, & \text{se } u > a \\ u, & \text{se } -a \leq u \leq a \\ -1, & \text{se } u < -a \end{cases} \quad (4)$$

d) Função logística

Conhecida também como sigmoide, o resultado desta função possui apenas valores entre 0 e 1 e é dado pela Eq.5, sendo β uma constante relacionada à inclinação da função perante o ponto de inflexão. A Figura 7d apresenta a função logística com a variação de β .

$$g(u) = \frac{1}{1+e^{-\beta \cdot u}} \quad (5)$$

e) Função tangente hiperbólica

Assume valores reais entre -1 e 1 de acordo com a Eq.6, em que β também está relacionado à inclinação do gráfico perante o ponto de inflexão. A variação de β é mostrada na Figura 7e.

$$g(u) = \frac{1-e^{-\beta \cdot u}}{1+e^{-\beta \cdot u}} \quad (6)$$

f) Função gaussiana

Esta função possui resultados iguais para valores de potencial de ativação que estejam equidistantes da média (Eq.7), sendo c o centro e σ o desvio padrão da função gaussiana (Figura 7f).

$$g(u) = e^{-\frac{(u-c)^2}{2\sigma^2}} \quad (7)$$

2.5.3 Primeiras modelagens neurais e treinamentos

Os primeiros neurônio artificiais foram o modelo *Perceptron* e o modelo *Adaline*, que são utilizados como base para os atuais modelos de neurônio e treinamento que das RNAs

2.5.3.1 *Perceptron*

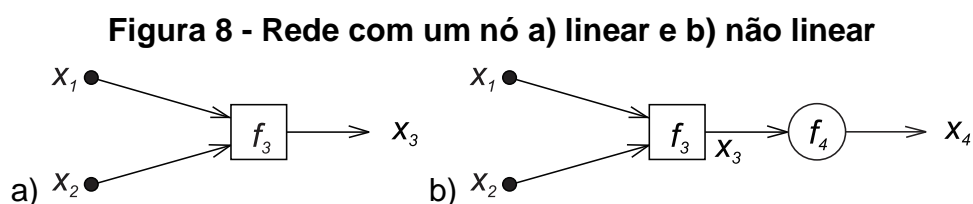
Criado por Rosenblatt em 1957, o *Perceptron* modela um neurônio a partir da soma ponderada das entradas. Na rede adaptável linear da Figura 8a., em que x_1, x_2 são entradas e a_1, a_2, a_3 são parâmetros variáveis, a saída x_3 define um plano $x_1 - x_2 - x_3$ no espaço e é dada pela Eq.8.

$$x_3 = f_3(x_1, x_2, a_1, a_2, a_3) = a_1x_1 + a_2x_2 + a_3 \quad (8)$$

Uma rede *Perceptron* não linear, como na Figura 8b, pode ser feita a partir da rede linear para a saída x_3 com valores 0 e 1, pela Eq.9.

$$x_4 = f_4(x_3) = \begin{cases} 1 & \text{se } x_3 > 0 \\ 0 & \text{se } x_3 < 0 \end{cases} \quad (9)$$

em que f_3 é uma função parametrizada linearmente e f_4 é uma função degrau. Esta é uma rede classificadora linear que cria uma linha $x_1 - x_2$ no espaço pela função f_3 e f_4 indica em qual lado do plano o vetor de entrada está (JANG, SUN e MIZUTANI, 1997, p.203-204; ROSA, 2011, p.176-178).



Fonte: Jang, Sun e Mizutani (1997).

O processo de treinamento de um *Perceptron* é feito pela regra de aprendizado de Hebb (Eq.10), cujo processo para cada amostra de treinamento (x_i) é resumido em incrementar os pesos sinápticos para sinais excitatórios – em que o valor (y) da saída se aproxima do valor desejado (d_i) – e decrementa-los para sinais inibitórios. Dessa forma, o ajuste dos pesos sinápticos (w_i) e do limiar (θ_i) do neurônio são expressos pela Eq.20 (SILVA, SPATTI e FLAUZINO, 2010).

$$\begin{cases} w_{atual} = w_{anterior} + \eta \cdot (d_i - y) \cdot x_i \\ \theta_{atual} = \theta_{anterior} + \eta \cdot (d_i - y) \cdot x_i \end{cases} \quad (10)$$

2.5.3.2 Adaline

O modelo de neurônio Adaline surgiu quase simultaneamente com o modelo *Perceptron*. Embora esses dois modelos baseiem-se em somas ponderadas e comparação com um valor limiar, o treinamento de um neurônio Adaline é fundamentado no algoritmo regra delta. A regra delta ou gradiente descendente é um algoritmo baseado no ajuste dos pesos sinápticos ao minimizar o erro quadrático médio entre a saída desejada e a resposta gerada pela rede, considerando todas as amostras. E a busca do valor ótimo do erro quadrático médio em relação aos pesos é expressa pela Eq.11 (BRAGA, CARVALHO, LUDERMIR, 2007; SILVA, SPATTI, FLAUZINO, 2010):

$$\frac{dE}{dW} = \sum_{i=1}^p (d_i - (W^T \cdot x_i - \theta)) \cdot (-x_i) \quad (11)$$

onde E é o erro, W é o vetor de pesos, p é o número de amostras, d_i é o valor desejado da i -ésima amostra de treinamento, x_i é a i -ésima amostra de treinamento e θ é o valor do limiar. A adaptação dos pesos é oposta ao gradiente do erro, conforme Eq. 12 e 13, em que η corresponde à taxa de aprendizado.

$$\Delta W = -\eta \cdot \frac{dE}{dW} \quad (12)$$

$$W_{atual} = W_{anterior} - \eta \cdot \frac{dE}{dW} \quad (13)$$

Essa rede é constituída por apenas uma camada neural, portanto, apresenta apenas um único neurônio. Além disso, é utilizada em problemas de classificação de padrões com apenas duas classes, devido a sua simplicidade estrutural, semelhantemente à rede *Perceptron* (SILVA; SPATTI; FLAUZINO, 2010).

2.5.3.3 Comparação entre modelos neurais *Perceptron* e *Adaline*

No Quadro 4 encontra-se a comparação entre os dois modelos *Perceptron* e *Adaline*. Apesar de simples, a rede *Adaline* permitiu um avanço importante para a evolução na área das redes neurais, pois seu algoritmo de treinamento foi um precursor da regra delta generalizado ou *backpropagation*, utilizado no treinamento de uma rede *Perceptron* de múltiplas camadas (SILVA; SPATTI; FLAUZINO, 2010).

Quadro 4 - Comparação entre modelo *Perceptron* e *Adaline*

Modelo	<i>Perceptron</i>	<i>Adaline</i>
Algoritmo de treinamento	Regra Hebb	Regra Delta
Quantidade de classes para a classificação de padrões	2	2
Quantidade de neurônios	1	1
Quantidade de camadas intermediárias	1	1
Tipos de problemas abordados	Linearmente separáveis	Linearmente e não-linearmente separáveis
Quantidade de épocas	Depende da inicialização dos valores dos pesos e da disposição das amostras	Depende da inicialização dos valores dos pesos, da disposição das amostras e da taxa de aprendizado

Fonte: Elaborado pela autora.

2.5.4 Arquiteturas de redes neurais artificiais

Uma rede neural é formada por camadas de neurônios, sendo a mais simples composta apenas pelas camadas de entrada e saída. Entretanto, por haver processamento apenas na camada de saída, é chamada rede de camada única (Figura 9a). Em contrapartida, a rede de múltiplas camadas possui ao menos uma camada oculta entre as camadas de entrada e saída, fazendo com que as entradas não estejam diretamente ligadas às saídas. Ao adicionar camadas ocultas, a rede passar a ser capaz de lidar com estatísticas de ordens superiores. Uma rede é totalmente conectada quando cada nó de uma camada está conectado a todos os nós

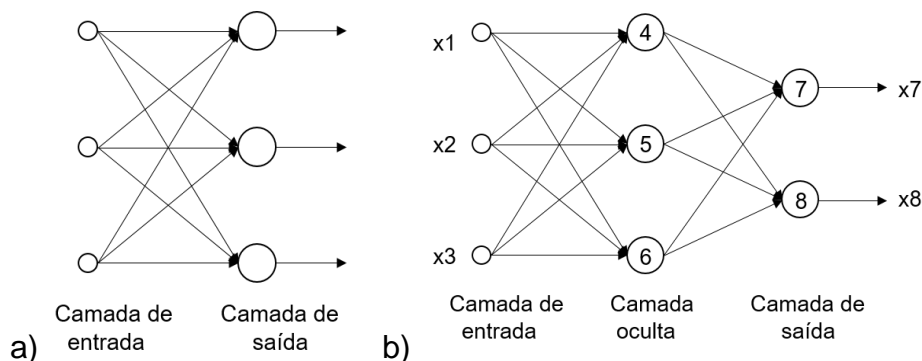
da camada seguinte, caso contrário é uma rede parcialmente conectada (HAYKIN, 2007).

Em uma rede como a da Figura 9b, é possível que cada camada de nós tenha uma função distinta de ativação, em que, por exemplo, a camada oculta seja composta de nós com função linear e a de saída tenha função sigmoideal. Dessa forma, a saída x_7 seria uma composição das funções, dada pela Eq. 14 (JANG, SUN, MIZUTANI, 1997, p.204).

$$x_7 = \frac{1}{1 + \exp[-(w_{4,7}x_4 + w_{5,7}x_5 + w_{6,7}x_6 + t_7)]} \quad (14)$$

em que x_4 , x_5 e x_6 são as respectivas saídas dos nós 4, 5 e 6 e os parâmetros relativos ao nó 7 são $\{w_{4,7}, w_{5,7}, w_{6,7}\}$.

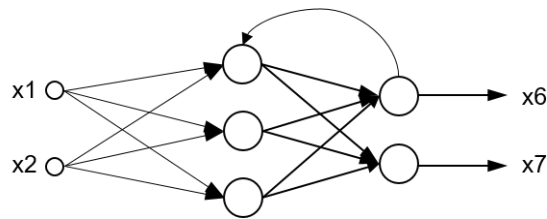
Figura 9 - Rede de a) camada única e b) de múltiplas camadas



Fonte: Adaptado de Haykin (2007) e Jang, Sun e Mizutani (1997).

A arquitetura das RNAs podem variar segundo o tipo de conexão dos neurônios. Nas redes *feedforward* todos os neurônios possuem conexões da esquerda para a direita, ou seja, o fluxo segue unidirecionalmente da entrada até a saída, como na Figura 9. Já na arquitetura recorrente, as saídas de ao menos um neurônio é realimentada na entrada de outro (Figura 10), formando um caminho circular (SILVA, SPATTI e FLAUZINO, 2010; JANG, SUN e MIZUTANI, 1997, p.200-202).

Figura 10 - Arquitetura de uma rede recorrente

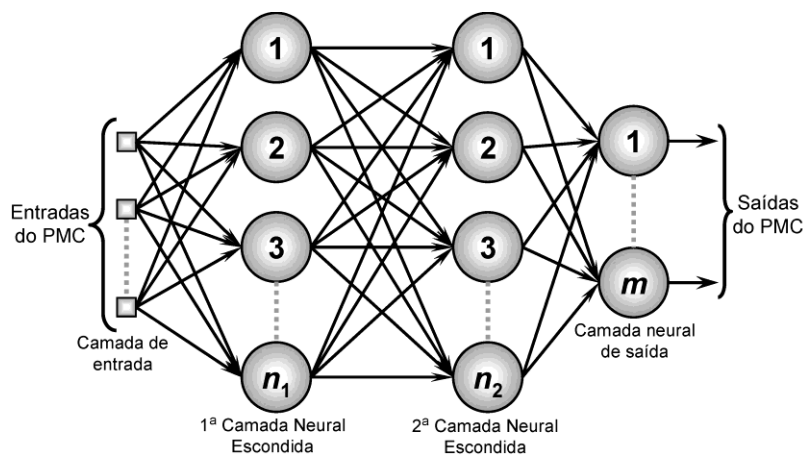


Fonte: Adaptado de Jang, Sun e Mizutani (1997, p.202).

2.5.5 Perceptron de múltiplas camadas (PMC)

As redes PMC possuem, além da camada de saída, no mínimo uma camada intermediária de neurônios com arquitetura *feedforward* (Figura 11), possibilitando o aprendizado de tarefas mais complexas, e podem ser empregadas para aproximação de funções e classificação de padrões, dentre outras aplicações. Esta rede apresenta também um alto grau de conectividade, ou seja, para alterar a conectividade é preciso modificar as conexões sinápticas ou seus pesos, e seus neurônios são modelados a partir de uma função de ativação não-linear totalmente diferenciável, como a função sigmoide (HAYKIN, 2007).

Figura 11 - Arquitetura de uma rede PMC



Fonte: Silva, Spatti e Flauzino (2010, p.92).

As entradas são propagadas para a primeira camada intermediária ou escondida, em seguida para a segunda e, se houver, para as seguintes, até a camada de saída, composta pela quantidade de neurônios tal qual for a quantidade de saídas. O treinamento desta rede é supervisionado. A topologia da rede pode ser variada,

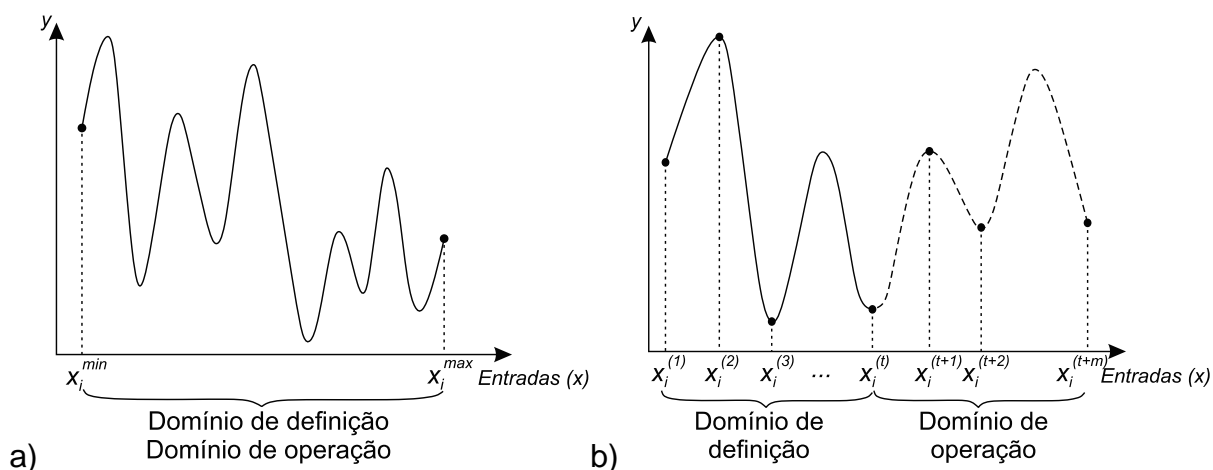
permitindo alteração do número de neurônios ou de camadas intermediárias, das funções de ativação dos neurônios e dos métodos de treinamento. A operação da rede PMC consiste em: a partir de uma amostra, assumindo os pesos sinápticos ajustados em treinamento, obter as respostas da(s) camada(s) intermediária(s) e, por fim, disponibilizar as saídas da rede (SILVA; SPATTI; FLAUZINO, 2010).

2.5.6 Aplicação da rede PMC em sistemas dinâmicos

As redes PMC podem ser aplicadas a sistemas que possuem o comportamento dependente do tempo, em que as entradas e saídas da rede em um determinado instante estão vinculadas aos seus valores em momentos anteriores (Aguirre, 2000; Ljung, 1998 apud SILVA, SPATTI E FLAUZINO, 2010, p.138).

Em um problema de aproximação de funções, o domínio de definição coincide com o domínio de operação, ou seja, tanto as entradas como as respostas da PMC estão compreendidas entre x_i^{min} e x_i^{max} , como mostra a Figura 12a. Quando se trata de uma aplicação em sistemas dinâmicos, ambos os domínios são variantes do tempo. Dessa forma, os parâmetros da rede são ajustados – com valores do domínio de definição – para que esteja apropriada a estimar valores futuros – no domínio de operação da Figura 12b.

Figura 12 - Domínios de definição e de operação de PMC para a) aproximação de funções e b) sistemas dinâmicos



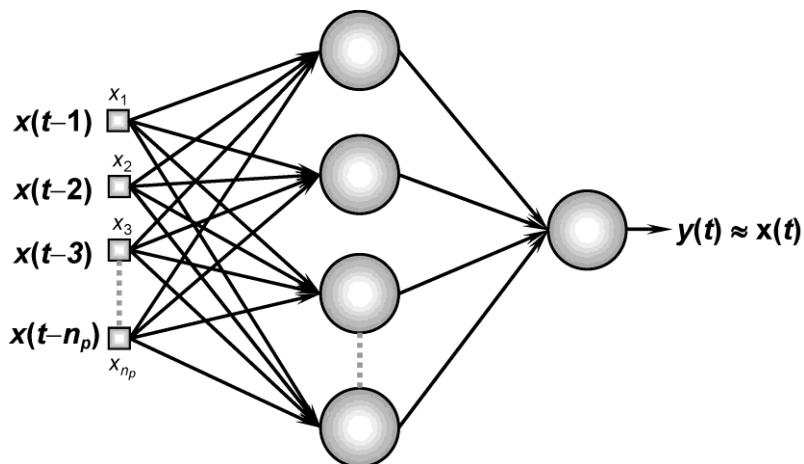
Fonte: Adaptado de Silva, Spatti e Flauzino (2010, p.138-139).

Visto que as saídas da rede PMC aplicadas a sistemas dinâmicos dependem das respostas anteriores, a rede pode ser feita de duas formas, de acordo com Silva, Spatti e Flauzino (2010): com saídas recorrentes às entradas ou com entradas atrasadas no tempo, sendo esta última a forma abordada neste estudo, uma vez que é preciso adaptar apenas a forma de entrada das amostras na rede para aplicá-la, sem a necessidade de modificar a estrutura da rede para uma rede recorrente.

2.5.6.1 PMC com entradas atrasadas no tempo

A rede PMC com entradas atrasadas no tempo possui arquitetura *feedforward* de múltiplas camadas sem realimentação da saída nas entradas. Na configuração da Figura 13, os atrasos temporais encontram-se apenas na camada de entrada, como uma memória temporal, designado TDNN com configuração focada ou concentrada (do inglês, *focused time-lagged feedforward network*), de acordo com Haykin (2007).

Figura 13 - PMC com entradas atrasadas no tempo



Fonte: Silva, Spatti e Flauzino (2010, p.140).

A predição de valores após o instante t é dada em função de valores antecedentes, sendo n_p a quantidade de amostras precedentes, conforme a Eq.15. O resultado $x(t)$ da predição é comparado com a saída $y(t)$ durante o treinamento, obtendo a Eq.16 de minimização do erro $E(t)$, em que N é a quantidade total de amostras sequencialmente coletadas (SILVA, SPATTI E FLAUZINO, 2010).

$$x(t) = f(x(t-1), x(t-2), \dots, x(t-n_p)) \quad (15)$$

$$E(t) = x(t) - y(t), \text{ onde } (n_p + 1) \leq t \leq N \quad (16)$$

Considerando uma amostra conforme a Eq.17 os dados são utilizados consoante uma operação janela que se movimenta em uma unidade a cada iteração de t , com uma largura de n_p amostras, de acordo com Silva, Spatti e Flauzino (2010) – como são mostradas as janelas destacadas no Quadro 5 para $n_p = 3$. Pode-se aumentar a ordem de predição (n_p) a fim de se obter maior precisão na previsão.

$$x(t) = [x(1) \ x(2) \ x(3) \ x(4) \ x(5) \ x(6)] \quad (17)$$

Quadro 5 - Operação janela deslizante

Janela	Tempo	$x(t)$					
Janela 1	$t = 3$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$
Janela 2	$t = 4$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$
Janela 3	$t = 5$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$
Janela 4	$t = 6$	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$

Fonte: Baseado em Silva, Spatti e Flauzino (2010).

À medida em que se executa a predição de valores futuros no domínio da operação, cada resultado vai sendo inserido como entrada no instante posterior. Para a Janela 4 do Quadro 5, a estimativa de $x(7)$ dada pelo resultado $y(7)$ é feita a partir de $\{x(4) \ x(5) \ x(6)\}$. No instante posterior ($t = 7$), para a estimativa de $x(8)$ a Janela 5 seria composta por $\{x(5) \ x(6)\}$ e também pelo valor estimado anteriormente $x(7)$.

2.5.7 Técnicas de treinamento

Existem diversas técnicas para o treinamento de redes neurais, divididas em treinamento supervisionado ou não supervisionado, que consistem respectivamente em ter o conhecimento das entradas e saídas desejadas ou utilizar apenas os dados de entrada para o treinamento. Por este trabalho ter como objetivo a utilização de um treinamento supervisionado, visto que são conhecidas as entradas e saídas desejadas da rede neural, apenas estes serão abordados (SILVA, SPATTI E FLAUZINO, 2010).

O processo de treinamento de uma RNA consiste na atualização dos pesos sinápticos até que um critério de parada estabelecido seja alcançado. Cada iteração

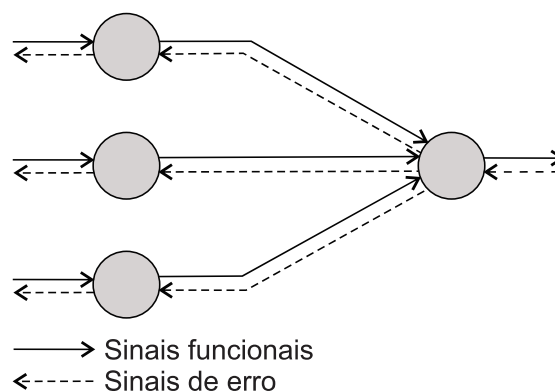
deste processo é denominada época, ou seja, uma época corresponde à cada utilização de todo o conjunto de amostras de treinamento para a alteração dos pesos sinápticos. Dessa forma, um dos critérios de parada pode ser o número de épocas, mas podem ser empregados outros critérios simultaneamente, como alcançar determinado valor de erro médio quadrático, em busca da convergência da RNA (HAYKIN, 2007).

O passo a passo dos cálculos necessários para a aplicação dos treinamentos supervisionados *Backpropagation* e *Momentum* encontra-se no APÊNDICE I.

2.5.7.1 *Backpropagation*

Segundo Silva, Spatti e Flauzino (2010), o algoritmo *backpropagation* é um algoritmo pela qual uma rede neural *Perceptron* é treinada. Esse algoritmo também é conhecido pela regra delta generalizada e se dá por duas fases bem específicas: a fase de propagação adiante (*forward*) e a fase de propagação reversa (*backward*), correspondentes respectivamente aos sinais funcionais e aos sinais de erro da Figura 14.

Figura 14 - Sinais funcionais e de erro de uma rede PMC



Fonte: Haykin (2007).

A derivação da regra delta generalizada tem uma função de custo para ser minimizada, que é a função da soma do erro quadrático médio. Matematicamente, ela é expressa pela Eq.18 (BRAGA; CARVALHO; LUDERMIR, 2007).

$$E = \frac{1}{2} \sum_p \sum_{i=1}^k (d_i^p - y_i^p)^2 \quad (18)$$

onde E é o erro total, p é número total de amostras destinadas para o treinamento da rede, k corresponde ao número de neurônios na camada de saída ou da última camada neural, d_i é a i -ésima saída desejada e y_i é a i -ésima saída gerada pela rede neural.

A fase *forward* ou fase da propagação adiante é definida pela resposta da rede neural frente aos sinais de entrada provenientes do conjunto de amostras para o treinamento da rede. Como o processo de aprendizagem da rede é supervisionado, as respostas da rede são comparadas com as respostas desejadas. De acordo com Braga, Carvalho e Ludermir (2007), essa fase pode ser mais explicitada pelos seguintes passos:

- 1) Os sinais de entrada são recebidos pela primeira camada da rede neural;
- 2) As repostas geradas na primeira camada são conectados às entradas da segunda camada neural, e as saídas da segunda camada são conectadas as entradas da próxima camada e assim por diante;
- 3) As respostas da última camada neural são comparadas às saídas desejadas.

Em função do erro gerado entre as respostas da última camada com os sinais desejados, a fase *backward* ou propagação reversa se inicializa, na qual os pesos sinápticos da rede neural são alterados, de forma que os primeiros pesos ajustados são os da última camada neural. Esses pesos já ajustados servem para o ajuste dos pesos das camadas anteriores, e assim sucessivamente até a primeira camada neural. De forma mais explícita, os passos dessa fase são determinados por (SILVA, SPATTI E FLAUZINO, 2010; BRAGA, CARVALHO E LUDERMIR, 2007):

- 1) A partir da última camada, até a primeira camada
 - i. Ajuste dos pesos para redução dos seus erros;
 - ii. Ponderar os pesos da próxima camada, utilizando o cálculo do erro da camada seguinte, que está conectado a ele.

A essência do algoritmo *backpropagation* encontra-se justamente na segunda fase, pois é onde o erro da última camada com a resposta desejada é retropropagado para as camadas anteriores, ponderando-os com os valores dos pesos previamente

ajustados. Em termos gerais, o algoritmo *backpropagation* segue as etapas (SILVA, SPATTI E FLAUZINO, 2010; BRAGA, CARVALHO E LUDERMIR, 2007):

- 1) Definir os pesos iniciais e os parâmetros da rede;
- 2) Repetir até o erro ser mínimo ou até uma determinada quantidade de iterações:
 - i. Para cada amostra do conjunto de treinamento;
 - ii. Definir a saída da rede com a fase *forward*;
 - iii. Comparar as saídas produzidas com as desejadas;
 - iv. Ajustar os pesos das camadas através da fase *backward*.

2.5.7.2 Momentum

É o termo $\delta(W_{atual} - W_{antigo})$ inserido na equação de atualização dos pesos:

$$W_{novo} = W_{atual} + \eta \frac{dE}{dW} + \delta(W_{atual} - W_{antigo}) \quad (19)$$

$$W_{novo} = W'_{novo} + \delta(W_{atual} - W_{antigo}) \quad (20)$$

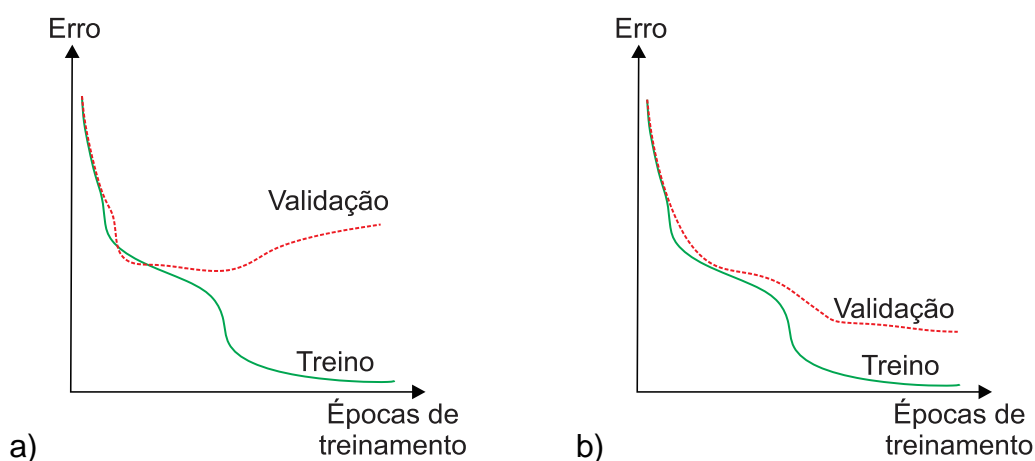
sendo δ definido como a taxa de momentum entre 0 e 1. Este é um método mais simples de ser efetuado no algoritmo. A variação dos pesos de acordo com uma taxa é somada com W'_{novo} . O momentum acelera a rede numa razão de $\eta/(1 - \delta)$. Silva; Spatti e Flauzino (2010) recomendam que δ esteja entre 0,05 e 0,75 e η esteja entre 0 e 0,9.

2.5.8 Desempenho das RNAs

O sobre-ajuste e o sobre-treinamento são parâmetros relacionados ao desempenho de uma RNA quanto à sua capacidade de generalização, visto que o objetivo é que a rede neural seja aplicada para amostras desconhecidas e diferentes das utilizadas no treinamento. Independentemente da regra de decisão utilizada, o desempenho da RNA depende da quantidade de amostras, dos valores dessas amostras, da arquitetura da rede neural e da complexidade do problema.

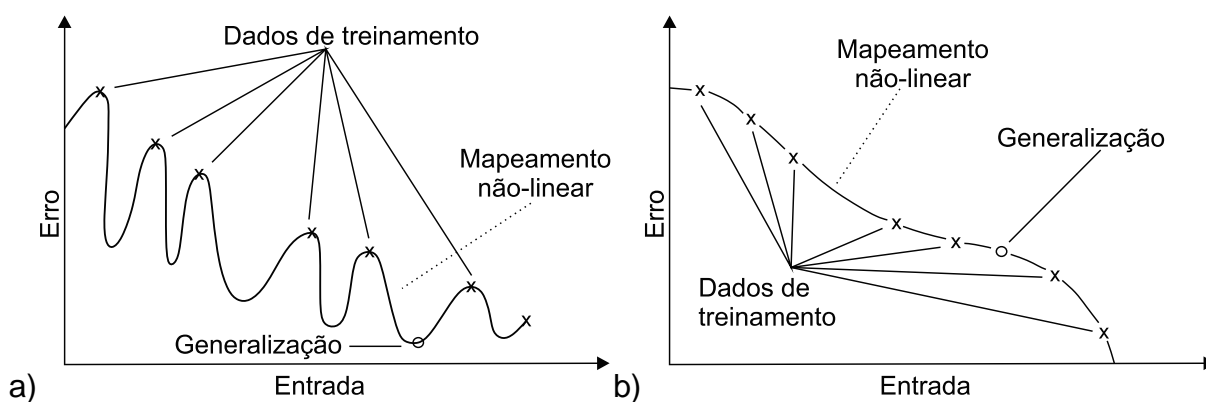
O sobre-ajuste, do inglês *overfitting*, ocorre quando o classificador fica mais complexo que o demandado e perde poder de generalização, como na Figura 15a. Isto tende a acontecer se a amostra é composta por uma grande quantidade de parâmetros livres – ou seja, fundamentais no ajuste de decisão da rede. O sobre-treinamento, do inglês *overtraining*, é decorrente da utilização de uma grande amostra de treinamento com pouca variação de valores, ou de um número elevado de épocas de treinamento, sucedendo também na perda de capacidade de generalização da rede, como mostra a Figura 16a (JAIN, DUIN E JIANCHANG MAO, 2000).

Figura 15 - Curva do erro de treinamento a) com sobre-ajuste e b) normal



Fonte: Simas (2011).

Figura 16 - a) Ajuste em excesso e b) boa generalização



Fonte: Haykin (2007).

Para encontrar valores satisfatórios para as taxas de aprendizagem (η) e de momentum (δ), busca-se atender a um dos critérios estabelecidos com o menor número de épocas: taxas que em média confluem para a topologia de rede com

melhor generalização; taxas que em média conduzem para um mínimo local do erro; e taxas que no pior caso ou em média levam à convergência para um mínimo global do erro – que é a opção mais interessante para uma pesquisa, segundo Haykin (2007).

Com o objetivo de se obter o tamanho suficiente do conjunto de amostras para treinamento para uma boa generalização, ou o tamanho do conjunto de treinamento sofre alteração e a arquitetura da rede é mantida; ou o tamanho do conjunto é fixo e pretende-se encontrar a melhor arquitetura (HUSH, HORNE, 1993 apud HAYKIN, 2007).

2.5.9 Divisão dos dados para treinamento

O conjunto de dados de treinamento pode ser subdividido em dois conjuntos – treino e teste – ou em três conjuntos – treino, validação e teste –, sendo destinado a cada um deles um determinado percentual de amostras para que se possa iniciar o processo de treinamento da RNA.

O conjunto de validação deve ser empregado quando se pretende utilizar o critério de parada antecipada do treinamento, que objetiva garantir a generalização da rede. Para isso, as amostras destinadas à validação devem ser representativas de todo o conjunto de amostras de treinamento (MATHWORKS, 2018).

Em MathWorks (2018), 90% dos dados foram destinados para teste e 10% para treino no exemplo dado. Segundo Simas (2011), quando há muitas amostras disponíveis para o treinamento, deve-se separar um terço para cada um dos três conjuntos. Entretanto, para poucas amostras, 50% deve ser destinada ao treino, 25% à validação e 25% ao teste. Já para Louzada (2013), 70% dos dados normalmente são reservados para o treino, 20% para a validação e 10% para o teste.

Não há um consenso sobre o percentual destinado a cada conjunto, pois este número pode variar de acordo com a quantidade de dados e da abrangência dos valores de amostras. É aconselhado por Silva, Spatti e Flauzino (2010) que 60 a 90% dos dados sejam destinados aleatoriamente ao conjunto de treinamento.

2.5.10 Escolha da rede neural

De acordo com Silva, Spatti e Flauzino (2010), a determinação da uma topologia, ou seja, a quantidade de neurônios e número de camadas intermediárias de uma rede neural é geralmente feita empiricamente, visto que dependem do problema abordado, do algoritmo de aprendizado, da disponibilidade dos pesos iniciais e da distribuição das amostras no espaço.

Pode haver problemas que duas camadas intermediárias favorecem o treinamento da rede neural. Mas, por outro lado, uma grande quantidade de camadas intermediárias pode tornar a rede menos precisa (BRAGA; CARVALHO; LUDERMIR, 2007).

Segundo Braga, Carvalho e Ludermir (2007), existem dois métodos mais utilizados para definir o número de neurônios nas camadas intermediárias, que são:

- a) Determinar a quantidade de neurônios através da relação do número de entradas e de saídas;
- b) Empregar um número de conexões dez vezes menor que o número de amostras.

Conforme Silva, Spatti e Flauzino (2010), uma técnica estatística para a escolha da topologia é o *Cross-validation* (Validação Cruzada), que é uma técnica de avaliação de desempenho das topologias candidatas frente ao conjunto de amostras. Mas essa técnica subdivide-se em três outros métodos:

- a) Validação Cruzada por Amostragem Aleatória (*Random Subsampling Cross-validation*)

Um conjunto de amostras é dividido, aleatoriamente, em dois grupos, o subconjunto teste e o subconjunto treinamento. Para esse último, é abrangido 60 a 90% do conjunto amostral. Depois, cada rede neural é treinada várias vezes e, em seguida, obtido o desempenho médio dela. A topologia que apresentar o maior desempenho será escolhida para a solução do problema.

b) *K-fold Cross-validation* (Validação Cruzada com k-partições)

O conjunto amostral é dividido em k partições, sendo que $(k-1)$ dos dados amostrais é destinado para o subconjunto teste e o restante para o subconjunto treinamento. Em seguida, o treinamento das topologias repete em k vezes, até que todas as partições tenham participado do subconjunto teste. O valor de k é compreendido entre 5 a 10. Do mesmo modo que o primeiro método, o desempenho global é obtido e avaliado.

c) *Leave-one-out Cross-validation* (Validação Cruzada por Unidade)

Utiliza-se apenas uma única amostra para o subconjunto teste, e os demais valores do conjunto amostral são destinados ao subconjunto de treinamento. Então, a rede é treinada até que todas as amostras tenham sido utilizadas no teste. Em seguida o desempenho da rede é retirado e avaliado.

Segundo Silva, Spatti e Flauzino (2010), o passo a passo para a implementação do *Cross-validation* abrangendo os três métodos citados é:

- i. Definir as topologias candidatas para o problema;
- ii. Formar o subconjunto teste e treinamento;
- iii. Treinar todas as topologias da rede candidatas, utilizando o subconjunto de treinamento;
- iv. Aplicar o subconjunto teste às topologias para a avaliação do desempenho das redes;
- v. Obter o desempenho final de cada topologia em função dos ensaios realizados;
- vi. Selecionar a topologia que obteve o melhor desempenho final;
- vii. Se o desempenho final da topologia não está de acordo com a precisão do problema (como por exemplo a taxa de acerto da rede), definir novas topologias e voltar ao passo (iii). Se esse desempenho estiver com a precisão requerida, finalizar o processo.

Em contrapartida ao método *Cross-validation*, existem duas regras para determinar a topologia de uma rede neural com apenas uma camada intermediária, apropriadas para problemas bem comportados (SILVA; SPATTI; FLAUZINO, 2010):

a) Método de Kolmogorov

$$n_1 = 2 \cdot n + 1 \quad (21)$$

b) Método de Fletcher-Gloss

$$2 \cdot \sqrt{n} + n_2 \leq n_1 \leq 2 \cdot n + 1 \quad (22)$$

onde n é o número de entradas da rede, n_1 é número de neurônios na camada intermediária e n_2 é o número de neurônios na camada de saída.

2.5.11 Intervalo de confiança de uma Rede Neural Artificial

Intervalo de confiança se relaciona com nível de confiança que representa uma estimativa da certeza de que o intervalo de confiança contém o verdadeiro valor de um parâmetro dado. O nível de confiança é dado em percentual e normalmente são adotados os valores de 90%, 95% e 99%. Ao lançar mão dos conceitos de redes neurais é comum o uso de diferentes métodos para a definição do intervalo de confiança visando garantir qualidade de seu poder preditivo (IZBICKI, SANTOS, 2018).

Um dos principais métodos para o cálculo do intervalo de confiança para redes neurais, que é fundamental para técnicas computacionais relacionadas à inteligência artificial e mineração de dados, é o da máxima verossimilhança. Esse método leva em conta que existem duas fontes de incertezas para um modelo preditivo. Seriam eles a variância do ruído (σ_v^2) e a variância da incerteza (σ_w^2) sendo que a primeira representa a variância dos erros do conjunto de validação cruzada no treinamento da rede e o outro é a variância da incerteza do modelo (MENEZES, VALENÇA, 2016).

Assumindo que as duas são independentes, a variância total pode ser calculada pela Eq.23.

$$\sigma_{total}^2 = \sigma_v^2 + \sigma_w^2 \quad (23)$$

No processo de validação cruzada calcula-se σ_v pela Eq.24.

$$\sigma_v = \frac{1}{n-1} \sum_{i=1}^n (Erro_i - \widehat{Erro})^2 \quad (24)$$

em que n é a quantidade de valores, $Erro_i$ é o erro referente a entrada i , \widehat{Erro} é a média dos erros, e σ_w^2 é calculado a partir dos erros para todas as entradas do treinamento da rede. Estes erros são armazenados e usados como o valor desejado de uma nova rede e, então, é repetido o processo de cálculo do σ_v para essa nova rede. Com os valores de σ_v^2 e σ_w^2 é possível calcular o σ_{total}^2 , que será usado para encontrar o intervalo de confiança a partir da Eq.25.

$$f(x) - t * \sigma_{total}^2 < f(x) < f(x) + t * \sigma_{total}^2 \quad (25)$$

em que $f(x)$ é o valor calculado pela rede e t é o valor extraído da tabela de *t Student* para o maior grau de liberdade possível e um intervalo definido que geralmente é 90%, 95% ou 99% de chance de conter o valor real.

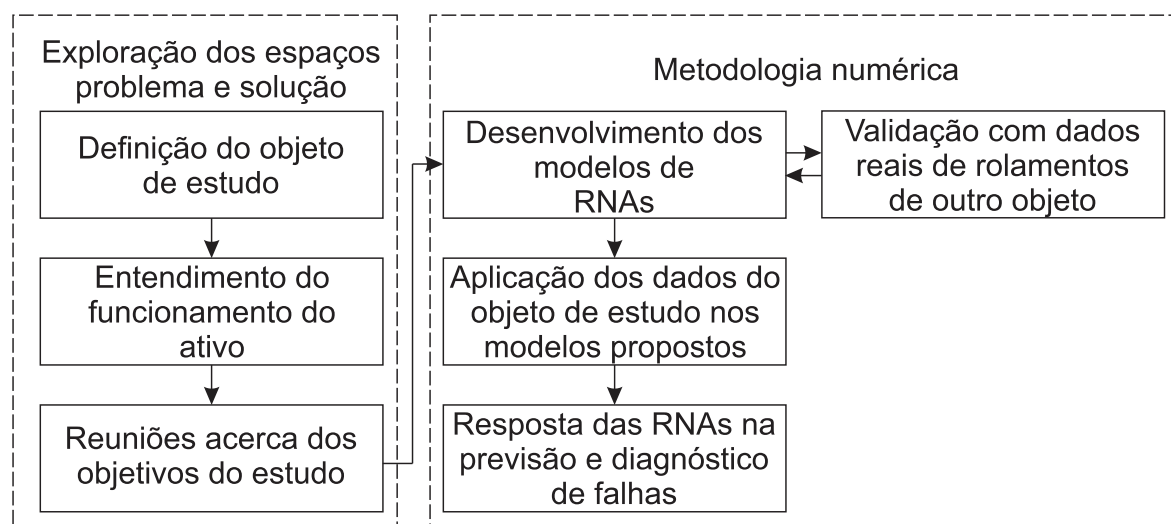
3 METODOLOGIA

Este capítulo contém as etapas seguidas para se alcançar os objetivos propostos pela pesquisa. São apresentados os fluxogramas do curso de trabalho, com ênfase no processo de modelagem da rede neural.

3.1 Etapas do trabalho

A metodologia deste estudo seguiu as etapas descritas no fluxograma da Figura 17, a partir da exploração dos espaços problema e solução, em que fazem parte os objetivos, metodologia e resultados que se pretende alcançar, e do aprofundamento na metodologia numérica, com o desenvolvimento dos modelos de RNAs para diagnóstico e previsão de falhas.

Figura 17 - Fluxograma do curso de trabalho



Fonte: Elaborada pela autora.

A definição do objeto de estudo ocorreu de acordo com as necessidades da indústria, sendo escolhido o Virador de Vagões, por ser o ativo que apresentou o maior custo e tempo de paradas em manutenções nos últimos dois anos dentre os equipamentos portuários de mineração, de acordo com o histórico de falhas analisado. Em seguida, foi realizada uma visita *in loco*, reuniões com especialistas e análise do histórico do ativo para entender suas condições de operação. Para aplicação da metodologia, foram escolhidos pelos especialistas os motores do sistema de

Alimentadores do VV. O banco de dados com o histórico das medições obtidas pelo sensoriamento do ativo foi utilizado para alimentar os modelos de rede neural desenvolvidos.

3.2 Banco de dados do motor do Alimentador do Virador de Vagões

O banco de dados do motor dos Alimentadores é composto por medições vibracionais e dados operacionais retirados do PIMS (Quadro 6). Os dados vibracionais apresentam intervalo de coleta aproximadamente semanal, compreendidos entre Julho de 2017 e Agosto de 2018. Os dados operacionais foram retirados com o mesmo intervalo de coleta deste mesmo período, para que pudessem ser utilizados em conjunto como entradas das RNAs. Em complemento a estes dados, foi acrescentado o tempo corrido entre paradas para manutenção do componente. Foram disponibilizados também documentos relativos às ordens de manutenção no ativo, histórico de trocas e previsão de manutenção.

Quadro 6 - Variáveis do motor dos Alimentadores

LOCAL DA MEDIÇÃO	VARIÁVEL	TIPO DE VARIÁVEL
Lado acoplado do motor do alimentador	Envelope	Vibração
	Velocidade	
	Aceleração	
Motor do alimentador	Rotação	Operação
	Corrente	
	Velocidade de referência	
	Horímetro	
	Tempo até parada	Manutenção

Fonte: Elaborada pela autora.

O banco de dados foi usado de diferentes formas com a finalidade de comparar os resultados das RNAs, no que diz respeito ao número de épocas (iterações), erro quadrático médio, compreendendo as seguintes configurações de entrada:

- 8 variáveis de entrada, com a ordem de grandeza original (Alimentador 1);
- 5 variáveis de entrada, com a ordem de grandeza original, excluindo-se as variáveis que não apresentaram tendência de alteração dos valores ao longo do tempo (Ambos os alimentadores);

- 5 variáveis de entrada, como na configuração anterior, com a ordem de grandeza normalizada, ou seja, os dados foram tratados para que a variação dos valores permanecesse de 0 a 1, e variável Horímetro tratada para considerar a diferença de horas entre medições em vez da hora absoluta da medição (Ambos os alimentadores).

3.3 Modelos das RNAs

Foram desenvolvidos em Matlab® três modelos de RNAs distintos no que se refere ao resultado derivado de cada um deles. Os modelos possuem em comum a arquitetura da rede – *feedforward*, de múltiplas camadas – e os treinamentos aplicados – *Backpropagation* e *Momentum*. Os modelos são:

- Classificação de padrões, cuja resposta da rede é a classificação da falha, ou seja, se há ou não há a ocorrência no momento atual de determinada(s) falha(s) em um ativo;
- Aproximação de funções, cuja resposta é uma faixa de valores que indicam a criticidade de uma falha em um ativo no momento atual;
- Rede com entradas variantes no tempo, em que é feita como resposta uma previsão de um momento posterior a partir de entradas de momentos anteriores.

Nos modelos Classificador e Aproximador foram aplicados dados referentes a medições vibracionais em rolamentos, coletados experimentalmente pela Case Western Reserve University¹ e tratados com a técnica do envelope e fatores estatísticos (RMS, Fator de crista e Fator K) por Alvarenga, Rodrigues e Silva (2017). Estes dados reais de rolamentos foram utilizados para validação da programação da rede, no que tange ao treinamento e teste. Estes resultados foram comparados com os obtidos pelo *Neural Net Fitting tool*, ferramenta pronta do Matlab® para modelagem de Redes Neurais Artificiais. Como os três modelos propostos compartilham da mesma lógica de programação para treinamento e teste, com mesma arquitetura e funções de ativação, alterando-se apenas a quantidade de dados de entrada e a

¹ Disponibilizado por Bearing Data Center Case Western Reserve University ([s.d.]).

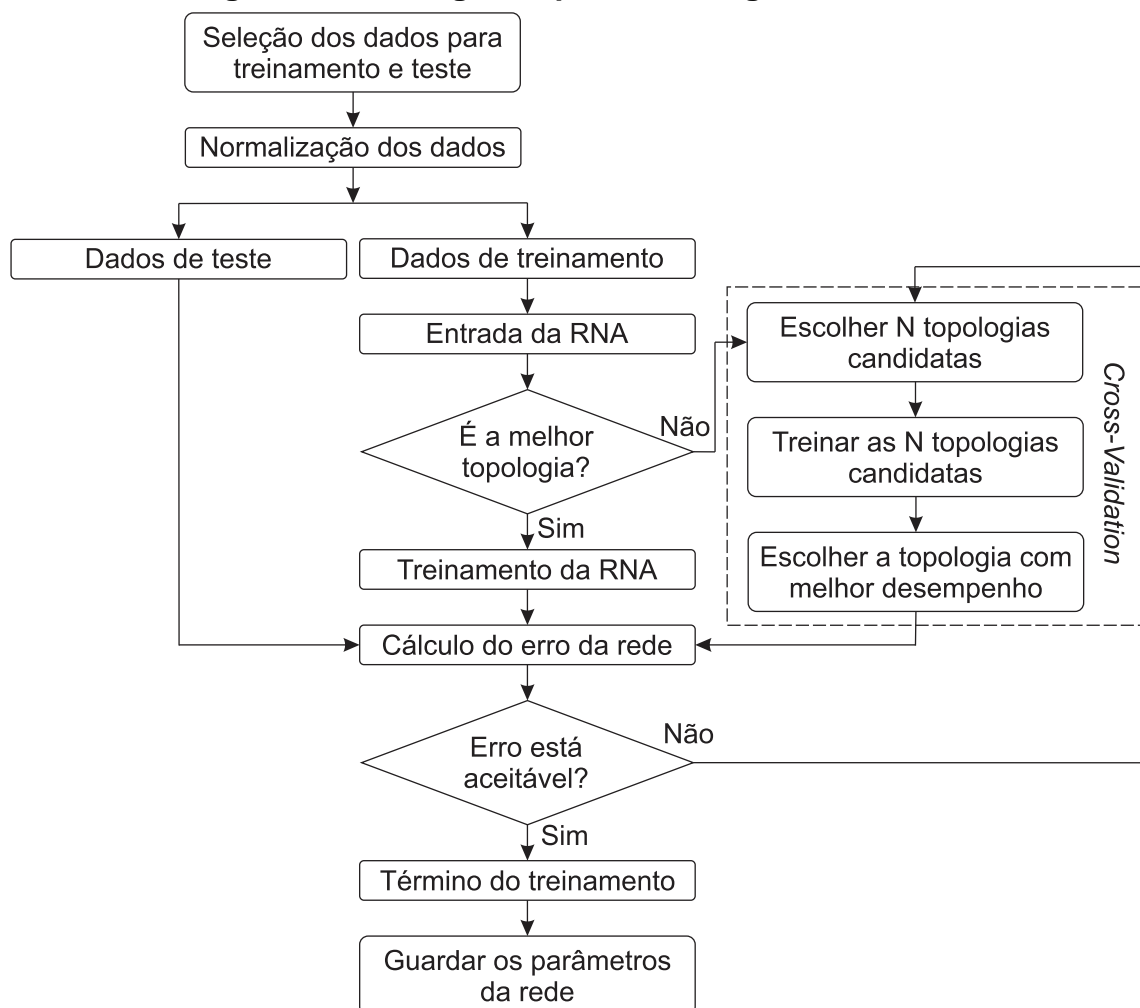
interpretação da saída, esta avaliação da programação da rede foi considerada para todos os modelos.

Para aplicação das amostras do banco de dados com o histórico de medições vibracionais e operacionais dos motores dos Alimentadores do VV, foi avaliada a possibilidade de utilização para todos os modelos, visando a obtenção de uma resposta completa acerca do objeto de estudo: classificação e criticidade de falhas em um momento atual, bem como a previsão da necessidade de uma parada para manutenção. Foram feitas comparações entre treinamentos com banco de dados utilizado das três diferentes maneiras já apresentadas, e também mantendo o banco de dados e alterando apenas o tamanho da janela deslizante da rede em 2, 3 e 4.

3.4 Metodologia de implementação das RNAs

Para a modelagem da RNA (Figura 18), os dados do sensoriamento foram normalizados para que a faixa de variação de seus valores fosse contida no intervalo entre 0 e 1, visando otimizar seu processamento, e foram feitos treinamentos e testes com diferentes topologias de redes e métodos de treinamento (*Cross-Validation*), a fim de escolher a topologia com melhor taxa de acerto para ser utilizada no estudo (SILVA; SPATTI; FLAUZINO, 2010).

Figura 18 - Fluxograma para modelagem das RNAs

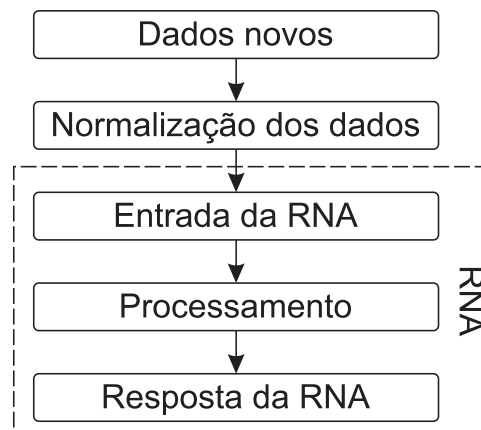


Fonte: Elaborada pela autora.

Após definidos e normalizados os dados históricos conhecidos para entrada e saída da rede, estes foram devidamente selecionados e organizados, de forma que um percentual foi destinado ao treinamento e o restante ao teste. Para encontrar a melhor topologia de rede neural, foi utilizado o método *Cross-validation*, que consiste na escolha e no treinamento de diversas topologias de redes neurais, com diferentes características, como, por exemplo, número de camadas, número de neurônios em cada camada e método de treinamento empregado. Por fim, os dados para teste foram usados no cálculo do erro da rede neural com a melhor topologia. Assim que encontrado um resultado satisfatório, o treinamento é encerrado e os parâmetros do modelo da rede neural selecionada são guardados. Caso contrário, são testadas novas topologias.

Após modelada a rede neural, a operação do sistema (Figura 19) consiste em entrar com as amostras do banco de dados normalizadas na RNA com os parâmetros previamente ajustados, onde são processados e resultam, como saída, na classificação, criticidade ou previsão do modo de falha, de acordo com o modelo.

Figura 19 - Operação das RNAs



Fonte: Elaborada pela autora.

O banco de dados vibracionais e de operação do ativo foram utilizados para o treinamento dos modelos, trazendo uma noção do avanço do desgaste no ativo. Então, devido às RNAs serem aplicadas na classificação de padrões, aproximação de função e para previsão, a partir da análise não-linear de um extenso banco de dados para extração de novas informações, foram escolhidas as redes PMC. Este tipo de RNA pode ser usada como classificadora de função, aproximadora universal e para previsões temporais (TIAN, 2009; JAIN, DUIN E JIANCHANG MAO, 2000; HAYKIN, 2007).

As amostras foram normalizadas para que dados de maior grandeza não mascarem dados de menor grandeza, haja visto que o conjunto de dados é constituído por variáveis de diferentes ordens de grandeza. Segundo Silva, Spatti e Flauzino (2010), a normalização evita a saturação das funções de ativação dos neurônios.

Após a normalização, foram destinadas pseudoaleatoriamente, pelo comando "rand" do Matlab®, um percentual do número de amostras para os subconjuntos de treino e teste, sendo entre 60 e 90% das amostras para treino e o restante para teste, consoante o tamanho do conjunto de treinamento disponível (SIMAS, 2011; SILVA, SPATTI, FLAUZINO, 2010).

Como, a princípio, a melhor topologia de RNA é desconhecida, foi aplicado o método *Cross-validation* para encontrá-la. As diversas topologias abarcaram:

- Rede neural PMC;
- Duas técnicas de treinamento: *Backpropagation* e *Momentum*;
- Uma ou duas camadas intermediárias;
- Variação do número de neurônios em cada camada intermediária, segundo método de Fletcher-Gloss para redes de uma camada;
- Variação das taxas de aprendizado (η) de 0 a 0,9 e, quando aplicável, variação da taxa de *Momentum* (δ) de 0,05 até 0,75, de acordo com Silva, Spatti e Flauzino (2010, p.162).

Durante o treinamento de cada RNA ocorreu o ajuste dos pesos dos neurônios, conforme a técnica utilizada. O critério de parada deste processo foi primeiramente o erro quadrático médio (MSE), obtido pelas amostras destinadas ao teste, serem inferiores a um valor determinado. Quando este critério não for atendido, o processo termina ao atingir o número máximo de épocas estipulado.

O método *Cross-Validation* por amostragem aleatória foi utilizada para determinar a RNA com melhor desempenho após o processo de treinamento. Para isso, foi obtida a média dos valores de MSE de cinco treinamentos de cada topologia. Em seguida, a topologia com menor MSE e maior média de acerto foi treinada mais dez vezes novamente, visando obter a RNA com o melhor desempenho individualmente (MATHWORKS, 2018).

A RNA com melhor desempenho foi utilizada para classificação das falhas, aproximação de funções ou previsão temporal para o ativo. Cada amostragem de dados de entrada deve ser normalizada, de maneira semelhante à fase de treinamento. Para cada amostra de entrada, a RNA faz o processamento e, em seguida, apresenta como resultado a classificação de falhas, criticidade de falhas ou previsão temporal, de acordo com o propósito da rede.

3.5 Metodologia de implementação do intervalo de confiança

Após da escolha da melhor topologia, foi calculado o intervalo de confiança a 95% da RNA para as fases de treinamento e de teste com o método da Máxima

Verossimilhança, a fim de permitir a avaliação quantitativa do resultado fornecido pela rede. A implementação dos intervalos de confiança de treinamento e teste segue os seguintes passos (MENEZES, VALENÇA, 2016):

1. Dividir a base de dados em treinamento e teste
2. Normalizar dados
3. Treinar a rede
4. Calcular a variância do erro σ_v^2 pela Eq.(24)
5. Desnormalizar dados
6. Calcular os erros $Erro_i = (Calculado_i - Desejado_i)$
7. Normalizar dados
8. Treinar nova rede com mesmas entradas e as saídas sendo os erros
9. Calcular a variância do erro da segunda rede σ_w^2 pela Eq.(24)
10. Desnormalizar dados
11. Calcular $\sigma_{total}^2 = \sigma_v^2 + \sigma_w^2$
12. Selecionar o grau de confiança e seu valor apropriado na tabela t student
13. Calcular o intervalo para os valores obtidos com os dados de teste, conforme Eq.(25)

4 RESULTADOS

Este capítulo apresenta os resultados deste estudo e está dividido em etapas relacionadas ao fluxo de trabalho descrito no Capítulo 3, que contém a metodologia. São expostos os resultados das redes neurais primeiramente com dados de rolamentos não relacionados ao objeto de estudo, para verificar sua eficácia quanto ferramenta. Em seguida, as redes neurais são aplicadas ao objeto de estudo, com histórico de dados referentes ao Virador de Vagões.

4.1 Rede neural artificial PMC programada

Foram feitas três redes PMC distintas: Classificadora de Padrões, Aproximadora de Funções e com entradas variantes no tempo. As três redes são treinadas com o mesmo algoritmo, e os dados entram da mesma forma: os dados são subdivididos de forma aleatória entre os conjuntos de treinamento e teste conforme percentual escolhido para cada conjunto.

Os modelos de redes neurais artificiais foram programados em Matlab®. As redes Classificadora de Padrões e Aproximadora de Funções foram testadas inicialmente com dados coletados em rolamentos por um experimento da Case Western Reserve University, que consistiu em induzir falhas nas pistas interna e externa e no elemento rolante de rolamentos e fazer coletas vibracionais no momento da ocorrência dessas falhas para três tamanhos de trincas nestes componentes. As medições vibracionais foram tratadas no estudo de Alvarenga, Rodrigues e Silva (2017), resultando em um banco de dados com 80 amostras referentes à existência e ao tamanho de falhas em elementos de rolamento.

O treinamento das redes foi feito primeiramente com estes dados, que não têm relação direta com o objeto de estudo, portanto, foram treinadas com a finalidade de testar o código feito em Matlab® e, principalmente, de observar a capacidade de generalização das redes.

4.1.1 Rede Classificadora de Padrões

As RNAs foram treinadas com um banco de dados de 80 amostras de histórico de falhas em rolamentos, conforme a Tabela 1. As entradas são dados de falha que

advém de sinais vibracionais medidos nos elementos do rolamento para falha na pista interna, externa ou no elemento rolante, que são as saídas da rede. Desta forma, foram seis as entradas para a rede neural e três as saídas correspondentes – sendo 0 uma situação sem falha no respectivo componente e 1 uma situação de falha. A Pista Interna, a Pista Externa e o Elemento Rolante referem-se à aceleração dos acelerômetros instalados nas respectivas partes do rolamento, enquanto o valor RMS, Fator de Crista e Fator K foram utilizados durante o processamento por Alvarenga, Rodrigues e Silva (2017) para identificar a existência de falhas.

Tabela 1 - Exemplos de amostras para rede Classificadora de Padrões

Entradas						Saídas (Falha)		
Pista Interna (g)	Pista Externa (g)	Elemento Rolante (g)	RMS (g)	Fator de Crista (g)	Fator K (g)	Pista Interna	Pista Externa	Elemento Rolante
0,00365	0,08816	0,00315	0,76926	4,35356	2,57625	0	1	0
0,00988	0,00414	0,00232	0,0936	4,22926	0,03705	0	1	0
0,04802	0,03105	0,01478	0,16307	11,3706	0,30236	1	0	0
0,01278	0,02896	0,00225	0,11794	4,88655	0,06797	0	0	1

Fonte: Dados retirados de Alvarenga, Rodrigues e Silva (2017).

A escolha da quantidade de neurônios na camada intermediária ocorreu segundo a Eq. 22, definindo-se assim que, para uma camada intermediária, a variação de neurônios n_1 foi de 8 até 13 (Eq.26). Para duas camadas intermediárias foi utilizada a mesma variação do número de neurônios em cada camada.

$$2.\sqrt{6} + 3 \leq n_1 \leq 2.6 + 1 \therefore 7,89 \leq n_1 \leq 13 \quad (26)$$

Foram testadas topologias de redes neurais classificatórias PMC com as seguintes características já descritas na metodologia: uma ou duas camadas intermediárias com as possíveis combinações de quantidade de neurônios por camada; dois tipos de treinamento – *Backpropagation* e *Momentum* –, variação da taxa de aprendizado e, quando aplicável, da taxa de *Momentum*.

Para cada topologia foi feita a média do resultado de cinco treinamentos com valores de época, ou seja, quantas iterações foram necessárias até a convergência, e o erro quadrático médio (MSE) de teste. O critério de parada do treinamento está relacionado ao erro quadrático médio durante o treinamento e à quantidade de épocas

realizadas. A taxa de acerto foi calculada com base na validação dos resultados com o banco de dados utilizado. Foram direcionadas aleatoriamente 75% das amostras para treinamento e 25% para teste, de acordo com o critério sugerido por Silva, Spatti e Flauzino (2010).

Foram treinadas 3.780 topologias de redes para a combinação das condições estipuladas, sendo 324 redes com uma camada e 3.456 com duas camadas intermediárias, pois a rede de duas camadas possui mais possibilidades de combinação. A Tabela 2 e a Tabela 3 apresentam os valores da média de 5 treinamentos feitos com cada uma das melhores topologias para uma e duas camadas intermediárias, respectivamente.

Tabela 2 - Melhores topologias da rede Classificadora de Padrões com uma camada intermediária

Tipo de Treinamento	Neurônio/camada	Taxa de aprendizado	Taxa de <i>Momentum</i>	Épocas	MSE Teste	Taxa de acerto (%)
<i>Momentum</i>	12	0,1	0,15	3030,6	0,152	78
<i>Backpropagation</i>	11	0,3	-	1576,8	0,131	78
<i>Backpropagation</i>	8	0,1	-	2320,2	0,131	77
<i>Momentum</i>	12	0,1	0,15	2317,2	0,140	76
<i>Momentum</i>	10	0,1	0,45	3402,8	0,159	76

Fonte: Elaborada pela autora.

Tabela 3 - Melhores topologias da rede Classificadora de Padrões com duas camadas intermediárias

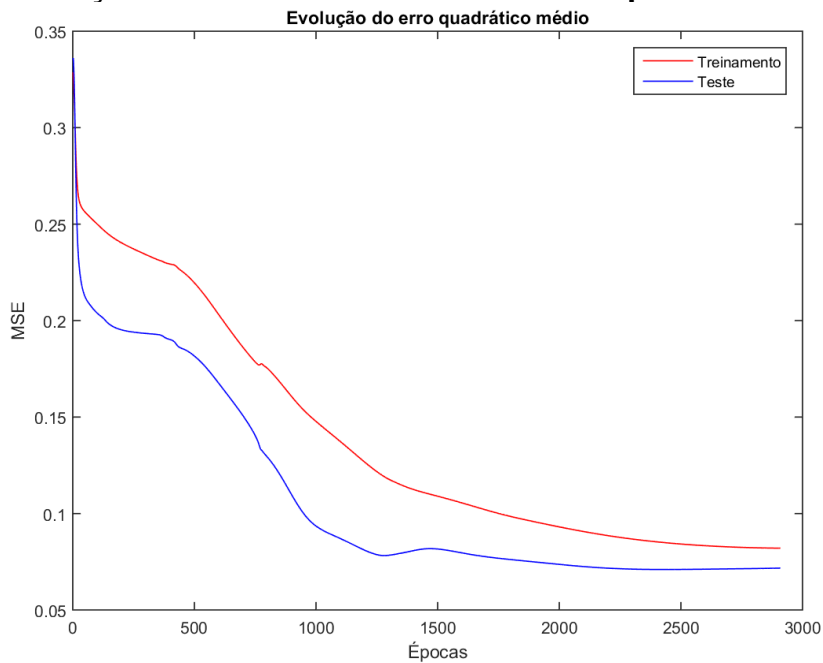
Tipo de Treinamento	Neurônio/camada	Taxa de aprendizado	Taxa de <i>Momentum</i>	Épocas	MSE Teste	Taxa de acerto (%)
<i>Momentum</i>	10/13	0,1	0,75	2631,4	0,153	77
<i>Backpropagation</i>	8/8	0,1	-	2376,2	0,148	75
<i>Momentum</i>	12/12	0,1	0,6	1943,8	0,164	75
<i>Momentum</i>	10/11	0,1	0,15	1930	0,180	74
<i>Momentum</i>	11/12	0,1	0,15	1761,6	0,166	74

Fonte: Elaborada pela autora.

O resultado mostrou que os melhores resultados das redes Classificadoras apresentaram acerto de até 78% da classificação de falhas na pista interna, externa e elemento rolante de um rolamento. Houve um acerto maior para as redes com uma camada intermediária.

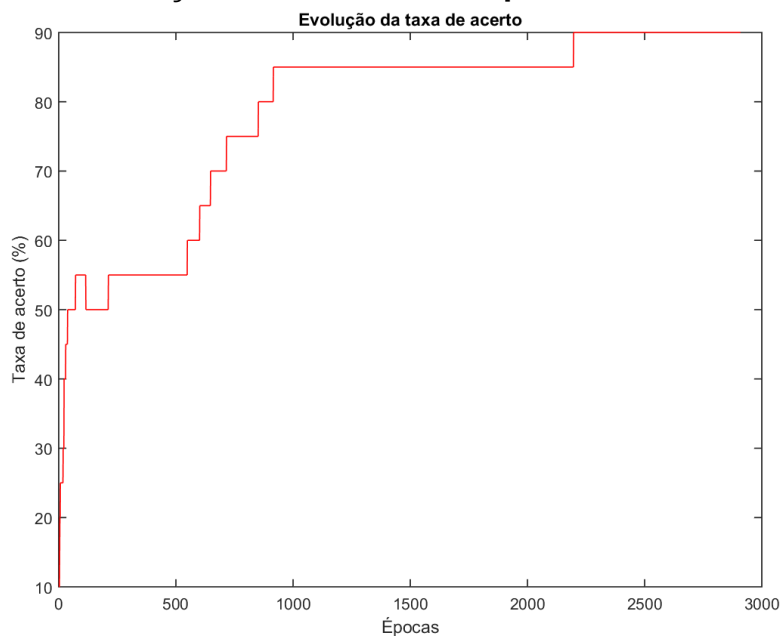
A rede PMC que foi capaz de classificar corretamente as falhas com a maior taxa de acerto, de 78%, possui uma camada intermediária com 12 neurônios, taxa de aprendizado de 0,1 e taxa *Momentum* 0,3, utilizando o método de treinamento *Momentum*, como mostra a Tabela 2. Então esta mesma rede foi treinada novamente e escolhido o melhor de 5 resultados para apresentar a evolução do MSE (Figura 20) e da taxa de acerto no decorrer das épocas (Figura 21).

Figura 20 - Evolução do MSE de treinamento e teste para rede Classificadora



Fonte: Elaborada pela autora.

Figura 21 - Evolução da taxa de acerto para rede Classificadora



Fonte: Elaborada pela autora.

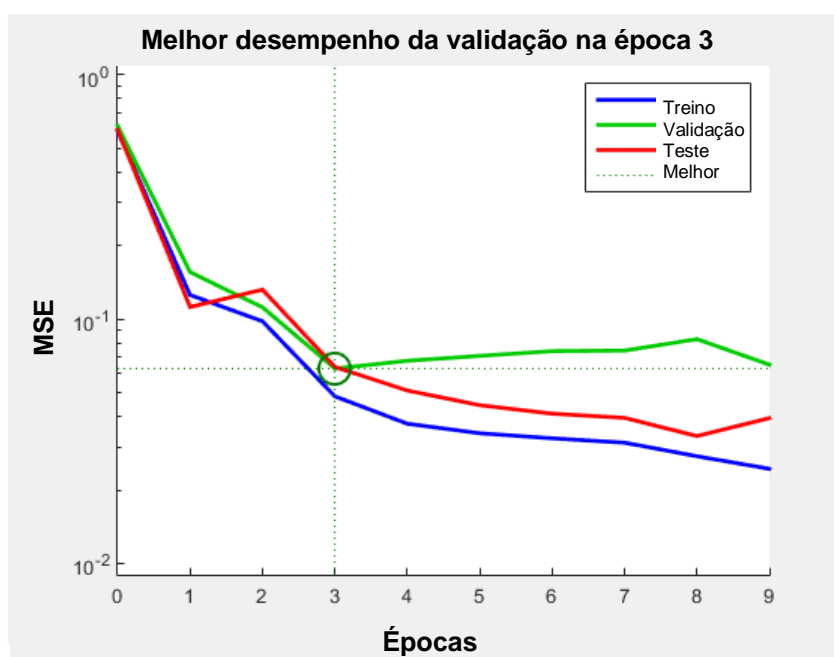
Neste treinamento a rede alcançou 90% de acerto após 2909 épocas, com MSE de treinamento de 0,082 e MSE de teste de 0,071. O gráfico apresentou características de uma boa generalização, com a convergência de ambas as curvas de erro de treinamento e de teste, mantendo a tendência de diminuição do erro.

Para o treinamento feito com o Toolbox do Matlab®, assim como para o treinamento realizado para as redes programadas, foram definidas as condições de treinamento:

- Redes de apenas uma camada intermediária, com alteração no número de neurônios desta camada de 8 a 12;
- Percentual de amostras para treinamento (75%), teste (20%) e validação (5%);

Neste treinamento, o melhor desempenho da rede foi encontrado na época 3, com MSE de validação de 0,063, MSE de treinamento de 0,048 e MSE de teste de 0,064, como mostra a Figura 22, com 85% de aproximação entre o resultado desejado e o obtido.

Figura 22 - Evolução do MSE de treinamento e teste para rede Classificadora com Toolbox do Matlab®



Fonte: Elaborada pela autora.

A RNA Classificadora programada se mostrou apta para generalização de padrões, com uma taxa de acertos aceitável quando encontrada a melhor topologia de rede, dadas as características dos dados de entrada e saída. Os valores de MSE da rede programada ficaram na mesma ordem de grandeza da rede feita no Toolbox, entretanto, a rede do Toolbox apresentou um número de épocas quase mil vezes menor. A eficiência desta rede pode se dever à utilização do treinamento por *Levenberg-Marquardt*, que, segundo Silva, Spatti e Flauzino (2010), faz o treinamento convergir normalmente até 100 vezes mais rápido que o *Backpropagation*.

4.1.2 Rede Aproximadora de Funções

A rede Aproximadora de Funções foi treinada com o mesmo banco de dados de 80 amostras relativas às falhas em rolamentos. Diferentemente da Classificadora, esta rede contém apenas uma única saída relativa ao tamanho conhecido da falha no rolamento, como mostra a Tabela 4. Foi usada a rede neural PMC como uma Aproximadora de Funções, uma vez que a finalidade foi estimar a criticidade ou tamanho da falha do rolamento, ou seja, estimar um valor não binário que varia dentro de um intervalo [0, 1].

Tabela 4 - Exemplos de amostras para rede Aproximadora de Funções

Entradas							Saída
Pista Interna (g)	Pista Externa (g)	Elemento Rolante (g)	X RMS (g)	Fator de Crista (g)	Fator K (g)	Carga (HP)	Tamanho (mm)
0,00365	0,08816	0,00315	0,76926	4,35356	2,57625	0	0,1778
0,00988	0,00414	0,00232	0,0936	4,22926	0,03705	1	0,3556
0,04802	0,03105	0,01478	0,16307	11,3706	0,30236	2	0,3556
0,01278	0,02896	0,00225	0,11794	4,88655	0,06797	3	0,5334

Fonte: dados retirados de Alvarenga, Rodrigues e Silva (2017).

Foram feitos dois tipos de treinamento – *Backpropagation* e *Momentum* – com variação das taxas de aprendizado e *Momentum* de 0,1 a 0,9 e de 0 a 0,75 com passos de 0,1 e 0,15, respectivamente. As topologias das redes possuem uma ou duas camadas intermediárias, com variação no número de neurônios n_1 dessas camadas de 7 a 15 (Eq. 27). A Tabela 5 e a Tabela 6 apresentam o resultado das melhores

topologias da rede encontradas para esta aplicação com uma e com duas camadas intermediárias, respectivamente.

$$2.\sqrt{7} + 1 \leq n_1 \leq 2.7 + 1 \therefore 6,29 \leq n_1 \leq 15 \quad (27)$$

Tabela 5 - Melhores topologias da rede Aproximadora de Funções com uma camada intermediária

Tipo de Treinamento	Neurônios por camada	Taxa de aprendizado	Taxa <i>Momentum</i>	Épocas	MSE Teste (mm ²)	Erro relativo médio (%)
<i>Momentum</i>	12	0,6	0,75	176,8	0,0085	34,41
<i>Momentum</i>	12	0,2	0,75	328,2	0,0103	34,63
<i>Momentum</i>	11	0,8	0,75	271,2	0,0077	35,04
<i>Momentum</i>	15	0,1	0,3	790,2	0,0085	35,47
<i>Momentum</i>	7	0,5	0,75	143,8	0,0105	35,71

Fonte: Elaborada pela autora.

Tabela 6 - Melhores topologias da rede Aproximadora de Funções com duas camadas intermediárias

Tipo de Treinamento	Neurônios por camada	Taxa de aprendizado	<i>Momentum</i>	Épocas	MSE Teste (mm ²)	Erro relativo médio (%)
<i>Momentum</i>	8/7	0,4	0,15	799,2	0,0086	31,66
<i>Momentum</i>	8/9	0,7	0,75	614	0,0101	32,49
<i>Momentum</i>	14/11	0,2	0,6	714	0,0096	32,64
<i>Momentum</i>	7/8	0,8	0,3	183,8	0,0091	33,11
<i>Momentum</i>	8/15	0,9	0,75	325,8	0,0119	33,25

Fonte: Elaborada pela autora.

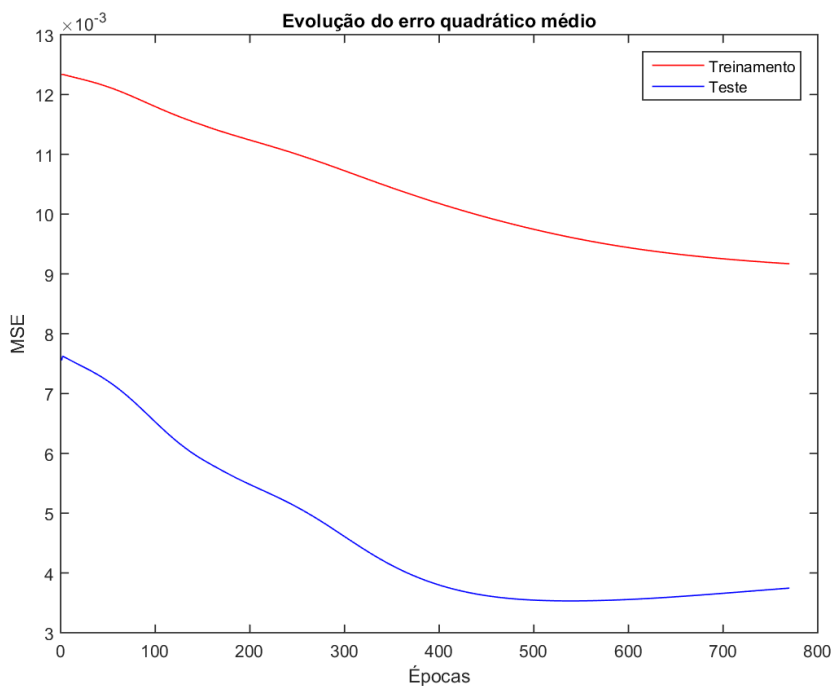
Os melhores resultados apresentados – com menor MSE – foram de redes com duas camadas intermediárias, com tipo de treinamento *Momentum*. O erro relativo médio foi de 31,66%. Este valor, que foi considerado alto para o erro, pode ter ocorrido devido à natureza do banco de dados apresentar apenas 3 valores distintos como saída da rede (0,1778, 0,3556 e 0,5334 mm) em vez de valores distintos para as amostras no intervalo entre 0 e 1. Ou seja, a repetição de um mesmo valor de saída para diferentes valores de entradas de várias amostras pode ter levado a rede a uma dificuldade na generalização.

O desempenho do tipo de rede tem ligação não apenas com suas próprias características, mas também com os tipos de dados de entrada e saída utilizados.

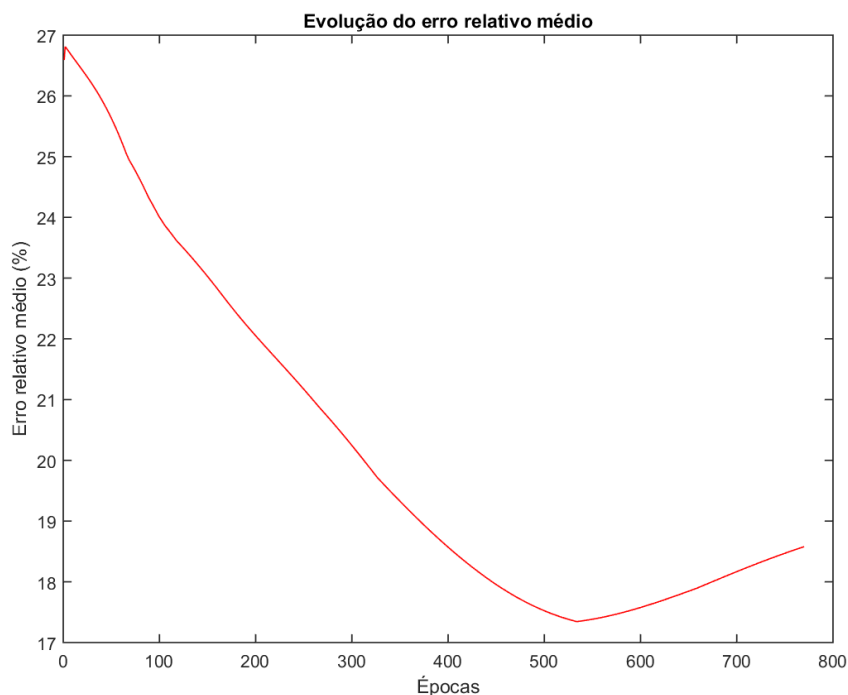
Assim sendo, pode ocorrer, para outros tipos de dados, que uma rede com características distintas das apresentadas como melhores neste estudo obtenha o melhor desempenho ou erros maiores, ou seja, resultados diferentes, reforçando a importância da aplicação do método *Cross-validation* para o alcance de resultados mais assertivos quanto possível.

A melhor topologia de rede da Tabela 6 foi treinada mais cinco vezes. O resultado apresentou melhora, visto que a tabela contém o valor da média de cinco treinamentos. Os valores foram de 18,58% para o erro relativo médio e MSE de teste de 0,0037mm² e de treinamento de 0,0092mm² para 770 épocas. A Figura 23 e a Figura 24 mostram a evolução destes erros ao longo as épocas, apresentando boa generalização. Entretanto, o treinamento poderia ter sido interrompido próximo a 530 épocas, momento em que ocorreu o menor valor para o MSE de teste e para o erro relativo médio. Para isto, seria necessário aprimorar a programação considerando também um subconjunto de validação, a fim de cessar o treinamento em um mínimo local cujo MSE de validação estivesse inferior a determinado valor programado.

Figura 23 - Evolução do MSE para rede Aproximadora



Fonte: Elaborada pela autora.

Figura 24 - Evolução da taxa de erro para rede Aproximadora

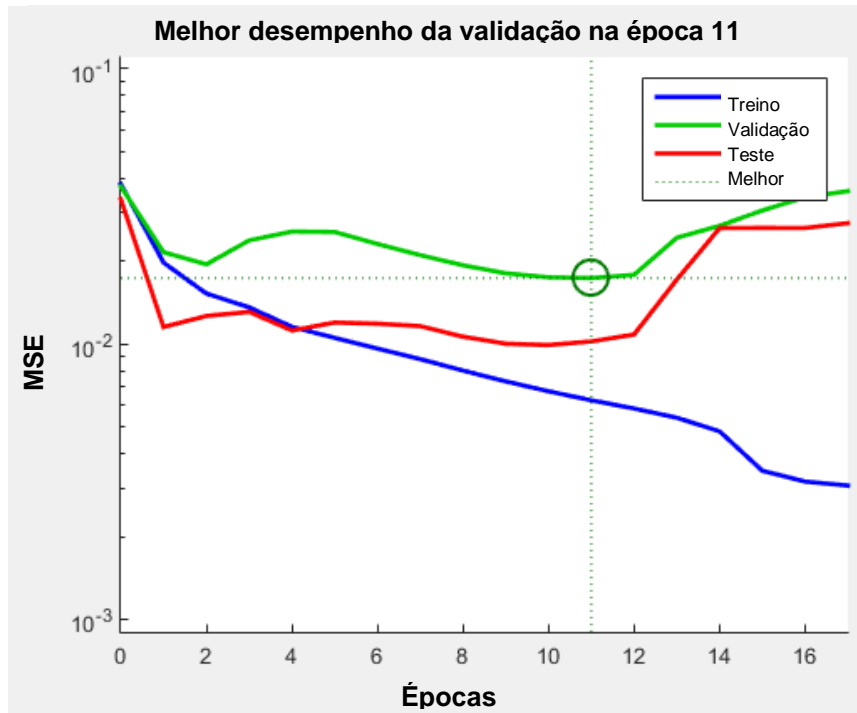
Fonte: Elaborada pela autora.

Para o treinamento feito com o Toolbox do Matlab®, assim como para o treinamento realizado para as redes programadas, foram definidas as condições de treinamento:

- Redes de apenas uma camada intermediária, com alteração no número de neurônios desta camada de 8 a 12;
- Percentual de amostras para treinamento (75%), teste (20%) e validação (5%);

A Figura 25 apresenta o melhor resultado deste treinamento: uma rede de uma camada de 11 neurônios, cujo treinamento obteve melhor resposta para 11 épocas, com MSE de validação de 0,017, MSE de treinamento de 0,006mm² e MSE de teste de 0,010mm², e 68,9% de aproximação entre o resultado desejado e o obtido.

Figura 25 - Evolução da taxa de acerto para rede Aproximadora com Toolbox do Matlab®



Fonte: Elaborada pela autora.

A RNA Aproximadora programada se mostrou apta para generalização de padrões, com um erro relativo inferior ao resultado do Toolbox do Matlab®, para a melhor topologia de rede. Os valores de MSE da rede programada ficaram cerca de dez vezes menores que os obtidos pelo Toolbox, entretanto, a rede do Toolbox, assim como para a Classificadora de padrões, apresentou um número de épocas 70 vezes menor. A eficiência desta rede também pode se dever à utilização do treinamento por *Levenberg-Marquardt*, que, segundo Silva, Spatti e Flauzino (2010), faz o treinamento convergir normalmente até 100 vezes mais rápido que o *Backpropagation*.

4.2 Definição do objeto de estudo e condições de operação

A definição do objeto de estudo levou em consideração fatores como tempo parado para manutenções preventiva e corretiva e gastos com pessoal de manutenção associados, e a disponibilidade do ativo. A partir da comparação destes aspectos entre os diversos ativos portuários de mineração, foi definido pelos especialistas como objeto de estudo o Virador de Vagões, como o da Figura 26. Este ativo é um gargalo para a atividade portuária, tendo em vista que é responsável pela

primeira etapa do minério no porto: a transposição do carregamento dos vagões para o pátio de estocagem.

Figura 26 - Virador de Vagões



Fonte: VALE@INFORMAR (2007).

Em meio aos sistemas que compõem o VV, foram selecionados como objeto de estudo o sistema de dois Alimentadores, cada um composto por um motor, um redutor e um redutor planetário. A aplicação deste estudo foi feita para os dados históricos dos motores dos alimentadores. Ambos os alimentadores são simétricos quanto à disposição dos componentes e sensores.

O funcionamento do virador de vagões ocorre na sequência de etapas a seguir: os vagões são posicionados no interior do VV, onde são rotacionados para que o minério seja vertido nas esteiras que o conduzem até o pátio. Os Alimentadores são o sistema responsável pelo acionamento destas esteiras. Neste sistema são monitoradas variáveis vibracionais e operacionais.

4.3 Banco de dados do objeto de estudo

As medições vibracionais coletadas são armazenadas a princípio semanalmente e, com o passar do tempo, as coletas são realizadas em intervalos cada vez menores até uma intervenção para manutenção. Os dados operacionais são armazenados conforme coleta, e foram disponibilizados dados do período correspondente coletados a cada 5 segundos, de modo a possuir amostras que contemplem as etapas do giro do VV. No período de Agosto de 2017 a Agosto de

2018, foram disponibilizados 108 dados vibracionais referentes ao Alimentador 1 e 101 ao Alimentador 2.

Para que os dados fossem normalizados², foi calculada ou informada por especialistas a faixa de variação dos dados de cada variável e feita a adequação dos valores para o intervalo de 0 a 1. O percentual de tempo até a parada foi calculado com base nas ordens de manutenção, como na Eq. (28), sendo Dia_{up} o dia da última parada, como marco zero, Dia_{atual} o número de dias até a coleta do dado, Dia_{pp} o dia até a próxima parada.

$$\%Tempo = \frac{Dia_{atual} - Dia_{up}}{Dia_{pp} - Dia_{up}} * 100\% \quad (28)$$

4.4 Banco de dados aplicado na rede com entradas variantes no tempo

O banco de dados do objeto de estudo foi aplicado na RNA variante no tempo conforme Seção 3.2. Foi decidido inicialmente utilizar uma janela deslizante de tamanho $np = 3$ para comparação entre os resultados das redes com os diferentes usos do banco de dados para ambos os alimentadores. O Quadro 7 contém a correspondência entre as variáveis utilizadas para treinamento das redes e as tabelas com seus respectivos resultados para o Alimentador 1 (AL1) ou para o Alimentador 2 (AL2).

Quadro 7 - Correspondência entre variáveis de treinamento e resultados

Variáveis	Resultados		Tamanho da janela
Quadro 8	Tabela 7	Resultado 1 (AL1)	$np = 3$
Quadro 9	Tabela 8	Resultado 2 (AL1)	$np = 3$
	Tabela 10	Resultado 4 (AL2)	$np = 3$
Quadro 10	Tabela 9	Resultado3 (AL1)	$np = 3$
	Tabela 11	Resultado 5 (AL2)	$np = 3$
	Tabela 14	Resultado 6 (AL1)	$np = 2$
	Tabela 15	Resultado 7 (AL1)	$np = 4$
	Tabela 16	Resultado 8 (AL2)	$np = 2$
	Tabela 17	Resultado 9 (AL2)	$np = 4$

Fonte: Elaborado pela autora.

² Os dados e cálculos realizados para a normalização não são expostos neste estudo por serem dados sigilosos da indústria.

O Quadro 8 mostra as variáveis de entrada e saída para a RNA com 8 entradas variantes no tempo, mantendo as grandezas da forma que foram disponibilizadas no banco de dados para o Alimentador 1.

Quadro 8 - Variáveis utilizadas no treinamento da RNA com 8 entradas variantes no tempo

Entradas	Rotação
	Corrente
	Envelope
	Velocidade
	Aceleração
	Velocidade Ref
	Horímetro
Saída	% Tempo

Fonte: Elaborado pela autora.

Após o *Cross-validation*, as melhores topologias encontradas (Tabela 7) para o treinamento realizado com as variáveis do Quadro 8 possuem duas camadas intermediárias e apresentaram um erro relativo médio superior a 60%, resultado pior que o esperado para a previsão temporal.

Tabela 7 - Resultado 1 - Alimentador 1 para RNA com 8 entradas e $np=3$

Parâmetros da topologia da rede				Resultados		
Tipo de treinamento	Nº neur./cam.	Momentum	Aprendizagem	Épocas	MSE	Erro relativo médio (%)
<i>Momentum</i>	16/14	0,15	0,1	683	0,01786	60,82834
<i>Momentum</i>	12/14	0,30	0,9	167,6	0,08128	76,16396
<i>Momentum</i>	15/12	0,30	0,4	343,6	0,04191	80,25506
<i>Momentum</i>	13/15	0,45	0,3	492,4	0,03187	80,85098

Fonte: Elaborada pela autora.

Devido ao Resultado 1 ter sido considerado ruim, ou seja, com um erro relativo médio alto, foram desconsideradas três variáveis de entrada para o treinamento seguinte, pois seus valores não apresentaram uma tendência de evolução ao longo do tempo. São elas: Rotação, Corrente e Velocidade Ref. O Quadro 9 mostra, então, as variáveis que foram utilizadas para o Resultado 2 (Tabela 8). Este resultado apresentou melhora em relação ao primeiro, com um MSE de 0,055 e erro relativo médio de 52,95%, entretanto estes valores ainda estavam superiores à expectativa, uma vez que os dados utilizados para treinamento não haviam sido tratados e normalizados, como sugerido na literatura.

Quadro 9 - Variáveis utilizadas no treinamento da RNA com 5 entradas variantes no tempo

Entradas	Envelope
	Velocidade
	Aceleração
	Horímetro
	% Tempo
Saída	% Tempo

Fonte: Elaborado pela autora.

Tabela 8 - Resultado 2 - Alimentador 1 para RNA com 5 entradas e $np=3$

Parâmetros da topologia da rede				Resultados		
Tipo de treinamento	Nº neur./cam.	Momentum	Aprendizagem	Épocas	MSE	Erro relativo médio (%)
<i>Momentum</i>	10/9	0,45	0,6	345,8	0,055065	52,95
<i>Momentum</i>	14/9	0,15	0,8	116	0,061861	56,53
<i>Momentum</i>	12/11	0,45	0,8	390	0,079327	57,35
<i>Momentum</i>	9/11	0,30	0,9	176,6	0,072893	58,85

Fonte: Elaborada pela autora.

O Quadro 10 aponta as variáveis tratadas e normalizadas que foram utilizadas para a obtenção do Resultado 3. A normalização feita nos valores de Envelope, Velocidade e Aceleração consistiu primeiramente em identificar – pelos valores do banco de dados e pelo conhecimento dos especialistas – os valores máximo e mínimo que cada variável poderia alcançar no processo. Feito isso, esta faixa de valores foi transformada para variar entre 0 e 1. Para o Horímetro, que não apresentava inicialmente uma faixa de valores definida, haja visto que estes podem crescer indefinidamente, foi calculada a diferença de horas entre amostras. Como a coleta ocorre no máximo semanalmente, sua faixa de valores ficou dentro da quantidade de horas que há no período de uma semana, podendo ser, em seguida, normalizada entre 0 e 1.

Quadro 10 - Variáveis normalizadas utilizadas no treinamento da RNA com 5 entradas variantes no tempo

Entradas	Envelope
	Velocidade
	Aceleração
	Diferença Horímetro
	% Tempo
Saída	% Tempo

Fonte: Elaborado pela autora.

Tabela 9 - Resultado 3 - Alimentador 1 para RNA com 5 entradas normalizadas e np=3

Parâmetros da topologia da rede				Resultados		
Tipo de treinamento	Nº neur./cam.	Momentum	Aprendizagem	Épocas	MSE	Erro relativo médio (%)
<i>Momentum</i>	14/12	0,60	0,1	363,8	0,000532	8,83
<i>Momentum</i>	11/10	0,15	0,1	575,8	0,000411	10,37
<i>Momentum</i>	12/13	0,30	0,2	572	0,002501	10,52
<i>Momentum</i>	9/14	0,60	0,9	559,2	0,000538	10,72

Fonte: Elaborada pela autora.

O Resultado 3 (Tabela 9) mostrou uma queda significativa nos valores de MSE e erro relativo médio de previsão da rede em relação aos resultados anteriores. O melhor resultado foi uma rede com duas camadas intermediárias, cujo erro relativo médio foi inferior a 9%, ou seja, cinco vezes menor que para o Resultado 2 e seis vezes menor que para o Resultado 1.

Para o Alimentador 2 foram obtidos resultados apenas para treinamentos com 5 variáveis e com 5 variáveis normalizadas, uma vez que os resultados do Alimentador 1 para 8 variáveis apresentaram erros significativamente maiores que para os com 5 variáveis. Além disto, o tempo de processamento é menor utilizando menos variáveis, portanto não foi realizado o treinamento com 8 variáveis para o Alimentador 2.

Tabela 10 - Resultado 4 - Alimentador 2 para RNA com 5 entradas 3 np=3

Parâmetros da topologia da rede				Resultados		
Tipo de treinamento	Nº neur./cam.	Momentum	Aprendizagem	Épocas	MSE	Erro relativo médio (%)
<i>Momentum</i>	12/13	0,15	0,8	230,6	0,0256	58,47
<i>Momentum</i>	9/11	0,75	0,6	87	0,0176	68,78
<i>Momentum</i>	9/9	0,30	0,8	224,8	0,0371	71,56
<i>Momentum</i>	9/14	0,15	0,5	201,8	0,0171	73,30

Fonte: Elaborada pela autora.

O Resultado 4, apresentado na Tabela 10, refere-se às variáveis de entrada do Quadro 9. A rede de melhor topologia apresentou MSE de 0,0256 e erro relativo médio de 58,47%, considerado alto neste estudo para tarefa de previsão temporal para auxiliar a manutenção no planejamento das paradas de forma verossímil.

Tabela 11 - Resultado 5 - Alimentador 2 para RNA com 5 entradas normalizadas e $n_p=3$

Parâmetros da topologia da rede				Resultados		
Tipo de treinamento	Nº neur./cam.	Momentum	Aprendizagem	Épocas	MSE	Erro relativo médio (%)
<i>Momentum</i>	12/13	0,60	0,7	427,8	0,00263	16,51
<i>Momentum</i>	9/11	0,45	0,8	613	0,007274	16,74
<i>Momentum</i>	13/10	0,60	0,7	551,8	0,004015	17,60
<i>Momentum</i>	11/9	0,75	0,4	542,2	0,001256	20,06

Fonte: Elaborada pela autora.

Então foi obtido o Resultado 5 para o Alimentador 2 (Tabela 11) com as 5 variáveis de entrada tratadas e normalizadas, conforme o Quadro 10. Assim como ocorreu para o Alimentador 1, houve uma queda significativa de mais de um terço do valor do erro relativo médio do Resultado 4 para o 5, indicando melhora na previsão temporal.

A Tabela 13 mostra a comparação entre os resultados anteriores para janela deslizante de tamanho 3. São comparados resultados referentes à duração de um treinamento, número de épocas, MSE de teste, erro relativo médio e intervalos de confiança a 95% para treinamento e teste.

Para cada resultado, a rede que apresentou a melhor topologia foi treinada cinco vezes e foi escolhido o treinamento com menor erro para o cálculo do intervalo de confiança da rede para treinamento e para teste, seguindo os passos da Seção 3.5. A Tabela 12 apresenta a tabela de t-student, utilizada para o cálculo do intervalo de confiança. O valor para infinitos graus de liberdade com probabilidade de 95% é 1,96. Este é o valor de t para aplicação da Eq.(25), que resulta no intervalo de confiança, tanto para o treinamento, quanto para o teste, para as redes de ambos os Alimentadores.

Tabela 12 - Tabela de t-student

GL	Área na cauda superior (monocaudal)					
	0,250	0,10	0,05	0,025	0,01	0,005
1	1,000	3,078	6,314	12,706	31,821	63,657
5	0,727	1,476	2,015	2,571	3,365	4,032
10	0,700	1,372	1,812	2,228	2,764	3,169
20	0,687	1,325	1,725	2,086	2,528	2,845
∞	0,674	1,282	1,645	1,960	2,326	2,576

Fonte: Recorte de Costa (2012).

Tabela 13 - Comparação entre os Resultados 1 a 5 das RNAs com entradas variantes no tempo com $np=3$

Resultado	1 (AL1)	2 (AL1)	3 (AL1)	4 (AL2)	5 (AL2)
Tempo de treinamento (s)	16,37	7,48	12,27	7,21	25,85
Número de épocas	332	238	258	172	609
Erro relativo médio (%)	59,46	32,69	4,35	90,50	9,99
MSE	0,0287	0,0883	0,000451	0,040	0,00143
Intervalo de confiança do treinamento (95%)	±16,06%	±56,73%	±3,11%	±11,47%	±0,20%
Intervalo de confiança do teste (95%)	±17,56%	±42,75%	±0,11%	±13,32%	±0,38%

Fonte: Elaborada pela autora.

Os resultados revelaram a importância de fazer o tratamento e a normalização dos dados, pois, para os dados de ambos os alimentadores, foi obtida a queda significativa do erro relativo médio e intervalo de confiança da rede.

A Tabela 14 até a Tabela 17 contém os melhores resultados das redes para o banco de dados do Quadro 10, ou seja, com 5 entradas cujos valores foram normalizados. O objetivo dos Resultados 6 a 9 é fazer uma comparação de resultados entre redes com janelas deslizantes de tamanhos distintos, sendo que os resultados 6 e 7 se referem ao AL1 e o 8 e 9 ao AL2.

O Resultado 6 apresenta $np = 2$, em que a melhor topologia de rede possui um erro relativo médio de apenas 5,56%. Enquanto para o Resultado 7, com $np = 4$ este valor subiu para 10,47%. Houve pequena variação do MSE entre eles, mas ainda assim se manteve baixo para todas as melhores topologias destas tabelas.

Tabela 14 - Resultado 6 - Alimentador 1 para RNA com 5 entradas normalizadas e $np=2$

Parâmetros da topologia da rede				Resultados		
Tipo de treinamento	Nº neur./cam.	Momentum	Aprendizagem	Épocas	MSE	Erro relativo médio (%)
<i>Momentum</i>	11/9	-	0,3	260,6	0,000360	5,56
<i>Momentum</i>	9/12	0,3	0,4	193,8	0,000671	6,11
<i>Momentum</i>	13/8	0,45	0,4	154,8	0,000443	6,18
<i>Momentum</i>	10/9	0,75	0,5	88,8	0,000509	6,24

Fonte: Elaborada pela autora.

Tabela 15 - Resultado 7 - Alimentador 1 para RNA com 5 entradas normalizadas e $np=4$

Parâmetros da topologia da rede				Resultados		
Tipo de treinamento	Nº neur./cam.	Momentum	Aprendizagem	Épocas	MSE	Erro relativo médio (%)
<i>Momentum</i>	12/10	0,45	0,9	432,8	0,000473	10,47
<i>Momentum</i>	13/12	0,60	0,7	604,4	0,001995	10,57
<i>Momentum</i>	12/14	0,75	0,8	294,2	0,000707	10,91
<i>Momentum</i>	15/14	0,60	0,9	444,4	0,001731	11,41

Fonte: Elaborada pela autora.

Para o Alimentador 2, os resultados se comportaram de forma semelhante ao Alimentador 1, com um erro relativo médio mais baixo para uma janela de tamanho 2 (Tabela 16) do que de tamanho 4 (Tabela 17).

Tabela 16 - Resultado 8 - Alimentador 2 para RNA com 5 entradas normalizadas e $np=2$

Parâmetros da topologia da rede				Resultados		
Tipo de treinamento	Nº neur./cam.	Momentum	Aprendizagem	Épocas	MSE	Erro relativo médio (%)
<i>Momentum</i>	8/10	0,60	0,5	646	0,001139	8,07
<i>Momentum</i>	13/8	0,75	0,5	456,8	0,004482	15,08
<i>Momentum</i>	8/12	0,60	0,7	454,6	0,001845	17,01
<i>Momentum</i>	13/11	0,75	0,1	1186,4	0,002299	17,07

Fonte: Elaborada pela autora.

Tabela 17 - Resultado 9 - Alimentador 2 para RNA com 5 entradas normalizadas e $np=4$

Parâmetros da topologia da rede				Resultados		
Tipo de treinamento	Nº neur./cam.	Momentum	Aprendizagem	Épocas	MSE	Erro relativo médio (%)
<i>Momentum</i>	13/13	0,75	0,7	543,6	0,004112	13,06
<i>Momentum</i>	10/13	0,75	0,7	347,6	0,005353	17,40
<i>Momentum</i>	15/14	0,75	0,9	441,4	0,005924	20,67
<i>Momentum</i>	14/12	0,75	0,7	587,6	0,005028	21,53

Fonte: Elaborada pela autora.

A Tabela 18 traz a comparação entre todos os resultados obtidos para o banco de dados no Quadro 10 com diferentes tamanhos de janela deslizante para ambos os Alimentadores.

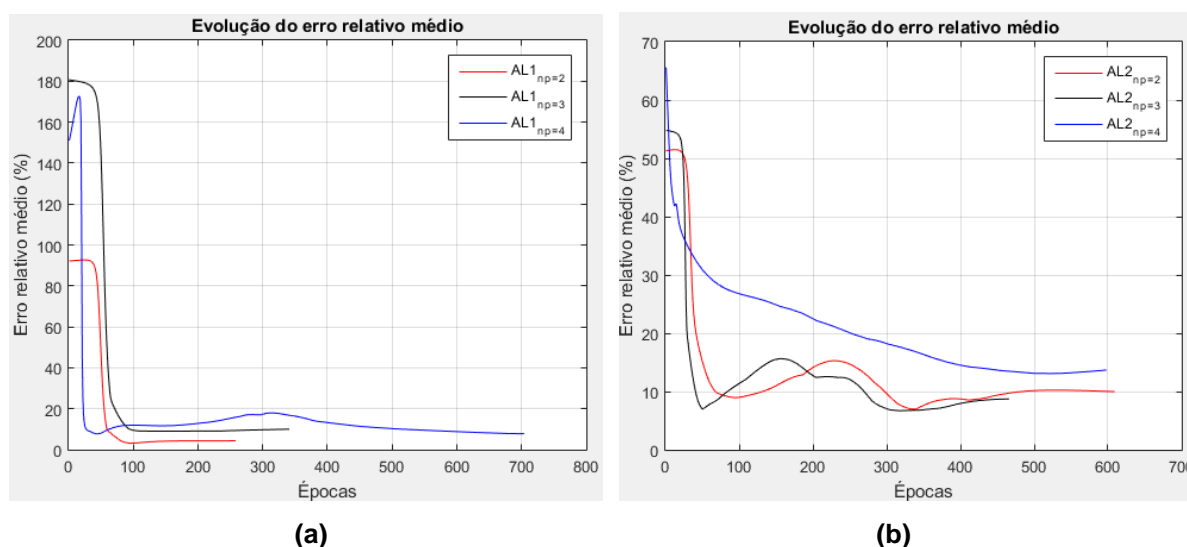
Tabela 18 - Comparação entre os Resultados 3, 5, 6, 7, 8 e 9 das RNAs com entradas variantes no tempo

Resultado	3 (AL1)	6 (AL1)	7 (AL1)	5 (AL2)	8 (AL2)	9 (AL2)
Janela deslizante (np)	2	3	4	2	3	4
Tempo de treinamento (s)	12,27	19,00	33,98	25,85	23,75	17,82
Número de épocas	258	341	704	609	466	598
Erro relativo médio (%)	4,35	10,02	7,78	9,99	8,72	13,69
MSE	0,000451	0,000352	0,00108	0,00143	0,00146	0,00435
Intervalo de confiança do treinamento a 95% (%)	$\pm 3,11$	$\pm 2,98$	$\pm 0,07$	$\pm 0,20$	$\pm 0,17$	$\pm 0,07$
Intervalo de confiança do teste a 95% (%)	$\pm 0,11$	$\pm 0,07$	$\pm 0,25$	$\pm 0,38$	$\pm 0,35$	$\pm 1,09$

Fonte: Elaborada pela autora.

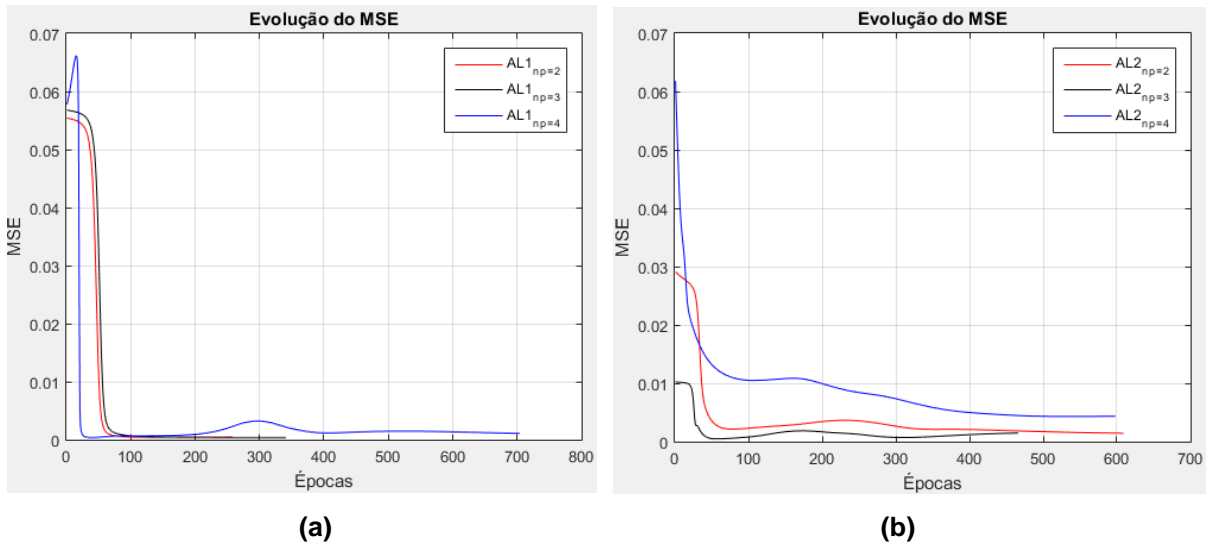
O tempo de treinamento não está necessariamente relacionado com o tamanho da janela deslizante. Por possuir mais entradas – 5 para cada momento, neste caso – ao aumentar o tamanho da janela, há uma tendência na melhora da precisão da rede, conforme Silva, Spatti e Flauzino (2010). Isto ocorreu para o intervalo de confiança de treinamento de ambos os alimentadores, entretanto não se repetiu para o teste. O intervalo de confiança foi calculado com base na variância da resposta obtida, não sendo diretamente proporcional ao erro relativo médio e ao MSE. No entanto, estes parâmetros são relevantes para a seleção das topologias que podem apresentar melhor desempenho.

Figura 27 - Evolução do erro relativo médio para diferentes tamanhos de janela deslizante para (a) AL1 e (b) AL2



Fonte: Elaborada pela autora.

Figura 28 - Evolução do MSE de teste para diferentes tamanhos de janela deslizante para (a) AL1 e (b) AL2



Fonte: Elaborada pela autora.

A Figura 27 e a Figura 28 mostram respectivamente a comparação da evolução do erro relativo médio e do MSE de teste para treinamentos com diferentes tamanhos de janela deslizante para o Alimentador 1 (a) e para o Alimentador 2 (b). Os treinamentos relativos ao AL1 apresentaram bom desempenho, entretanto, para o AL2 pode ser observado o ajuste excessivo para $np = 2$ e $np = 3$.

Os resultados obtidos para a previsão do tempo até a próxima parada para manutenção dos alimentadores utilizando 5 dados de entrada normalizados apresentaram MSE inferior a 0,005 e erro relativo médio inferior a 14%, o que foi considerado um resultado satisfatório para este estudo. O intervalo de confiança de teste a 95% das redes variou entre $\pm 0,07\%$ e $\pm 1,09\%$, mostrando que as RNAs apresentaram potencial para obter resultados de previsão assertivos para auxiliar no planejamento da manutenção.

4.5 Banco de dados aplicado nas redes Classificadora e Aproximadora

Assim como para a rede com entradas variantes no tempo (Seção 4.4), os dados disponíveis para a entrada das redes Classificadora e Aproximadora advém das coletas vibracionais e do PIMS, conforme Quadro 6.

Para montar o banco de dados de saída da RNA para classificação de falhas, foi disponibilizado o documento de ordens de manutenção, que possui como principais campos a serem preenchidos pelos técnicos, as lacunas a seguir:

- Datas de entrada da ordem, de início programado e do fim do chamado;
- Texto breve com limitação de caracteres para explicação do local e tipo da falha;
- Status da ordem, informando se realmente ocorreu ou foi cancelada;
- Denominação do ativo a qual a nota se refere;
- Número da nota da ordem, que é um outro documento, no qual constam campos para preenchimento padrão (apenas escolha de itens já pré-definidos do *software*), dentre eles a marcação do subsistema, componente e tipo de falha que ocorreu.

Os dados disponibilizados, entretanto, não foram suficientes para a montagem do banco de dados de treinamento e teste da rede Classificadora, uma vez que apenas metade das ordens de manutenção estava direcionada a uma nota, de tal maneira que não havia disponibilidade de informações sobre qual falha ocorreu para o outra metade; e um terço das notas existentes apontavam como “não encontrado” a falha e seu local de ocorrência.

A aplicação dos dados do objeto de estudo na rede Aproximadora, visando obter a criticidade de determinada falha, também foi inviabilizada pela falta de dados a elas relacionados: com a falta de dados coletados em momento de falha, como ocorreu no caso de objetivar a avaliação do desbalanceamento e desalinhamento do motor e não haver dado vibracional coletado nestas situações.

Outras dificuldades apresentadas para a montagem do banco de dados, além das informações incompletas ou indisponíveis, foram as incertezas sobre a data exata em que uma manutenção ocorreu, uma vez que o preenchimento do campo das datas mostra quando o chamado foi aberto, para quando estava programado e o término da ordem de manutenção, e não de quando ocorreu efetivamente a parada do ativo. A data real de manutenção poderia ser encontrada na nota, entretanto muitas ordens não estão relacionadas a nenhuma nota. A descentralização dos documentos é um obstáculo também para a montagem do banco de dados, uma vez que é preciso garimpar dados em diversos setores.

Com o intuito de ser possível a realização da classificação de falhas e avaliação de sua criticidade pelas RNAs de forma verossímil, é necessário possuir um banco de dados completo e correto. Para cada amostra do banco de dados é imprescindível que a coleta/obtenção de todas as variáveis de entrada e saída tenha sido realizada no mesmo momento e é essencial o conhecimento da relação das variáveis de entrada com as de saída. Sendo assim, é possível implementar RNAs que objetivam a classificação ou a avaliação da criticidade de falhas a partir de banco de dados para treinamento e teste da rede que se assemelhe aos modelos A e B do Quadro 11, respectivamente. Para o treinamento e teste de RNAs, é importante que o banco de dados que se pretende utilizar contemple todos os casos de saída a serem abordados, a fim de que a rede tome os dados relativos a toda a gama de saídas como referência para os cálculos de atualização dos pesos e, assim sendo, apresente um resultado mais exato para a pluralidade de cenários que podem ocorrer.

Quadro 11 - Modelos A e B de banco de dados para treinamento da RNA

Modelo A – Classificação de falhas – Rede Classificadora							
Entradas				Saídas			
Var 1	Var 2	...	Var N	Falha 1	Falha 2	...	Falha N
0 - 1	0 - 1	...	0 - 1	0 / 1	0 / 1	...	0 / 1

Modelo B – Criticidade de falhas – Rede Aproximadora							
Entradas				Saídas			
Var 1	Var 2	...	Var N	Falha 1	Falha 2	...	Falha N
0 - 1	0 - 1	...	0 - 1	0 - 1	0 - 1	...	0 - 1

Fonte: Elaborada pela autora.

As variáveis Var 1 a N devem ser escolhidas de forma criteriosa, para considerar no treinamento da rede apenas as variáveis que, de alguma forma, tenham relação com as falhas abordadas na saída. Variáveis com comportamento aleatório ou que se tenha o conhecimento de antemão que não possuam tendência de variação de seu valor que seja indicativo de qualquer falha da saída da rede devem ser evitadas. Para os dados do objeto de estudo, ao desconsiderar as variáveis que não apresentaram relação da variação de seus valores com a saída pretendida, o desempenho das redes aumentou significativamente.

Para ambas as redes, é recomendado que tanto as variáveis de entrada, quanto as de saída tenham seus valores normalizados, de acordo com a faixa de

valores assumida pela função de ativação escolhida para os neurônios da camada que sucede a camada de entrada. Na rede Classificadora (Modelo A), o valor das saídas – Falhas 1 a N – assume os valores 0 ou 1, considerando uma função de ativação sigmoidal, para representar uma situação de operação “sem falha” e “com falha”, respectivamente. Já para a rede Aproximadora (Modelo B), as saídas possuem valores entre 0 e 1, que podem corresponder à criticidade de uma falha, como, por exemplo, tamanho de uma trinca, ou a um valor não binário da saída que se deseja obter como resultado da rede.

CONCLUSÃO

Este estudo contribuiu com uma metodologia aplicável a banco de dados para o monitoramento de danos em ativos utilizando Redes Neurais Artificiais Perceptron de Múltiplas Camadas. Neste estudo, esta metodologia foi aplicada a rolamentos e ao sistema de Alimentadores de um Virador de Vagões. Para classificação de falhas e obtenção da criticidade destas falhas, foram aplicados nas RNAs programadas dados vibracionais de rolamentos, em que as redes alcançaram respectivamente 90% de acerto e 81,42% de concordância entre resposta desejada e obtida. Ambas as redes apresentaram resultados de MSE próximos ou inferiores aos adquiridos pelas redes do Toolbox do Matlab®, entretanto o treinamento das redes programadas teve um número de épocas significativamente maior (de 70 a 1000 vezes) que as do Toolbox.

As RNAs se mostraram aptas para a classificação de falhas e obtenção da criticidade de falhas em ativos, com boa generalização para o banco de dados real de rolamentos. Como os treinamentos apresentaram um número de épocas muito elevado, sugere-se como melhoria a aplicação do treinamento de Levenberg-Marquardt, que apresenta a possibilidade de redução das iterações de 10 a 100 vezes.

Para a previsão temporal, também foram utilizadas as RNAs PMC, mas com entradas variantes no tempo, em que foi feito o janelamento do banco de dados de um motor do sistema Alimentador de um Virador de Vagões. Neste contexto, foram treinadas redes com banco de dados original e normalizado, além da comparação entre diferentes tamanhos de janelamento dos dados, como foi mostrado nos resultados.

Para ambos os Alimentadores, houve diminuição significativa do erro relativo médio – de mais de 50% para menos de 17% – e do MSE – com redução de ao menos 10 vezes – ao considerar apenas entradas que apresentavam tendência de aumento dos valores ao longo das coletas e ao normalizar os valores das entradas. As redes convergiram e se mostraram aptas também quando utilizadas para previsão temporal. Possuir o conhecimento sobre as variáveis que se deseja abordar para a entrada e saída das RNAs revelou-se importante, uma vez que variáveis de entrada que não possuem relação direta com a saída pretendida podem dificultar a convergência da rede e contribuir com o aumento do erro da resposta obtida.

A metodologia de RNAs para monitoramento e previsão de falhas pode ser passível de replicação para diferentes ativos, uma vez que ela pode ser empregada

para diversos bancos de dados, compostos por vários tipos de sinais, sendo necessário apenas um novo treinamento com os respectivos dados para a aplicação em outro ativo. As RNAs se mostraram aplicáveis para diferentes objetivos, como classificação de falhas, estimação do nível de criticidade de uma falha e previsão temporal de valores com a aplicação de dados reais de ativos. A partir disso, é possível auxiliar o departamento de manutenção no planejamento de paradas para manutenção, visto que utilizadas em conjunto, as três redes em conjunto podem informar, por exemplo, se há uma falha e qual seu estado atual de criticidade, e uma previsão de tempo até uma parada.

As RNAs apresentaram aptidão para utilização com diferentes objetivos, entretanto, para que seja possível a implementação da rede para determinado ativo, é importante que o banco de dados deste equipamento esteja completo e confiável. Caso contrário, poderá haver dificuldade ou impossibilidade de sua implementação pela falta de integridade ou disponibilidade de dados, como ocorreu neste estudo. O banco de dados incompleto e com informações incertas impossibilitou a classificação e obtenção da criticidade das falhas no motor do Alimentador do VV.

Então, este trabalho trouxe uma contribuição para a indústria com a exposição de como os dados devem ser armazenados em função do que se deseja obter de resposta (detalhamento da montagem do banco de dados), de uma metodologia para auxiliar o planejamento da manutenção, no que diz respeito a qual modo de falha está mais propenso a acontecer, em qual intensidade (criticidade) e em quanto tempo se dá a evolução dos danos, podendo auxiliar a implementação da Indústria 4.0 na mineração.

Mesmo com a falta de dados experimentais – ponto de desvantagem desta metodologia, devido à necessidade prévia de um banco de dados completo e íntegro – é passível de utilização um banco de dados advindos de simulações do ativo inicialmente como alternativa. A utilização destes dados como alternativa inicial aos dados de coleta incompletos vem como sugestão para posteriormente fazer a substituição dos dados de simulação por medições à medida em que se vai fazendo a coleta, ou seja, treinar a rede agregando às amostras de simulação, amostras de coleta real do equipamento. Então, esta metodologia também pode ser aplicada para dados de simulação para suprir a falta de dados experimentais inicialmente. Neste caso, seria necessária a criação de um banco de dados a partir de simulações de diferentes cenários de operação do ativo. Dessa forma pode ser possível a obtenção

de uma configuração inicial que permite a utilização da ferramenta antes mesmo de haver coleta experimental. Esta sugestão, no entanto, não foi aplicada, pois não houve tempo hábil para que as simulações feitas pela indústria gerassem os dados necessários para este estudo.

Para trabalhos futuros é sugerida a implementação de outros tipos de treinamento, como o Levenberg-Marquardt, para acelerar esta etapa; da lógica *Fuzzy*, para incorporar incertezas; a aplicação de metodologias auxiliares para exploração dos dados, como KDD e D3M; a utilização de outras técnicas de cálculo de intervalo de confiança de redes neurais, como *bootstrap* e regressão não linear; a utilização de dados reais para previsão temporal até a falha do ativo com base em coletas de variáveis dinâmicas e o desenvolvimento de outras técnicas de IA para a previsão do tempo até a falha em ativos. Esta metodologia poderia futuramente ser aplicada a outros ativos, para diversos modos de falha e diferentes contextos operacionais.

REFERÊNCIAS

ABNT. **NBR 5462: Confiabilidade e manutenibilidade - Terminologia** Associação Brasileira De Normas Técnicas, 1994.

ALVARENGA, A. G.; RODRIGUES, L. DE M.; SILVA, R. M. DA. **Desenvolvimento de um aplicativo em Matlab para o auxílio de manutenção preditiva em rolamentos**. [s.l.] Pontifícia Universidade Católica de Minas Gerais, 2017.

AMARAL, J. L. M. DO. **SISTEMAS IMUNOLÓGICOS ARTIFICIAIS APLICADOS À DETECÇÃO DE FALHAS**. Rio de Janeiro, Brasil: PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO, 16 mar. 2006.

Bearing Data Center Case Western Reserve University. Disponível em: <<https://csegroups.case.edu/bearingdatacenter/pages/welcome-case-western-reserve-university-bearing-data-center-website>>. Acesso em: 26 ago. 2018.

BRAGA, A. DE P.; CARVALHO, A. P. DE L. F. DE; LUDERMIR, T. B. **Redes neurais artificiais: teoria e aplicações**. 2. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos, 2007.

CAI, S.; HU, J.; ZHANG, L. **Risk analysis of refining equipment based on fuzzy theory and bow-tie model**. 2016 35th Chinese Control Conference (CCC). **Anais...IEEE**, jul. 2016 Disponível em: <<http://ieeexplore.ieee.org/document/7554896/>>. Acesso em: 17 jul. 2017.

CHENG, F.; QU, L.; QIAO, W. Fault Prognosis and Remaining Useful Life Prediction of Wind Turbine Gearboxes Using Current Signal Analysis. **IEEE Transactions on Sustainable Energy**, p. 1–1, 2017.

COSTA, G. G. DE O. **Curso de estatística inferencial e probabilidades: teoria e prática**. São Paulo: 2012.

DEFILTRO, V. F.; GOMES, W. J. S. ANÁLISE DE CONFIABILIDADE ESTRUTURAL DE PROBLEMAS BASEADOS NA MECÂNICA DOS SÓLIDOS. **REEC - Revista Eletrônica de Engenharia Civil**, v. 12, n. 2, 4 jul. 2016.

DNPM. **Sumário Mineral - 2015**. Brasília, DNPM, , 2016. Disponível em: <<http://www.anm.gov.br/dnpm/publicacoes/serie-estatisticas-e-economia-mineral/sumario-mineral/sumario-mineral-brasileiro-2015>>. Acesso em: 20 jun. 2017.

DRATH, R.; HORCH, A. Industrie 4.0-hit or hype. **IEEE Industrial Electronics Magazine**, v. 8, n. 2, p. 56–58, 2014.

ELSHEIKH, A.; YACOUT, S.; OUALI, M.-S. Bidirectional handshaking LSTM for remaining useful life prediction. **Neurocomputing**, v. 323, p. 148–156, 5 jan. 2019.

GAO, Z. et al. **Fuzzy time-dependent reliability modeling and analysis method for mechanisms based on strength degradation**. 2014 Reliability and Maintainability Symposium. **Anais...IEEE**, jan. 2014Disponível em: <<http://ieeexplore.ieee.org/document/6798441/>>. Acesso em: 17 jul. 2017.

HAYKIN, S. **Redes Neurais: Princípios e Prática**. 2. ed. Porto Alegre: Bookman, 2007.

HERMANN, M.; PENTEK, T.; OTTO, B. **Design Principles for Industrie 4.0 Scenarios**. 2016 49th Hawaii International Conference on System Sciences (HICSS). **Anais...IEEE**, jan. 2016Disponível em: <<http://ieeexplore.ieee.org/document/7427673/>>. Acesso em: 18 set. 2018.

IZBICKI, R.; SANTOS, T. M. DOS. **Machine Learning sob a ótica estatística: Uma abordagem preditiva para estatística com exemplos em R**. [s.l.], 2018.

JAIN, A. K.; DUIN, P. W.; JIANCHANG MAO. Statistical pattern recognition: a review. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 22, n. 1, p. 4–37, 2000.

JANG, J.-S. R.; SUN, C.-T.; MIZUTANI, E. **Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence**. Upper Saddle River, New Jersey: Prentice Hall, 1997.

KAERCHER, A. R.; LUZ, D. F. DA. **Gerenciamento de riscos**. 1. ed. Rio de Janeiro: 2016.

KARDEC, A.; NASCIF, J. **Manutenção: função estratégica**. 3. ed. Rio de Janeiro: Qualitymark, 2010.

LIMA, F. P. A. et al. **Monitoramento da Integridade de Estruturas Aeronáuticas Utilizando um Método Inteligente Baseado em Sistemas Imunológicos Artificiais**. Proceeding Series of the Brazilian Society of Computational and Applied Mathematics. **Anais...8 dez. 2014**Disponível em: <<http://proceedings.sbmac.org.br/sbmac/article/view/265>>. Acesso em: 17 jul. 2017.

LOUZADA, D. R. **Detecção e caracterização de danos estruturais através de sensores a rede de Bragg e Redes Neurais Artificiais**. [s.l.] Pontifícia Universidade Católica do Rio de Janeiro, 2013.

MAN, J.; ZHOU, Q. Prediction of hard failures with stochastic degradation signals using Wiener process and proportional hazards model. **Computers & Industrial Engineering**, v. 125, p. 480–489, 1 nov. 2018.

MATHWORKS. **Improve Neural Network Generalization and Avoid Overfitting**. Disponível em: <<https://www.mathworks.com/help/nnet/ug/improve-neural-network-generalization-and-avoid-overfitting.html>>. Acesso em: 8 maio. 2018.

MCDERMOTT, R. E.; MIKULAK, R. J.; BEAUREGARD, M. R. **The basics of FMEA**. [s.l.] CRC Press, 2009.

MECHEFSKE, C. K. **Machine Condition Monitoring and Fault Diagnostics** Queen's University, 2005.

MENEZES, B.; VALENÇA, M. **Construindo Intervalos de Confiança na Previsão Da Potência do Vento Utilizando Reservoir Computing**. Anais do 11. Congresso Brasileiro de Inteligência Computacional. **Anais...SBIC**, 11 mar. 2016. Disponível em: <http://abricom.org.br/eventos/cbic_2013/bricsccicbic2013_submission_238>. Acesso em: 28 nov. 2018.

MOHAMMADI, A.; TAVAKOLAN, M. **Construction project risk assessment using combined fuzzy and FMEA**. 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS). **Anais...IEEE**, jun. 2013. Disponível em: <<http://ieeexplore.ieee.org/document/6608405/>>. Acesso em: 17 jul. 2017.

MONTE ALEGRE, D. [UNESP]. Metodologia dos observadores de estado para diagnose de falhas em sistemas contendo elementos finitos de placas de Kirchoff. **Aleph**, p. 131 f. : il. (algumas color.), 5 fev. 2009.

MOUBRAY, J. **Reliability-centered maintenance**. [s.l.] Butterworth Heinemann, 1997.

PEREIRA, R. R. et al. **On-line monitoring induction motors by fuzzy logic: A study for predictive maintenance operators**. 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). **Anais...IEEE**, ago. 2016 Disponível em: <<http://ieeexplore.ieee.org/document/7603373/>>. Acesso em: 17 jul. 2017.

PITAKRAT, T. et al. Hora: Architecture-aware online failure prediction. **Journal of Systems and Software**, v. 137, p. 669–685, 1 mar. 2018.

POKORADI, L.; SZAMOSI, B. **Fuzzy failure modes and effects analysis with summarized center of gravity defuzzification**. 2015 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI). **Anais...IEEE**, nov. 2015. Disponível em: <<http://ieeexplore.ieee.org/document/7382911/>>. Acesso em: 25 jul. 2017.

PYLE, D. **Data preparation for data mining**. [s.l.] Morgan Kaufmann Publishers, 1999.

ROSA, J. L. G. **Fundamentos da inteligência artificial**. Rio de Janeiro: 2011.

SANTOS, F. M. DA C.; SILVA, I. N. DA; SUETAKE, M. Sobre a aplicação de sistemas inteligentes para diagnóstico de falhas em máquinas de indução - uma visão geral. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, v.

23, n. 5, p. 553–569, out. 2012.

SCHMIDTKE, N. et al. **Technical potentials and challenges within internal logistics 4.0**. 2018 4th International Conference on Logistics Operations Management (GOL). **Anais...IEEE**, abr. 2018. Disponível em: <<https://ieeexplore.ieee.org/document/8378072/>>. Acesso em: 18 set. 2018.

SILVA, I. N. DA; SPATTI, D. H.; FLAUZINO, R. A. **Redes neurais artificiais para engenharia e ciências aplicadas**. São Paulo: Artliber, 2010.

SIMAS, E. **Introdução às Redes Neurais Artificiais**. Salvador, Semana de Engenharia e Tecnologia do IFBA, 2011. Disponível em: <<http://www.dee.eng.ufba.br/home/simas/introdRNA.pdf>>. Acesso em: 20 set. 2017.

SOUZA, S. S. DE; LIMA, C. R. C. **Manutenção centrada em confiabilidade como ferramenta estratégica**, 2003. Disponível em: <http://ftp.demec.ufpr.br/disciplinas/TM285/Conte%FAdos/Complementos/Artigo MCC como ferram estrat%E9g ENEGEP2003_TR0109_1353.pdf>. Acesso em: 22 set. 2017.

TEIXEIRA, F. H. **Metodologia para definição de intervenção de manutenção em estruturas baseada no seu nível de segurança por meio da confiabilidade estrutural – Manutenção Baseada na Condição Probabilística**. [s.l.] Pontifícia Universidade Católica de Minas Gerais, 2016.

TIAN, Z. **An artificial neural network approach for remaining useful life prediction of equipments subject to condition monitoring**. 2009 8th International Conference on Reliability, Maintainability and Safety. **Anais...IEEE**, jul. 2009. Disponível em: <<http://ieeexplore.ieee.org/document/5270220/>>. Acesso em: 10 out. 2017.

TIAN, Z.; WORG, L.; SAFAEI, N. A neural network approach for remaining useful life prediction utilizing both failure and suspension histories. **Mechanical Systems and Signal Processing**, v. 24, n. 5, p. 1542–1555, 1 jul. 2010.

TUPA, J.; SIMOTA, J.; STEINER, F. Aspects of Risk Management Implementation for Industry 4.0. **Procedia Manufacturing**, v. 11, p. 1223–1230, 1 jan. 2017.

VALE@INFORMAR. **Dispositivo criado por empregados aumenta produtividade de viradores de vagões**. Disponível em: <<https://valeinformar.valeglobal.net/BR/MA/Paginas/16-12-07-Dispositivo-criado-por-empregados-aumenta-produtividade-de-viradores-de-vagões.aspx>>. Acesso em: 17 jan. 2019.

WANG, X. et al. **Intelligent fault diagnosis and prediction technologies for condition based maintenance of track circuit**. 2016 IEEE International Conference on Intelligent Rail Transportation (ICIRT). **Anais...IEEE**, ago. 2016. Disponível em:

<<http://ieeexplore.ieee.org/document/7588745/>>. Acesso em: 17 jul. 2017.

WOOD, A. S. et al. **Fuzzy criticality assessment for process equipments maintenance**. International Conference on Automatic Control and Artificial Intelligence (ACAI 2012). **Anais...**Institution of Engineering and Technology, 2012. Disponível em: <<http://digital-library.theiet.org/content/conferences/10.1049/cp.2012.1168>>. Acesso em: 17 jul. 2017.

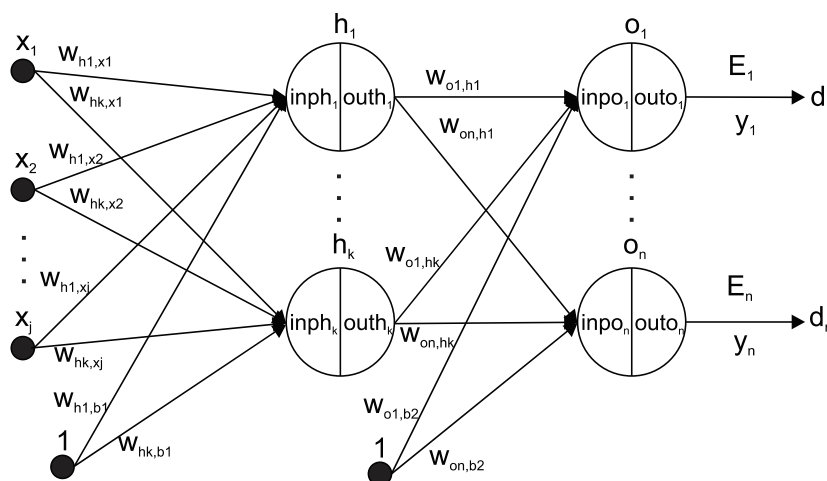
WU, B.; TIAN, Z.; CHEN, M. Condition-based Maintenance Optimization Using Neural Network-based Health Condition Prediction. **Quality and Reliability Engineering International**, v. 29, n. 8, p. 1151–1163, dez. 2013.

ZHENG, X. et al. **Drive system reliability analysis of wind turbine based on fuzzy fault tree**. 2016 35th Chinese Control Conference (CCC). **Anais...**IEEE, jul. 2016. Disponível em: <<http://ieeexplore.ieee.org/document/7554422/>>. Acesso em: 30 jun. 2017.

APÊNDICE I

Matematicamente, a sequência de operações em forma matricial para a execução do treinamento *Backpropagation* e *Momentum* para uma rede com apenas uma camada intermediária (Figura 29) ocorreu conforme as Eq.29 até 47.

Figura 29 - Variáveis de rede com uma camada escondida



Fonte: Elaborada pela autora.

- *Passo forward:*

A matriz w^1 se refere aos pesos sinápticos entre as entradas (vetor x) e os neurônios da camada intermediária.

$$w^1 = \begin{bmatrix} w_{h1,x1} & w_{h1,x2} & \dots & w_{h1,xj} & w_{h1,b1} \\ w_{hk,x1} & w_{hk,x2} & \dots & w_{hk,xj} & w_{hk,b1} \end{bmatrix} \quad (29)$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_j \\ 1 \end{bmatrix} \quad (30)$$

Inicialmente, faz-se o somatório ponderado das entradas na entrada de cada neurônio da camada intermediária (*inph*), como na Eq. 31.

$$inph = \begin{bmatrix} inph_1 \\ inph_2 \\ \vdots \\ inph_k \end{bmatrix} = w^1 X x = \begin{bmatrix} w_{h1,x1} & w_{h1,x2} & \cdots & w_{h1,xj} & w_{h1,b1} \\ & & & \vdots & \\ & & & w_{hk,xj} & w_{hk,b1} \\ w_{hk,x1} & w_{hk,x2} & \cdots & & \end{bmatrix} X \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_j \\ 1 \end{bmatrix} \quad (31)$$

Em seguida, aplica-se a função de ativação como saída da camada intermediária (*outh*).

$$outh = f(inph) = \begin{bmatrix} outh_1 \\ \vdots \\ outh_k \end{bmatrix} \quad (32)$$

A matriz w^2 se refere aos pesos sinápticos entre as saídas da camada intermediária (vetor h) e os neurônios da camada de saída.

$$h = \begin{bmatrix} outh_1 \\ \vdots \\ outh_k \\ 1 \end{bmatrix} \quad (33)$$

$$w^2 = \begin{bmatrix} w_{o1,h1} & \cdots & w_{o1,hk} & w_{o1,b2} \\ & & \vdots & \\ w_{on,h1} & \cdots & w_{on,hk} & w_{on,b2} \end{bmatrix} \quad (34)$$

Na entrada dos neurônios da camada de saída (*inpo*) também é feito o somatório ponderado entre h e w^2 .

$$inpo = w^2 X h = \begin{bmatrix} w_{o1,h1} & \cdots & w_{o1,hk} & w_{o1,b2} \\ & & \vdots & \\ w_{on,h1} & \cdots & w_{on,hk} & w_{on,b2} \end{bmatrix} X \begin{bmatrix} outh_1 \\ \vdots \\ outh_k \\ 1 \end{bmatrix} = \begin{bmatrix} inpo_1 \\ \vdots \\ inpo_n \end{bmatrix} \quad (35)$$

Aplica-se a função de ativação na camada de saída, obtendo-se o resultado da rede neural (*outo*).

$$outo = f(inpo) = \begin{bmatrix} outo_1 \\ \vdots \\ outo_n \end{bmatrix} \quad (36)$$

- Cálculo do erro quadrático médio:

$$E = \sum_{i=1}^n \frac{1}{2} * (d_i - outo_i)^2 = E_1 + \dots + E_n = \frac{1}{2} * (d_1 - outo_1)^2 + \dots + \frac{1}{2} * (d_n - outo_n)^2 \quad (37)$$

- Passo *backward*:

Inicialmente, calcula-se a diferença entre o valor desejado (d) e o resultado dado pela rede ($outo$), como na Eq. 38. Então, calcula-se a derivada da função de ativação do neurônio (Eq. 39). Na Eq. 40 multiplica-se o resultado das Eq. 38 e 39, obtendo α^2 . Por fim, na Eq. 41 obtém-se o erro $\partial E / \partial w^2$.

$$\frac{\partial E}{\partial outo} = -(d - outo) = \begin{bmatrix} outo_1 - d1 \\ outo_2 - d2 \\ \vdots \\ outo_n - dn \end{bmatrix} \quad (38)$$

$$\frac{\partial outo}{\partial inpo} = \begin{bmatrix} f'(inpo_1) \\ f'(inpo_2) \\ \vdots \\ f'(inpo_n) \end{bmatrix} \quad (39)$$

$$\alpha^2 = \frac{\partial E}{\partial outo} \cdot \frac{\partial outo}{\partial inpo} = \begin{bmatrix} (outo_1 - d1) * f'(inpo_1) \\ (outo_2 - d2) * f'(inpo_2) \\ \vdots \\ (outo_n - dn) * f'(inpo_n) \end{bmatrix} = \begin{bmatrix} \delta_1^2 \\ \delta_2^2 \\ \vdots \\ \delta_n^2 \end{bmatrix} \quad (40)$$

$$\frac{\partial E}{\partial w^2} = \alpha^2 X outh' = \begin{bmatrix} \alpha_1^2 \\ \alpha_2^2 \\ \vdots \\ \alpha_n^2 \end{bmatrix} X \begin{bmatrix} outh_1 & outh_2 & \dots & outh_k & 1 \end{bmatrix} = \begin{bmatrix} \alpha_1^2 * outh_1 & \alpha_1^2 * outh_2 & \dots & \alpha_1^2 * outh_k & \alpha_1^2 \\ \alpha_2^2 * outh_1 & \alpha_2^2 * outh_2 & \dots & \alpha_2^2 * outh_k & \alpha_2^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_n^2 * outh_1 & \alpha_n^2 * outh_2 & \dots & \alpha_n^2 * outh_k & \alpha_n^2 \end{bmatrix} \quad (41)$$

Para realizar a atualização dos pesos na camada de saída, aplica-se a Eq. 42, obtendo-se uma nova matriz de pesos (w_{novo}^2), sendo η a taxa de aprendizagem e δ a taxa de *Momentum*. No treinamento *Backpropagation*, δ vale zero.

$$w_{novo}^2 = w^2 - \eta * \frac{dE}{dw^2} + \delta(w^2 - w_{antigo}^2) \quad (42)$$

O processo para a atualização dos pesos na camada intermediária ocorre seguindo a mesma lógica da camada de saída. As Eq. 43 a 44 mostram o passo a passo para o cálculo do erro.

$$\frac{\partial E}{\partial outh} = (\alpha^2)' X w^2 = [\alpha_1^2 \quad \alpha_2^2 \quad \dots \quad \alpha_n^2] X \begin{bmatrix} w_{o1,h1} & w_{o1,h2} & \dots & w_{o1,hk} & w_{o1,b2} \\ w_{o2,h1} & w_{o2,h2} & \dots & w_{o2,hk} & w_{o2,b2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{on,h1} & w_{on,h2} & \dots & w_{on,hk} & w_{on,b2} \end{bmatrix} \quad (43)$$

$$\frac{\partial outh}{\partial inph} = \begin{bmatrix} f'(inph_1) \\ f'(inph_2) \\ \vdots \\ f'(inph_k) \end{bmatrix} \quad (44)$$

$$\alpha^1 = \left(\frac{\partial E}{\partial outh} \right)' \cdot \frac{\partial outh}{\partial inph} = \begin{bmatrix} (\alpha_1^1 * w_{o1,h1} + \dots + \alpha_n^1 * w_{o1,b2}) * f'(inph_1) \\ (\alpha_1^1 * w_{o2,h1} + \dots + \alpha_n^1 * w_{o2,b2}) * f'(inph_2) \\ \vdots \\ (\alpha_1^1 * w_{on,h1} + \dots + \alpha_n^1 * w_{on,b2}) * f'(inph_k) \end{bmatrix} \quad (45)$$

$$\frac{\partial E}{\partial w^1} = \alpha^1 X x' = \begin{bmatrix} \alpha_1^1 \\ \alpha_2^1 \\ \vdots \\ \alpha_n^1 \end{bmatrix} X [x_1 \quad x_2 \quad \dots \quad x_j \quad 1] =$$

$$\begin{bmatrix} \alpha_1^1 * x_1 & \alpha_1^1 * x_2 & \dots & \alpha_1^1 * x_j & \alpha_1^1 \\ \alpha_2^1 * x_1 & \alpha_2^1 * x_2 & \dots & \alpha_2^1 * x_j & \alpha_2^1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_n^1 * x_1 & \alpha_n^1 * x_2 & \dots & \alpha_n^1 * x_j & \alpha_n^1 \end{bmatrix} \quad (46)$$

Os pesos da camada intermediária são atualizados conforme a Eq. 47.

$$w_{novo}^1 = w^1 - \eta \cdot \frac{dE}{dw^1} + \delta(w^1 - w_{antigo}^1) \quad (47)$$