



**PROGNÓSTICO DE FALHAS BASEADO EM
REDES NEURAIIS COM ESTADOS DE ECO**

EDGAR JHONNY AMAYA SIMEÓN

**TESE DE DOUTORADO EM SISTEMAS MECATRÔNICOS
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**PROGNÓSTICO DE FALHAS BASEADO EM REDES
NEURAIS COM ESTADOS DE ECO**

EDGAR JHONNY AMAYA SIMEÓN

ORIENTADOR: ALBERTO JOSÉ ÁLVARES

TESE DE DOUTORADO EM SISTEMAS MECATRÔNICOS

PUBLICAÇÃO: ENM.TD-10/15

BRASÍLIA/DF: AGOSTO – 2015

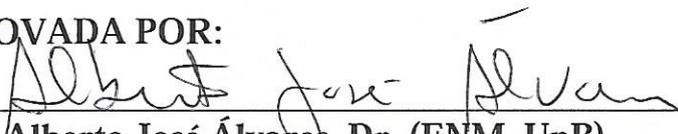
UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

PROGNÓSTICO DE FALHAS BASEADO EM REDES NEURAIS
COM ESTADOS DE ECO

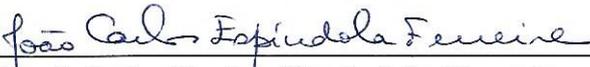
EDGAR JHONNY AMAYA SIMEÓN

TESE SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA
MECÂNICA DA FACULDADE DE TECNOLOGIA DA
UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM
SISTEMAS MECATRÔNICOS.

APROVADA POR:



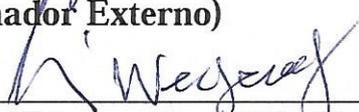
Prof. Alberto José Álvares, Dr. (ENM-UnB)
(Orientador)



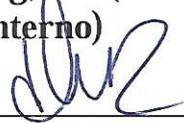
Prof. João Carlos Espíndola Ferreira, Dr. (UFSC)
(Examinador Externo)



Prof. Alexandre Ricardo Soares Romariz, Dr. (ENE-UnB)
(Examinador Externo)



Prof. Li Weigang, Dr. (CIC-UnB)
(Examinador Interno)



Prof. José Alfredo Ruiz Vargas, Dr. (ENE-UnB)
(Examinador Interno)

BRASÍLIA/DF, 28 DE AGOSTO DE 2015

FICHA CATALOGRÁFICA

SIMEÓN, EDGAR JHONNY AMAYA

Prognóstico de Falhas Baseado em Redes Neurais com Estados de Eco [Distrito Federal] 2015. xix, 125p., 210 x 297 mm (ENM/FT/UnB, Doutor, Sistemas Mecatrônicos, 2015).

Tese de Doutorado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

- | | |
|--------------------------|-------------------------------------|
| 1. Prognóstico de Falhas | 2. Redes Neurais com Estados de Eco |
| 3. Algoritmo ABC | 4. Prognóstico Baseado em Dados |
| 5. Otimização | 6. Computação com Reservatórios |
| I. ENM/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

AMAYA, E. J. (2015). Prognóstico de Falhas Baseado em Redes Neurais com Estados de Eco. Tese de Doutorado em Sistemas Mecatrônicos, Publicação ENM.TD-10/15, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 125p.

CESSÃO DE DIREITOS

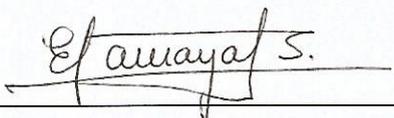
AUTOR: Edgar Jhonny Amaya Simeón.

TÍTULO: Prognóstico de Falhas Baseado em Redes Neurais com Estados de Eco.

GRAU: Doutor

ANO: 2015

É concedida à Universidade de Brasília permissão para reproduzir cópias desta tese de doutorado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa tese de doutorado pode ser reproduzida sem autorização por escrito do autor.



Edgar Jhonny Amaya Simeón

QI. 09 BL. E AP. 211

71.020-058 Brasília - DF - Brasil.

DEDICATÓRIA

*A Deus que iluminou o meu caminho durante esta caminhada.
A meus pais Crisanto e Maurelia pelo apoio e educação exemplar
A todos meus irmãos por torcerem, para a realização desse objetivo
A todos meus amigos e colegas, pelas alegrias e tristezas compartilhadas.*

AGRADECIMENTOS

A Deus, pela sua luz, paz e amor que me proporciona todos os dias, mesmo nos momentos mais difíceis, nunca me abandonou cuidando dos menores detalhes em minha existência;

À minha família, que mesmo estando longe, sempre torceu muito por mim;

Ao professor Alberto José Álvares, pela motivação, competência e pelos conhecimentos transmitidos, orientação e apoio em todo o processo de doutorado;

Ao coordenador do programa PPMEC, professor Edson Paulo da Silva, pela confiança, motivação e compreensão;

Ao Grupo de Automação e Controle (GRACO) e ao programa de pós-graduação em sistemas mecatrônicos da Universidade de Brasília pelos recursos físicos fornecidos;

A todos os professores que formam o corpo docente do programa de pós-graduação em Sistemas Mecatrônicos (PPMEC);

A todos os amigos em Brasília e no mundo, assim como aos colegas de laboratório pelo apoio, incentivo e pelos momentos de distração.

À comunidade peruana em Brasília e ao povo Brasileiro que me fizeram sentir como em casa.

Ao CNPq, FAP-DF, DPP-UnB e FINATEC, pelo suporte financeiro.

RESUMO

PROGNÓSTICO DE FALHAS BASEADO EM REDES NEURAIIS COM ESTADOS DE ECO

Autor: Edgar Jhonny Amaya Simeón

Orientador: Alberto José Álvares

Programa de Pós-graduação em Sistemas Mecatrônicos

Brasília, agosto de 2015

A rapidez e precisão do prognóstico de falhas pode reduzir os custos de manutenção e diminuir a probabilidade de acidentes. Uma das abordagens de prognóstico é baseada em dados históricos coletados de sensores que monitoram as condições de operação de máquinas. A maioria de métodos de prognóstico baseados em dados utilizam Redes Neurais Artificiais (RNA), e entre eles, as Redes Neurais Recorrentes (RNR) constituem uma importante ferramenta para lidar com problemas de natureza dinâmica, devido aos laços de realimentação entre suas camadas. A rede com estados de eco (ESN – *Echo State Networks*) é um tipo de RNR que se caracteriza por possuir um reservatório de dinâmicas gerado aleatoriamente, apenas são treinados os pesos da camada de saída. No entanto, é necessário ajustar os parâmetros e a topologia para gerar uma ESN idônea para uma determinada aplicação. Neste trabalho, é desenvolvida uma abordagem para prognóstico chamada ESN-ABC, um sistema híbrido baseado em ESN e o algoritmo de colônia de abelhas artificiais (ABC – *Artificial Bee Colony*). O algoritmo ABC através da minimização da função de aptidão, busca simultaneamente os melhores valores dos parâmetros e pesos do reservatório da ESN. A ESN com os parâmetros definidos e os pesos treinados é utilizada para estimar a vida útil remanescente (RUL – *Remaining Useful Life*). Um estudo de caso foi implementado para verificar o método desenvolvido aplicado ao prognóstico de RUL, utilizando um conjunto de dados de sinais multivariáveis coletados a partir de um processo de simulação dinâmica de turbinas turbofan, disponível no repositório da NASA. Os resultados obtidos pela abordagem ESN-ABC são comparados com os obtidos pela ESN clássica e por outros autores que utilizaram o mesmo conjunto de dados. A comparação quantitativa através de métricas de prognóstico mostraram que a abordagem desenvolvida tem um melhor desempenho no prognóstico de RUL neste caso particular.

RESUMEN

PRONÓSTICO DE FALLAS BASADO EN REDES NEURONALES CON ESTADOS DE ECO

Autor: Edgar Jhonny Amaya Simeón

Supervisor: Alberto José Álvares

Programa de Pós-graduação em Sistemas Mecatrônicos

Brasília, agosto de 2015

La rapidez y precisión del pronóstico de fallas puede reducir los costos de mantenimiento y disminuir la probabilidad de accidentes. Uno de los enfoques de pronóstico es basado en datos históricos colectados de sensores que monitorean las condiciones de operación de máquinas. La mayoría de los métodos de pronóstico basados en datos utilizan Redes Neuronales Artificiales (RNA), y entre ellos, las Redes Neuronales Recurrentes (RNR) constituyen una importante herramienta para tratar con problemas de naturaleza dinámica, debido a los lazos de realimentación entre sus capas. La red con estados de eco (ESN – *Echo State Networks*) es un tipo de RNR que se caracteriza por poseer un reservorio de dinámicas generado aleatoriamente, apenas son entrenados los pesos de la capa de salida. Sin embargo, es necesario ajustar los parámetros y la topología para generar una ESN idónea para una determinada aplicación. En este trabajo, es desarrollado un enfoque para pronóstico llamado ESN-ABC, un sistema híbrido basado en ESN y el algoritmo de colonia de abejas artificiales (ABC – *Artificial Bee Colony*). El algoritmo ABC a través de la minimización de la función de aptitud, busca simultáneamente los mejores valores de los parámetros y pesos del reservorio de la ESN. La ESN con los parámetros definidos y los pesos entrenados es utilizada para estimar la vida útil remaneciente (RUL – *Remaining Useful Life*). Un estudio de caso fue implementado para verificar el método desarrollado aplicado al pronóstico de RUL, utilizando un conjunto de datos de señales multi-variables colectados a partir de un proceso de simulación dinámica de turbinas turbofan, disponible en el repositorio de la NASA. Los resultados obtenidos por el enfoque ESN-ABC son comparados con los obtenidos por la ESN clásica y por otros autores que utilizaron el mismo conjunto de datos. La comparación cuantitativa a través de métricas de pronóstico demostró que el enfoque desarrollado tiene un mejor desempeño en pronóstico de RUL en este caso particular.

ABSTRACT

FAILURE PROGNOSTIC BASED ON ECHO STATE NETWORK

Author: Edgar Jhonny Amaya Simeón

Supervisor: Alberto José Álvares

Programa de Pós-graduação em Sistemas Mecatrônicos

Brasília, august of 2015

Quick and precise prognostic of failures can reduce maintenance costs and decrease accidents probabilities. One of prognostic approaches is based on historical data collected from sensors that monitor operating conditions of machines. Most prognostic methods based on data use Artificial Neural Networks (ANN), and among them, the Recurrent Neural Networks (RNN) are an important tool for dealing with dynamic nature problems, due to feedback loops between their layers. The Echo State Networks (ESN) is a type of RNN characterized by having a dynamic reservoir randomly generated, and only are trained the weights of the output layers. However, it is necessary to adjust the parameters and the topology to generate a suitable ESN for a particular application. In this work an approach of prognostic called ESN-ABC is developed as a hybrid system based on ESN algorithm and Artificial Bee Colony (ABC). The ABC algorithm through minimizing the fitness function, searches simultaneously the best parameter values and weights of the ESN's reservoir. The ESN with defined parameters and trained weights is used to estimate a Remaining Useful Life (RUL). A case study was implemented to verify the developed method applied to the RUL prognostic, using a dataset of multivariate signals collected from a dynamic simulation process of turbofan engines, available in the repository of the NASA. The results obtained by the ESN-ABC approach are compared with the obtained by classical ESN and other authors who used the same dataset. The quantitative comparison through prognostic metrics showed that the approach developed has better prognostic performance RUL in this particular case.

SUMÁRIO

1	INTRODUÇÃO.....	1
1.1	RESUMO ORIENTATIVO DA TESE.....	3
1.1.1	Hipótese a comprovar.....	3
1.1.2	Objetivo Geral.....	3
1.1.3	Objetivos específicos.....	3
1.2	ORIGINALIDADE E CONTRIBUIÇÕES.....	4
1.2.1	Parametrização e Treinamento Simultâneo de Reservatório.....	4
1.2.2	Implementação de um Método para Otimização de Reservatório.....	4
1.2.3	Desenvolvimento de uma ferramenta para prognóstico.....	5
1.3	ESTRUTURA DO DOCUMENTO.....	5
2	FUNDAMENTAÇÃO TEÓRICA.....	7
2.1	PROGNÓSTICO DE FALHAS.....	7
2.2	ABORDAGENS DE PROGNÓSTICO.....	7
2.2.1	Abordagens baseadas em experiências.....	10
2.2.2	Abordagens baseadas em dados.....	11
2.2.3	Abordagens baseadas em modelos.....	13
2.2.4	Abordagens híbridas.....	14
2.3	VIDA ÚTIL REMANESCENTE.....	15
2.4	AQUISIÇÃO E PROCESSAMENTO DE DADOS.....	16
2.4.1	Aquisição de dados.....	17
2.4.2	Processamento de dados.....	17
2.4.2.1	Extração de características.....	18
2.4.2.2	Seleção de características.....	18
2.5	ALGORITMOS PARA PROGNÓSTICO.....	19
2.6	ALGORITMO DE OTIMIZAÇÃO ABC.....	23

2.6.1	Fase de Inicialização.....	25
2.6.2	Fase de Abelhas Operárias	25
2.6.3	Fase de Abelhas Seguidoras	26
2.6.4	Fase de Abelhas Escoteiras.....	26
2.7	SÍNTESE DO CAPÍTULO	27
3	REVISÃO DE LITERATURA: REDES COM ESTADOS DE ECO.....	28
3.1	INTRODUÇÃO.....	28
3.2	NEURÔNIO ARTIFICIAL	28
3.3	ARQUITETURAS DE REDES NEURAIS ARTIFICIAIS	29
3.3.1	Redes Neurais Progressivas.....	30
3.3.2	Redes Neurais Recorrentes.....	31
3.4	REDES COM ESTADO DE ECO.....	33
3.4.1	Propriedades do Estado de Eco (ESP).....	35
3.4.2	Geração do Reservatório	35
3.5	TREINAMENTO DE UMA ESN.....	36
3.5.1	Inicialização dos Pesos	36
3.5.2	Amostragem Dinâmica de Treinamento.....	37
3.5.3	Treinamento do <i>Readout</i>	37
3.6	DESAFIOS NAS REDES COM ESTADO DE ECO.....	38
3.6.1	Otimização do Reservatório	38
3.6.2	Camada de saída não-linear.....	40
3.6.3	Aprendizado Incremental	41
3.7	APLICAÇÕES DE RNA EM PROGNÓSTICO.....	42
3.7.1	Prognóstico usando RNA	43
3.7.2	Prognóstico usando ESN	43
3.8	SÍNTESE DO CAPÍTULO	44

4	ABORDAGEM DESENVOLVIDA DE PROGNÓSTICO DE RUL BASEADO EM ESN	46
4.1	MOTIVAÇÃO.....	46
4.2	ABORDAGEM DESENVOLVIDA	47
4.2.1	Aquisição de Dados	47
4.2.2	Representação da solução.....	49
4.2.3	Função de Aptidão.....	51
4.2.4	Parâmetros	51
4.2.5	Algoritmo de otimização	51
4.2.6	Prognóstico de RUL	53
4.3	MODELAGEM FUNCIONAL IDEF0.....	55
4.3.1	Atividade Aquisição de Dados	56
4.3.2	Atividade Otimização e Treinamento.....	57
4.3.3	Atividade Prognóstico de Falhas	57
4.4	SÍNTESE DO CAPÍTULO	63
5	ESTUDO DE CASO: PROGNÓSTICO DE RUL DE TURBINAS TURBOFAN	64
5.1	DESCRIÇÃO DOS DADOS	64
5.2	ESTRUTURA DOS DADOS.....	65
5.3	SELEÇÃO DO SUBCONJUNTO DE SENSORES.....	66
5.4	RESULTADOS DE PROGNÓSTICO DE RUL.....	72
5.4.1	ESN clássica	72
5.4.2	ESN-ABC.....	75
5.5	TRABALHOS BASEADOS NO REPOSITÓRIO DE PROGNÓSTICO.....	78
5.6	AVALIAÇÃO DE DESEMPENHO	79
5.7	SÍNTESE DO CAPÍTULO	81
6	CONCLUSÕES E TRABALHOS FUTUROS.....	82
6.1	CONCLUSÕES.....	82

6.2 TRABALHOS FUTUROS	83
REFERÊNCIAS BIBLIOGRÁFICAS	85
APÊNDICE A – MÉTRICAS DE PROGNÓSTICO	110
A.1 INTRODUÇÃO.....	110
A.2 MÉTRICAS DE EXATIDÃO.....	111
A.2.1 Exatidão	111
A.2.7 PHM08.....	111
A.2.8 MSE, MAE e RMSE.....	112
A.2.8 Taxa Falsos Positivos e Falsos Negativos (FPR e FNR)	112
A.2.3 Porcentagem do Erro Médio Absoluto (MAPE).....	113
A.3 MÉTRICAS DE PRECISÃO.....	113
A.2.2 Erro Médio (ME)	113
A.2.2 Desvio Médio Absoluto (MAD)	113
A.2.2 Precisão	114
A.4 MÉTRICAS DE PROGNÓSTICO	114
A.2.4 Horizonte de prognóstico (PH)	114
A.2.5 Medida de desempenho (α - λ)	114
A.2.6 Medida de exatidão relativa (ER)	115
A.2.7 Convergência (CV)	115
APÊNDICE B – IMPLEMENTAÇÃO COMPUTACIONAL	117
B.1 AQUISIÇÃO DE DADOS	117
B.2 TREINAMENTO CLÁSSICO DE UMA ESN.....	118
B.2.1 Seleção do Conjunto de dados	120
B.2.2 Parametrização e Treinamento	120
B.3 OTIMIZAÇÃO ABORDAGEM HÍBRIDA ESN-ABC.....	121
B.4 RESULTADOS DE PROGNÓSTICO	122

APÊNDICE C – PUBLICAÇÕES REALIZADAS E A PUBLICAR.....	124
C.1 TRABALHOS PUBLICADOS	124
C.1.1 Manutenção Baseada em Condição	124
C.1.2 Diagnóstico de Falhas	124
C.1.3 Prognóstico de Falhas	125
C.2 TRABALHO SUBMETIDO EM JOURNAL INTERNACIONAL	125

LISTA DE TABELAS

TABELA 2.1- PRINCÍPIOS, VANTAGENS E LIMITAÇÕES DAS ABORDAGENS DE PROGNÓSTICO. ..	9
TABELA 2.2- PRINCIPAIS ALGORITMOS PARA PROGNÓSTICO (ADAPTADO DE LEE <i>ET AL.</i> , 2014).	21
TABELA 4.1- PARÂMETROS DO ALGORITMO ABC (KARABOGA, 2005).....	52
TABELA 4.2- PARÂMETROS DA ESN (LUKOSEVICIUS, 2012).....	52
TABELA 5.1- CARACTERÍSTICAS DOS QUATRO CONJUNTOS DE DADOS.....	66
TABELA 5.2- DESCRIÇÃO DAS VARIÁVEIS MONITORADAS.....	67
TABELA 5.3- EXEMPLO DA ESTRUTURA DOS DADOS DA TURBINA 1.	67
TABELA 5.4- PUBLICAÇÕES PARA CADA CONJUNTO DE DADOS.	68
TABELA 5.5- PUBLICAÇÕES BASEADOS COM DADOS COMPLETOS TREINAMENTO/TESTE.	69
TABELA 5.6- AJUSTE DE PARÂMETROS DA ESN CLÁSSICA.	73
TABELA 5.7- PARÂMETROS AJUSTADOS PELA ABORDAGEM ESN-ABC.....	76
TABELA 5.8- PUBLICAÇÕES FOCADAS EM DETECÇÃO E PROGNÓSTICO.....	78
TABELA 5.9- MÉTODOS COM ENTRADAS DE SENSORES E SAÍDA RUL.	79
TABELA 5.10- PUBLICAÇÕES E AS MÉTRICAS DE DESEMPENHO USADAS.	80
TABELA 5.11- COMPARAÇÃO DOS RESULTADOS ATRAVÉS DE MÉTRICAS DE PROGNÓSTICO..	81

LISTA DE FIGURAS

FIGURA 2.1- CLASSIFICAÇÃO DAS ABORDAGENS DE PROGNÓSTICO DE FALHAS (ADAPTADA DE VACHTSEVANOS <i>ET AL.</i> , 2006).....	8
FIGURA 2.2- ABORDAGENS HÍBRIDAS DE PROGNÓSTICO (LIAO E KOTTIG, 2014).....	15
FIGURA 2.3- DEFINIÇÃO DA VIDA ÚTIL REMANESCENTE.	16
FIGURA 2.4- FENÔMENO DE PICO (THEODORIDIS E KOUTROUMBAS, 2006).	19
FIGURA 2.5- ALGORITMOS USADOS NAS ABORDAGENS BASEADAS EM DADOS (HUIGUO <i>ET AL.</i> , 2009).....	19
FIGURA 2.6- ALGORITMO ABC.	25
FIGURA 3.1- MODELO DE UM NEURÔNIO ARTIFICIAL (BOCCATO, 2013).	29
FIGURA 3.2- REDE PROGRESSIVA.	30
FIGURA 3.3- REDE NEURAL ARTIFICIAL RECORRENTE.....	32
FIGURA 3.4- ESTRUTURA GENÉRICA DE UMA ESN (PENG <i>ET AL.</i> , 2012).....	34
FIGURA 3.5- ESTRUTURA DO MODELO RNA USADA PARA PREDIÇÃO DE RUL (TIAN <i>ET AL.</i> , 2010).....	43
FIGURA 3.6- ESN MODIFICADA POR PENG <i>ET AL.</i> (2012A).	44
FIGURA 4.1- ARQUITETURA DA ABORDAGEM ESN-ABC.	48
FIGURA 4.2- ELEMENTOS DO VETOR SOLUÇÃO S.	50
FIGURA 4.3- ESTRUTURA DE UMA ATIVIDADE OU PROCESSO E SEUS DETALHES.....	55
FIGURA 4.4- DIAGRAMA IDEF0: SISTEMA DE PROGNÓSTICO DE RUL NÍVEL A0.	58
FIGURA 4.5- DIAGRAMA IDEF0: PRINCIPAIS ATIVIDADES DO SISTEMA DE PROGNÓSTICO DE RUL.	59
FIGURA 4.6- DIAGRAMA IDEF0: ATIVIDADES AQUISIÇÃO DE DADOS.	60
FIGURA 4.7- DIAGRAMA IDEF0: ATIVIDADES OTIMIZAÇÃO E TREINAMENTO.	61
FIGURA 4.8- DIAGRAMA IDEF0: ATIVIDADES PROGNÓSTICO DE FALHAS.....	62
FIGURA 5.1- DIAGRAMA SIMPLIFICADO DE UMA TURBINA TURBOFAN (SAXENA E GOEBEL, 2008).....	65
FIGURA 5.2- MÓDULOS E CONEXÕES DO MODELO DE SIMULAÇÃO (SAXENA E GOEBEL, 2008).	65
FIGURA 5.3- MEDIÇÕES DO SENSOR 2 DAS 100 TURBINAS E DISTRIBUIÇÃO DE VIDA ÚTIL.	68
FIGURA 5.4- CONJUNTO DE TODAS VARIÁVEIS MONITORADAS PARA A TURBINA 7.	70
FIGURA 5.5- SUBCONJUNTO DE VARIÁVEIS SELECIONADAS E A RUL DA TURBINA 7.....	71

FIGURA 5.6- SEQUÊNCIA DE TREINAMENTO USANDO ESN.....	74
FIGURA 5.7- SEQUÊNCIA DE TESTE USANDO ESN.....	74
FIGURA 5.8- RUL ESTIMADA ATRAVÉS DA ESN.....	75
FIGURA 5.9- SEQUÊNCIA DE TREINAMENTO USANDO ESN-ABC.....	77
FIGURA 5.10- SEQUÊNCIA DE TESTE USANDO ESN-ABC.....	77
FIGURA 5.11- RUL ESTIMADA ATRAVÉS DA ABORDAGEM ESN-ABC.....	78
FIGURA A.1- LIMIAR DE FALSOS POSITIVOS E NEGATIVOS.....	113
FIGURA A.2- HORIZONTE DE PREDIÇÃO.....	115
FIGURA B.1- ANÁLISE DE TENDÊNCIAS DAS VARIÁVEIS MONITORADAS.....	117
FIGURA B.2- ANÁLISE DA OCORRÊNCIA DE RUL.....	118
FIGURA B.3- ANÁLISE DOS MODOS DE OPERAÇÃO.....	119
FIGURA B.4- TREINAMENTO DE ESN ATRAVÉS DO MÉTODO CLÁSSICO.....	119
FIGURA B.5- AQUISIÇÃO E SELEÇÃO DO CONJUNTO DE DADOS.....	120
FIGURA B.6- PARAMETRIZAÇÃO DA ESN.....	120
FIGURA B.7- CRIAÇÃO DA REDE ESN.....	121
FIGURA B.8- TREINAMENTO E ARMAZENAMENTO DA ESN.....	121
FIGURA B.9- TREINAMENTO DE ESN ATRAVÉS DO ALGORITMO ABC.....	121
FIGURA B.10- FUNÇÃO DE APTIDÃO DA ABORDAGEM ESN-ABC.....	122
FIGURA B.11- PROGNÓSTICO DE RUL.....	123

LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

ABC	– Colônia de Abelhas Artificiais (<i>Artificial Bee Colony</i>)
APSO	– Otimização por Enxame de Partículas Adaptativo (<i>Adaptive Particle Swarm Optimization</i>)
ARMA	– Auto-Regressivo de Média Móvel (<i>Autoregressive-Moving Average</i>)
ART	– Teoria de Ressonância Adaptativa (<i>Adapative Resonance Theory</i>)
ARTMAP	– Mapa da Teoria de Ressonância Adaptativa (<i>Adapative Resonance Map</i>)
BPDC	– Decorrelação de Retropropagação (<i>Backpropagation Decorrelation</i>)
BPTT	– Retropropagação através do Tempo (<i>Backpropagation-Through-Time</i>)
CBM	– Manutenção Baseada Em Condição (<i>Condition Based Maintenance</i>)
C-MAPSS	– Sistema de Simulação de Aero Propulsão Modular Comercial (<i>Commercial Modular Aero Propulsion System Simulation</i>)
CNC	– Controle Numérico Computadorizado (<i>Computerized Numerical Control</i>)
CV	– Convergência (<i>Convergence</i>)
DBN	– Redes Bayesianas Dinâmicas (<i>Dynamic Bayesian Networks</i>)
DEA	– Algoritmo de Evolução Diferencial (<i>Differential Evolution Algorithm</i>),
DWNN	– Rede Neural Wavelet Dinâmico (<i>Dynamic Wavelet Neural Networks</i>)
EKF	– Filtro de Kalman Estendido (<i>Extended Kalman Filter</i>)
ELM	– Aprendizado de Máquina Extremo (<i>Extreme Learning Machine</i>)
EPNL	– Elementos de Processamento Não Linear (<i>Nonlinear Processing Elements</i>)
EPUS-PSO	– Estratégia de Utilização Eficiente de População para Otimização por Enxame de Partículas (<i>Efficient Population Utilization Strategy for Particle Swarm Optimizer</i>)
ER	– Exatidão Relativa (<i>Relative Accuracy</i>)
ESN	– Rede com Estados de Eco (<i>Echo State Networks</i>)
ESN-ABC	– Rede com Estados de Eco otimizada por Colônia de Abelhas Artificiais (<i>Echo State Networks Optimized by Artificial Bee Colony</i>)
ESP	– Propriedade do Estado de Eco (<i>Echo State Property</i>)
ETTF	– Tempo Estimado até a Falha (<i>Estimated Time To Failure</i>)
FFT	– Transformada Rápida de Fourier (<i>Fast Fourier Transform</i>)
FNN	– Redes Neurais Progressivas (<i>Feedforward Neural Networks</i>)

FN	– Falsos Negativos (<i>False Negatives</i>)
FP	– Falsos Positivos (<i>False Positives</i>)
GA	– Algoritmos Genéticos (<i>Genetic Algorithms</i>)
GUI	– Interface Gráfica de Usuário (<i>Graphical User Interface</i>)
HMM	– Modelo Oculto de Markov (<i>Hidden Markov Models</i>)
HPC	– Compressor de Alta Pressão (<i>High Pressure Compressor</i>)
HPT	– Turbina de Alta Pressão (<i>High Pressure Turbine</i>)
IA	– Inteligência Artificial (<i>Artificial Intelligence</i>)
IDEFO	– Definição Integrada 0 (<i>Integrated DEFinition 0</i>)
IGMM	– Modelo Incremental de Misturas Gaussianas (<i>Incremental Gaussian Mixture Model</i>)
IGMN	– Redes Incrementais de Misturas Gaussianas (<i>Incremental Gaussian Mixture Network</i>)
ISO	– Organização internacional para padronização (<i>International Organization for Standardization</i>)
JESS	– Linguagem de Sistema Especialista em Java (<i>Java Expert System Shell</i>)
KF	– Filtro de Kalman (<i>Kalman Filter</i>)
LPC	– Compressor de Baixa Pressão (<i>Low Pressure Compressor</i>)
LPT	– Turbina de Baixa Pressão (<i>Low Pressure Turbine</i>)
LSM	– Máquinas de Estado Líquido (<i>Liquid State Machines</i>)
MAE	– Erro Absoluto Médio (<i>Mean Absolute Error</i>),
MAPE	– Porcentagem do Erro Médio Absoluto (<i>Mean Absolute Percentage Error</i>)
MC	– Monitoração de Condição (<i>Condition Monitoring</i>)
ME	– Erro Médio (<i>Mean Error</i>)
MSE	– Erro Quadrático Médio (<i>Mean Square Error</i>)
MTBF	– Tempo Médio entre Falhas (<i>Mean Time Between Failure</i>)
MTTF	– Tempo Médio até a Falha (<i>Mean Time To Failure</i>)
NMSE	– Erro Quadrático Médio Normalizado (<i>Normalized Mean Square Error</i>)
NRMSE	– Raiz do Erro Quadrático Médio Normalizado (<i>Normalized Root-Mean-Square Error</i>)
PCA	– Análise de Componentes Principais (<i>Principal Component Analysis</i>)
PMC	– Perceptron Multi-Camadas (<i>Multi-Layers Perceptron</i>)
PF	– Filtro de Partículas (<i>Particle Filtering</i>)

PH	– Horizonte de Prognóstico (<i>Prognostic Horizon</i>)
PHM	– Prognóstico e Gestão de Saúde (<i>Prognostic and Health Management</i>)
PHM08	– Métrica de competição do congresso PHM no ano 2008
PNN	– Rede Neural Polinomial (<i>Polynomial Neural Network</i>)
PSO	– Otimização por Enxame de Partículas (<i>Particle Swarm Optimization</i>)
RBC	– Raciocínio Baseado em Casos (<i>Case Based Reasoning</i>)
RBF	– Função de Base Radial (<i>Radial Basis Function</i>)
RC	– Computação com Reservatórios (<i>Reservoir Computing</i>)
RNA	– Redes Neurais Artificiais (<i>Artificial Neural Networks</i>)
RNR	– Redes Neurais Recorrentes (<i>Recurrent Neural Networks</i>)
ROI	– Retorno sobre Investimento (<i>Return on Investment</i>)
RCDESIGN	– Projeto de Computação com Reservatórios (<i>Reservoir Computing Design</i>)
RTF	– Funcionamento até a Falha (<i>Run To Failure</i>)
RTRL	– Aprendizado Recorrente em Tempo Real (<i>Real-Time Recurrent Learning</i>)
RUL	– Vida Útil Remanescente (<i>Remaining Useful Life</i>)
RPM	– Radianos por minuto (<i>Radians Per Minute</i>)
SIMPREBAL	– Sistema Inteligente de Manutenção Preditiva de Balbina
SOM	– Mapas Auto-Organizáveis (<i>Self-Organized Maps</i>)
SRPF	– Filtro de Partículas Sensível ao Risco (<i>Risk Sensitive Particle Filtering</i>)
STFT	– Transformada de Fourier de Tempo Curto (<i>Short Time Fourier Transform</i>)
SVM	– Máquinas de Vetor Suporte (<i>Support Vector Machines</i>)
TRA	– Ângulo Resolver do Regulador de Pressão (<i>Throttle Resolver Angle</i>),
TTF	– Tempo até a Falha (<i>Time To Failure</i>)
WA	– Algoritmo de Minhoca (<i>Worm Algorithm</i>)
WNN	– Rede Neural Wavelet (<i>Wavelet Neural Network</i>)

1 INTRODUÇÃO

A Vida Útil Remanescente (RUL – *Remaining Useful Life*) é definida como a diferença entre o tempo atual e o fim da vida útil de um equipamento. O prognóstico da RUL é uma das tarefas mais importantes entre as várias rotinas e funcionalidades de um Sistema de Prognóstico e Gestão de Saúde (PHM – *Prognostic and Health Management*) (Dong e He, 2007a; Pecht, 2008; Pecht e Jaai, 2010; Gasperin *et al.*, 2011) e da Manutenção Baseada Em Condição (CBM – *Condition Based Maintenance*) (Cui *et al.*, 2004; Lee *et al.*, 2006; Wang, 2007; Wang e Zhang, 2008). Quando implementado, um sistema PHM permite avaliar as condições atuais de funcionamento de um equipamento ou sistema, prever o início de uma falha, e mitigar os riscos associados ao comportamento anormal do equipamento (Schwabacher, 2005 e Heng *et al.*, 2009a).

Tradicionalmente, um sistema PHM é constituído por três módulos principais, os quais são: detecção, diagnóstico e prognóstico. Nas últimas décadas, os módulos de detecção e diagnóstico (isolação) foram amplamente estudados e bem estabelecidos. Por outro lado, ultimamente, as técnicas relativas ao módulo de prognóstico atraíram a atenção da comunidade científica de PHM. O crescente interesse no desenvolvimento de métodos de prognóstico objetiva aumentar a produtividade e a competitividade, e reduzir as paradas não programadas e os custos de indisponibilidade (Kothamasu *et al.*, 2006). As falhas inesperadas em equipamentos podem gerar custos elevados em atividades de manutenção, e perdas no sistema produtivo. Conhecer antecipadamente a RUL de um equipamento auxiliaria a tomada de decisões relativas à manutenção.

O prognóstico de falhas costuma ser estimado em modo *off-line*; este cálculo se baseia em modelos estatísticos, usando as propriedades mecânicas dos materiais, as condições de operação, paradas de emergência, etc. No entanto, com o progresso da área de prognóstico, a RUL pode ser estimada em tempo real. A avaliação da RUL enquanto o equipamento está em operação é criticamente importante, uma vez que tem impacto no planejamento das atividades de manutenção, estoque de peças sobressalentes, desempenho operacional, e na rentabilidade do proprietário do equipamento (Jardine *et al.*, 2006; Altay e Green, 2006; Elwany e Gebraeel, 2008; Wang *et al.*, 2009; Kim e Kuo, 2009; Papakostas *et al.*, 2010).

Foram propostas diferentes abordagens para a tarefa de prognóstico de RUL (Vachtsevanos *et al.*, 2006). As abordagens propostas se encaixam em abordagens baseadas em experiências, baseadas em modelos matemáticos, baseadas em dados, e baseadas em modelos híbridos (fusão de duas ou mais abordagens) (Liao e Kottig, 2014). A escolha da abordagem depende de vários fatores como a precisão e o custo de implementação. Nesta tese preferiu-se aplicar a abordagem baseada em dados, uma vez que os dados coletados de históricos de falhas e de monitoração de condição (MC) são representativos dos fatores e características de funcionamento. A abordagem baseada em dados precisa apenas de dados históricos suficientes, estatisticamente, para realizar o prognóstico de RUL. Além disso, a abordagem baseada em dados demanda algoritmos rápidos e precisos.

As abordagens de prognóstico baseadas em dados históricos, segundo Ying *et al.* (2010) usam duas técnicas: estatísticas e Inteligência Artificial (IA). A maioria das abordagens IA empregam os seguintes tipos de Redes Neurais Artificiais (RNA): rede neural polinomial (PNN – *Polynomial Neural Networks*), rede neural *wavelet* dinâmico (DWNN – *Dynamic Wavelet Neural Networks*), mapas auto-organizáveis (SOM – *Self-Organized Maps*), Perceptron Múlti-Camadas (PMC), e Redes Neurais Recorrentes (RNR) (Ying *et al.*, 2010). Dos diferentes tipos de RNA, as RNR constituem uma importante ferramenta para lidar com problemas de natureza dinâmica, devido ao fato de que dispõem de laços de realimentação entre as diferentes camadas de neurônios. No entanto, o processo de treinamento desse tipo de rede se mostra relativamente complexo e apresenta dificuldades de robustez e convergência (Haykin, 1998; Lukosevicius e Jaeger, 2009).

Nos últimos anos, foi apresentada uma nova área de pesquisa, denominada Computação com Reservatórios (RC – *Reservoir Computing*), abrindo interessantes perspectivas para a aplicação de estruturas recorrentes, ao introduzir uma significativa simplificação no seu processo de treinamento. Uma rede pertencente à RC proposta por Jaeger (2001) é a chamada Rede com estados de Eco (ESN – *Echo State Networks*), que surge como uma solução para duas características muitas vezes adversas: simplicidade do modelo matemático; e capacidade de aproximar comportamentos dinâmicos não-lineares (Bocato, 2013).

Uma ESN tem basicamente três camadas: entrada, reservatório e saída (Jaeger, 2001). Sendo fixos os pesos da camada de entrada e do reservatório, superam-se as dificuldades do processo de treinamento das RNR, preservando o potencial dos laços de realimentação. O processo de treinamento é relativamente simples, consistindo basicamente em adaptar de

forma supervisionada os parâmetros da camada de saída, o que usualmente equivale a um combinador linear (Boccatto, 2013).

Apesar dessas vantagens, a definição da topologia e parâmetros de uma ESN requer experiência do usuário. Para a aplicação da ESN no prognóstico de RUL, propõe-se no presente trabalho uma abordagem que inclui como contribuição, propor um método para definir parâmetros e pesos otimizados de uma ESN através do algoritmo de Colônia de Abelhas Artificiais ABC (*Artificial Bee Colony*), inspirado nos trabalhos de Ferreira e Ludermir (2009), Boedecker *et al.* (2009b), e Sergio e Ludermir (2014). Isto permitirá um processo automatizado na definição dos parâmetros e treinamento dos pesos do reservatório de dinâmicas de uma ESN.

1.1 RESUMO ORIENTATIVO DA TESE

1.1.1 Hipótese a comprovar

Uma das necessidades no prognóstico de falhas é a rapidez de processamento de informações e a precisão dos resultados. Para atender estas demandas propõe-se a otimização dos parâmetros e pesos do reservatório de uma rede neural com estados de eco através do algoritmo de colônias de abelhas artificiais. A rede treinada será utilizada para estimar a Vida útil Remanescente de uma máquina, equipamento ou Sistema.

1.1.2 Objetivo Geral

Este trabalho tem como principal objetivo: Propor uma abordagem para prognóstico de vida útil remanescente baseada em dados históricos usando redes neurais com estados de eco com parâmetros e pesos do reservatório de dinâmicas otimizados através do algoritmo de colônias de abelhas artificiais.

1.1.3 Objetivos específicos

Pretende-se alcançar os seguintes objetivos específicos:

- Apresentar uma arquitetura de prognóstico de RUL baseado em dados históricos;
- Aplicar a rede ESN para estimação de RUL;
- Propor um método de otimização dos parâmetros e pesos do reservatório de uma ESN;
- Implementar um sistema híbrido ESN-ABC;
- Implementar o estudo de caso utilizando dados de degradação de turbinas *turbofan* adquiridos do repositório de prognóstico da NASA;

- Avaliar os resultados de prognóstico usando métricas de prognóstico como indicadores de comparação quantitativa;
- Comparar os resultados alcançados pela abordagem desenvolvida com os resultados alcançados por outros pesquisadores que usaram os mesmos conjuntos de dados do repositório de prognóstico.

1.2 ORIGINALIDADE E CONTRIBUIÇÕES

O estado da arte das técnicas de prognóstico de RUL de equipamentos foi revisado. Esta síntese de revisão da literatura permitiu identificar lacunas de conhecimento com alertas dos pesquisadores para oportunidades de contribuições chaves nesta área de pesquisa. Com base nessa revisão, nesta tese apresentam-se as seguintes contribuições originais na área de prognóstico de RUL baseado em dados históricos e ESN:

1.2.1 Parametrização e Treinamento Simultâneo de Reservatório

Projeto, implementação e a análise de uma abordagem para a parametrização e treinamento simultâneo do reservatório de dinâmicas de uma ESN que possibilite um processo de treinamento preciso, proporcionando agilidade ao processo de prognóstico de falhas de máquinas, especificamente ao prognóstico de RUL de turbinas, cujos dados foram gerados a partir do Sistema de Simulação de Aero Propulsão Modular Comercial (C-MAPSS – *Commercial Modular Aero Propulsion System Simulation*). Comparação dos resultados obtidos pelo método ESN-ABC com o resultado de outros modelos de prognóstico.

1.2.2 Implementação de um Método para Otimização de Reservatório

A camada de reservatórios de uma ESN consiste de uma grande quantidade de neurônios com pesos fixos e gerados de forma aleatória. Entre os parâmetros tem-se o número de neurônios do reservatório, raio espectral, escala de entrada e saída, deslocamento de entrada e saída, e os pesos do reservatório. Esta abordagem consiste no desenvolvimento e implementação de um método para otimização do reservatório de dinâmicas de uma ESN utilizando os parâmetros clássicos para integração com o toolbox de redes com estados de eco que está disponível em ToolboxESN (2015).

1.2.3 Desenvolvimento de uma ferramenta para prognóstico

Desenvolvimento de algoritmos em *Matlab* que permitam testar e comparar algoritmos de treinamento de ESN. Essa ferramenta permitirá realizar um estudo comparativo de diferentes algoritmos, assim como auxiliará em futuras pesquisas em áreas relacionadas ao prognóstico de RUL.

1.3 ESTRUTURA DO DOCUMENTO

Este documento foi estruturado em seis capítulos, e três apêndices. Nos seguintes itens apresenta-se a descrição dos capítulos e apêndices.

Capítulo 1: Este capítulo é a introdução ao trabalho onde se apresenta a hipótese a comprovar, o objetivo geral, os objetivos específicos a serem alcançados, as contribuições a serem desenvolvidas, assim como a estrutura do documento.

Capítulo 2: Apresenta a fundamentação teórica com uma revisão da literatura de prognóstico de falhas, as abordagens de prognóstico, o processo de prognóstico, os algoritmos mais usados em prognósticos e os trabalhos correlatos de prognóstico de RUL baseados em dados históricos, o uso de técnicas de IA, e o algoritmo de otimização ABC.

Capítulo 3: Descreve os fundamentos das redes neurais artificiais, os tipos de RNA, as redes neurais com estado de eco, os componentes de uma ESN, criação do reservatório, treinamento, o estado da arte em otimização de reservatórios, as vantagens e desvantagens das saídas lineares e não-lineares, os tipos de aprendizado (supervisionado, não-supervisionado e incremental), os métodos de aprendizado incremental, e os trabalhos correlatos dos diferentes tipos de redes neurais, em especial das ESN, usados no prognóstico de RUL.

Capítulo 4: Apresenta a abordagem de uma rede com estado de eco para prognóstico de RUL incluindo as contribuições mencionadas no capítulo 1. O algoritmo ABC para otimizar os parâmetros e pesos de uma ESN, a arquitetura para prognóstico de RUL e a modelagem através de Definição Integrada 0 (IDEF0 – *Integrated DEFINITION 0*).

Capítulo 5: Mostra o estudo de caso para a abordagem desenvolvida. São utilizados dados de degradação de turbinas turbofan do repositório de prognóstico da NASA. Os resultados

obtidos são comparados com os resultados obtidos por outros pesquisadores através de métricas quantitativas descritas no Apêndice A.

Capítulo 6: Apresentam-se as conclusões dos resultados desta tese e as sugestões de trabalho futuros sobre o tema apresentado.

Apêndice A: são ilustradas as métricas de prognóstico que podem ser usadas como indicadores para realizar uma comparação quantitativa, descreve as métricas de exatidão, precisão e de prognóstico.

Apêndice B: Apresenta-se a implementação computacional em Matlab desenvolvida para as etapas de aquisição de dados, treinamento e teste clássico de uma ESN, e o treinamento otimizado através do algoritmo ABC de uma ESN. Também são mostradas as telas de resultados produzidos pela implementação.

Apêndice C: Apresentam-se a relação de trabalhos publicados e o trabalho submetido a um *journal* internacional com os resultados da tese.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo apresentam-se os conceitos necessários para a compreensão do processo de prognóstico de RUL. Descrevem-se as três abordagens mais usadas em prognósticos de falhas, os princípios, as aplicações, e as vantagens e desvantagens de cada uma delas, as etapas prévias ao prognóstico, como a aquisição e o processamento de dados. Também é apresentada a fundamentação teórica do algoritmo de otimização ABC. Finalmente, listam-se os principais algoritmos que usam dados históricos para estimar a RUL, dando maior ênfase às redes neurais artificiais.

2.1 PROGNÓSTICO DE FALHAS

A Organização internacional para padronização (ISO – *International Organization for Standardization*) através da ISO 13381 (2004), define o prognóstico como: a “estimação do tempo até a falha e risco para um ou mais modos de falha existentes e futuros”. Numericamente, o conceito mais utilizado em prognóstico é RUL; quando estimada de forma precisa, resulta um fator importante na minimização dos custos de manutenção. Conhecer antecipadamente a RUL pode auxiliar na tomada de decisões sobre ações de manutenção, determinando a sua data de início e a data para a substituição de peças degradadas, diminuindo desta forma as ações de manutenção desnecessárias e as falhas inesperadas.

Os equipamentos ou máquinas e seus componentes em geral passam por um processo de degradação antes de falhar completamente (Lee *et al.*, 2006). A estimação de RUL é realizada monitorando a degradação do equipamento. O processo de degradação pode ser acompanhado por um sistema de prognóstico, com o propósito de monitorar a evolução de um ou mais tipos de falha, e estimar a RUL, uma vez que uma condição de falha seja detectada, isolada e identificada. A estimação da RUL é realizada por três abordagens: baseadas em modelos, em dados e em experiências, além de modelos híbridos que surgem a partir da combinação das abordagens anteriores.

2.2 ABORDAGENS DE PROGNÓSTICO

As pesquisas sobre prognósticos de falhas têm empregado um vasto número de técnicas em diversas áreas do conhecimento, tais como análise de regressão, previsão de séries temporais, inteligência artificial, sistemas especialistas, lógica *fuzzy*, etc. As diferentes técnicas

apresentadas foram classificadas por Vachtsevanos *et al.* (2006) em três categorias, as quais representam três níveis de abordagens de prognóstico: abordagens baseadas em modelos, baseadas em dados, e baseadas em experiências, como mostrado na Figura 2.1. Nas abordagens próximas ao topo da pirâmide observa-se um aumento do custo relativo de desenvolvimento e também um aumento da precisão desejada; em contrapartida há perda de generalidade da abordagem, o que a torna mais específica em sua aplicação. Cada uma dessas abordagens apresenta vantagens e desvantagens, descritas na Tabela 2.1.

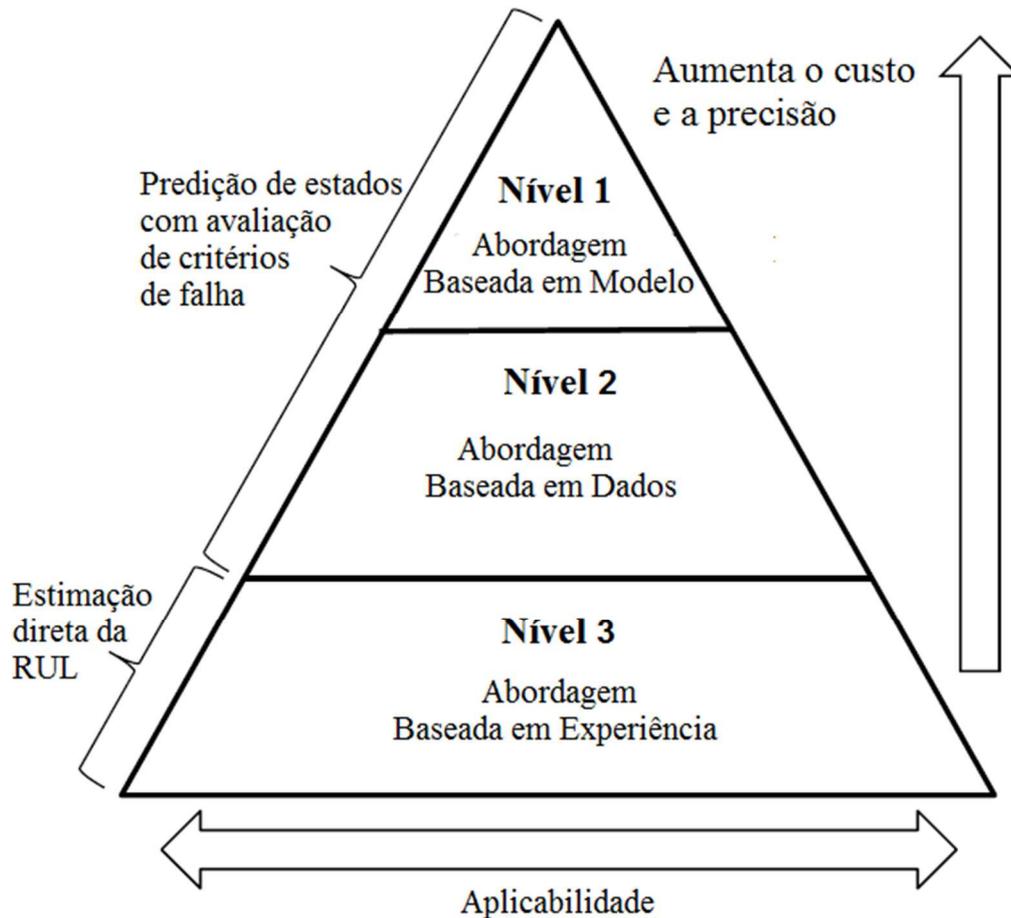


Figura 2.1- Classificação das abordagens de prognóstico de falhas (adaptada de Vachtsevanos *et al.*, 2006).

As abordagens de prognóstico baseadas em modelos e em dados se baseiam na predição de estados e na avaliação de um critério ou limiar de falha; nesses dois casos pode-se considerar duas fases: 1) estimativa do estado de saúde do equipamento (índice de saúde, estado de degradação, etc.); e 2) predição ou extrapolação do estado do equipamento até que atinja um limiar de falha, definido previamente.

As abordagens baseadas em modelos fazem previsões usando os modelos matemáticos do equipamento, planta ou sistema, enquanto que as abordagens baseadas em dados realizam previsões através de modelos treinados com dados adquiridos de MC. Estas abordagens requerem um critério ou limiar de falha conhecidos *a priori*. Caso isso não seja possível, primeiro devem ser estimados a partir dos dados de casos de falhas.

Tabela 2.1- Princípios, vantagens e limitações das abordagens de prognóstico.

Abordagens	Princípio	Vantagem	Limitações
Baseadas em experiências	Exploração das funções de probabilidade ou processos estocásticos de degradação, onde os parâmetros são determinados por especialistas ou por dados de experiência armazenados.	Fácil implementação caso haja experiências suficientes e significativas.	As funções de probabilidade são específicas para cada conjunto de componentes. Resultados de prognóstico de baixa precisão.
Baseada em dados	Baseado na exploração de sintomas e indicadores de degradação. A evolução futura dos sintomas é determinada usando métodos estatísticos ou de IA. Utiliza o monitoramento de dados direto ou indireto mediante de indicadores (observados, medidos ou calculados)	Compensação entre aplicabilidade e precisão	Tempo de aprendizado, e disponibilidade de um sistema de monitoramento de condição.
Baseadas em modelos	Modelo analítico da função de degradação do sistema.	Boa precisão nos resultados de prognóstico.	O modelo de degradação não é facilmente obtido. Dificuldade de aplicação a sistemas complexos.
Híbridas	Resultado da fusão de duas ou mais abordagens descritas anteriormente	Aproveita o melhor das abordagens individuais	Em algumas aplicações não é trivial a fusão de duas ou mais abordagens

A aplicação dessas abordagens é significativa apenas para equipamentos ou sistemas com comportamentos de degradação evolutivos, não se aplicando a equipamentos com falhas abruptas ou aleatórias. Por outro lado, as abordagens baseadas em experiências são aplicadas para estimar a RUL, sendo necessários apenas dados históricos de falhas e distribuições de probabilidade que permitam estimar o Tempo Médio até Falha (MTTF – *Mean Time To Failure*) ou o Tempo Médio entre Falhas (MTBF – *Mean Time Between Failure*).

No desenvolvimento de prognósticos de falhas lida-se com incertezas associadas ao prognóstico de RUL que, muitas vezes, depende de parâmetros relacionados ao modelo, ao seu processamento ou aos dados adquiridos (Goebel *et al.*, 2008a). A escolha da metodologia de prognóstico a ser implementada deve levar em consideração as incertezas, a fim de obter um modelo confiável para prognóstico de RUL. Também devem ser considerados os recursos disponíveis como históricos de dados de MC, históricos de falhas, modelos matemáticos, etc. Além disso, deve-se considerar a criticidade do equipamento, pois cada abordagem requer certo nível de complexidade no tratamento dos dados, informações históricas e modelos, a fim de predizer com exatidão a condição futura de um equipamento (Bizarria, 2009).

2.2.1 Abordagens baseadas em experiências

Este tipo de abordagem de prognóstico de RUL possui um baixo custo de implementação, baixa precisão e grande aplicabilidade como mostrado na Figura 2.1, fácil implementação (Tabela 2.1). Esta abordagem correlaciona o conhecimento especialista e a experiência de engenharia com situações observadas e históricos de medições e falhas para obter uma estimação de RUL. O conhecimento especialista e a experiência de engenharia geralmente estão documentados na forma de regras de produção SE-ENTÃO, coletados diretamente do especialista (Liao e Kottig, 2014). Os dois modelos mais usados, considerados como modelos baseados em experiências, são os sistemas especialistas e a lógica *fuzzy*, devido a que ambos os métodos dependem do conhecimento especialista. Biagetti e Sciubba (2004) apresentaram um sistema especialista que incorpora uma base de conhecimento para detecção de falhas, definindo regras SE-ENTÃO com indicadores para estimativa de falhas, e representação *fuzzy* para diagnóstico de falhas. O prognóstico de RUL foi calculado estimando-se o estado futuro a partir do estado atual e das condições de operação.

Amaya *et al.* (2009) apresentaram um sistema de manutenção baseado em sistemas especialistas usando regras SE-ENTÃO e sua implementação na Linguagem de Sistema Especialista em Java (JESS – *Java Expert System Shell*). As regras de produção foram obtidas através de um processo de engenharia de conhecimento, por meio de entrevistas com as equipes especializadas em manutenção, e também através do uso dos manuais dos equipamentos, dos históricos de falhas, e dos históricos de manutenção. Essa metodologia foi implementada em um sistema computacional chamado SIMPREBAL (Sistema Inteligente de Manutenção Preditiva de Balbina) aplicado a usinas hidrelétricas; os detalhes

e resultados do sistema podem ser encontrados em Amaya e Alvares (2010). Nesse sistema, o prognóstico de falhas é realizado mediante projeção futura do histórico de MTTF.

Em plantas industriais de grande porte é comum encontrar um grande número de equipamentos. Para avaliar a saúde e estimar a RUL dos equipamentos de uma planta desse porte, pode ser necessário implementar centenas ou milhares de regras de produção. Para lidar com esse problema, Amaya e Alvares (2012) apresentaram uma abordagem que torna eficiente o desempenho computacional da máquina de inferência. Essa abordagem, chamada de meta-regras, permitiu obter um melhor desempenho do processamento computacional no modelo baseado em experiências.

De maneira similar aos sistemas de regras de produção, os sistemas de lógica *fuzzy* também usam regras SE-ENTÃO baseadas em dados empíricos para realizar prognósticos. No entanto, as regras são definidas com base em conjuntos de pertinência (ex. SE {o atrito é alto} e {a temperatura sobe rapidamente} ENTÃO {esfriar}). Os modelos de lógica *fuzzy* são mais efetivos quando as variáveis de entrada são contínuas, quando o modelo matemático não está disponível, ou quando os dados de operação possuem alto nível de ruído (Cox, 1992). Assim como os sistemas especialistas, os sistemas *fuzzy* também são baseados em conhecimentos especializados do equipamento que permitam definir as regras *fuzzy*. Nesta abordagem, Zio e Maio (2010) apresentaram um método para prognóstico baseado em similaridade *fuzzy*. Majidian e Saidi (2007) propõem um método para a comparação entre a lógica *fuzzy* e as redes neurais artificiais, aplicado à estimação de RUL de tubos de aquecimento de caldeiras, tendo como entradas as medidas de espessura e o tempo de vida do tubo, obtendo-se resultados muito próximos nas duas abordagens.

2.2.2 Abordagens baseadas em dados

As abordagens baseadas em dados como mostra-se na Figura 2.1 possui um balanço entre aplicabilidade e precisão. Estas abordagens são eficazes quando o modelo matemático que representa a natureza física do equipamento é desnecessário ou difícil de deduzir. Essas abordagens englobam métodos que modelam o comportamento de degradação usando algoritmos de treinamento. A abordagem tradicional de prognóstico baseado em dados, consiste em desenvolver múltiplos candidatos a algoritmos, onde a fase de aprendizado é realizada usando um conjunto de dados de treinamento, e a avaliação dos seus respectivos desempenhos é realizada usando um conjunto de dados de teste, sendo selecionado aquele

com o melhor desempenho (Hu *et al.*, 2011). Inicialmente, devem ser identificados os valores nominais das grandezas medidas do equipamento ou sistema, e monitoradas continuamente durante a operação (Bailey *et al.*, 2010). A predição com esses algoritmos treinados é realizada usando as medições da condição do equipamento. Dessa forma, o algoritmo terá a capacidade de prever mudanças nas grandezas monitoradas do equipamento, comparando-as com as condições nominais de operação. As abordagens baseadas em dados dependem unicamente dos dados atualmente observados e dos dados históricos armazenados; através da aplicação de um algoritmo são capazes de prever os comportamentos futuros.

A estimação da RUL é realizada recursivamente por um algoritmo de prognóstico, depois de detectado um desvio nas condições de operação. Obtém-se como resultado a progressão das características do sistema até a interseção entre os dados extrapolados e um limiar de falha estabelecido *a priori*. A ideia do uso de dados de MC para prever comportamentos futuros de um equipamento foi investigada amplamente na literatura. Ghasemi *et al.* (2010) estimaram a RUL baseados no modelo de risco proporcional para um sistema de múltiplos estados discretos, com transições de estado específicas, para todos os níveis desde o estado saudável até o estado de falha.

Para obter precisão no prognóstico de RUL, é importante contar com modelos confiáveis de evolução de falha e dados históricos de MC, estatisticamente suficientes para treinamento e teste dos algoritmos de prognóstico. A modelagem de correlação é abordada principalmente por métodos probabilísticos, IA, entre outros. Também podem ser encontradas técnicas específicas como predição baseada em redes bayesianas (Ferreiro *et al.*, 2012), *fuzzy*-Kalman adaptativo (Tian *et al.*, 2011), modelo auto-regressivo (Xin *et al.*, 2012), filtragem neuro-*fuzzy* (Li *et al.*, 2013a), Raciocínio Baseado em Casos (RBC) (Berenji, 2006), e RBC distribuído (Pla *et al.*, 2013). Entretanto, ainda existem desafios a superar no desenvolvimento de algoritmos para prognóstico usando técnicas de IA.

Técnicas como o Modelo Oculto de Markov (HMM – *Hidden Markov Models*) foram aplicadas para problemas de prognóstico. Um trabalho recente, que foi apresentado por Pen e Dong (2011), consiste em uma abordagem de prognóstico baseada em HMM utilizada para prever o estado de saúde de bombas hidráulicas. O tradicional HMM foi estendido por um fator de idade para considerar a deterioração do equipamento. Medjaher *et al.* (2012) apresentaram uma aplicação para predição da RUL usando Redes Bayesianas Dinâmicas

(DBN – *Dynamic Bayesian Networks*). A DBN proposta foi usada para avaliar o estado de saúde dos rolamentos a partir das características extraídas dos sensores de vibração no domínio do tempo ou da frequência. Revisões de literatura relativas às abordagens de prognóstico baseada em dados de MC foram publicados por Katipamula e Brambley (2005), Jardine *et al.* (2006), Heng *et al.* (2009), Peng *et al.* (2010), Sikorska *et al.* (2011), Jouin *et al.* (2013) e Lee *et al.* (2014).

2.2.3 Abordagens baseadas em modelos

Estas abordagens assumem que o modelo matemático do processo de degradação do equipamento ou sistema está disponível ou que é factível elaborá-lo; a combinação deste modelo com os dados medidos permite caracterizar quantitativamente o comportamento do sistema usando princípios físicos e matemáticos, com o objetivo de estimar a RUL. Os modelos são baseados em equações matemáticas e filtragem estocástica. Os primeiros são modelos analíticos do processo de degradação do equipamento. Os segundos são *modelos de espaço de estado*, os quais são modelos de processo e observação. Estes modelos consistem de parâmetros que precisam ser identificados. A identificação dos parâmetros de um modelo requer experimentos e dados empíricos. Em prognóstico de falhas, dados de MC são frequentemente usados para identificação e atualização dos parâmetros do modelo.

A principal vantagem das abordagens baseadas em modelos é a sua capacidade para incorporar descrições físicas fundamentais do equipamento monitorado. Portanto, a precisão das abordagens baseadas em modelos é significativamente maior do que a das abordagens baseadas em dados e baseadas em experiências. No entanto, em muitos casos, a obtenção do modelo detalhado do equipamento é complexa, custosa e às vezes impossível, o que limita a sua aplicação. Na literatura são encontradas muitas aplicações dessa abordagem usando várias técnicas em diferentes áreas. Zhang *et al.* (2011) apresentaram uma abordagem para predição da RUL de engrenagens, usando o modelo de crescimento de fissuras. Esse modelo foi testado para fornecer uma distribuição normal da longitude das fissuras. Mediante o uso de uma arquitetura bayesiana atualizaram-se os parâmetros do modelo de degradação. Swanson *et al.* (2000) apresentaram um método baseado no filtro de Kalman para modelar o crescimento de fissuras em correias de aço tensionadas. A RUL foi derivada usando a predição do estado da frequência modal quando ultrapassava o limiar predefinido. Sun *et al.* (2011) apresentaram uma arquitetura de múltiplas escalas para direcionar diferentes escalas

de tempo para estados de carga, a fim de estimar o funcionamento de baterias de íons de lítio.

Não limitado por sistemas lineares ou suposição de ruídos gaussianos, o Filtro de Partículas (PF – *Particle Filtering*) tornou-se popular na predição de RUL. Daigle e Goebel (2010) superaram o problema dos dados limitados de sensores, aplicando prognósticos baseados em modelos, e usando PF com aplicação em válvulas solenoides. Além dessas aplicações, outros métodos foram propostos por diferentes autores que usaram PF para o prognóstico de RUL (Daroogheh *et al.*, 2014; Celaya *et al.*, 2008; Saha *et al.*, 2009a; Saha *et al.*, 2009b; Orchard *et al.*, 2005; Orchard e Vachtsevanos, 2009). Uma abordagem que combina o filtro de partículas, o filtro exato e o filtro múltiplo é a apresentada por Liang *et al.* (2010). Devido à ampla aplicação de PF, Bin *et al.* (2010) propuseram algoritmos de verificação e validação baseados em filtragem exata e método de Monte Carlo para algoritmos de prognóstico baseada em PF. Com o intuito de melhorar o tempo de processamento, Compare e Zio (2014) apresentaram uma nova abordagem de PF chamado de Filtro de Partículas Sensível ao Risco (SRPF – *Risk Sensitive Particle Filtering*). Na literatura é comum encontrar aplicações para prognóstico de carga de baterias de íons de lítio (Weiming *et al.*, 2014; Yinjiao *et al.*, 2012; Chaochao e Pecht, 2012; Olivares *et al.*, 2013). Outras variantes como combinações com outras técnicas assim como PF e neuro-*fuzzy* são apresentadas por Chaochao *et al.* (2011).

2.2.4 Abordagens híbridas

As abordagens híbridas de prognóstico para predição de RUL são resultado da fusão de duas ou mais abordagens descritas anteriormente, gerando diferentes combinações das três abordagens. Liao e Kottig (2014) apresentaram cinco tipos de abordagens híbridas apresentadas na Figura 2.2, onde H1, H2, H3, H4 e H5 são abordagens híbridas como resultado da combinação de duas ou mais abordagens.

A aplicação de abordagens híbridas é encontrada em publicações, por exemplo, Kumar *et al.* (2008) apresentaram uma abordagem híbrida de prognóstico baseada na combinação das abordagens baseadas em dados e em modelos, para diagnóstico e prognóstico de saúde de componentes eletrônicos. As vantagens e desvantagens das abordagens baseadas em dados e em modelos são apresentadas por Huiguo *et al.* (2009).

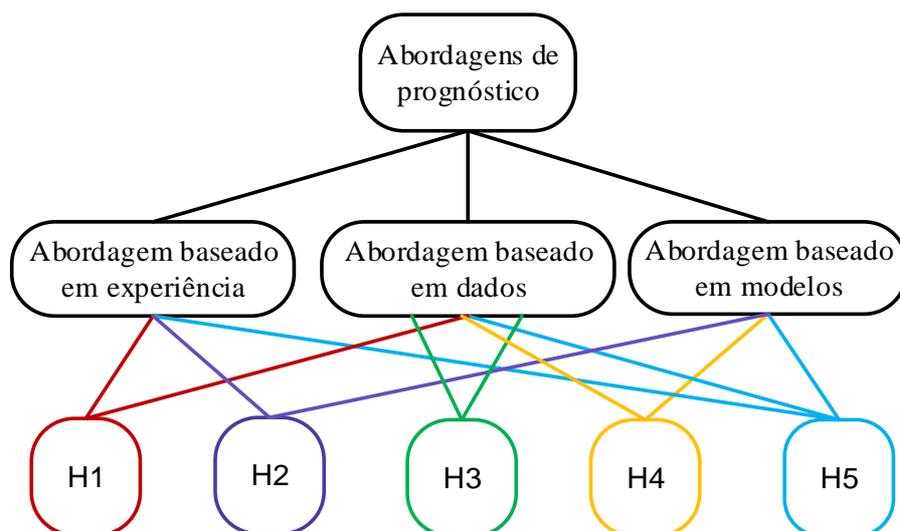


Figura 2.2- Abordagens híbridas de prognóstico (Liao e Kottig, 2014).

2.3 VIDA ÚTIL REMANESCENTE

A RUL é definida como o tempo remanescente, em termos de horas de operação, ciclos, etc., que o equipamento ou sistema continuará operando dentro de suas especificações de projeto; também é conhecida como Tempo Estimado até a Falha (ETTF – *Estimated Time To Failure*) (Vachtsevanos *et al.*, 2006; ISO 13381, 2004). A RUL refere-se também ao tempo restante (desde a aparição de um sintoma) até que um determinado limiar de falha seja atingido, como mostrado na Figura 2.3. Sikorska *et al.* (2011) definem o prognóstico de falha como a estimativa da RUL de um equipamento. Para Cheng *et al.* (2010), o prognóstico de falha consiste de métodos, algoritmos e tecnologias que permitem avaliar a confiabilidade de um equipamento no seu estado atual e determinar as possíveis falhas futuras. A RUL é uma medida que possui incertezas e indica o processo de degradação da saúde de um equipamento, e é estimada baseada em análises de observações passadas (Engel *et al.*, 2000).

A precisão na estimativa de RUL é fundamental para a otimização dos custos de manutenção. As medidas de prognóstico podem auxiliar nas decisões de manutenção, na determinação do tempo de início de uma manutenção, e na estimativa do tempo para a substituição de peças ou equipamentos degradados, evitando dessa forma manutenções desnecessárias e falhas inesperadas. Nos últimos anos houve um interesse significativo em propostas de metodologias para prognóstico de RUL de equipamentos ou sistemas utilizando diferentes abordagens e algoritmos. Com respeito à revisão de literatura de pesquisas recentes nessa área, as vantagens e as desvantagens dos vários modelos disponíveis podem ser encontradas

em Si *et al.* (2011). Os pontos fortes e fracos e a aplicabilidade dos principais modelos de prognóstico foram apresentados por Sikorska *et al.* (2011).

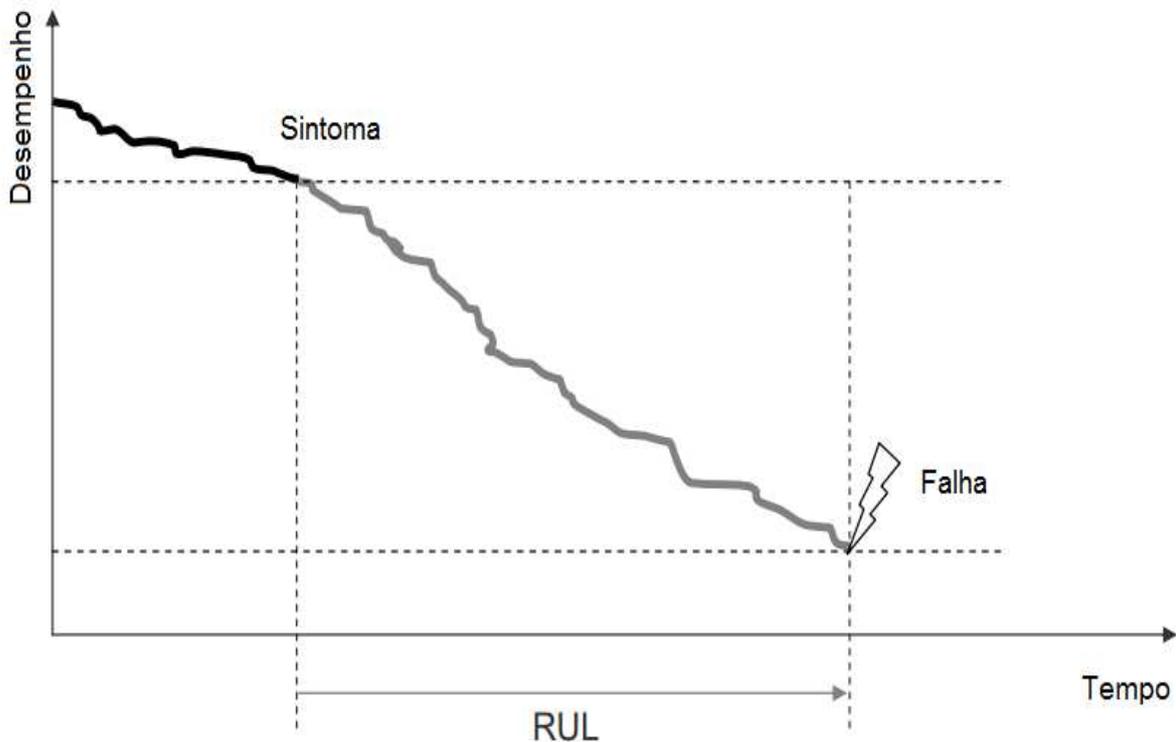


Figura 2.3- Definição da vida útil remanescente.

2.4 AQUISIÇÃO E PROCESSAMENTO DE DADOS

Devido ao fato que o processo de predição inclui incertezas, resulta um grande desafio no desenvolvimento de sistemas de prognóstico (Greitzer e Pawlowski, 2002). Essas incertezas, na maioria das vezes, estão presentes na aquisição de dados. Os fatores que podem gerá-las, na aquisição de dados, são ruídos, distúrbios, degradação do instrumento de medida, e erros humanos (Zedda e Singh, 2002). A precisão dos resultados de um sistema de prognóstico é altamente dependente da precisão dos dados adquiridos, sendo um dos motivos pelos quais nos últimos anos as técnicas de MC receberam destacada atenção, levando à incorporação de sensores inteligentes, e à inclusão do processamento de sinais e extração de características. Dois fatores fundamentais para obter um prognóstico preciso são: a aquisição e o processamento de dados relevantes, que inclui a extração e a seleção de características. Das *et al.* (2011) apresentam as etapas essenciais para o desenvolvimento de um efetivo sistema PHM, descrevendo as características extraídas no domínio do tempo e da frequência.

2.4.1 Aquisição de dados

A aquisição de dados é a primeira fase de um processo de prognóstico. Essa etapa consiste em selecionar as medições mais relevantes para a monitoração da saúde de um equipamento ou sistema. Os sensores e as estratégias de sensoriamento são fundamentais para o prognóstico. O processo de prognóstico está associado ao tipo, número e localização dos sensores, seu uso, peso, custo, faixa dinâmica entre outras características, sendo que eles podem ter ligação com fio e sem fio (Vachtsenavos *et al.*, 2006). Os dados coleados pelos dispositivos transdutores raramente são usados na sua forma bruta. Esses dados devem ser processados apropriadamente para extrair informações úteis, resultando em uma versão reduzida dos dados originais, mas que preservam a maior quantidade de atributos destas características ou indicadores de falha, que também são indicadores de eventos de falhas e que procuram detectar, isolar e prever a evolução ao longo do tempo. O processamento de dados consiste em realizar filtragem, compressão e correlação para remover impurezas e reduzir níveis de ruídos e volume de dados a serem processados.

A maioria das grandes empresas dispõem de grandes históricos de dados, os quais não são efetivamente usados. Esses dados podem ser utilizados para prever e identificar os defeitos de equipamentos antes da ocorrência de uma falha. A aquisição desses dados de MC é realizada a partir de sensores instalados nos equipamentos. Esses sensores podem medir temperatura, umidade, vazão, pressão, etc. Outros tipos de dados, em muitos casos disponíveis, tais como dados de históricos de falhas e informações do fabricante podem ajudar na modelagem de prognóstico de falhas; esses dados serão chamados de dados de eventos.

2.4.2 Processamento de dados

Geralmente, os dados coletados possuem sinais brutos contendo ruídos e sinais irrelevantes. A eliminação desses ruídos e sinais é necessária para obter um sistema de prognóstico confiável. Essa etapa inclui a remoção de picos, a normalização de dados, a remoção de ruídos e sinais irrelevantes. Uma análise sem um processamento adequado de sinais pode levar a falsos alarmes. Depois da obtenção de sinais relevantes e sem presença de ruído, são necessárias outras duas etapas: a extração e seleção de características.

2.4.2.1 Extração de características

As características são as propriedades heurísticas individuais medíveis do fenômeno em observação, sendo usualmente valores numéricos, como por exemplo, a média, a variância e o valor máximo de uma série de sinais (Theodoridis e Koutroumbas, 2006). São informações que ajudam a entender o estado de saúde de um equipamento (Theodoridis e Koutroumbas, 2006), e que podem ser extraídas usando técnicas no domínio do tempo, frequência e tempo-frequência.

As técnicas no domínio do tempo são utilizadas para sinais não periódicos ou quando a periodicidade de um sinal não é significativa. Entre as características presentes no domínio do tempo estão a média, a variância, o mínimo, o máximo, os coeficientes polinomiais do sinal, etc. (Kim *et al.*, 2007; Sreejith *et al.*, 2008; Zhang e Randall, 2009).

No que diz respeito aos sinais periódicos, a característica pode ser extraída usando técnicas no domínio da frequência, como por exemplo a Transformada Rápida de Fourier (FFT – *Fast Fourier Transform*). A amplitude de uma frequência pode ser usada como uma característica (Theodoridis e Koutroumbas, 2006). Entretanto, a transformada de Fourier é indicada apenas para a transformação de um sinal estacionário. Para sinais não estacionários podem ser usadas a Transformada de Wavelet (Daubechies, 1990) ou a Transformada de Fourier de Tempo Curto (STFT – *Short Time Fourier Transform*) (Zhu *et al.*, 2007).

2.4.2.2 Seleção de características

Em primeira análise, extrair a maior quantidade de características é sempre melhor, já que mais características podem fornecer mais informações. Entretanto, a presença de características irrelevantes e redundantes complica o modelo de prognóstico e aumenta o custo computacional, e um fator ainda mais importante, a presença de um grande número de características, diminui a capacidade de generalização do modelo de prognóstico.

Na Figura 2.4 indica-se que o desempenho nem sempre melhora com o aumento do número de características (Trunk, 1979). Neste cenário, o aumento do número de características pode melhorar apenas o desempenho inicial, depois de um número crítico de características, o desempenho diminui: a isto denomina-se *fenômeno de pico* (Theodoridis e Koutroumbas, 2006). É possível observar também que somente para um conjunto de dados infinito ou suficientemente grande, o aumento do número de características pode melhorar o modelo de

prognóstico. Contudo, a geração de um conjunto de dados suficientemente grande não é possível na maioria dos casos.

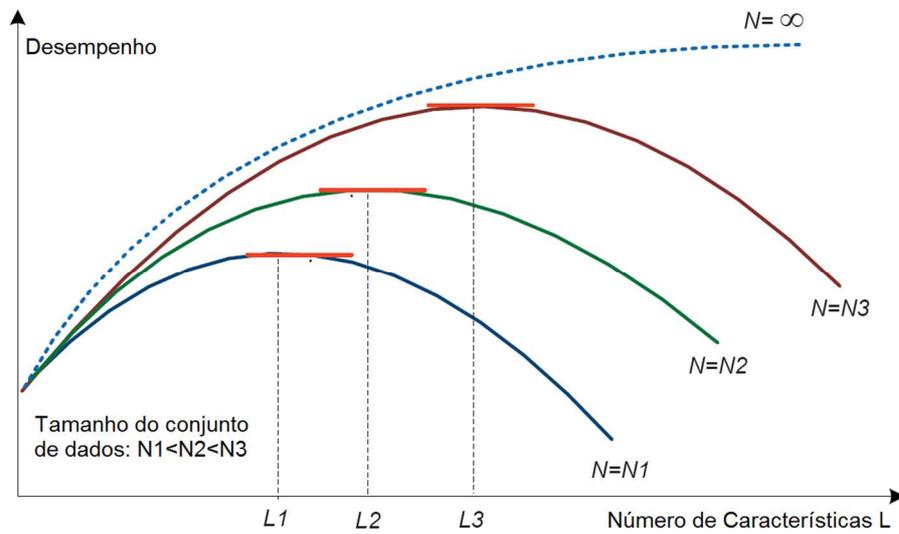


Figura 2.4- Fenômeno de pico (Theodoridis e Koutroumbas, 2006).

2.5 ALGORITMOS PARA PROGNÓSTICO

Nesta seção será apresentada uma revisão dos principais algoritmos para prognóstico de RUL, usados nas abordagens baseadas em dados, foco deste trabalho. Os algoritmos para estas abordagens foram classificados por Huiguo *et al.* (2009) e apresentados na Figura 2.5. Basicamente dividem-se em dois grandes grupos: aprendizagem de máquina e métodos estatísticos.

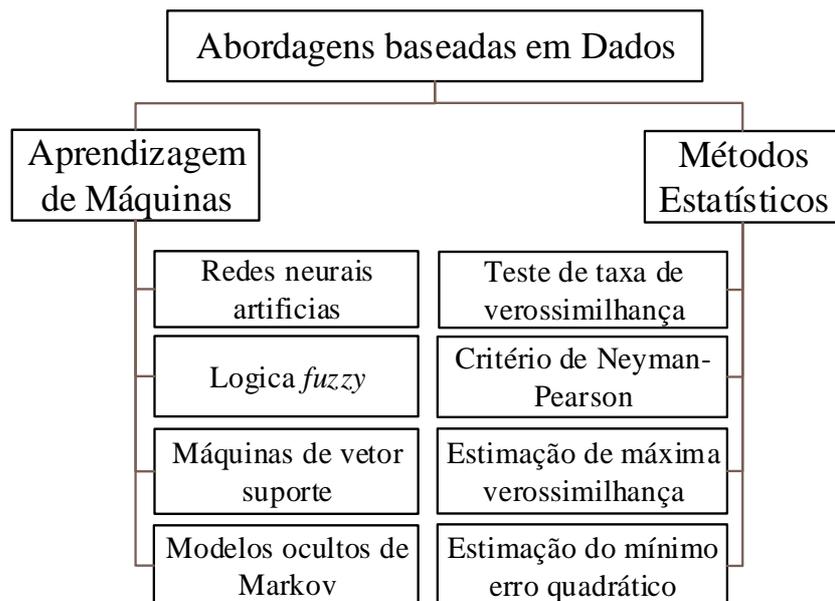


Figura 2.5- Algoritmos usados nas abordagens baseadas em dados (Huiguo *et al.*, 2009).

A aprendizagem de máquina é muito flexível e pode se adaptar facilmente aos câmbios, que podem ocorrer na sua estrutura interna ou nas condições de operação. Outra vantagem do aprendizado de máquina é o fato de serem apropriadas para todos os níveis, desde os componentes, equipamentos, subsistemas, até o sistema em si. No entanto, os dados de treinamento são uma parte essencial de um aprendizado de máquina que precisam passar por uma etapa de pré-processamento apropriada. Também precisam de algoritmos de aprendizado eficientes, rápidos e de atualização em tempo real.

A abordagem estatística é relativamente simples, e economicamente mais factível a sua implementação, embora sejam necessários grandes recursos computacionais. No entanto, os algoritmos estatísticos não consideram o ambiente de uso, as condições de operação e os mecanismos de falhas atuais, além disso, precisam de grandes quantidades de dados para sua implementação.

Dos algoritmos de IA encontrados na literatura, verificou-se que é necessário desenvolver algoritmos específicos para aplicações em situações específicas. A eficácia desses algoritmos depende do tipo e da qualidade dos dados disponíveis, e das suposições inerentes ao algoritmo. Um critério importante para a seleção do algoritmo apropriado depende da informação disponível no limiar da falha: se os dados do indicador de condição são objetivos ou subjetivos, se estão ou não disponíveis, e dependem também da disponibilidade dos dados de tempo até a falha (TTF – *Time To Failure*).

Como se observa na Tabela 2.2, o prognóstico de falhas é abordado por várias técnicas, desde estimações Bayesianas e outros métodos estatísticos e probabilísticos até ferramentas de inteligência artificial. Essas tecnologias incluem o filtro *Kalman* (Carr e Wang, 2011; Andre *et al.*, 2013), modelos auto-regressivos de média móvel (Long *et al.*, 2013), modelos *Weibull* (Groer, 2000; Zhang *et al.*, 2014), e métodos de estimação de parâmetros (Moghaddass e Zuo, 2012). Foram encontradas também metodologias baseadas na inteligência artificial, raciocínio baseado em casos (Zhuang *et al.*, 2009), modelos inteligentes baseados em decisões, assim como grafos *min-max* que foram considerados como métodos potenciais para algoritmos de prognóstico. Outros métodos como redes de *Petri*, RNA, sistemas de lógica nebulosa (*fuzzy*) e sistemas híbridos como *neuro-fuzzy* (Wang *et al.*, 2004; Chen *et al.*, 2012) apresentam grande utilidade como ferramentas de prognóstico.

Tabela 2.2- Principais algoritmos para prognóstico (Adaptado de Lee *et al.*, 2014).

Algoritmo	Descrição	Vantagens	Desvantagens
Filtro de partículas (Daroogheh <i>et al.</i> , 2014; Weiming <i>et al.</i> , 2014; Yinjiao <i>et al.</i> , 2012; Chaochao e Pecht, 2012)	-Abordagem bayesiana para obter a estimação de estado que represente a função distribuição de probabilidade de uma trajetória de estado definido por amostras de partículas recursivas.	-Aplicável em sistemas não lineares e ruído não gaussiano; -Alta precisão comparada com outros algoritmos de filtragem; -A amostragem de importância sequencial ajuda a incrementar a precisão e evitar a degeneração.	-Os modelos da dinâmica do sistema e de medição precisam ser definidos; -Maior custo computacional para sistemas de alta dimensão ou com mais partículas.
Filtro de Kalman (Baraldi <i>et al.</i> , 2012; Saikiran <i>et al.</i> , 2013; Daigle <i>et al.</i> , 2012; Laslett <i>et al.</i> , 2014)	-Técnica bayesiana que estima o estado de um processo e minimiza a covariância da estimação incorporando medidas relativas ao estado.	-Capaz de estimar o estado atual e prever os estados futuros; -Corrige a estimação com a última medida para manter a covariância mínima do erro de estado.	-Os modelos do sistema e de medida precisam ser definidos; -Níveis de ruídos em ambos os modelos podem afetar o desempenho e estabilidade do algoritmo; -Trabalha unicamente com sistemas lineares e ruído gaussiano.
Mapas auto organizáveis (Hai e Lee, 2004; Lall <i>et al.</i> , 2010; Xiaochuang <i>et al.</i> , 2012)	-Representar uma característica do espaço multidimensional em um espaço de baixa dimensão preservando as propriedades da topologia do espaço de entrada.	-Método de aprendizado não supervisionado; -Boa capacidade de visualização.	-Falta de um algoritmo padrão para determinar a estrutura e a forma do mapa.
Redes bayesianas (Murphy, 2002; Mosallam <i>et al.</i> , 2013; Lin <i>et al.</i> , 2013)	-Ferramenta gráfica acíclica para representar a estrutura da relação de interdependência condicional e distribuições de probabilidade entre variáveis de um sistema.	-Reduz o número de parâmetros para apreender a estrutura de domínio através da marginalização das distribuições de probabilidade condicional; -Visualiza a dependência entre cada par de variáveis.	-O aprendizado de uma estrutura desconhecida pode ser complexo; -É necessária certa quantidade de conhecimento a priori do domínio.
Redes neurais artificiais (Xiangjun e Tongmin, 2012; Ghavami e Kapur, 2012; Zhou <i>et al.</i> , 2012; Morando <i>et al.</i> , 2013)	-Modelar e simular funções e estruturas de redes neurais biológicas; -Aprendizagem de conhecimento mediante modelagem das relações complexas entre entradas e saídas busca de padrões nos dados.	-Para sistemas complexos onde envolve comportamento não linear e processo instável; -Sistema adaptativo.	-Inexistência de métodos padrões para determinar a estrutura da RNA; -Requer altos recursos computacionais.
Auto-Regressivo de Média Móvel (ARMA- <i>Autoregressive-Moving Average</i>) (Pandit e Wu, 1983)	-Consiste de duas partes, parte AR auto regressivo e parte MA média móvel, para modelagem e predição de valores futuros em dados de séries de tempo.	-Aplicável a sistemas lineares e invariantes no tempo com comportamento estacionário; -Requer pequena quantidade de dados históricos.	-Não fornece boa predição em grandes espaços de tempo; -Limitado para processos dinâmicos e não estacionários.

Lógica <i>fuzzy</i> (Ross, 2004; Chang-Yu <i>et al.</i> , 2012)	-Representar e processar incertezas de modo a fazer um sistema complexo administrável; -Tolera incertezas e pode utilizar linguagem vago para oferecer robustez, modelo tolerante falhas ou predição quando não estão disponíveis entradas precisas.	-Pode lidar com dados incompletos, ruidosos ou imprecisos; -Útil no desenvolvimento de modelos de dados incertos; -Mais compatível como o processo de raciocínio humano do que a abordagem simbólica tradicional. -Apropriado para sistemas complexos e/ou desconhecidos.	-Impraticável quando funções de pertinência são complicadas de determinar.
Máquina vetor suporte (Sloukia <i>et al.</i> 2013, Soualhi <i>et al.</i> , 2014)	-Para projetar o espaço de características em um espaço de alta dimensão por uma função kernel; -Para encontrar um hiperplano de separação otimizada no espaço projetado para maximizar a fronteira de decisão.	-Obtem melhor precisão em decisões devido à sua fronteira de decisão maximizada; -Eficiente para um grande conjunto de dados e análise em tempo real.	-Não possui um método padrão para escolha da função kernel, o qual é um processo chave para SVM.
Modelo oculto de Markov (Tobon-Mejia <i>et al.</i> , 2011; Geramifard <i>et al.</i> , 2012; Li <i>et al.</i> , 2013a)	-Modelo estatístico, onde o modelo do sistema é assumido como de Markov com parâmetros de espaço estado desconhecidos.	-Pode ser usado para diagnóstico de degradação e falha em sinais não estacionários e sistemas dinâmicos; - Apropriado para múltiplos modos de falha.	-Inapropriado quando os estados de falha são observáveis; -Demanda grande quantidade de dados para uma modelagem precisa.

Um grande número de abordagens existentes baseadas em dados usa Redes Neurais Artificiais (RNA) para modelar um sistema de prognóstico (Parker *et al.*, 1993; Wang e Vachtsevanos, 2001; Shimanek, 2003; Brotherton *et al.*, 2000). As RNA são um tipo de modelo inspirado na estrutura neural do cérebro, onde o processamento de informação é realizado por elementos interconectados chamados de neurônios. Os pesos das conexões entre os neurônios são ajustados para maximizar a relação do modelo como os dados de treinamento (Bishop, 1995).

Desde a apresentação do modelo de neurônio por McCulloch e Pitts (1943) as RNA foram continuamente evoluindo. Nas últimas décadas com o aumento da disponibilidade e da capacidade de processamento dos computadores, as RNA emergiram como uma ferramenta poderosa para lidar com problemas de classificação e regressão, usados em diferentes aplicações. A popularidade das RNA resulta da sua habilidade em modelar as não linearidades complexas entre as entradas e saídas de um conjunto de dados de treinamento. Diferentes tipos de RNA foram desenvolvidas, entre elas encontram-se as redes Perceptron de Múltiplas Camadas (PMC), Redes Neurais de Função de Base Radial (RBF – *Radial*

Basis Function), os mapas auto organizáveis (SOM – *Self Organized Maps*) de Kohonen, as Redes de Hopfield, e etc.

Os desenvolvimentos atuais das RNA emergiram como novas ferramentas para lidar com os problemas de diagnóstico e prognóstico existentes. Neste sentido surgiram as primeiras aplicações na área da medicina (Liang *et al.*, 1988). Uma aplicação de redes neurais polinomiais para a detecção, isolamento e estimação de falhas foi apresentada por Parker *et al.* (1993). O método foi aplicado no prognóstico de falhas de transmissão de helicópteros. Berenji e Yan (2006) apresentaram uma Rede Neural Wavelet (WNN – *Wavelet Neural Network*) e uma Rede Neural Wavelet Dinâmica (DWNN – *Dynamic Wavelet Neural Network*) para diagnóstico e prognóstico aplicados ao sistema de resfriamento *chiller*, usando Matlab/Simulink, obtendo um bom desempenho na reconstrução de sinais e detecção de falhas. Para a etapa de treinamento de uma WNN, Lei *et al.* (2007) propõem um método que utiliza Algoritmos Genéticos (GA – *Genetic Algorithms*).

Ke-Xu *et al.* (2011) apresentaram uma RNA otimizada por um enxame de partículas para prognóstico de aeronaves analisando os tipos e características de falha para obter um limiar de falha. Malhi *et al.* (2011) apresentaram uma abordagem baseada no aprendizado competitivo para prognóstico do estado de máquinas de longo prazo, usando sinais de vibração, e observando que a técnica desenvolvida é mais precisa do que a técnica de treinamento incremental para previsão da progressão de defeitos em rolamentos. Heimes (2008) propôs uma arquitetura para estimar a RUL baseada em uma RNR, treinada por retropropagação, filtro de Kalman estendido e algoritmos evolutivos para gerar um algoritmo compacto. Zhigang e Zou (2009) propuseram a aplicação de uma RNR para previsão da condição de saúde de uma caixa de engrenagens.

2.6 ALGORITMO DE OTIMIZAÇÃO ABC

O comportamento no processo de coleta de alimentos, atributos de aprendizagem, memorização e compartilhamento de informações são características destacáveis na inteligência de enxames. As colônias de abelhas estão entre os algoritmos bioinspirados mais estudados devido à sua alta capacidade organizacional. O algoritmo ABC proposto inicialmente por Karaboga (2005), foi inspirado no comportamento social de colônias de abelhas durante a coleta de alimentos. Cada agente (abelha) registra a posição geográfica, cor, dança, forma e odor de uma florada. Por outro lado, o conjunto de abelhas (enxame)

desenvolve uma inteligência coletiva, permitindo a tomada de decisões mediante o intercâmbio de informação usando danças, substâncias químicas, sons e estímulos eletromagnéticos (Tereshko e Loengarov, 2005; Serapião, 2009).

Os modelos matemáticos desenvolvidos a partir das observações e estudos do comportamento de colônia de abelhas usam três elementos: 1) fontes de alimento, 2) abelhas operárias, e 3) abelhas não-operárias (Cazamine e Sneyd, 1991; Tereshko, 2000). A escolha de uma fonte de alimento depende da proximidade à colmeia, qualidade e concentração de néctar. As fontes de alimento estão associadas às abelhas operárias, e estas transportam informações sobre a qualidade, a distância e a orientação das posições do néctar. Após a coleta de néctar a abelha operária volta à colmeia e compartilha a informação sobre a fonte de alimento por meio de uma dança. A vivacidade e orientação da dança representam a distância até a fonte de alimento e a orientação em relação ao sol, respectivamente.

Existem de dois tipos de abelhas não operárias, as escoteiras e as seguidoras. A função das escoteiras é percorrer o ambiente aleatoriamente em busca de novas fontes de alimento. Depois de esgotada uma fonte de alimento, uma abelha operária vira escoteira. As seguidoras têm como missão: a) observar a dança de mais de uma operária no interior de colmeia e escolher com determinada probabilidade, uma fonte de alimento considerando a qualidade do néctar e a proximidade à colmeia, e b) tornar-se escoteira e explorar aleatoriamente ao redor da colmeia. Por outro lado, as operárias uma vez dentro da colmeia podem: a) dançar e recrutar seguidoras, b) continuar explorando a fonte de alimento, c) abandonar a fonte de alimento e se tornar uma seguidora. A formação de conhecimento coletivo é devido ao intercâmbio de informação entre as abelhas.

No algoritmo ABC, uma fonte de alimento representa uma possível solução ao problema de otimização, e a qualidade da fonte de alimento denotado por uma quantidade numérica, em geral representa a função de aptidão ou de custo. O número de abelhas operárias é igual ao número de fontes de alimentos (soluções) uma vez que cada abelha operária está associada a uma única fonte de alimento. O diagrama de fluxo do algoritmo ABC é mostrado na Figura 2.6, e consiste de quatro fases que serão descritas a seguir.

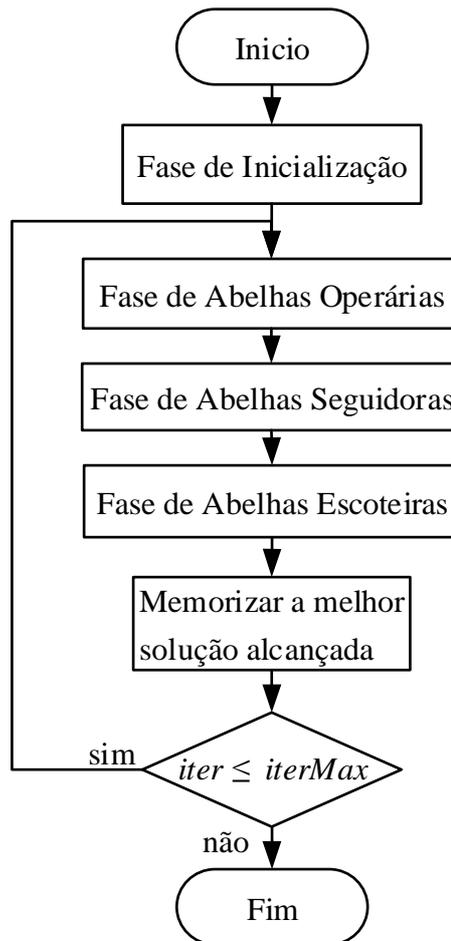


Figura 2.6- Algoritmo ABC.

2.6.1 Fase de Inicialização

Inicialização por abelhas escoteiras de todos os vetores da população de fontes alimentos (\vec{x}_m), onde ($m = 1 \dots SN$, SN : tamanho da população). Também são definidos os parâmetros de controle. Cada fonte de alimento (\vec{x}_m) é um vetor solução para o problema de otimização, cada vetor (\vec{x}_m) possui n componentes, (x_{mi} , $i = 1, 2, \dots, n$), que devem ser otimizados de forma a minimizar a função de aptidão. A Equação 2.1 pode ser utilizada para fins de inicialização.

$$x_{mi} = l_i + (u_i - l_i) * rand(0,1) \quad (2.1)$$

Onde: l_i e u_i é o limite inferior e superior do parâmetro x_{mi} , respectivamente, e $rand(0,1)$ é um valor aleatório entre 0 e 1.

2.6.2 Fase de Abelhas Operárias

Abelhas operárias têm como missão procurar novas fontes de alimento (\vec{v}_m) com mais néctar dentro da vizinhança da fonte de alimento (\vec{x}_m), descrito pela Equação 2.2. Ao encontrar uma próxima fonte de alimento, imediatamente é avaliada a sua qualidade ($fitness$).

$$v_{mi} = x_{mi} + \phi_{mi}(x_{mi} - x_{ki}) \quad (2.2)$$

Onde: \vec{x}_k é uma fonte de alimento selecionada aleatoriamente, i é um índice escolhido aleatoriamente e ϕ_{mi} é um número aleatório no intervalo $[-a, a]$. Depois de produzir a nova fonte de alimento (\vec{v}_m), sua aptidão é calculada usando a Equação 2.3 para problemas de minimização.

$$fit_m(\vec{x}_m) = \begin{cases} \frac{1}{1 + f_m(\vec{x}_m)} & \text{Se: } f_m(\vec{x}_m) \geq 0 \\ 1 + |f_m(\vec{x}_m)| & \text{Se: } f_m(\vec{x}_m) < 0 \end{cases} \quad (2.3)$$

Onde: $f_m(\vec{x}_m)$ é o valor da função objetivo da solução \vec{x}_m .

2.6.3 Fase de Abelhas Seguidoras

No caso das abelhas seguidoras tem-se dois grupos: Abelhas seguidoras e escoteiras. As abelhas operárias compartilham suas informações sobre a fonte de alimento com as abelhas seguidoras que esperam na colmeia, as abelhas seguidoras probabilisticamente escolhem suas fontes de alimento, dependendo desta informação. No algoritmo ABC, uma abelha seguidora escolhe uma fonte de alimento, dependendo dos valores de probabilidade calculados usando os valores de aptidão fornecida pelas abelhas operárias. Para esta finalidade, uma técnica de seleção baseada na aptidão pode ser utilizada. O valor da probabilidade p_m com a qual \vec{x}_m é escolhido por uma abelha seguidora pode ser calculada utilizando a Equação 2.4.

$$p_m = \frac{fit_m(\vec{x}_m)}{\sum_{m=1}^{SN} fit_m(\vec{x}_m)} \quad (2.4)$$

Depois de escolher probabilisticamente uma fonte de alimento (\vec{x}_m) por uma abelha seguidora, uma vizinhança (\vec{v}_m) é determinada por meio da Equação 2.2, e seu valor de aptidão é calculado. Assim, mais seguidoras são recrutadas para as fontes mais ricas e aparece o comportamento de realimentação positiva.

2.6.4 Fase de Abelhas Escoteiras

As abelhas seguidoras que escolhem as fontes de alimentos de forma aleatória são chamadas de escoteiras. Abelhas operárias cujas soluções não podem ser melhoradas através de um predeterminado número de ciclos, especificado do pelo usuário do algoritmo ABC e chamado de "limite" ou "critérios de abandono", tornam-se escoteiras e suas soluções são abandonados. Em seguida, as operárias convertidas em escoteiras começam a procurar novas soluções, de forma aleatória. Por exemplo, se a solução (\vec{x}_m) foi abandonada, a nova solução

descoberta pela abelha escoteira que era a abelha operária de (\vec{x}_m) pode ser definido pela Equação 2.1. Essas fontes de alimento que são inicialmente pobres ou terem sido empobrecidas por exploração são abandonadas e o comportamento de realimentação negativa surge para equilibrar a realimentação positiva.

2.7 SÍNTESE DO CAPÍTULO

Neste capítulo faz-se uma introdução aos conceitos necessários para o desenvolvimento de prognósticos que permitam estimar a RUL. Os principais conceitos incluem a aquisição e o processamento de dados, as três abordagens principais, e algoritmos para as abordagens de prognóstico baseadas em dados de MC. Das abordagens de prognóstico apresentadas, as abordagens baseadas em dados são as mais factíveis de implementar, considerando que atualmente muitas empresas usam controle digital com sistemas de supervisão e armazenamento de histórico. Essas abordagens precisam apenas de dados históricos, que podem ser grandezas (dados de MC) ou eventos (falhas). Esses dados devem passar por um processo de extração e seleção de características, etapas importantes para o prognóstico de falhas. Dentre a ampla gama de algoritmos utilizados para o prognóstico de falhas, chega-se à conclusão de que as RNA são um dos algoritmos com maior potencial, especialmente quando um modelo físico preciso ou um modelo matemático não está disponível, mas uma grande quantidade de dados está.

3 REVISÃO DE LITERATURA: REDES COM ESTADOS DE ECO

Neste capítulo apresenta-se o fundamento teórico das Redes Neurais Artificiais, os tipos, as vantagens e desvantagens principais das redes neurais progressivas e redes neurais recorrentes. Logo é realizada uma revisão das redes com estados de eco, da sua arquitetura, das camadas, dos algoritmos e dos tipos de aprendizado. Também são apresentados os desafios a desenvolver nas redes com estados de eco, em especial na otimização do reservatório de dinâmicas. Finalmente, apresenta-se a aplicação dos tipos de RNA no prognóstico de vida útil remanescente, e os trabalhos correlatos que usam a redes com estados de eco para prognóstico de falhas.

3.1 INTRODUÇÃO

O desenvolvimento das Redes Neurais Artificiais começou a partir das publicações iniciais mais relevantes, desenvolvidas por McCulloch e Pitts (1943) que introduziram o primeiro modelo algébrico de um neurônio artificial; Hebb (1949) no livro “*The Organization of Behavior*” definiu o conceito de atualização de pesos sinápticos; e Roseblatt (1958) implementou o primeiro modelo de neurônio artificial denominado *perceptron* além de uma metodologia de treinamento supervisionado inspirado no trabalho de Hebb (1949). Minsky e Papert (1969), no livro “*Perceptrons: an Introduction to Computational Geometry*”, mostraram que com um perceptron de camada única não é possível representar problemas não linearmente separáveis, colocando como exemplo o operador XOR.

Depois de mais de uma década com poucas contribuições na área de RNA, a década de 1980 foi apontada como o reinício das RNA, mediante a publicação dos trabalhos de Hopfield (1982) sobre a utilização de redes simétricas para otimização, e os trabalhos de Rumelhart, Hinton e Williams (Rumelhart *et al.*, 1986) que introduziram o método *Backpropagation*. A partir dessas e outras publicações, as RNA recobram interesse devido à sua eficiência na modelagem de problemas complexos, podendo ser usadas em problemas de classificação, onde a saída é uma variável categórica; ou para regressão, onde a saída é uma variável contínua.

3.2 NEURÔNIO ARTIFICIAL

O neurônio artificial exibido na Figura 3.1 é a unidade básica de processamento de sinais de uma RNA. Essa unidade recebe os sinais provenientes da camada de entrada ou de outros

neurônios, e realiza uma transformação para obter um sinal na camada de saída. O processo que ocorre entre a saída de um e a entrada de outro neurônio artificial é denominado de *sinapse*. Cada sinapse está associada a um peso que armazena informação correspondente ao neurônio.

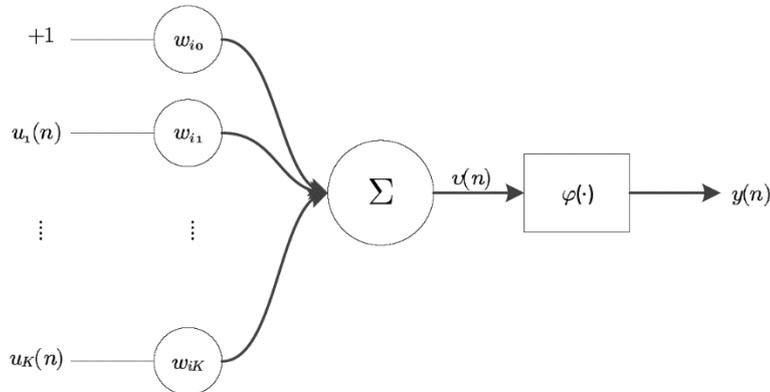


Figura 3.1- Modelo de um neurônio artificial (Bocato, 2013).

A Equação 3.1 exemplifica o cálculo para a saída $y(n)$ de um neurônio artificial. Esta é obtida através da soma ponderada entre os sinais de entrada $u_k(n)$, $k = 1, 2, \dots, K$, junto com os pesos sinápticos w_{ik} e as viés de valor +1. Este resultado é avaliado por uma função de ativação $\varphi(\cdot)$, que em geral é uma função não linear.

$$y(n) = \varphi \left(\sum_{k=1}^K w_{ik} u_k(n) + w_{i0} \right) \quad (3.1)$$

Onde: i é o índice do neurônio e u_k é o sinal ao qual o peso está associado.

3.3 ARQUITETURAS DE REDES NEURAIS ARTIFICIAIS

As RNA são classificadas basicamente por três características: 1) pelo modelo do neurônio que utilizam, 2) pela arquitetura da RNA que descreve a conectividade entre os neurônios, e 3) pelo algoritmo de treinamento ou aprendizado, que consiste na estratégia de ajuste dos pesos sinápticos. Existem diversas arquiteturas de RNA, algumas delas são: as Redes Perceptron Multi-Camadas (PMC), as de Função de Base Radial (RBF – *Radial Basis Function*), os Mapas Auto Organizáveis (SOM – *Self Organized Maps*), as Redes de Hopfield (Haykin, 1998; Hopfield, 1982; Kohonen, 2000), etc. Segundo a sua arquitetura, as RNA podem ser classificadas em dois grupos: as Redes Neurais Progressivas (FNN – *Feedforward Neural Networks*), onde os sinais são propagados em um único sentido, e as Redes Neurais Recorrentes (RNR); nesta rede existem laços de realimentação que se deslocam desde os neurônios de uma camada até os neurônios de camadas anteriores.

3.3.1 Redes Neurais Progressivas

A rede PMC é uma das principais redes do tipo FNN. Um exemplo desta rede é mostrado na Figura 3.2, e basicamente consiste de três componentes: 1) uma camada de entrada, onde os padrões são apresentados à rede; 2) uma camada intermediária, consideradas como extratoras de características e onde é realizada a maior parte do processamento através de conexões ponderadas que realizam o mapeamento não-linear dos sinais de entrada através da função de ativação; e 3) uma camada de saída, onde é apresentado o resultado final. Em uma PMC as informações se movem unicamente para a frente desde a camada de entrada, passando através da camada oculta e chegando à camada de saída.

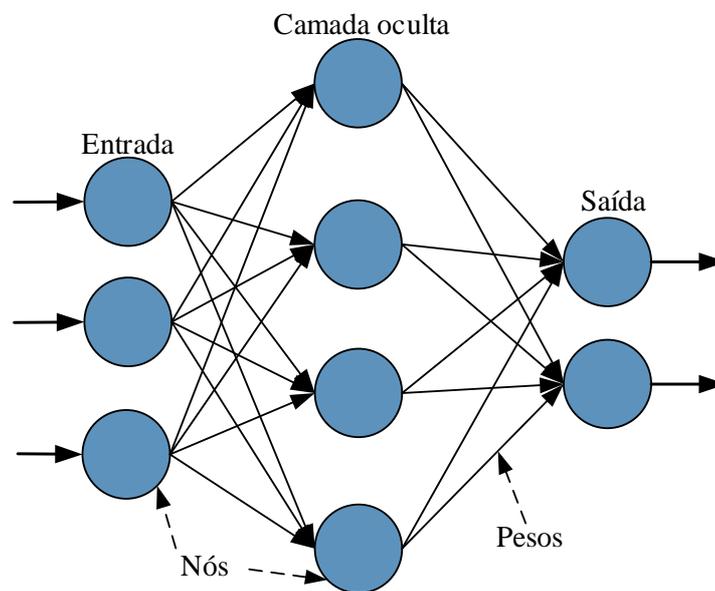


Figura 3.2- Rede Progressiva.

O processo de treinamento de uma rede PMC consiste no ajuste de todos os pesos sinápticos tanto das camadas ocultas quanto da camada de saída, tendo como objetivo encontrar os pesos que representem o melhor mapeamento da relação entrada/saída. No final do treinamento a rede estará em capacidade de fornecer valores próximos aos valores desejados. Neste sentido, o processo de treinamento pode ser formulado como a minimização do erro entre as saídas e os valores desejados. Sendo assim, é possível usar técnicas de otimização não linear, tais como os métodos iterativos de primeira e segunda ordem (Haykin, 1998; Luenberger, 2003), entre os quais está o conhecido algoritmo *backpropagation*.

Com os pesos sinápticos treinados, cada padrão de entrada é mapeado em um único conjunto de sinais de saída. Isto torna as redes PMC capazes de lidar com problemas de aproximação

de funções e classificação de padrões (Haykin, 1998; Duda *et al.*, 2001). Foi demonstrado por Cybenko (1989) e Hornik *et al.* (1989) que esse tipo de rede possui capacidade de aproximação universal, o que significa que existe uma rede PMC que consegue aproximar uma função contínua em um domínio compacto. No entanto, este teorema não especifica o número de neurônios ocultos e o algoritmo de treinamento que deve ser utilizado.

Aplicações deste tipo de redes podem ser encontradas em predição, identificação e controle de sistemas dinâmicos (Suykens *et al.* 1996; Kusnetsov *et al.*, 1998), e filtragem adaptativa (Haykin, 1996). No entanto, em muitos problemas, é uma exigência que o modelo seja capaz de memorizar e acessar o histórico de sinais de entradas e saídas, além dos estados internos da rede. Uma alternativa para abordar os problemas que demandam memória é representar o tempo de processamento e fornecer ao sistema propriedades dinâmicas que respondam a perturbações temporais. Isto pode ser possível inserindo laços de realimentação entre as camadas e/ou neurônios, passando a rede a ter uma espécie de memória, e tendo, portanto, a capacidade de guardar informações para usá-las posteriormente. Estas características podem ser encontradas nas RNR.

3.3.2 Redes Neurais Recorrentes

As RNR possuem laços de realimentação entre suas camadas, como mostrado na Figura 3.3. A partir dessa característica surge o seu maior potencial, a memorização, a capacidade de acessar ao histórico de sinais presentes na rede. Foi demonstrado por Funahashi e Nakamura (1993), e Schafer e Zimmermann (2007) que esta arquitetura também apresenta a capacidade de aproximação universal. Estes fatos demonstram que as RNR são ferramentas poderosas com boas perspectivas de aplicação em problemas dinâmicos e temporais de aprendizagem de máquinas.

As RNR são RNA com conexões realimentadas, o oposto às redes progressivas. Matematicamente as RNR são sistemas dinâmicos, enquanto que as redes progressivas são funções. Na teoria, as RNR podem aproximar sistemas dinâmicos com arbitraria precisão (Doya, 1993) e processar tanto sinais contínuos como discretos. Apesar das capacidades mencionadas, a presença de laços de realimentação faz com que as RNR tenham um comportamento dinâmico não linear bastante complexo, sendo necessário um ajuste delicado dos seus parâmetros a fim de conseguir o objetivo desejado. Devido à realimentação, pequenas alterações nos parâmetros podem desestabilizar a dinâmica da rede, causando o

aumento rápido do erro. O uso desses fatores acaba inibindo a grande capacidade intrínseca desta arquitetura de rede.

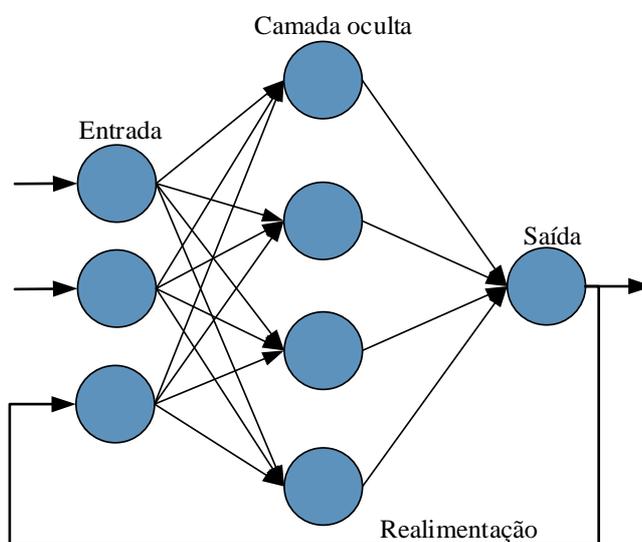


Figura 3.3- Rede neural artificial recorrente.

No âmbito do aprendizado supervisionado, o processo de ajuste é realizado através de métodos iterativos, visando a minimização da função de erro entre as saídas e os valores desejados; alguns desses algoritmos são: Retropropagação através do Tempo (BPTT – *Backpropagation-Through-Time*) (Werbos, 1990), o Aprendizado Recorrente em Tempo Real (RTRL – *Real-Time Recurrent Learning*) (Williams e Zipser, 1989), assim como algoritmos de segunda ordem (dos Santos e Von Zuben, 2000). O algoritmo BPTT proposto por Werbos (1990) é uma extensão do algoritmo *backpropagation* adaptado para RNR.

O algoritmo de treinamento através do método padrão de gradiente descendente se configura como complexo e demanda um alto custo computacional (Lukosevicius, 2012). Além de calcular o gradiente da função de erro através das camadas intermediárias, como ocorre nas redes PMC, também é necessário considerar a influência dos sinais de realimentação. Estas dificuldades são listadas por Jaeger (2002b). Entre elas estão: a convergência lenta do algoritmo de aprendizado, e a dificuldade na escolha dos parâmetros do algoritmo.

Neste contexto, as Redes Neurais com Estados de Eco (ESN – *Echo State Networks*) apresentadas por Jaeger (2001) surgem como uma abordagem capaz de superar as dificuldades de treinamento das RNR. Uma ESN é, basicamente, uma RNR cuja camada oculta é denominada de *reservatório de dinâmicas*. Os pesos sinápticos da camada de

entrada e do reservatório são definidos prévia e aleatoriamente; os pesos da camada de saída são definidos por meio de combinações lineares das ativações do reservatório. Mantendo fixos os pesos das conexões recorrentes do reservatório, o processo de treinamento consiste em ajustar os pesos da camada de saída, o que pode ser realizado por qualquer método de mínimos quadrados (Jaeger, 2001; Lukosevicius e Jaeger, 2009). As ESN além de aproveitar as vantagens de uma estrutura recorrente, conseguem simplificar significativamente o processo de treinamento.

3.4 REDES COM ESTADO DE ECO

Nos últimos anos, as novas abordagens de modelos e algoritmos de treinamento de RNR atraíram atenção dos pesquisadores. Esses algoritmos foram propostos independentemente por Wolfgang Maass sob o nome de Máquinas de Estado Líquido (LSM – *Liquid State Machines*) (Maass *et al.*, 2002; Natschläger *et al.*, 2002), e por Herbert Jaeger sob o nome de Redes de Estados de Eco (ESN – *Echo State Networks*) (Jaeger, 2001; Jaeger, 2002b; Jaeger e Haas, 2004). As LSM e ESN, juntamente com o método mais recente de aprendizado de RNR chamado Decorrelação de Retropropagação (BPDC – *Backpropagation Decorrelation*) (Steil, 2004), deram lugar ao termo Computação com Reservatórios (RC – *Reservoir Computing*) (Verstraeten *et al.*, 2007; Schrauwen *et al.*, 2007).

Um dos métodos pioneiros de RC é a ESN originalmente proposta por Jaeger (2001), e se baseiam na observação das propriedades algébricas de uma RNR aleatória, e que treinando somente a saída (*readout*) em geral é suficiente para conseguir bons resultados em aplicações práticas. Parte de uma ESN é uma RNR não treinada, chamada de reservatório de dinâmicas, onde os estados resultantes $x(n)$ são denominados ecos dos sinais de entrada (Jaeger, 2001). Quando uma ESN satisfaz a chamada Propriedade do Estado de Eco (ESP – *Echo State Property*), torna o reservatório um sistema dinâmico.

Em termos simples, as ESN mantêm a capacidade de processamento dinâmico inerente às RNR e possuem um processo de treinamento simplificado. A arquitetura dessa rede consiste de uma camada de entrada, uma camada densamente interconectada de Elementos de Processamento Não Linear (EPNL), os quais introduzem um rico repertório de comportamentos dinâmicos, e uma camada de saída responsável por combinar esses EPNLs. A estrutura genérica de uma ESN com K sinais de entrada, N unidades internas e L saídas é mostrada na Figura 3.4.

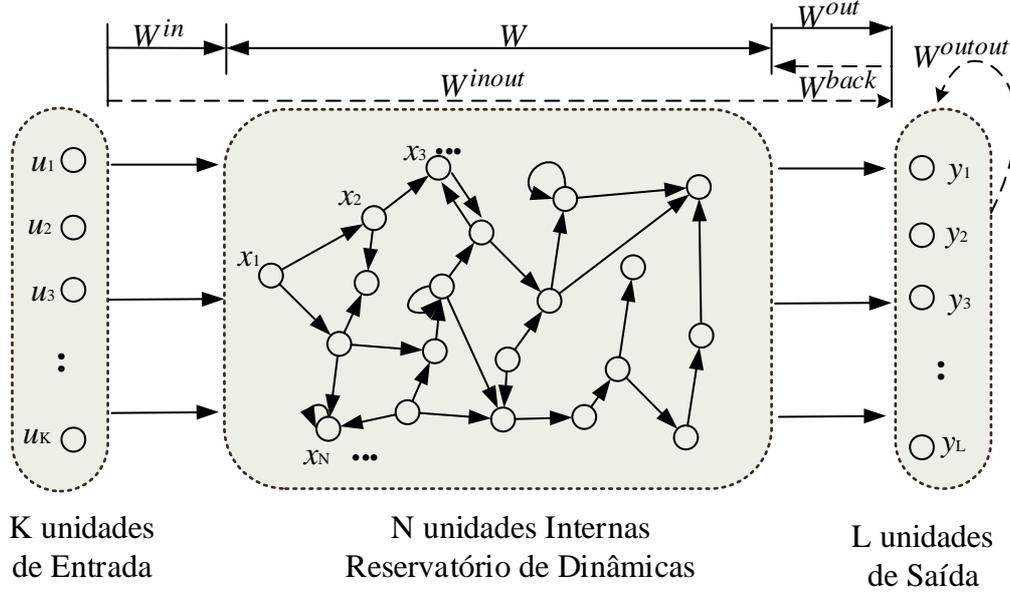


Figura 3.4- Estrutura genérica de uma ESN (Peng *et al.*, 2012).

A camada interna, denominada reservatório de dinâmicas, é composta por EPNL totalmente interconectadas cujas ativações correspondem aos estados da rede. A atualização dos estados é definida pela Equação 3.2 e a saída da rede pela Equação 3.3.

$$x(n + 1) = f(W^{in}u(n + 1) + Wx(n) + W^{back}y(n) + W^{bias}) \quad (3.2)$$

$$y(n + 1) = f^{out}(W^{inout}u(n + 1) + W^{out}x(n + 1) + W^{outout}y(n) + W^{biasout}) \quad (3.3)$$

Onde: $x(n)$ representa o estado do reservatório no instante n , f é a função de ativação das unidades do reservatório (geralmente funções sigmóides), $u(n)$ é a entrada no instante n , W^{in} os pesos sinápticos entre a camada de entrada e o reservatório, e W representa as conexões do reservatório, W^{back} representa as conexões entre o *readout* e o reservatório, W^{bias} representa as conexões entre viés e o reservatório, $y(n)$ é a saída da rede no instante n , f^{out} é a função de ativação das unidades da camada de saída, W^{inout} representa as conexões entre a camada de entrada e a camada de saída, W^{out} as conexões entre o reservatório e a camada de saída, e W^{outout} as conexões recorrentes da camada de saída e $W^{biasout}$ representa as conexões de viés e a camada de saída. Apenas as conexões no sentido à camada de saída são treinadas (W^{inout} , W^{out} , W^{outout} e $W^{biasout}$).

A adição do parâmetro, α (*leak rate*), aos neurônios sigmóides é chamada de *leaky integrator neurons* (Lukosevicius e Jaeger, 2009). Uma das possibilidades é a aplicação antes, e a outra depois da função de ativação. Se escolhido corretamente, o parâmetro α pode efetivamente ajustar a dinâmica da ESN para coincidir com a escala de tempo de entrada, melhorando o

desempenho da rede (Antonelo *et al.*, 2008). A Equação 3.4 apresenta as mudanças da Equação 3.2 de uma ESN depois de adicionar o parâmetro α antes da função de ativação.

$$x(n+1) = f\left((1-\alpha)x(n) + \alpha(W^{in}u(n+1) + Wx(n) + W^{back}y(n) + W^{bias})\right) \quad (3.4)$$

As equações do erro de predição (Equação 3.5) e da atualização do W^{out} (Equação 3.6) através de um algoritmo estocástico de gradiente descendente são descritos a seguir:

$$e(n) = y_d(n) - y(n) \quad (3.5)$$

$$\Delta W^{out} = \eta e(n)[u(n); x(n)]^T \quad (3.6)$$

Onde: $e(n)$ representa o erro, $y_d(n)$ os valores desejados, $y(n)$ as saídas da rede, W^{out} representa os pesos sinápticos entre o reservatório e a camada de saída, η é a taxa de aprendizado, $u(n)$ são as entradas da rede, e $x(n)$ os estados do reservatório no instante n .

3.4.1 Propriedades do Estado de Eco (ESP)

Foi comprovado por Jaeger (2001) que sob as condições listadas no Jaeger (2001), os estados $x(n)$ de uma ESN tornam-se assintoticamente independentes da condição inicial. Significa que se a rede é inicializada a partir de dois estados iniciais $x(0)$ e $\hat{x}(0)$, e é fornecida a mesma sequência de sinais de entrada, as sequências de estados resultantes $x(n)$ e $\hat{x}(n)$ convergem para valores próximos. Quando esta propriedade é satisfeita, os estados se mostram independentes das suas condições iniciais. Então a dinâmica do reservatório dependerá unicamente dos sinais de entrada, o que permite dizer que uma ESN possui estados de eco.

3.4.2 Geração do Reservatório

Uma guia genérica para produzir um bom reservatório dinâmico foi apresentada por Jaeger (2001) e Jaeger (2002b) em artigos que introduziram as ESN. Motivados por um intuitivo objetivo de produzir um rico conjunto dinâmico, a receita é descrita por Lukosevicius e Jaeger (2009) objetivando obter muitos sinais de ativação, escassamente acoplados e diferentes, seguindo os seguintes passos:

- Gerar um reservatório grande;
- Esparsa (em torno de 20% das possíveis conexões);
- Aleatoriamente conectada (distribuição uniforme com média 0 e variância 1).

3.5 TREINAMENTO DE UMA ESN

O processo de aprendizado consiste em determinar os pesos de uma ESN utilizando um conjunto de dados de treinamento. O conjunto de dados consiste de T amostras de pares, entradas e saídas desejadas $[\langle u(1), y_d(1) \rangle, \langle u(2), y_d(2) \rangle, \dots, \langle u(T), y_d(T) \rangle]$. Quando a ESN é alimentada com a entrada de treinamento $u(n)$, a saída $y(n)$ será próxima da saída desejada $y_d(n)$. A Equação 3.7 mostra uma função Raiz do Erro Quadrático Médio Normalizado (NRMSE – *Normalized Root-Mean-Square Error*), bastante usada, e deve ser minimizada durante o processo de treinamento.

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \quad (3.7)$$

Onde: o RMSE é calculado segundo a Equação 3.8 e y_{max} e y_{min} são os valores máximo e mínimo de $y(n)$, respectivamente.

$$RMSE = \sqrt{\frac{1}{T} \sum_{n=1}^T (e(n))^2} \quad (3.8)$$

Onde: e é o erro de predição (Equação 3.5). O treinamento de uma ESN é apresentado nas seguintes etapas.

3.5.1 Inicialização dos Pesos

Inicialmente é gerada uma RNR aleatória que satisfaça a propriedade do estado de eco (Jaeger, 2002b). Para a geração aleatória de pesos de uma ESN serão necessários dois parâmetros: o tamanho da rede N , e o raio espectral $\alpha \in [0, 1]$. O tamanho da rede deve ser selecionado segundo o tamanho dos dados de treinamento, quanto maior seja N , maior será a dificuldade do processo. Enquanto que o raio espectral determina o tamanho da memória da ESN. Nesta primeira etapa constrói-se uma RNR (W^{in} , W , W^{back}) não treinada, que satisfaça a propriedade de estado de eco, consiste dos seguintes passos:

- Gerar uma matriz de pesos aleatórios W_0 ;
- Normalizar a matriz W_0 gerando a matriz W_1 com raio espectral λ de W_0 , $W_1 = W_0/\lambda$. Desta maneira W_1 terá raio espectral unitário;
- Escalar a matriz W_1 para a matriz W com $\alpha < 1$, $W = \alpha.W_1$. Desta forma W terá raio espectral α ;
- Gerar as matrizes de pesos aleatórios W^{in} e W^{back} .

3.5.2 Amostragem Dinâmica de Treinamento

Utilizando as matrizes de pesos inicializados, executa-se a ESN introduzindo conjunto de dados de treinamento, $[u(1), u(2), \dots, u(T)]$ e aplicando a Equação 3.2. Em cada amostra, os estados das unidades de entrada e das unidades internas são armazenados na matriz de coleção de estados X de dimensão $T \times (K+N)$, sendo K , N e L os números de unidades de entrada, interna e saída, respectivamente.

3.5.3 Treinamento do *Readout*

Chamar-se-á W_{out} todos os pesos do *readout* (W^{inout} , W^{out} , W^{outout} e $W^{biasout}$). Teoricamente, a formação do *readout* a partir de um reservatório é uma tarefa supervisionada não-temporal de mapeamento dos estados do reservatório $x(n)$ para as saídas desejadas $d(n)$. Sendo este um domínio bastante investigado em aprendizado de máquina e existem vários métodos disponíveis. No método *batch*, o treinamento dos pesos do *readout* (W_{out}) pode ser definido como uma solução de um sistema de equações lineares da Equação 3.9.

$$W_{out}X = Y_d \quad (3.9)$$

Onde: $X \in \mathbb{R}^{N \times T}$ são todos os estados do reservatório e $Y_d \in \mathbb{R}^{L \times T}$ são todos os valores desejados $d(n)$, ambos armazenados durante o período de treinamento $n=1, \dots, T$.

O processo de treinamento consiste em minimizar o erro quadrático $E(Y_d, W_{out}X)$. Para resolver a Equação 3.9 utilizam-se com frequência métodos para encontrar as soluções dos mínimos quadrados em sistemas de equações lineares, conhecido como regressão linear. Os métodos comumente utilizados para o treinamento do *readout* são a pseudo-inversa de Moore-Penrose (Lukoseviciuse Jaeger, 2009) e a regressão *Ridge* (*ridge-regress*) (Bishop, 2006). Existem outros métodos como a decomposição ponderada ou a utilização de busca evolucionária (Lukoseviciuse Jaeger, 2009). Um método direto consiste em calcular a pseudo-inversa de Moore-Penrose X^+ de X e W_{out} usando a Equação 3.10.

$$W_{out} = Y_d X^+ \quad (3.10)$$

Onde: W_{out} representa todos os pesos do *readout*, Y_d contém todas as saídas desejadas do processo de treinamento, X^+ é a matriz pseudo-inversa de X .

O cálculo direto através da pseudo-inversa mostra alta estabilidade numérica, mas demanda um alto custo computacional para grandes matrizes de coleta de estados $X \in \mathbb{R}^{N \times T}$, limitando assim o tamanho do reservatório N e/ou o número de amostras do conjunto de dados de treinamento T . Este problema é resolvido formulando equações normais a partir da

Equação 3.9 obtendo a Equação 3.11 e finalmente a sua solução é expressada segundo a Equação 3.12.

$$W_{out}XX^T = Y_dX^T \quad (3.11)$$

$$W_{out} = Y_dX^T(XX^T)^{-1} \quad (3.12)$$

Nota-se que neste caso $Y_d \in \mathbb{R}^{L \times N}$ e $XX^T \in \mathbb{R}^{N \times N}$, o que indica que a solução não depende do tamanho do conjunto de treinamento T .

3.6 DESAFIOS NAS REDES COM ESTADO DE ECO

Além dos estudos da aplicação das ESN, a maior parte das pesquisas atualmente é dedicada ao modelo do reservatório ideal, ou aos algoritmos de otimização do reservatório de dinâmicas, que são detalhadas em Lukosevicius e Jaeger (2009). Outro desafio é a modelagem da camada de saída para uma estrutura não linear. Boccato (2013) propôs uma abordagem não linear para a camada de saída, conservando a simplicidade do processo de treinamento. Outro desafio identificado para a aplicação em prognósticos que permitam adicionar novos casos de falha à ESN é propor uma arquitetura para aprendizado supervisionado incremental (Pinto *et al.*, 2011; Ciarelli *et al.*, 2012).

3.6.1 Otimização do Reservatório

O desempenho de uma ESN depende basicamente dos seguintes parâmetros: fator de escala de entrada, tamanho do reservatório, e a topologia do reservatório. O fator de escala de entrada controla o impacto das entradas sob os estados do reservatório (Butcher *et al.*, 2013). Na literatura de RC, os autores recomendam reservatórios com grande quantidade de neurônios, escassamente conectadas. A projeção do reservatório num espaço maior aumenta a precisão do modelo, embora exista uma compensação para alcançar o tamanho do reservatório.

Um reservatório muito grande pode provocar o fenômeno conhecido como *overfitting*. O raio espectral impacta na estabilidade e na caoticidade do reservatório, como consequência influi na capacidade de memória do modelo. A estabilidade do reservatório ESN é garantida quando o raio espectral é menor do que 1. Esta condição de estabilidade foi estabelecida na propriedade de estado de eco (ESP) (Jaeger, 2001). De acordo com as experiências previas não é claro o impacto da densidade do reservatório na precisão do modelo. Como o processamento de informações de uma matriz esparsa é mais rápido do que de uma matriz

densa, como consequência, um reservatório esparso pode melhorar o tempo de processamento (Lukosevicius e Jaeger, 2009; Lukosevicius, 2012).

As estruturas iniciais do RC tais como ESN e LSM são baseadas em RNR com pesos fixos e gerados aleatoriamente. No entanto, Lukosevicius e Jaeger (2009) declararam que a geração aleatória dos pesos do reservatório e o treinamento da camada de saída através de uma função de regressão linear não necessariamente representam uma solução ótima para a RC. Então, surge a necessidade de procura por alternativas para geração do reservatório e o treinamento da camada de saída. Como alternativas para a geração do reservatório de dinâmicas são encontradas duas abordagens: inicializando os pesos de forma não-supervisionada ou a partir de um pré-treinamento supervisionado. O método não-supervisionado envolve a otimização de algumas medidas definidas no reservatório para uma entrada determinada, sem tomar em conta a saída desejada. Em contraste, o pré-treinamento supervisionado toma em consideração as saídas desejadas.

Recentemente foram usados algoritmos genéticos e evolutivos para otimizar os parâmetros globais e determinar a conectividade do reservatório (Ferreira e Ludermir, 2010; Ferreira e Ludermir, 2011; Ferreira *et al.*, 2013). A técnica de otimização por enxame de partículas (PSO – *Particle Swarm Optimization*) foi utilizada para definir o raio espectral e outros parâmetros principais do reservatório (Sergio e Ludermir, 2012). Além da adaptação do reservatório, outras formas de otimizar o reservatório podem ser consideradas. Ferreira e Ludermir (2009) apresentaram um método para otimizar os parâmetros globais usando GA para predição de séries temporais. Ferreira *et al.* (2013) desenvolveram um método para encontrar o melhor reservatório em predição de séries temporais chamado de Projeto de Computação com Reservatórios (RCDESIGN – *Reservoir Computing Design*), o método busca simultaneamente os melhores parâmetros globais da topologia da rede e os pesos, combinando o RC com um algoritmo evolutivo. Dois métodos de otimização adicionais foram implementados para comparar os resultados. “*RS Search*” procura otimizar o tamanho do reservatório, o raio espectral e a densidade da conexão. “*TR Search*” não considera a abordagem de sistemas lineares com RC. O RCDESIGN forneceu resultados satisfatórios nos conjuntos de dados estudados, resultando melhor do que os outros métodos comparados. Todas as abordagens foram usadas para predição de velocidade de ventos, o qual é uma importante tarefa para geração de energia eólica.

A PSO se destaca entre as abordagens para otimização de RC devido às suas vantagens comparado com os GA (Hassan *et al.*, 2005; Panda e Padhy, 2007). Duas extensões do algoritmo PSO foram aplicadas por Sergio e Ludemir (2012) com o objetivo de aumentar a eficiência, a Estratégia de Utilização Eficiente de População para Otimização por Enxame de Partículas (EPUS-PSO – *Efficient Population Utilization Strategy for Particle Swarm Optimizer*)(Hsieh *et al.*, 2009) e o outro Otimização por Enxame de Partículas Adaptativo (APSO – *Adaptive Particle Swarm Optimization*). Estes algoritmos foram usados para otimizar todo o conjunto de parâmetros do RC. Conclui-se que, em cinco séries usadas, o método proposto reduz o número de ciclos necessários para treinar a rede (Sergio e Ludemir, 2012). Os parâmetros otimizados foram o número de nós no reservatório, a função de ativação dos neurônios no reservatório, o raio espectral e a presença ou ausência de conexões opcionais.

Baseado no método RSDESIGN, Sergio (2013) propôs um método para otimização global dos parâmetros, topologia e pesos de RC usando PSO e as duas extensões EPUS-PSO e APSO. O método APSO obteve melhores resultados para o erro de predição em séries temporais, e o EPUS-PSO mostrou um desempenho melhor quando é requerido que o critério de ciclos de treinamento precisa superar valores ótimos. Com os bons resultados obtidos através da aplicação dos algoritmos bioinspirados tais como PSO e suas extensões, foi encontrada também uma abordagem baseada no Algoritmo de Minhoca (WA – *Worm Algorithm*) (Abdulrasool e Abbas, 2013); e baseada em algoritmos híbridos (Sergio e Ludemir, 2014).

3.6.2 Camada de saída não-linear

A função básica da camada de saída de uma ESN é mapear os estados do reservatório $x(n)$ nos sinais de saída $y(n)$, com o objetivo de aproximar um conjunto de saídas desejadas $y_d(n)$, correspondendo a uma tarefa supervisionada de aprendizado de máquina. A proposta inicial, realizada por Jaeger (2001), é a mais utilizada e consiste no uso de um combinador linear. A vantagem do uso de uma estrutura linear está na sua eficiência e simplicidade de treinamento. A principal desvantagem é a sua capacidade de aproximação em aplicações que envolvem sistemas não lineares. Neste contexto surgiram abordagens para ampliar as capacidades de uma ESN e lidar com as deficiências da camada de saída tradicional. Surgindo como alternativa propostas de estruturas não lineares.

O uso de uma rede PMC como camada de saída para uma LSM foi proposta por Maass *et al.* (2002), e Bush e Anderson (2005); e para uma ESN por Babinec e Pospichal (2006). A aplicação de PMC permitiu ganhar flexibilidade, no entanto, o treinamento em muitos casos teve um desempenho inferior se comparado com o combinador linear de uma ESN original. No processo de treinamento de uma PMC, foi mostrado por Huang *et al.* (2004) e Huang *et al.* (2006) que através do uso de Aprendizado de Máquina Extremo (ELM – *Extreme Learning Machine*) evitam-se a retropropagação do erro e os cálculos iterativos, representando simplicidade e eficiência de treinamento. Neste contexto, surge o uso de uma arquitetura híbrida que foi originalmente proposta por Butcher *et al.* (2010) e Butcher *et al.* (2013), onde uma ELM representa a camada de saída de uma ESN.

Outra abordagem é apresentada por Shi e Han (2007) com o uso de Máquinas de Vetor Suporte (SVM – *Support Vector Machines*). Esta abordagem foi utilizada para treinamento de uma ESN, modelando o reservatório como o *kernel* temporal de uma SVM, de modo a treinar a ESN usando os princípios de uma SVM. Também é usada na camada de saída uma combinação de diferentes camadas de saída por Bush e Anderson (2006). A aplicação de saídas hierárquicas foi proposta por Jaeger (2007), onde as saídas superiores são usadas como coeficientes das saídas dos níveis inferiores. Uma nova abordagem para uma camada de saída não linear usando o filtro de Volterra é apresentada por Boccato (2013).

3.6.3 Aprendizado Incremental

Na área de aprendizado incremental existem arquiteturas exclusivamente para este fim. Entre as mais conhecidas estão o Mapa da Teoria de Ressonância Adaptativa (ARTMAP – *Adapative Resonance Map*), uma família de arquiteturas propostas por Carpenter e Grossberg (1987), baseadas na Teoria de Ressonância Adaptativa (ART – *Adapative Resonance Theory*), que possui capacidade de aprendizado incremental supervisionado ou não supervisionado. Durante o treinamento, as redes ART ajustam as categorias aprendidas previamente em resposta a uma entrada parecida, e criam novas categorias dinamicamente, em resposta às entradas suficientemente diferentes das conhecidas previamente.

Dentro desta família tem-se também a *fuzzy* ARTMAP, que consiste de duas camadas com os nós totalmente conectados, uma camada de entrada F_1 de M nós, e uma camada competitiva F_2 de N nós, assim como um conjunto de valores reais de pesos W , que associa as conexões das camadas de F_1 e F_2 . Outra rede que difere em poucos aspectos da rede

anterior é a chamada Gaussian ARTMAP; esta representa as categorias como uma função de densidade gaussiana, definida por vetores com suas médias e desvios padrões.

As Redes Incrementais de Misturas Gaussianas (IGMN – *Incremental Gaussian Mixture Network*) usam o algoritmo de aproximação incremental (Dempster *et al.*, 1977), e o Modelo Incremental de Misturas Gaussianas (IGMM – *Incremental Gaussian Mixture Model*) (Engel e Heinen, 2011). Essa rede cria e ajusta continuamente os modelos probabilísticos para todos os dados já apresentados, depois da apresentação de um novo dado. O processo de aprendizado, chamado de “*one-shot*”, revela que é necessária uma única passada através dos dados para obter um modelo consistente.

A revisão desses trabalhos se dedica ao aprendizado incremental como um desafio a aplicar às ESN, dando lugar às redes chamadas ESN adaptativas, onde os pesos das camadas de saída são ajustados de forma *online* à medida que novos sinais de entrada vão aparecendo. Esta propriedade é interessante em prognósticos de falhas, uma vez que, em sistemas dinâmicos, a aparição de novas falhas é frequente, e os modelos precisam ser atualizados para incorporar os novos tipos de falha.

3.7 APLICAÇÕES DE RNA EM PROGNÓSTICO

Aplicações em prognóstico usando RNA podem ser encontradas em diferentes áreas. Uma das áreas com muitas aplicações é na medicina (Arizmendi *et al.*, 2011; Yonglin *et al.*, 2014). Outras aplicações são encontradas no prognóstico de falhas de equipamentos usando dados de monitoração de condição; nesta área, diferentes tipos de RNA foram vastamente explorados e aplicados (Liu *et al.*, 2010; Gebraeel e Lawley, 2008; Goebel *et al.*, 2008b; Tian, 2009).

Através da revisão de trabalhos, foram identificadas as duas principais vantagens de estimação de RUL usando RNA: 1) as RNA são técnicas bastante usadas na modelagem de funções complexas, ocultas e com alta dependência não linear, portanto são uma boa ferramenta para prognóstico de RUL; 2) as RNA são modelos baseados em exemplos de aprendizagem, mesmo quando há um ruído significativo nos dados de treinamento. Como desvantagem tem-se o alto custo computacional, especialmente na etapa de treinamento.

3.7.1 Prognóstico usando RNA

Tian *et al.* (2010), com o objetivo de melhorar a precisão de prognóstico, desenvolveram um método de predição baseado em RNA usando dados de falhas e dados suspensos. Considera-se dados de falhas quando no último dado coletado o equipamento parou de funcionar, e dados suspensos quando o equipamento ainda está em funcionamento. Uma das considerações levadas em conta foi a de que a relação entradas-saídas de uma RNR é a mesma para todo o histórico de dados de falha e suspensão. Com base nessa ideia, determinaram o prognóstico de vida otimizado, usando o histórico de dados suspensos através da minimização do erro quadrático médio. Na etapa de treinamento usaram o histórico de dados falhas e histórico de dados suspensos. A validação da abordagem foi realizada usando dados reais de MC de vibração coletados de rolamentos de bombas.

A arquitetura apresentada por Tian *et al.* (2010) é mostrada na Figura 3.5, onde t_i é o tempo de funcionamento do equipamento até o momento atual (instante i), e t_{i-1} é o tempo de funcionamento até a inspeção anterior (instante $i-1$), z_j^1 e z_{j-1}^1 são valores das medições da grandeza 1 nos instantes atual e prévio, respectivamente. z_j^2 e z_{j-1}^2 são valores das medições da grandeza 2 nos instantes atual e prévio, respectivamente.

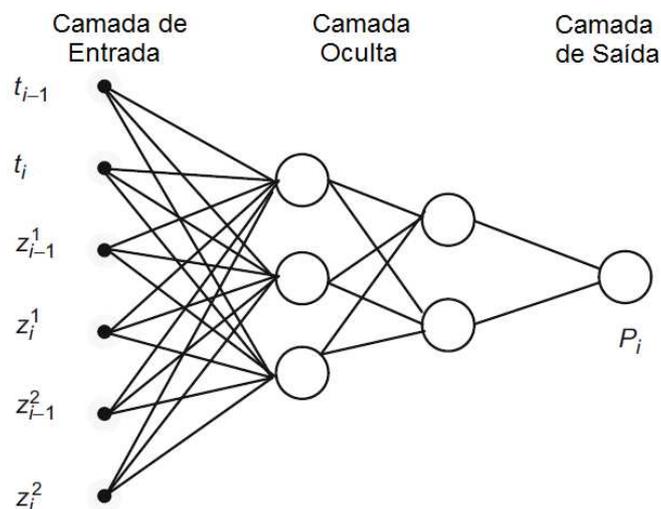


Figura 3.5- Estrutura do modelo RNA usada para predição de RUL (Tian *et al.*, 2010).

3.7.2 Prognóstico usando ESN

Peng *et al.* (2012a) apresentaram uma abordagem baseada em uma ESN modificada para estimar a RUL. A modificação resulta em uma abordagem com submodelos classificados da ESN, como mostrado na Figura 3.6. A realização deu-se no treinamento dos pesos de saída,

utilizando para esse propósito um filtro de Kalman. O método proposto foi testado, apresentando um resultado comparativo entre uma ESN clássica e uma ESN treinada pelo filtro de Kalman. O modelo consiste na criação de vários submodelos cada uma com uma ESN, agrupando os dados que melhor desempenho conseguem com um submodelo.

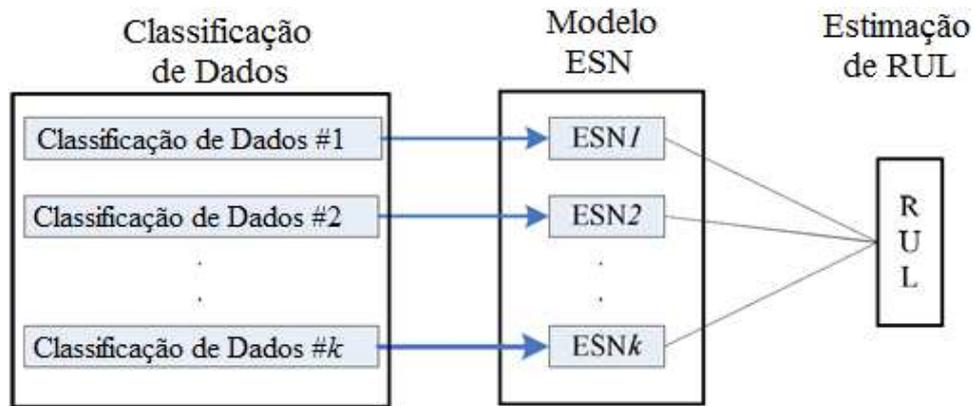


Figura 3.6- ESN modificada por Peng *et al.* (2012a).

Morando *et al.* (2013) apresentaram uma ferramenta para a otimização da vida útil de células de combustível baseada na ferramenta computacional de reservatório, e na Rede de Estados de Eco (ESN). Este sistema de prognóstico visa estimar a RUL de uma membrana de célula de combustível. Os autores mostraram o potencial da ESN em aplicações de prognóstico sem propor melhorias neste tipo de RNA (Morando *et al.*, 2014). Outra aplicação de ESN para predição de cargas em sistemas de potência é apresentada por Deihimi e Showkati (2012) mostrando a capacidade das ESN para aprendizado de dinâmicas complexas de séries de tempo de cargas elétricas e predição de futuras cargas com aceitável precisão. Aplicações de ESN na classificação de dados para predição de diálise em pacientes críticos são apresentadas por Ongena *et al.* (2013). Estudos comparativos entre ESN e ELM aplicados à predição foram apresentados por Bin *et al.* (2011) e Siqueira *et al.* (2012).

3.8 SÍNTESE DO CAPÍTULO

Neste capítulo foi apresentada uma introdução às RNAs. Mostrou-se que as RNR se destacam pela sua capacidade de memória interna e laços de realimentação, sendo apropriadas para a solução de problemas de natureza dinâmica, tal como prognóstico. No entanto, as RNR apresentam desvantagens no processo de treinamento, surgindo as ESN como uma alternativa para lidar com as desvantagens apresentadas. Uma RNR se torna uma ESN quando satisfaz a ESP, isto é, quando o reservatório da rede depende unicamente do

histórico das entradas. A maior vantagem encontrada nas ESN é a sua capacidade de processamento recorrente, com um treinamento relativamente simples, resumindo apenas aos pesos da camada de saída. O reservatório gerado de forma aleatória consiste de muitos neurônios, e o seu processamento muitas vezes precisa ser otimizado. Uma área de pesquisa está dedicada à otimização do reservatório de dinâmicas. Na revisão de trabalhos correlatos sobre a aplicação de prognóstico de RUL baseados em RNA, foram apresentados os trabalhos que usam RNR e ESN. As aplicações de ESN em prognóstico são poucas, mas os resultados alcançados são promissores, abrindo espaço para pesquisas a serem desenvolvidas na otimização do processamento e à sua aplicação em prognóstico de falhas.

4 ABORDAGEM DESENVOLVIDA DE PROGNÓSTICO DE RUL BASEADO EM ESN

As RNAs são ferramentas eficientes na solução de um grande número de problemas, mas elas apresentam várias limitações ou deficiências, que as torna insuficientes para resolver todos os problemas de IA. Uma única técnica de IA, devido às suas limitações ou deficiências, pode não ser capaz, por si só, de resolver um dado problema. Neste caso, a combinação de duas ou mais técnicas pode gerar uma solução mais robusta e eficiente (Braga *et al.*, 2007). Essa combinação de duas ou mais técnicas de IA é conhecida como sistema híbrido. Pesquisas em sistemas híbridos têm-se focado em combinar técnicas baseadas em dados com técnicas baseadas em conhecimento. Os algoritmos de otimização global também têm sido aplicados em sistemas híbridos, em especial na determinação dos valores ótimos de configuração de sistemas inteligentes. A abordagem desenvolvida neste capítulo apresenta um sistema híbrido para prognóstico de RUL, que resulta da combinação do algoritmo ABC para a otimização dos parâmetros e pesos do reservatório de uma ESN. O prognóstico de RUL através de ESN é baseado em dados históricos de MC e dados de falha.

4.1 MOTIVAÇÃO

Das várias técnicas de otimização, o algoritmo ABC destaca-se pela eficiência na busca de soluções ótimas de vários tipos de problemas, com a vantagem de utilizar poucos parâmetros de controle. Em comparação com as outras técnicas, Karaboga e Akay (2009) contrastaram o desempenho do algoritmo ABC com o desempenho das técnicas GA, PSO e Algoritmo de Evolução Diferencial (DEA – *Differential Evolution Algorithm*), os resultados mostraram que o desempenho do algoritmo ABC é melhor ou similar aos outros algoritmos citados. Os resultados obtidos por Turanoglu *et al.* (2011), Butani *et al.* (2011); Hossain e El-Shafie (2014), onde é realizada a comparação do algoritmo ABC e o algoritmo PSO, foi demonstrado que o ABC é mais eficiente na busca de soluções ótimas. Esses resultados motivaram a investigação da sua aplicação na escolha de parâmetros de uma ESN e posteriormente a aplicação em prognóstico de falhas. Geralmente o ajuste dos parâmetros de uma ESN é realizado de maneira exaustiva ou através de experimentos sistemáticos, demandando muito tempo na implementação e alto custo computacional.

Os principais parâmetros que determinam a dinâmica de uma ESN são: o número de neurônios, o raio espectral, o percentual de conexões, as escalas de entrada e saída, os deslocamentos de entrada e saída, e o tipo da função de ativação dos neurônios (Verstraeten *et al.*, 2007 e Ishii *et al.*, 2004). A busca pelos parâmetros de uma ESN utilizando um algoritmo de otimização como uma ferramenta de ajuste dos principais parâmetros foi usada por Ishii *et al.* (2004), Ferreira *et al.* (2008) e Ferreira e Ludermir (2009), as interconexões da rede e os pesos são gerados aleatoriamente resultando em uma busca dos parâmetros globais. A abordagem desenvolvida nesta tese adota este método embora demande um grande espaço de busca.

4.2 ABORDAGEM DESENVOLVIDA

O método de prognóstico de Vida Útil Remanescente proposto nesta tese é baseado em um sistema inteligente híbrido. Utiliza o algoritmo ABC para ajuste simultâneo dos parâmetros globais e dos pesos do reservatório de uma Rede com Estados de Eco. Esta abordagem híbrida utiliza dados históricos de monitoramento de condição e dados de eventos (dados de funcionamento até a falha) de um conjunto de máquinas, equipamento ou sistemas do mesmo tipo. A arquitetura da abordagem é mostrada na Figura 4.1, e inclui três módulos: 1) Aquisição de dados, 2) Treinamento e Otimização, e 3) Prognóstico de RUL.

4.2.1 Aquisição de Dados

Neste módulo realiza-se a coleta de dados de monitoração de condição e dados de eventos (dados de falha). Os dados coletados devem passar por um processo de extração e seleção de características visando obter variáveis que possam ser usadas em prognóstico. Se não se dispõe de conjuntos de dados de treinamento e teste, deve ser realizada a fragmentação do total de dados, em dados de treinamento e dados de teste.

Os dados históricos de MC consistem de grandezas coletadas dos sensores instalados em uma máquina ou sistema. As grandezas podem ser: temperatura, pressão, vazão, vibração, etc., essas grandezas são úteis para identificar as mudanças que indiquem a evolução de uma falha. Neste trabalho, o histórico MC será representado por uma matriz, $H = [T_{nx1}, C_{nxp}]$ de ordem $nx(p+1)$, a qual contém informações dos “ n ” ciclos de funcionamento durante a operação de uma máquina, onde T é o tempo ou ciclo de funcionamento e a matriz C contém as “ p ” grandezas de MC durante os “ n ” ciclos.

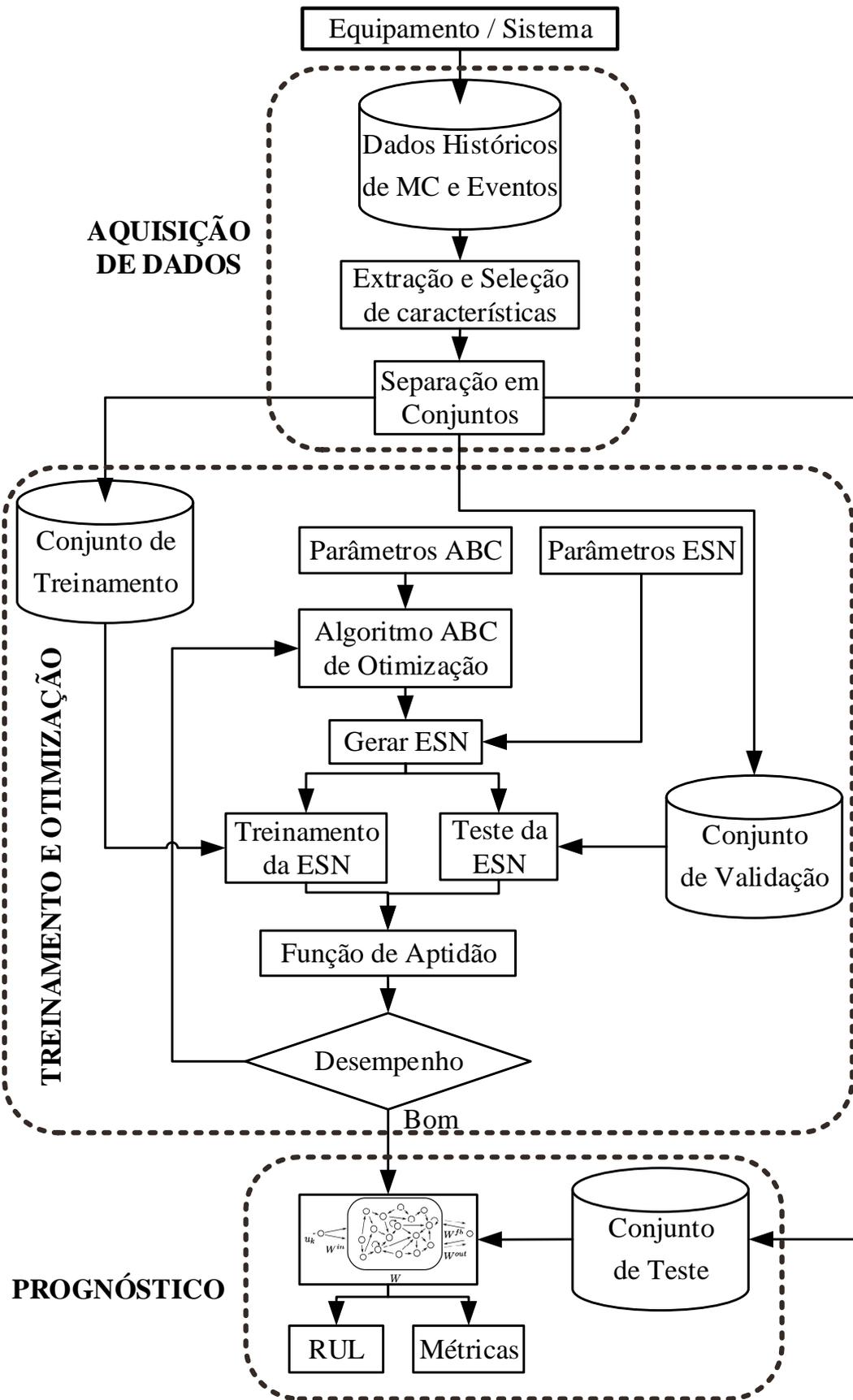


Figura 4.1- Arquitetura da abordagem ESN-ABC.

Considerando que esteja disponível o histórico de funcionamento de “ m ” máquinas do mesmo tipo, define-se o arranjo de matrizes $[H] = \{H_1, H_2, \dots, H_m\}$ onde são armazenados os dados de monitoração de condição necessários para prognóstico de RUL. O histórico de MC de uma nova máquina quando disponibilizado, pode ser anexado em $[H]$ em qualquer ordem. Os tamanhos das matrizes H não necessariamente têm que ser os mesmos, isto é, o histórico de uma máquina pode consistir de mais ou de menos medidas ou sinais de MC do que o histórico de outra máquina do mesmo tipo.

No histórico de dados de eventos ou dados de falha pode-se encontrar o tempo de funcionamento até a falha (RTF – *Run To Failure*) das “ m ” máquinas, sendo definido pelo vetor $RTF = [RTF_1, RTF_2, \dots, RTF_i, \dots, RTF_m]$ de ordem m . O valor de RTF_i é o tempo de funcionamento até a falha ou o tempo em que esteve em operação a máquina “ i ” até a ocorrência de uma determinada falha funcional que provocou a parada dessa máquina.

Em geral os históricos de MC contêm dados brutos, multidimensionais, de diferentes frequências e amplitudes de onda. Para obter informações úteis que indiquem o processo de degradação de uma máquina, os dados brutos têm que passar por um processo de extração e seleção. A abordagem de extração de características inclui diferentes técnicas de processamento que podem ser classificadas em três domínios: tempo, frequência e tempo-frequência. Dependendo do tipo de dados contidos no histórico de MC opta-se por um método em qualquer dos domínios. Por outro lado, a etapa de seleção de características consiste em identificar o melhor conjunto de características para treinamento e teste do algoritmo de prognóstico.

Finalmente, a separação de dados em conjuntos de treinamento, validação e teste é uma parte importante da avaliação desta abordagem. Normalmente, na separação é realizada de forma aleatória para um conjunto de dados em um conjunto de treinamento, um conjunto de validação, e um conjunto de teste. A maior parte dos dados é usada para treinamento, uma parte menor dos dados é usada para teste, e uma parte ainda menor é usada para validação.

4.2.2 Representação da solução

A abordagem desenvolvida aponta para a otimização dos parâmetros do reservatório dinâmico, especificamente seis deles e os pesos do reservatório W . Os parâmetros são: o tamanho do reservatório, o raio espectral, a escala de entrada, o deslocamento de entrada, a

escala de saída, o deslocamento de saída. Cada solução “S” é representada por um vetor segundo o formato da Equação 4.1.

$$S = [N, sr, is, if, os, of, \bar{W}_0] \quad (4.1)$$

Onde:

- N – Tamanho do reservatório que determina o tamanho da matriz de pesos W ;
- sr – Raio espectral $[0,1]$;
- is – Escala de entrada $[0, 1]$, valor que multiplica a um vetor unitário de tamanho igual ao número de nós de entradas da ESN;
- if – Deslocamento de entrada $[-1, 1]$, valor que multiplica a um vetor unitário de tamanho igual ao número de nós de entradas da ESN;
- os – Escala de saída $[0, 1]$, valor que multiplica a um vetor unitário de tamanho igual ao número de nós de saídas da ESN;
- of – Deslocamento de saída $[-1, 1]$, valor que multiplica a um vetor unitário de tamanho igual ao número de nós de saídas da ESN;
- \bar{W}_0 forma vetorial da matriz de pesos W_0 começando na posição 7 até a posição N^2+6 , como mostrado na Figura 4.2. A matriz W_0 é normalizado gerando uma matriz W_1 com raio espectral λ de W_0 , $W_1 = W_0/\lambda$. Fazendo com que W_1 tenha raio espectral unitário. Finalmente, escala-se a matriz W_1 para a matriz de pesos do reservatório $W = sr*W_1$. Desta forma W terá o valor de sr como raio espectral.

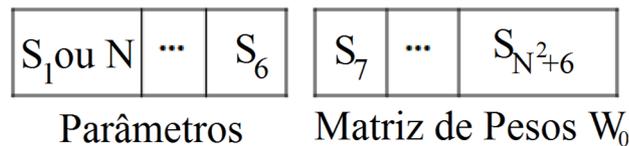


Figura 4.2- Elementos do vetor solução S.

O tamanho do reservatório dinâmico (N) representa a dificuldade do processo de treinamento. Um valor muito pequeno ou muito grande pode causar *overfitting*. Por outro lado, a escolha do raio espectral (sr) da matriz de pesos do reservatório é de muita importância para o sucesso do processo de treinamento. Isso porque sr está intimamente ligado à escala de tempo intrínseca da dinâmica do reservatório. Um valor pequeno de sr significa que a ESN tem um reservatório rápido, enquanto um valor grande de sr (próximo de 1) significa que a ESN tem um reservatório lento.

Os parâmetros que determinam o comportamento e o desempenho da rede estão definidos pela estrutura topológica que permite conectar as entradas, o reservatório, e as saídas; por um lado para o reservatório, e por outro para as saídas; com exceção de conexões do reservatório para as entradas. Numa topologia básica somente existirão conexões das entradas para o reservatório, do reservatório para o reservatório, e do reservatório para as saídas. A configuração básica pode ser estendida com conexões diretas entre as entradas e as saídas. Neste trabalho usa-se a topologia básica nos experimentos.

4.2.3 Função de Aptidão

Devido ao fenômeno de *overfitting*, se há duas redes treinadas para o conjunto de dados de treinamento, a rede com o maior erro durante o treinamento pode ter um desempenho melhor do que a rede com o menor erro, pois a rede com o menor erro pode ter se adaptado aos valores do conjunto de treinamento, perdendo desta forma a capacidade de uma boa generalização. Neste trabalho a função de aptidão considera o desempenho do conjunto de treinamento e validação, adaptando desta maneira o critério de perda de capacidade de generalização (*GL-Generalization Loss*) do Proben 1 (Prechelt, 1994). A função de aptidão usada é baseada na formula apresentada por Ferreira (2011), mostrada na Equação 4.2.

$$f = NRMSE_{Treinamento} + \left\| NRMSE_{Treinamento} - NRMSE_{Validação} \right\| \quad (4.2)$$

Onde: f é o valor a ser minimizado pelo algoritmo ABC, e os valores de NRMSE são calculados através da Equação 3.7.

4.2.4 Parâmetros

Ao utilizar o algoritmo ABC em conjunto com a rede ESN para prognóstico de falhas, é importante analisar como a escolha dos parâmetros utilizados pode influenciar no seu desempenho. Os parâmetros devem ser escolhidos segundo as necessidades do problema e os recursos disponíveis. Na Tabela 4.1 e na Tabela 4.2 encontram-se os principais parâmetros do algoritmo ABC e da rede ESN respectivamente.

4.2.5 Algoritmo de otimização

O processo de busca do algoritmo ABC envolve uma sequência de passos onde um conjunto de soluções passa por processos de seleção. Este processo é dividido em três passos: entradas, onde são coletados os dados, o processo de otimização onde será realizado o ajuste dos

parâmetros e pesos, e finalmente o resultado que mostra os melhores parâmetros e pesos da rede treinada. Os passos descritos são apresentados a seguir:

Tabela 4.1- Parâmetros do algoritmo ABC (Karaboga, 2005).

Sigla	Descrição	Valor ou Faixa
COL	Tamanho da colônia (Abelhas Operárias + Abelhas Seguidoras)	Valor diretamente proporcional às possíveis soluções, e inversamente ao tempo de processamento.
BN	Número inicial de abelhas operárias	COL/2
SN	Número de fontes de alimento	BN
BC	Número inicial de abelhas seguidoras	COL-BN
maxTrial	A fonte de alimento o qual não pode ser melhorada.	Depois de " maxTrial " tentativas é abandonada pelas abelhas operarias.
maxIter	O número de ciclos (Critério de parada)	Quantidade de Iterações.
D	Número de parâmetros a ser otimizado	Tamanho do vetor S (Equação 4.1)
f	Função de aptidão a ser minimizada	Equação 4.2
ub	Limite superior dos parâmetros	Valor máximo do vetor S.
lb	Limite inferior dos parâmetros	Valor mínimo do vetor S.

Tabela 4.2- Parâmetros da ESN (Lukosevicius, 2012).

Sigla	Descrição	Valor ou Faixa
NE	Número de Entradas da ESN	Número de variáveis de monitoração de condição
N	Número neurônios do reservatório	$[T/10, T/2]$, onde: T é número de dados de treinamento
IS	Escala de Entrada	[0.01, 1]
IF	Deslocamento de Entrada	[-1, 1]
OS	Escala de Saída	[0.01, 1]
OF	Deslocamento de Saída	[-1, 1]
sr	Raio Espectral	[0.01, 1]
c	Conectividade do reservatório	[0.1, 0.5]

Entradas: As entradas do algoritmo ABC são: o tamanho da colônia de abelhas (COL), o número inicial de abelhas operárias (BN) que será numericamente igual à metade do tamanho da colônia; o número de fontes de alimento (SN) que é igual a BN; o número inicial de abelhas seguidoras (BC), que é igual à diferença entre COL e BN; e o número de tentativas de liberar uma fonte de alimento (*maxIter*).

Processo de Otimização: Começa enviando aleatoriamente as abelhas operárias procurar por um conjunto de soluções S (parâmetros e pesos do reservatório da ESN), dentro de um espaço de busca. Avaliar esses conjuntos S de acordo com $f_{Aptidao}$, quanto menor for a $f_{Aptidao}$, melhor será o conjunto S . A partir dessa avaliação os melhores conjuntos S (melhores fontes de alimento), os quais possuem os melhores resultados para $f_{Aptidao}$ serão repartidas entre dois grupos: fontes ricas e fontes pobres. Depois, serão enviadas as abelhas seguidoras para as melhores fontes de alimento encontradas pelas operárias e determinar as quantidades de néctar $fit(A_i)$ coletadas por cada uma. A seguir, calcular o valor de probabilidade (P) das fontes que serão escolhidas pelas abelhas operárias. Interrompe-se o processo de exploração das fontes abandonadas pelas abelhas (fontes pobres). Envie as escoteiras, aleatoriamente, para a área de busca para descobrir novas fontes de alimento na vizinhança. O pseudocódigo de otimização baseado no algoritmo ABC é mostrado no Algoritmo 1, este a sua vez faz referência à função chamada “ $f_{Aptidao}$ ” cujo pseudocódigo é descrito no Algoritmo 2.

Resultados: A busca por novos conjuntos S e exploração de vizinhança continuará até que o conjunto S obtido como solução final (melhor conjunto de parâmetros e pesos) seja satisfatório para o problema em questão ou tenha atingido a quantidade máxima de ciclos do algoritmo ($iter > maxIter$), ou seja, se durante $maxIter$ tentativas as fontes de alimento não melhorarem então as abelhas escoteiras devem abandonar suas fontes atuais e buscar aleatoriamente novas fontes de alimento.

4.2.6 Prognóstico de RUL

Esta abordagem de prognóstico de RUL é baseado em dados históricos de MC. O processo de prognóstico é realizado por uma ESN, onde os seus parâmetros são definidos pelo algoritmo ABC. Após o processo de treinamento, a ESN, com os parâmetros ajustados e com os pesos treinados, estará em capacidade de realizar prognóstico de RUL. O prognóstico de RUL é realizado processando o conjunto de dados de teste. O resultado de prognóstico da abordagem ESN-ABC é comparado com a RUL real do conjunto de dados de teste. A comparação com os resultados alcançados por outros pesquisadores é realizada através de métricas de prognóstico mostrada no Apêndice A. Estas métricas resultam das equações matemáticas que tem como variáveis de entrada a RUL estimada e a RUL real.

Algoritmo 1: Pseudo-código para o algoritmo ESN-ABC

Entrada: SN, S, f, [x_{\min} ; x_{\max}], maxTrial, maxIter

Saída: Melhor solução S_g : parâmetros e pesos da ESN

01: **Início**

02: Gerar as posições aleatórias para as SN fontes de alimento (Equação 2.1) e determinar aptidões $f(S)$

03: **Repita**

04: // FASE ABELHAS OPERÁRIAS

05: Para cada solução i determinar um vizinho k e dimensão j

06: Criar uma nova solução usando a Equação 2.2

07: Calcular os valores de aptidão $f(S)$

08: Atualizar posições se $f(S_i)$ melhora o valor anterior

09: //FASE ABELHAS SEGUIDORAS

10: Calcular o vetor de probabilidades p_i usando Equação 2.4

11: **Para** $i = 1 : SN$ **faça**

12: **Se** $\text{rand}() > p_i$ **então**

13: Determinar um vizinho k e dimensão j

14: Criar uma nova solução S_i usando a Equação 2.2

15: Calcular o valor de aptidão $f(S_i)$

16: Atualizar a posição se $f(S_i)$ melhora o valor anterior

17: **Fim Se**

18: **Fim Para**

19: //FASE ABELHAS ESCOTEIRAS

20: Determine as soluções abandonadas e envie as abelhas escoteiras para buscar novas fontes de alimento

21: Atualize a melhor solução S_g segundo as aptidões

22: $\text{iter} = \text{iter} + 1$

23: **Até** $\text{iter} \leq \text{maxIter}$

24: **Fim**

Algoritmo 2: Pseudo-código da função *fAptidao*

Entrada: dTreinamento, dTeste, ParametrosESN

Saída: Valor da função de aptidão: f

1: **Início**

2: Carregar o conjunto de dados de Treinamento

3: Carregar o conjunto de dados de Validação

4: Criar a rede ESN com os parametrosESN

5: Treinar pesos da camada linear de saída (*readout*)

6: Calcular NRMSE do conjunto de treinamento (Equação 3.7)

7: Calcular NRMSE do conjunto de validação (Equação 3.7)

8: Calcular o valor da função de aptidão (Equação 4.2)

9: **Fim**

4.3 MODELAGEM FUNCIONAL IDEF0

O método IDEF0 é utilizado para criar uma descrição clara e detalhada de um sistema em uma forma gráfica estruturada (Kim *et al.*, 2002). Este método possui como elemento básico um retângulo para definir cada atividade ou processo. Uma atividade é mostrada na Figura 4.3, as quatro setas ao redor do retângulo são:

- **Entradas:** As matérias primas ou dados que são transformadas durante o processo;
- **Controles:** Direção de funcionamento do processo, como e quando uma entrada deve ser processada e executada;
- **Mecanismos:** O necessário para que a atividade ocorra, tais como pessoas, ferramentas, máquinas ou equipamentos, software, etc.;
- **Saídas:** Resultados de um processo que são transmitidas para outro processo ou utilizadas como resultado de todo o processo.

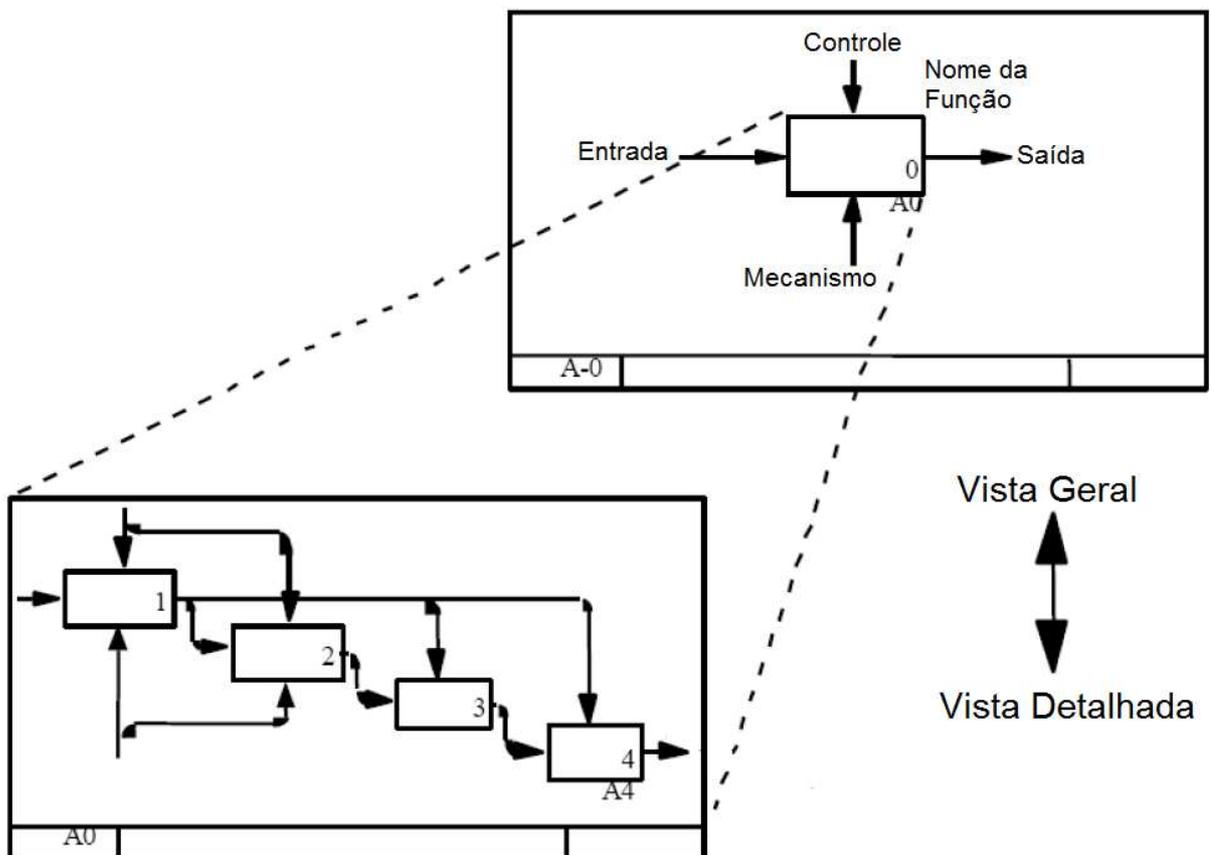


Figura 4.3- Estrutura de uma atividade ou processo e seus detalhes.

A abordagem desenvolvida nesta tese foi modelada através da metodologia IDEF0. A Figura 4.4 representa o nível 0 do modelo IDEF0 com todas as entradas, controles, mecanismos, saídas e funcionalidades que descreve a abordagem do sistema de prognóstico de vida útil remanescente. Como entradas temos dados de MC, dados de Falha e dados de operação. Os mecanismos são: a rede ESN, o algoritmo de otimização ABC, a ferramenta computacional Matlab, o banco de dados onde estão os dados históricos, e as equações para calcular as métricas de prognóstico. As saídas são: a RUL estimada e as métricas de prognóstico.

Na Figura 4.5 é apresentado o desdobramento do sistema em três camadas associadas aos módulos do sistema de prognóstico, sendo as atividades:

1. Aquisição de Dados;
2. Otimização e Treinamento;
3. Prognóstico de Falhas.

A atividade Aquisição de Dados tem como entrada dados de falhas, de monitoração de condição, e de operação; e como saídas dados de treinamento, validação e teste, que servem como entradas para a Otimização e Treinamento. A atividade Otimização e Treinamento tem como saída os parâmetros, pesos da RNA, que serão usados como entrada no processo de Prognóstico de Falhas. A atividade Prognóstico de Falhas possui como saídas a RUL estimada e as métricas de prognóstico. A seguir são apresentados os diagramas IDEF0 associados às três atividades descritas anteriormente.

4.3.1 Atividade Aquisição de Dados

O modelo IDEF0 da atividade Aquisição de Dados é apresentado na Figura 4.6. Esta atividade é constituída por duas atividades, que permite descrever a extração e seleção de características e a separação dos dados em conjunto de dados de treinamento, validação e teste:

1. Extração e Seleção de Características: Extrai e seleciona as características sejam apropriadas para a realização de prognóstico;
2. Separação de Dados: Segundo o valor de porcentagem definida, separa os dados em conjuntos de treinamento, validação e teste.

4.3.2 Atividade Otimização e Treinamento

A Figura 4.7 apresenta a atividade Otimização e Treinamento, e é dividida em quatro atividades, permitindo desta forma descrever o processo de otimização dos parâmetros da ESN e a determinação dos pesos:

1. Algoritmo de Otimização: Recebe como entrada do processo iterativo, a função objetivo formado pela NRMSE de treinamento e pela NRMSE de validação. Através de um processo iterativo visando a minimização da função objetivo, esta atividade fornecerá como saídas os parâmetros definidos e pesos treinados da ESN;
2. Treinamento da ESN: Com os parâmetros definidos e o conjunto de dados de treinamento fornece o NRMSE de treinamento;
3. Validação da ESN: Com a ESN treinada e o conjunto de dados de validação, fornecerá como saída o valor de NRMSE de validação;
4. Função de Aptidão: definida como uma função que depende do NRMSE de treinamento e o NRMSE de validação;
5. O processo volta ao passo 1 fechando o laço de iteração.

4.3.3 Atividade Prognóstico de Falhas

Esta atividade é dividida em duas atividades e é apresentada na Figura 4.8. Através desta atividade descreve-se o processo de prognóstico de RUL e o cálculo das métricas:

1. Prognóstico de RUL: Tendo como entradas os pesos do reservatório de dinâmicas da ESN, e o conjunto de dados de teste, o processamento de informações é realizada através da ESN cujos parâmetros foram definidos e os pesos treinamos pelo algoritmo ABC na etapa otimização e treinamento. A estimação de RUL é realizada para cada máquina, equipamento ou sistema.
2. Avaliação de Desempenho: Considerando como entradas a RUL real coletado do conjunto de teste e a RUL estimada fornecida pela atividade Prognostico de RUL, esta atividade calcula várias métricas de prognóstico úteis para avaliar o desempenho da abordagem desenvolvida, o que permitirá realizar uma comparação quantitativa com os resultados de outras abordagens.

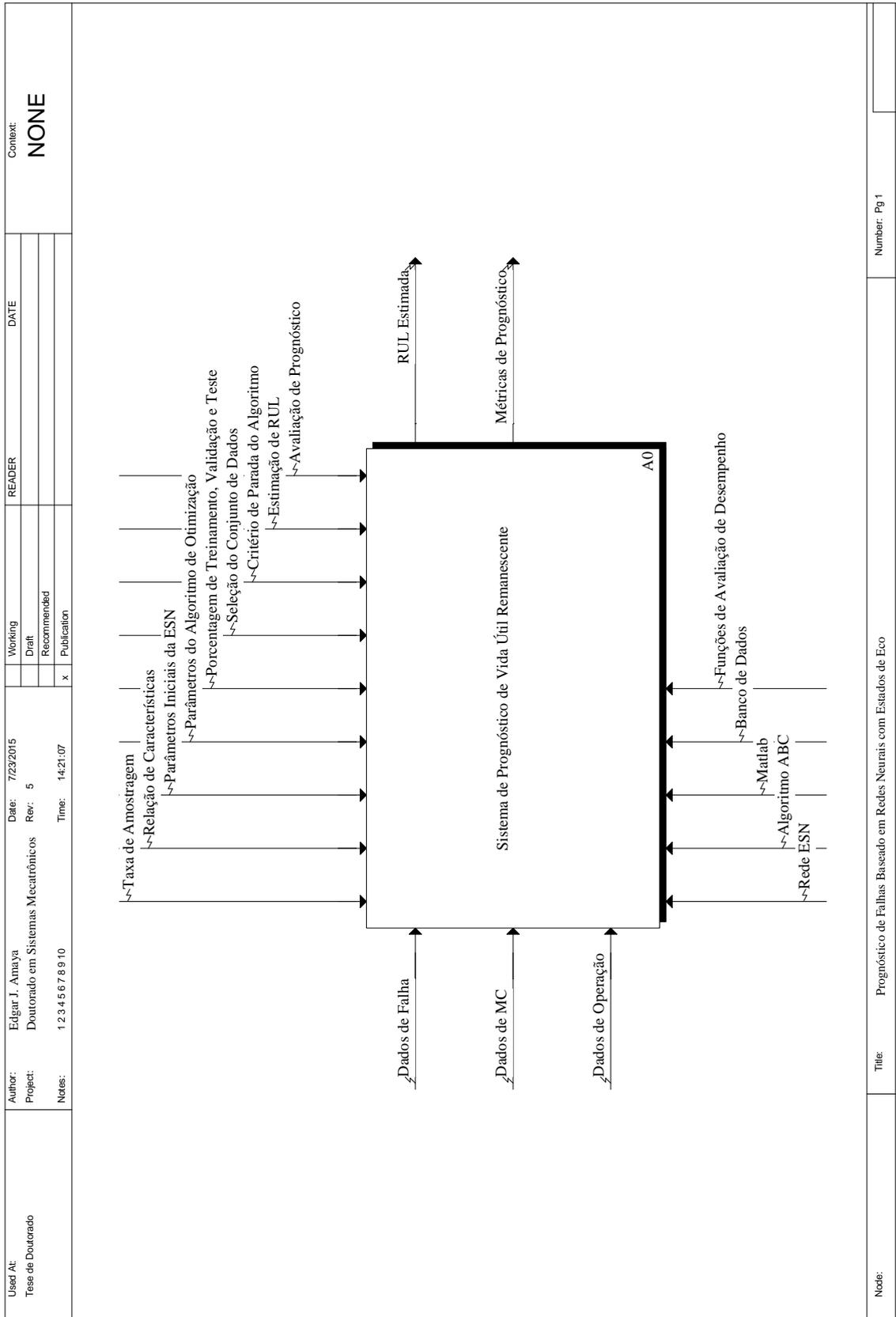


Figura 4.4- Diagrama IDEF0: Sistema de Prognóstico de RUL nível A0.

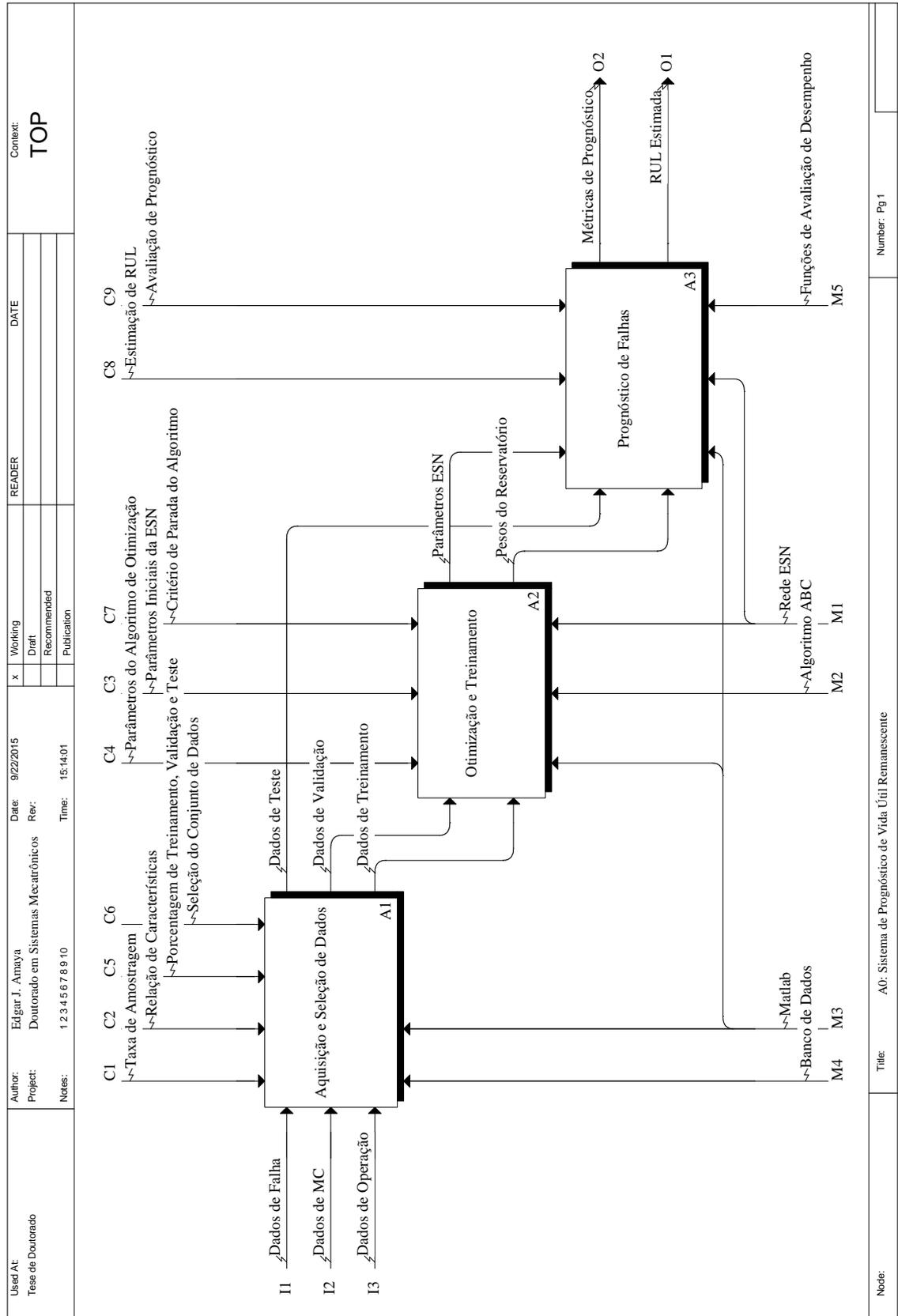


Figura 4.5- Diagrama IDEF0: Principais Atividades do Sistema de Prognóstico de RUL.

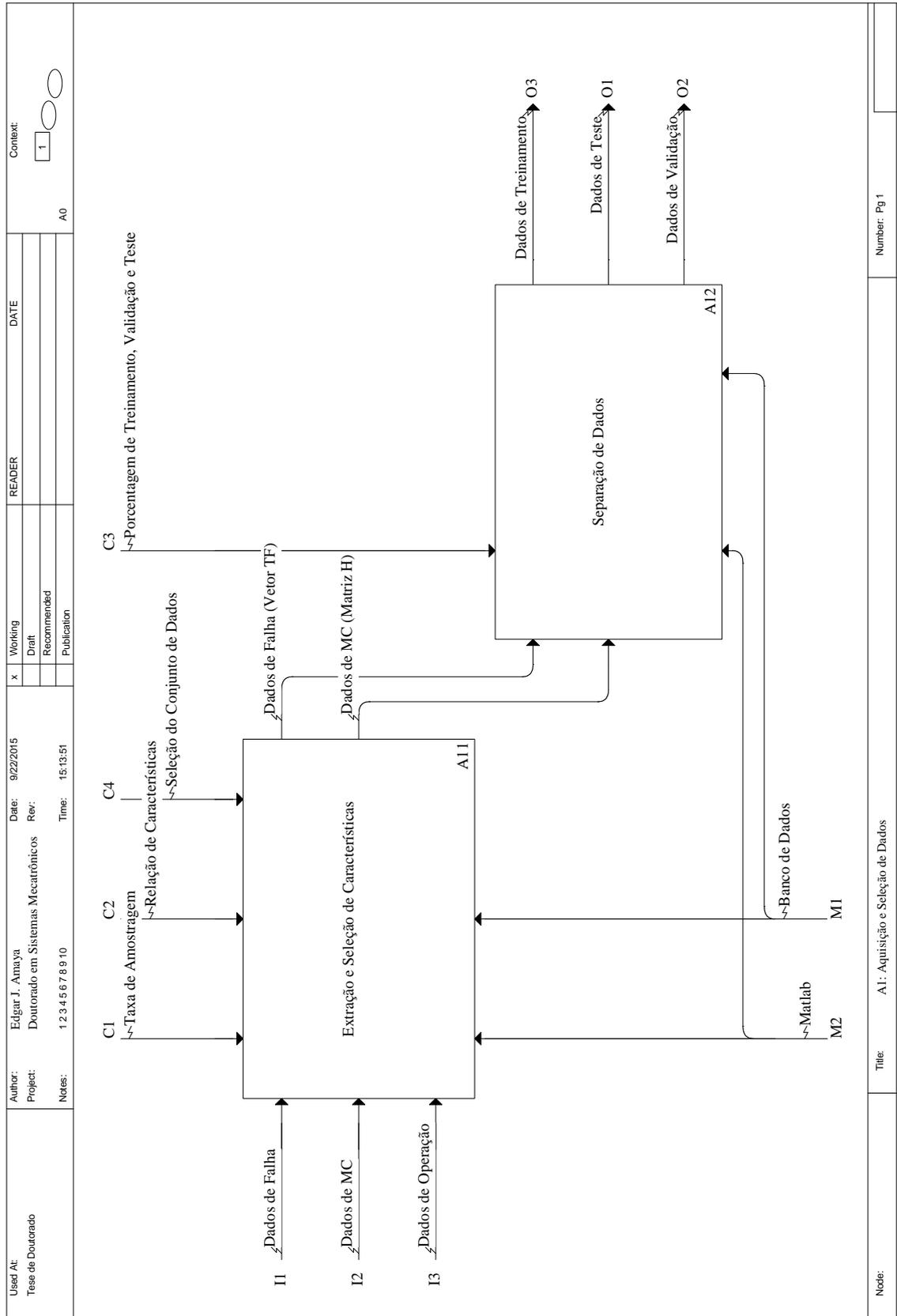


Figura 4.6- Diagrama IDEF0: Atividades Aquisição de Dados.

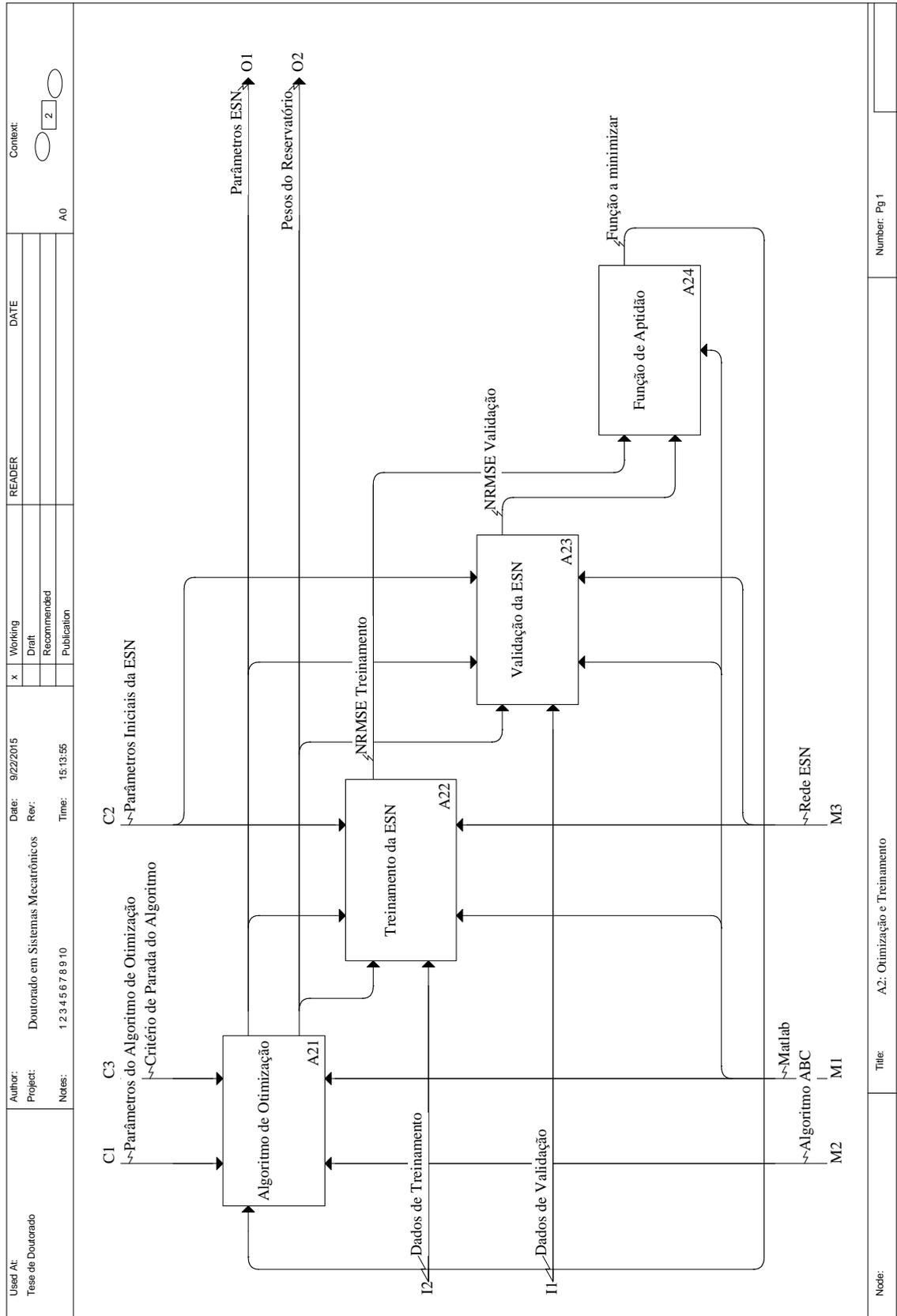


Figura 4.7- Diagrama IDEF0: Atividades Otimização e Treinamento.

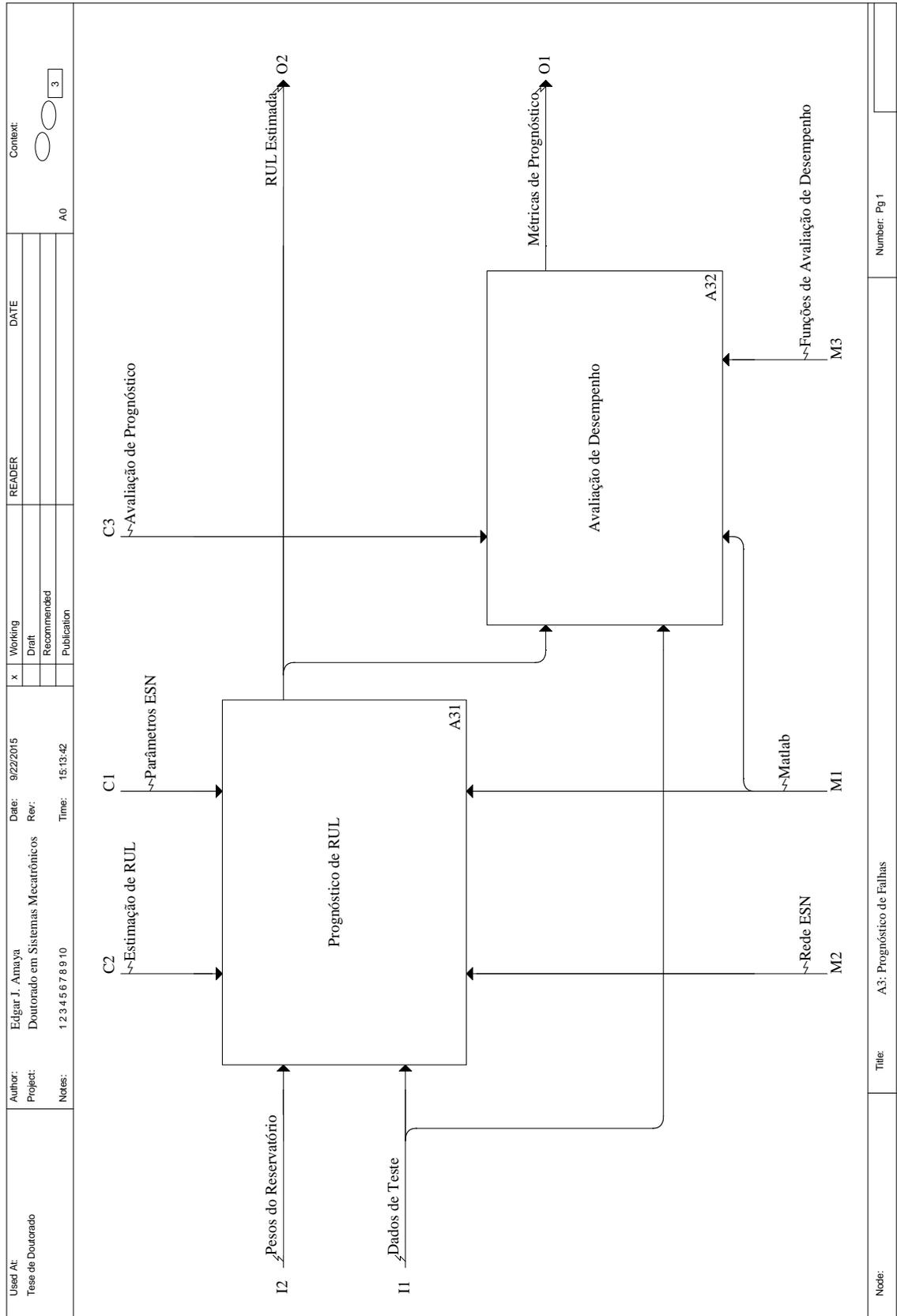


Figura 4.8- Diagrama IDEF0: Atividades Prognóstico de Falhas.

4.4 SÍNTESE DO CAPÍTULO

Neste capítulo foi apresentada uma abordagem para prognóstico de RUL baseada em ESN caracterizada pelo uso de uma estrutura recorrente onde a camada intermediária, conhecida como reservatório, é composta por pesos definidos aleatoriamente. Esta abordagem foi desenvolvida com o intuito de otimizar os parâmetros e pesos da ESN através do algoritmo ABC. Com esta abordagem espera-se ajustar os pesos de uma forma automatizada. A função de aptidão do algoritmo de otimização será baseada na NRMSE de treinamento e NRMSE de validação com o objetivo de evitar o *overfitting*. A etapa de prognóstico de RUL tem como entrada o conjunto de dados de teste e o processamento através da ESN treinada, resultando na saída da rede os valores estimados de RUL. Para realizar um estudo comparativo quantitativo são calculadas as métricas de prognóstico apresentadas no Apêndice A.

5 ESTUDO DE CASO: PROGNÓSTICO DE RUL DE TURBINAS TURBOFAN

Neste capítulo apresenta-se o estudo de caso da abordagem ESN-ABC usando dados históricos de monitoração de condição. Esta abordagem baseada em dados requer na etapa de treinamento ou modelagem um grande conjunto de dados. Em aplicações reais essa quantidade de dados na maioria das vezes não está disponível, não está em um formato adequado ou apresentam inconsistências. Para a realização do estudo de caso da abordagem desenvolvida optou-se por utilizar dados de degradação simulados de turbinas turbofan, disponibilizados no repositório de prognóstico da NASA (Saxena e Goebel, 2008). Devido a que estes dados foram amplamente utilizados para validar algoritmos de prognóstico, será realizada a comparação dos resultados obtidos neste trabalho com os resultados dos outros algoritmos. Para a realização dos experimentos desenvolveu-se uma ferramenta com interface gráfica apresentada no Apêndice B. Os algoritmos desenvolvidos são baseados na ToolboxESN (2015) e ToolboxABC (2015). O código fonte dos algoritmos desenvolvidos e vídeo tutorial encontra-se disponível no link: <https://sourceforge.net/projects/esn-abc/>.

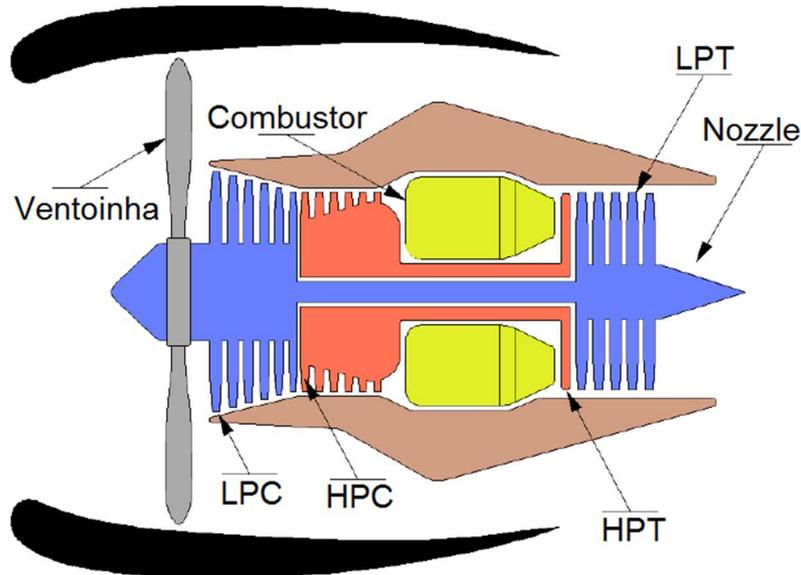
5.1 DESCRIÇÃO DOS DADOS

O conjunto de dados do repositório de prognóstico é resultado de uma série de simulações de funcionamento até a falha (RTF – *Run To Failure*) realizado por Saxena *et al.* (2008a). O modelo usado nas simulações foi implementado no C-MAPSS desenvolvido pela NASA (Frederick *et al.*, 2007).

A plataforma C-MAPSS é capaz de simular modelos de turbinas de 90.000 lb de impulso sob diferentes condições de operação: altitude, número Mach, e temperatura ao nível do mar. Além disso, o impulso da turbina pode ser controlado através do valor do Ângulo Resolver do Regulador de Pressão (TRA – *Throttle Resolver Angle*), este é considerado como outra condição de operação.

Através da variação dos parâmetros de entrada do C-MAPSS é possível simular os efeitos da degradação e defeitos em qualquer um dos cinco componentes rotativos da turbina, que incluem: ventoinha, compressor de baixa pressão (LPC – *Low Pressure Compressor*), compressor de alta pressão (HPC – *High Pressure Compressor*), turbina de alta pressão (HPT – *High Pressure Turbine*), e turbina de baixa pressão (LPT – *Low Pressure Turbine*).

Os componentes citados anteriormente podem ser visualizados na Figura 5.1, que mostra o diagrama e os principais elementos da turbina. Adicionalmente na Figura 5.2 mostra-se o fluxograma que corresponde à simulação e seus módulos.



Compressor de baixa pressão (LPC – *Low Pressure Compressor*)
 Compressor de alta pressão (HPC – *High Pressure Compressor*)
 Turbina de alta pressão (HPT – *High Pressure Turbine*)
 Turbina de baixa pressão (LPT – *Low Pressure Turbine*)

Figura 5.1- Diagrama simplificado de uma turbina turbofan (Saxena e Goebel, 2008).

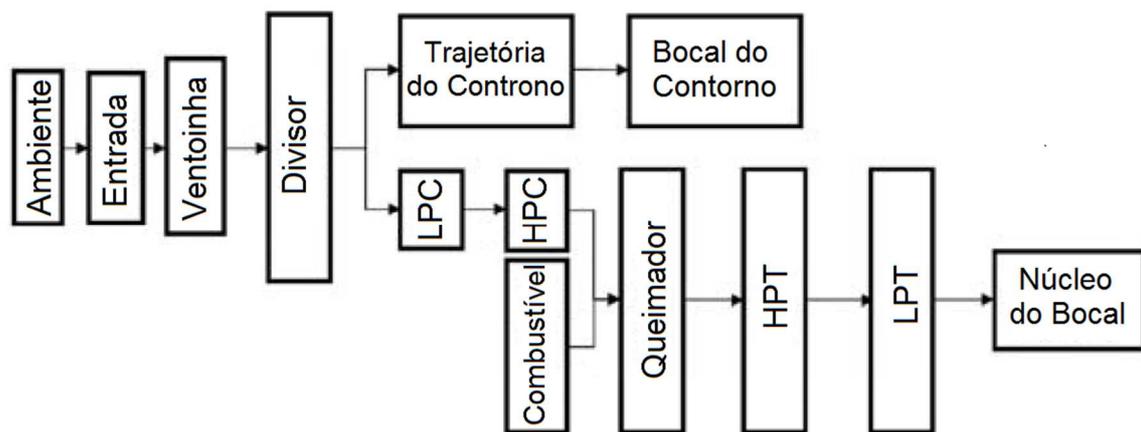


Figura 5.2- Módulos e conexões do modelo de simulação (Saxena e Goebel, 2008).

5.2 ESTRUTURA DOS DADOS

O repositório de prognóstico da NASA consiste de quatro conjuntos de dados, cujas características de simulação são mostradas na Tabela 5.1. Entre as características tem-se: o

número de modos de falha, os modos de operação, turbinas para treinamento, e turbinas para teste.

Tabela 5.1- Características dos quatro conjuntos de dados.

Quantidade	Conjuntos de dados			
	Nº 1	Nº 2	Nº 3	Nº 4
Modos de falha	1	1	2	2
Modos de operação	1	6	1	6
Turbinas para treinamento	100	260	100	249
Turbinas para teste	100	259	100	248

Entre as grandezas monitoradas tem-se: temperatura, pressão, velocidade, fluxo, entalpia, etc. A descrição das grandezas é mostrada na Tabela 5.2, detalhando o número do sensor, tag, descrição da variável e a unidade de medida. Cada conjunto de dados, tanto de treinamento, quanto de teste tem uma estrutura como a apresentada na Tabela 5.3, a primeira coluna é o número da turbina, para cada ciclo de funcionamento é coletado 3 grandezas que representam os modos de operação: (1) altitude sobre o nível do mar até 40.000 pés, (2) número Mach entre 0 à 0,90, e (3) temperatura ao nível do mar entre -60 à 103°F. Além disso, mostram-se as 21 variáveis de monitoramento de condição a partir dos sensores, cujas unidades de medição são mostradas na Tabela 5.2.

5.3 SELEÇÃO DO SUBCONJUNTO DE SENSORES

Para o desenvolvimento do modelo de prognóstico foi usado o conjunto de dados N°1 porque apresenta maior quantidade de publicações em comparação aos outros conjuntos de dados como mostrado na Tabela 5.4. Os dados do arquivo “train_FD001.txt” foram usados para treinamento, o 80% dos dados do arquivo “test_FD001.txt” para teste, e os restantes 20% para validação. Este conjunto de dados contém informações de 100 turbinas, consideradas do mesmo tipo. Cada registro do conjunto de dados corresponde a um ciclo de operação e estão contaminados com ruído. As turbinas têm diferenças na fabricação e começam com diferente desgaste.

No início todas as turbinas estão em operação normal, mas devido a diferentes fatores se produz a deterioração à medida que aumenta a operação. A degradação aumenta até a ocorrência de uma falha. As falhas são causadas pela degradação do compressor de alta pressão (HPC – *High Pessure Compressor*). Os valores do sensor 2 dos dados de treinamento

do conjunto N° 1 é mostrado no lado esquerdo da Figura 5.3. Os dados estão contaminados com ruído e precisam ser filtrados antes de serem introduzidos no modelo de prognóstico. A lado direito da Figura 5.3 mostra a distribuição de vida útil das 100 turbinas dos dados de treinamento.

Tabela 5.2- Descrição das variáveis monitoradas.

Sensor	Tag	Descrição	Unidade
1	T2	Temperatura na entrada da ventoinha	°R
2	T24	Temperatura na saída do LPC	°R
3	T30	Temperatura na saída do HPC	°R
4	T50	Temperatura na saída do LPT	°R
5	P2	Pressão na entrada da ventoinha	psia
6	P15	Pressão na entrada do duto by-pass	psia
7	P30	Pressão na saída do HPC	psia
8	Nf	Velocidade da ventoinha	rpm
9	Nc	Velocidade do núcleo	rpm
10	Epr	Taxa de pressão da turbina	-
11	Ps30	Pressão estática na saída do HPC	psia
12	Phi	Taxa de fluxo de combustível para Ps30	pps/psi
13	NRf	Velocidade corrigida do ventilador	rpm
14	NRc	Velocidade corrigida do núcleo	rpm
15	BPR	Taxa de by-pass	-
16	farB	Taxa de queima ar-combustível	-
17	htBleed	Entalpia de dreno	-
18	Nf_dmd	Velocidade demandada da ventoinha	rpm
19	PCNfR_dmd	Velocidade demandada corrigida da ventoinha	rpm
20	W31	Dreno do refrigerante HPT	lbm/s
21	W32	Dreno do refrigerante LPT	lbm/s

Tabela 5.3- Exemplo da estrutura dos dados da turbina 1.

Turbina N°	Ciclo	Modo de Operação 1	...	Modo de Operação 3	Sensor 1	...	Sensor 21
1	1	0,0070	...	100	518,67	...	23,4190
1	2	0,0019	...	100	518,67	...	23,4236
:	:	:	:	:	:	:	:
1	100	0,0009	...	100	518,67	...	22,9649

Tabela 5.4- Publicações para cada conjunto de dados.

Conjunto	Publicações
1	Siegel (2009), Ramasso (2009), Ramasso e Gouriveau (2010), Ramasso e Gouriveau (2013), Zhao e Willett (2011), El-Koujok <i>et al.</i> (2011), Liao e Sun (2011), Sun <i>et al.</i> (2012), Peng <i>et al.</i> (2012a), Javed <i>et al.</i> (2012), Serir <i>et al.</i> (2012), Gouriveau e Zerhouni (2012), Yu (2013), Ramasso <i>et al.</i> (2013), Liu <i>et al.</i> (2013), Lin <i>et al.</i> (2013), Javed <i>et al.</i> (2013), Ramasso e Denoeux (2013), Li <i>et al.</i> (2013b), Ishibashi e Nascimento Junior (2013), Gouriveau <i>et al.</i> (2013), Jianzhong <i>et al.</i> (2010), Ramasso (2014a), e Ramasso (2014b)
2	Zhao e Willett (2011), Peng <i>et al.</i> (2012b), Li <i>et al.</i> (2013b), Ramasso (2014a), e Ramasso (2014b)
3	Li <i>et al.</i> (2013b), Ramasso (2014a), e Ramasso (2014b)
4	Wang (2010), Li <i>et al.</i> (2013b), Ramasso (2014a), e Ramasso (2014b)

Depois de selecionar o conjunto de dados, uma opção é realizar o processamento usando completos treinamento/teste, dados de funcionamento até a falha no treinamento (“train_FD001.txt”), e suspensos para teste e validação (“test_FD001.txt”). A outra opção é utilizar unicamente dados de funcionamento até a falha particionando o conjunto de dados de treinamento (“train_FD001.txt”) em três subconjuntos: de treinamento e validação para estimação dos parâmetros do modelo, e teste para realizar o prognóstico de RUL.

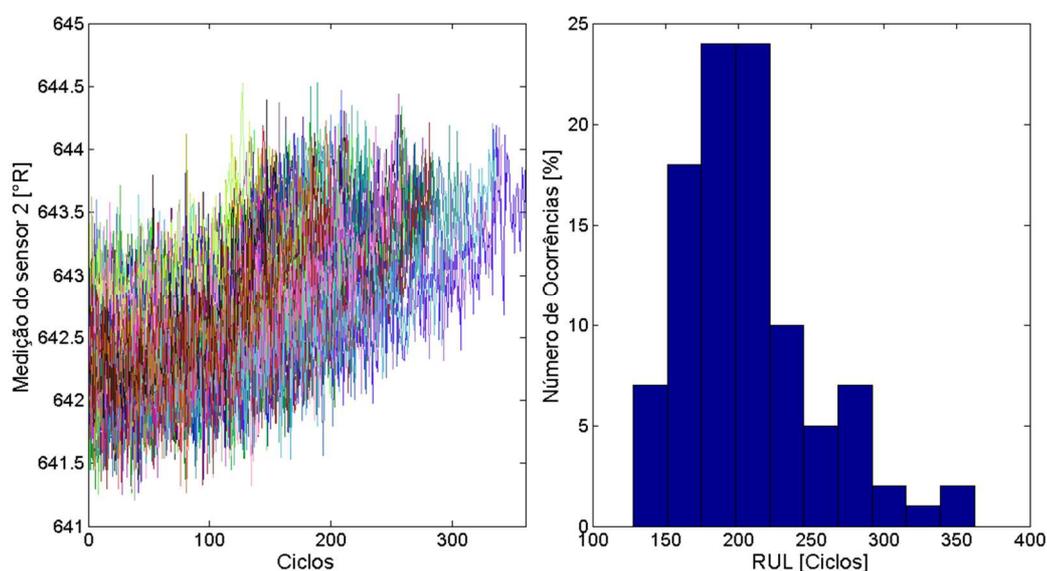


Figura 5.3- Medições do sensor 2 das 100 turbinas e distribuição de vida útil.

Nesta tese foram utilizados dados completos treinamento/teste para validar a abordagem desenvolvida. Os valores reais de RUL foi adquirido do arquivo “rul_FD001.txt” para comparação com os valores de RUL estimados por uma ESN e pela abordagem ESN-ABC.

Essa opção foi utilizada em cinco publicações, todas com dados do conjunto 1 como mostrado na Tabela 5.5.

Tabela 5.5- Publicações baseados com dados completos treinamento/teste.

Conjunto	Publicações
1	Peng <i>et al.</i> (2012a), Ramasso <i>et al.</i> (2013), Liu <i>et al.</i> (2013), Ramasso (2014a), e Ramasso (2014b)
2	Ramasso (2014a) e Ramasso (2014b)
3	Ramasso (2014a) e Ramasso (2014b)
4	Ramasso (2014a) e Ramasso (2014b)

A seleção das variáveis é baseada em previsibilidade, isto é, que apresentem tendências (variáveis que aumentam ou diminuem em função do tempo de operação de uma máquina) para a realização de prognóstico. Além disso, a seleção de variáveis reduz a dimensionalidade do problema. Algumas regras heurísticas são propostas por Ramasso (2014a), e aplicado aos quatro conjuntos de dados C-MAPSS. Alguns outros métodos são baseados na Análise de Componentes Principais (PCA – *Principal Component Analysis*) (Wang *et al.*, 2008) e outros procedimentos de seleção de sensores publicados por Heimes (2008), Peel (2008), e Ramasso (2014b).

Do total de variáveis que contém o conjunto de dados, diferentes combinações podem ser usadas (dentro as 21 variáveis). Javed *et al.* (2013) das 21 variáveis monitoradas, unicamente oito foram consideradas para os seus experimentos, estas são as medidas dos sensores {2, 3, 4, 8, 11, 13, 15, 17}. Por outro lado, Sarkar *et al.*, 2011 usaram o subconjunto formado pelas seguintes seis características {7, 8, 12, 16, 17, 20}. A partir da pesquisa realizada, o subconjunto mais usado é dos sensores {7, 8, 9, 12, 16, 17, 20}. Em todos os casos, o processo de seleção envolve os sensores chave para degradação de turbina.

Neste trabalho foi realizada graficamente a análise de previsibilidade, para isto foi realizada a plotagem de todas variáveis, e como exemplo mostra-se na Figura 5.4 as grandezas monitoradas da turbina 7, onde o termo “velocidade corrigida” significa a velocidade à qual rodará um componente, se a temperatura de entrada corresponde às condições ambientais ao nível do mar, em um dia padrão (ou seja, 15°C).

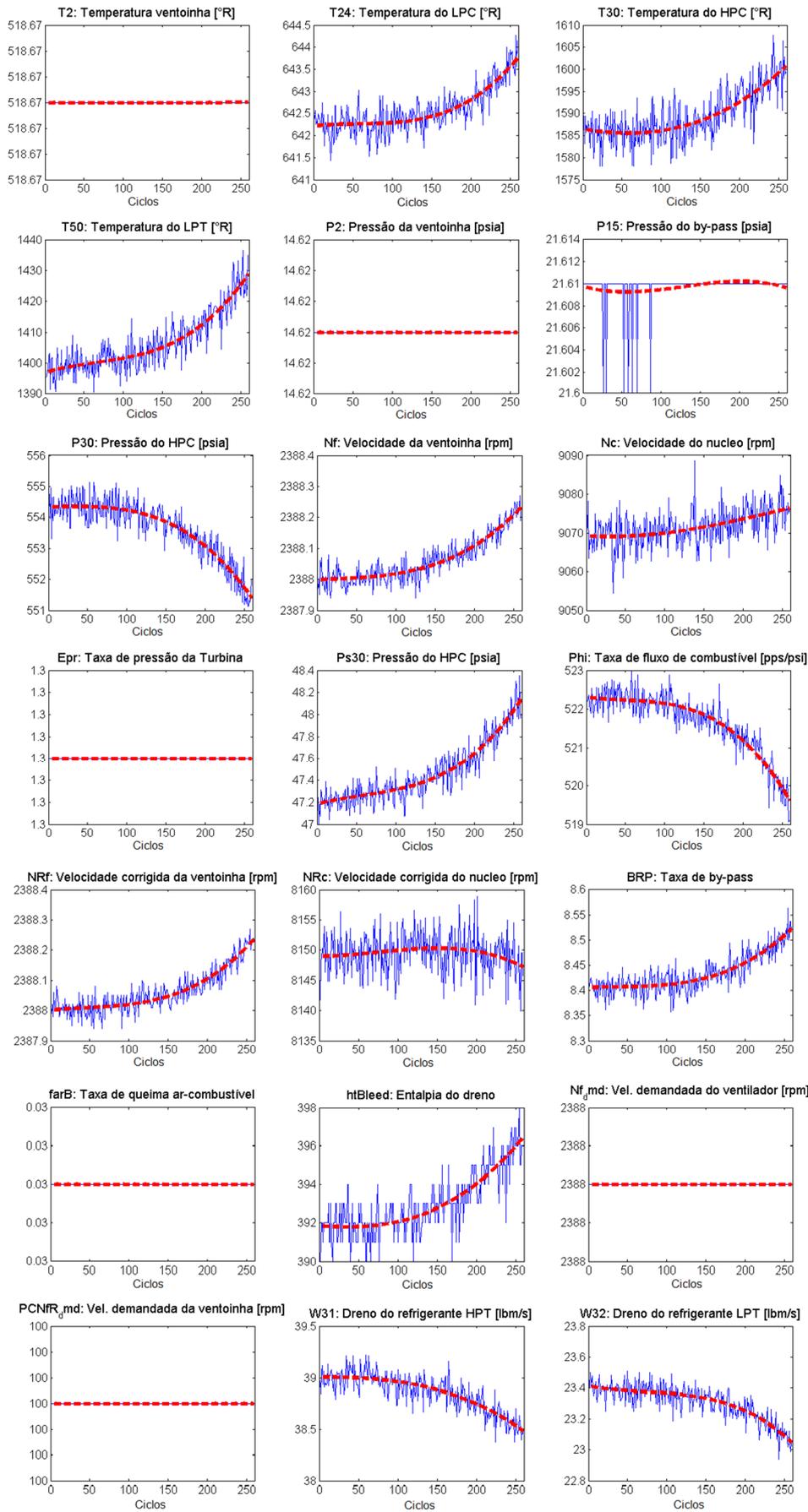


Figura 5.4- Conjunto de todas variáveis monitoradas para a turbina 7.

Foram selecionadas apenas as variáveis que apresentam tendência, quatorze características de degradação dos sensores {2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, 21}. Na Figura 5.5 são mostradas as quatorze variáveis com ruído e com atenuação de ruído, assim como também a curva da RUL da turbina 7.

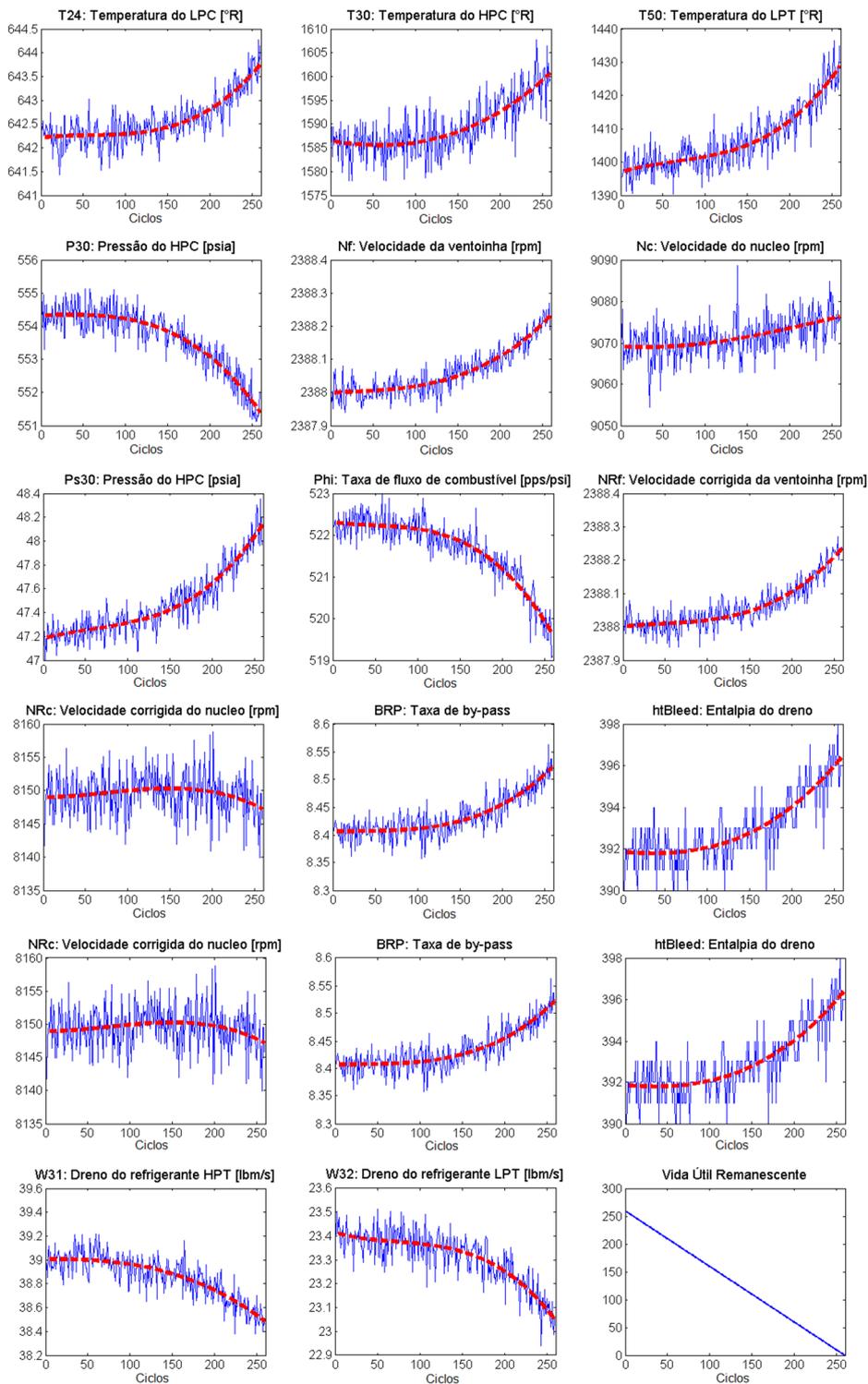


Figura 5.5- subconjunto de variáveis selecionadas e a RUL da turbina 7.

5.4 RESULTADOS DE PROGNÓSTICO DE RUL

Nesta seção descrevem-se os algoritmos usados, os parâmetros de configuração, a topologia, e os resultados de prognóstico obtidos por uma ESN clássica e pela abordagem híbrida ESN-ABC desenvolvida neste trabalho. Os algoritmos foram desenvolvidos em *Matlab* baseados no ToolboxABC (2015) e no ToolboxESN (2015) para o desenvolvimento do algoritmo de otimização ABC e da rede ESN, respectivamente. A etapa de treinamento é realizada usando os dados do arquivo “train_FD001.txt”. Para validação 20% conjunto de dados do arquivo “test_FD001.txt”. Na etapa de teste é utilizado 80% conjunto de dados do arquivo “test_FD001.txt” obtendo na saída a RUL estimada. Os valores de RUL obtidos na saída são comparados com as RUL reais que se encontram no arquivo “rul_FD001.txt”.

5.4.1 ESN clássica

O prognóstico de RUL do conjunto de turbinas é realizado através de uma ESN clássica. Considera-se como entradas grandezas de MC, e como saída uma variável contínua que representa o número de ciclos restantes até a ocorrência de uma falha. Esta variável representa a RUL e é considerada como a saída desejada durante o treinamento da ESN. Considerando que o final da sequência de dados representa o último ciclo antes da ocorrência da falha de uma turbina específica. A variável RUL é calculada como o número de ciclos que falta antes do final da sequência de dados. Os resultados obtidos com a ESN clássica serão utilizados como referência de comparação para a abordagem híbrida ESN-ABC. O experimento da definição de parâmetros de treinamento e teste da ESN clássica é controlado pelo Algoritmo 3.

A guia prática para aplicação de ESN publicada por Lukosevicius (2012) é utilizada como referência para sintonizar os parâmetros da ESN mostrados na Tabela 5.6. Esses parâmetros necessitam ser alterados simultaneamente considerando a faixa e os passos de variação dos valores. O parâmetro N cujos valores estão na faixa de 40 à 300, variando com passos de 5, pode assumir $(300-40)/5+1 = 53$ valores, executando o mesmo cálculo para os demais parâmetros, no total serão necessários $53*20*20*40*20*40 = 678'400.000$ experimentos, tornando grande o espaço de busca. No experimento optou-se por mudar duas variáveis mantendo as demais fixas, começando com o número de unidades do reservatório (N), o raio espectral (sr), mantendo fixos a escala e o deslocamento de entrada (IS , IF) e a escala e deslocamento de saída (OS , OF). Desta forma o número de experimentos resultaria

$53*20+20*40+20*40=2660$. Depois de vários experimentos, parâmetros com melhor desempenho são mostrados na última coluna da Tabela 5.6.

Algoritmo 3: Pseudo-código do Experimento ESN Clássica

Entrada: Dados de Treinamento, Dados de Teste, RUL real

Saída: RUL estimada, métricas de prognóstico

01: **Início**

02: Carregar o conjunto de dados de Treinamento;

03: Carregar o conjunto de dados de Teste;

04: Carregar o conjunto de dados de RUL real;

05: $n = 1$;

06: **Repita**

07: Definir Parâmetros ESN, mudar 2 e manter fixo o resto;

08: Gerar Pesos aleatórios;

09: Treinar a ESN;

10: Testar a ESN (Estimar a Vida Útil Remanescente);

11: Analisar desempenho (Métricas de Prognóstico);

12: **Até** ($n \leq n\text{Testes}$)

13: Salvar os Melhores Parâmetros, Pesos, RUL estimada e as métricas de prognóstico;

14: **Fim**

No entanto, os valores de W^{in} e W são gerados aleatoriamente usando distribuição randômica uniforme. Será realizado o escalonamento dos pesos W usando o raio espectral com o objetivo de garantir a ESP. Os pesos do *readout* (W^{out}) serão calculados usando o conjunto de treinamento e regressão linear. O experimento é repetido várias vezes mudando-se dois parâmetros e mantendo fixos os restantes, avaliando o desempenho da ESN para cada experimento.

Tabela 5.6- Ajuste de parâmetros da ESN clássica.

Parâmetros	Faixa	Passo	Valor
N	[40, 300]	5	150
sr	[0,01, 1]	0,05	0,5
IS	[0,01, 1]	0,05	0,05
IF	[-1, 1]	0,05	0,95
OS	[0,01, 1]	0,05	0,005
OF	[-1, 1]	0,05	-0,05

Durante a etapa de treinamento, como pode ser observado na Figura 5.6, a ESN conseguiu acompanhar a sequência de RUL das diferentes turbinas no conjunto de treinamento. A

Figura 5.7 mostra a sequência real e a sequência estimada pela ESN na etapa de teste. Na Figura 5.8 é mostrada graficamente a RUL real e a RUL estimada pela ESN clássica para as 100 turbinas do conjunto de dados de teste, assim como o erro de predição do 34% de turbinas encontra-se dentro de um limiar de falsos positivos e falsos negativos, +13 e -10 respectivamente.

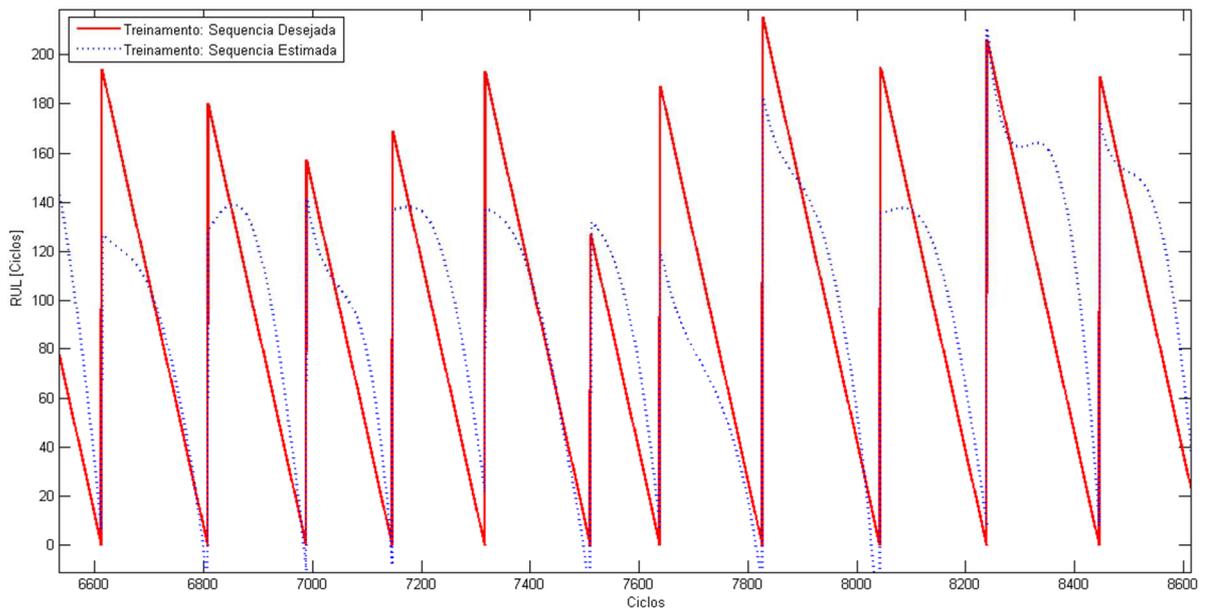


Figura 5.6- Sequência de treinamento usando ESN.

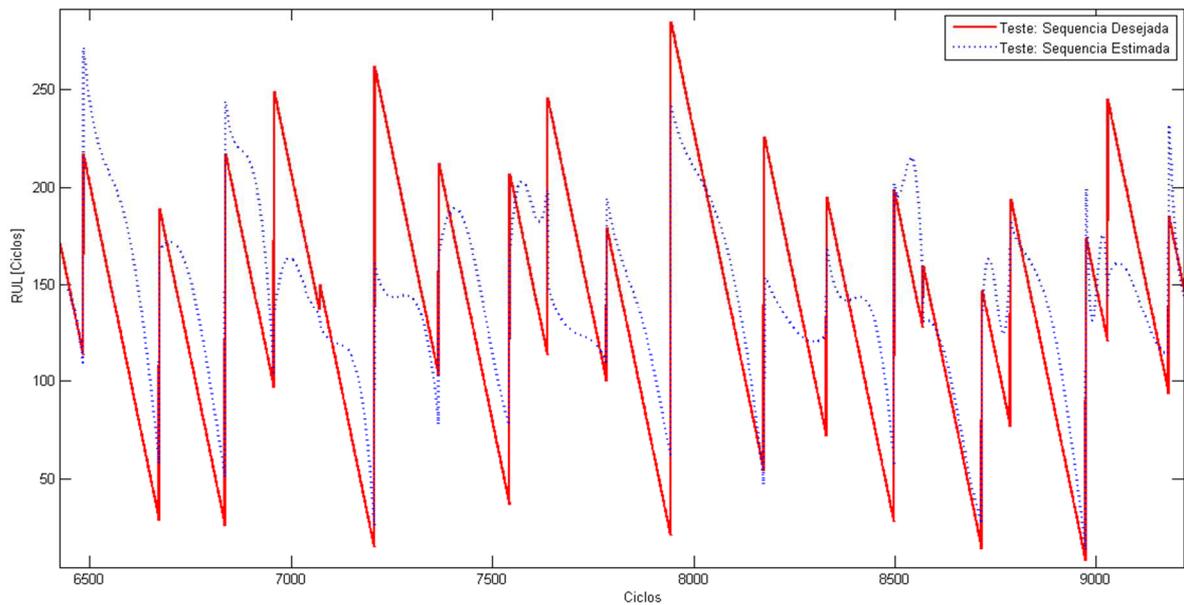


Figura 5.7- Sequência de teste usando ESN.

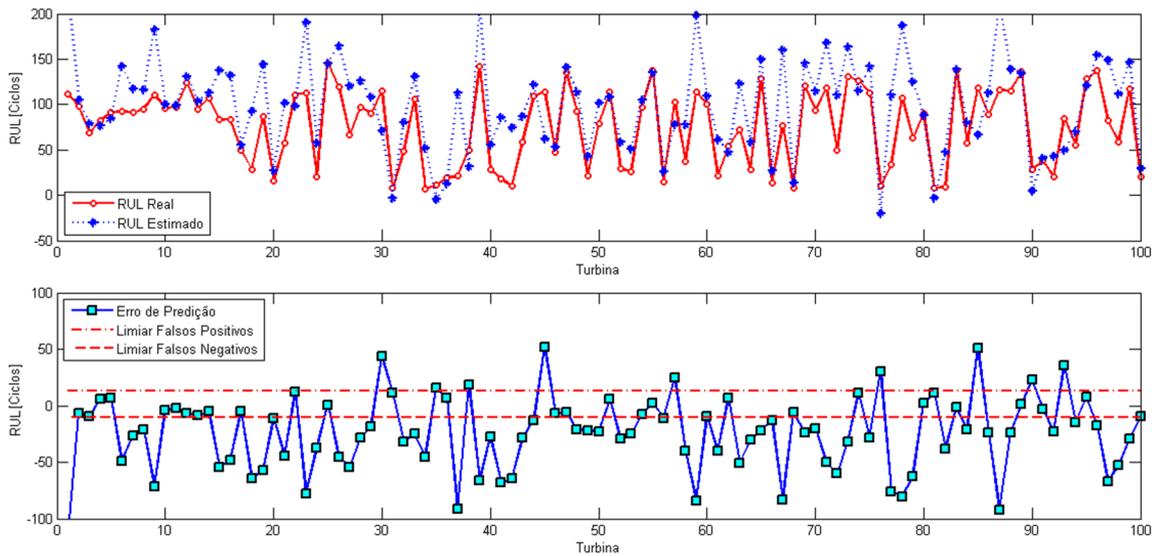


Figura 5.8- RUL estimada através da ESN.

5.4.2 ESN-ABC

O experimento da abordagem ESN-ABC para prognóstico de RUL baseado na rede ESN com parâmetros e pesos ajustados pelo algoritmo de otimização ABC é controlado pelo Algoritmo 4.

Algoritmo 4: Pseudo-código do Experimento ESN-ABC

Entrada: Dados de Treinamento, Dados de Teste, RUL real, Parâmetros ABC, Parâmetros ESN

Saída: RUL estimada, métricas de prognóstico

01: **Início**

02: Carregar o conjunto de dados de Treinamento;

03: Carregar o conjunto de dados de Teste e Validação;

04: Carregar o conjunto de dados de RUL real;

05: Inicializar Parâmetros ABC;

06: Inicializar Parâmetros e Pesos iniciais da ESN;

07: $n = 1$;

08: **Repita**

09: Treinamento da ESN;

10: Validação da ESN;

11: Calcular o valor da Função de Aptidão;

12: Executar o Algoritmo ABC;

13: Salvar os parâmetros e pesos calculados;

14: **Até** ($n \leq \text{maxIter}$)

15: Testar a ESN (Estimar a Vida Útil Remanescente);

16: Analisar desempenho (Métricas de Prognóstico);

17: **Fim**

Os pesos da camada de entrada W^{in} seguem uma distribuição randômica uniforme. Depois é inicializado o número de neurônios do reservatório, e mantendo fixo durante o experimento, e os demais parâmetros e os pesos do reservatório W foram inicializados com valores aleatórios. Depois, o algoritmo ABC inicia o processo iterativo de ajuste dos parâmetros e pesos do reservatório da ESN. A Equação 4.2 é considerada como função de aptidão do algoritmo ABC. Finalmente, usando o conjunto de treinamento são definidos os pesos do *readout*. Os resultados dos parâmetros ajustados são mostrados na Tabela 5.7.

Tabela 5.7- Parâmetros ajustados pela abordagem ESN-ABC.

Parâmetros	Faixa	Valor
N	[30, 200]	198
sr	[0,01, 1]	0,09
IS	[0,01, 1]	0,83
IF	[-1, 1]	0,38
OS	[0,01, 1]	0,14
OF	[-1, 1]	0,83

Nos experimentos comprovou-se que o processo de ajuste de parâmetros e pesos através do algoritmo ABC demanda um alto custo computacional (duração de 27h 21min para uma configuração de $N = 150$, $NP = 100$, $\max\text{Trial} = 50$, e $\max\text{Iter} = 100$). No entanto, o processo experimental realizado várias vezes de forma manual demandaria tempo muito maior e dependeria da experiência do especialista. Os resultados do processo manual não necessariamente refletiriam uma solução dos parâmetros e pesos otimizados.

Durante o ajuste dos parâmetros e pesos com o auxílio do algoritmo de otimização ABC, a ESN conseguiu acompanhar a sequência de RUL das diferentes turbinas no conjunto de dados treinamento como pode ser observado na Figura 5.9. Depois deste processo, a rede ESN estará em capacidade de processar novos dados. Os novos dados chamados dados de teste são introduzidos na camada de entrada da ESN para testar a capacidade de prognóstico da rede. A Figura 5.10 mostra a sequência real e a sequência estimada pela ESN na etapa de prognóstico.

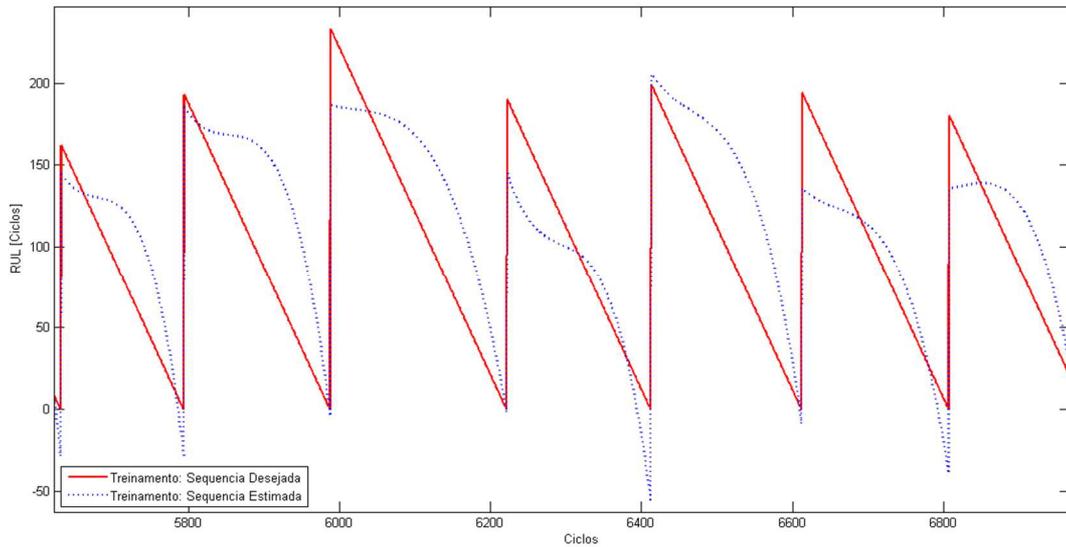


Figura 5.9- Sequência de treinamento usando ESN-ABC.

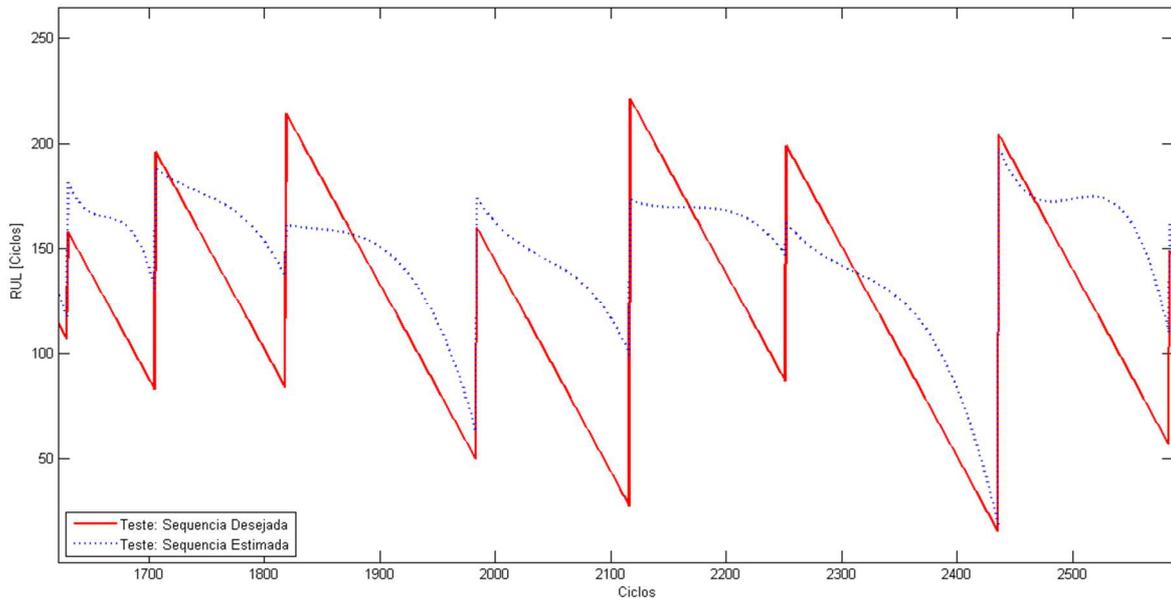


Figura 5.10- Sequência de teste usando ESN-ABC.

Os resultados obtidos mostrados na Figura 5.11 confrontam os valores de RUL estimados e os valores de RUL verdadeiros, assim como o erro de predição das 100 turbinas do conjunto de dados de teste. Observa-se que a abordagem ESN-ABC consegue obter bons resultados na estimação de RUL, considerando a métrica de exatidão MSE. Observando o gráfico do erro de predição observa-se que se encontram dentro de um limiar de falsos positivos e falsos negativos, +13 e -10 respectivamente. Em prognóstico de falhas é mais conveniente obter mais falsos positivos do que falsos negativos. Os falsos positivos indicam que o algoritmo consegue estimar uma RUL inferior à RUL real.

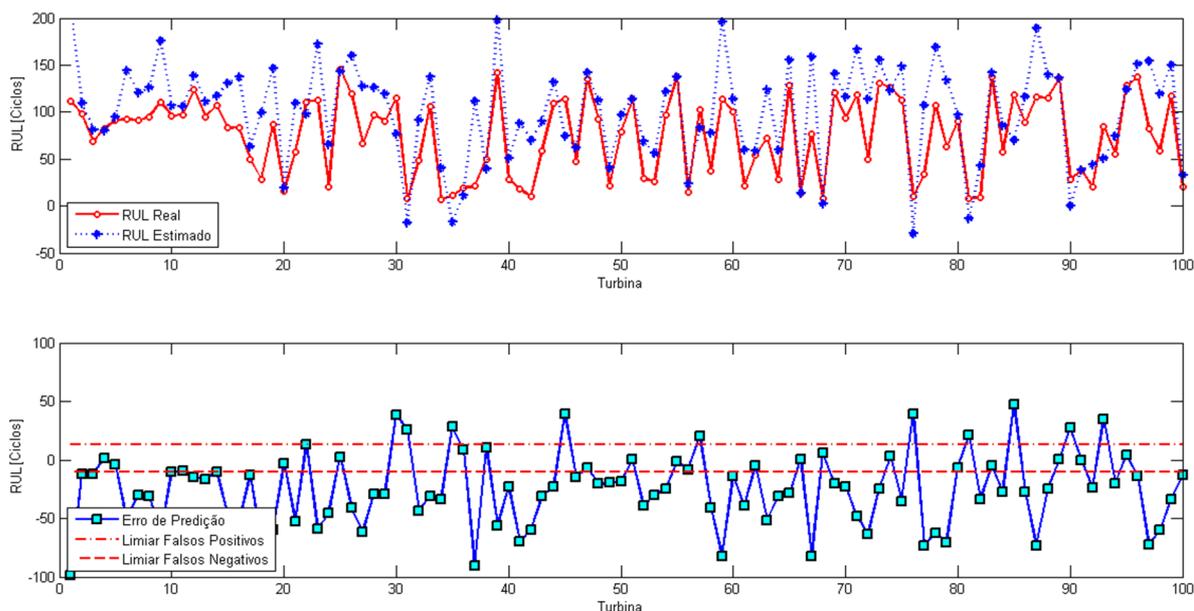


Figura 5.11- RUL estimada através da abordagem ESN-ABC.

5.5 TRABALHOS BASEADOS NO REPOSITÓRIO DE PROGNÓSTICO

Na Tabela 5.8 apresenta-se uma lista de publicações cujos resultados foram baseados usando o repositório de prognóstico da NASA. Pode ser observada que a maior quantidade de publicações é na área de prognóstico, existindo desta forma resultados nesta área com os quais comparar a abordagem neste trabalho.

Tabela 5.8- Publicações focadas em detecção e prognóstico.

Propósito	Publicações
Diagnóstico	Ramasso (2009), Ramasso e Gouriveau (2010), e Ramasso e Gouriveau (2013), Sarkar <i>et al.</i> (2011), Xue <i>et al.</i> (2011), Zhao e Willett (2011), Lin <i>et al.</i> (2013), Ramasso e Denoeux (2013), Li <i>et al.</i> (2013b), Tamilselvan e Wang (2013), Gouriveau <i>et al.</i> (2013)
Prognóstico	Wang <i>et al.</i> (2008), Heimes (2008), Peel (2008), Coble e Hines (2008), Coble (2010), e Coble e Hines (2011), Siegel (2009), Wang (2010), Riad <i>et al.</i> (2010), Abbas (2010), Ramasso e Gouriveau (2010), e Ramasso e Gouriveau (2013), Zhao e Willett (2011), El-Koujok <i>et al.</i> (2011), Wang <i>et al.</i> (2012), Son <i>et al.</i> (2012), Sun <i>et al.</i> (2012), Peng <i>et al.</i> (2012a), Hu <i>et al.</i> (2012), Peng <i>et al.</i> (2012b), Javed <i>et al.</i> (2012), Serir <i>et al.</i> (2012), Gouriveau e Zerhouni (2012), Ramasso <i>et al.</i> (2013), Liu <i>et al.</i> (2013), Son <i>et al.</i> (2013), Xi <i>et al.</i> (2013), Javed <i>et al.</i> (2013), Li <i>et al.</i> (2013b), Ishibashi e Nascimento Junior (2013), Gouriveau <i>et al.</i> (2013), Jianzhong <i>et al.</i> (2010), Zein-Sabatto <i>et al.</i> (2013) e Al-Salah <i>et al.</i> (2013), Ramasso (2014a) e Ramasso (2014b)

Como mostrado anteriormente, o conjunto de dados gerado pelo C-MAPSS foi usado amplamente em detecção e prognóstico. Especificamente em prognóstico, Ramasso e Saxena (2014) dividiram as abordagens em três categorias: (1) mapeamento funcional entre conjunto de entradas e RUL, (2) mapeamento entre indicador de saúde e RUL, e (3) correspondência baseada em similaridade. A abordagem desta tese se encaixa na categoria 1, as publicações nesta categoria são mostradas na Tabela 5.9, e observa-se que as redes neurais artificiais predominam como técnicas de processamento. Encontrou-se apenas uma publicação que usa como técnica a ESN, havendo espaço para explorar a ESN como técnica de prognóstico de este conjunto, e analisar as vantagens e desvantagens.

Tabela 5.9- Métodos com entradas de sensores e saída RUL.

Métodos	Publicações
RNR, EKF	Peel (2008)
PMC, RBF, KF	Peel (2008)
PMC	Riad <i>et al.</i> (2010)
RNA	Abbas (2010)
ESN	Peng <i>et al.</i> (2012a)
Regras <i>fuzzy</i> , algoritmos genéticos	Ishibashi e Nascimento Junior (2013)
PMC	Jianzhong <i>et al.</i> (2010)

5.6 AVALIAÇÃO DE DESEMPENHO

A determinação da precisão, exatidão e desempenho de um algoritmo de prognóstico é um assunto de discussão recente. A taxonomia destas métricas de desempenho para estimação de RUL foi proposto por Saxena *et al.* (2008b) e Saxena *et al.* (2010), onde foram apresentadas diferentes categorias baseadas em: exatidão, precisão e dedicadas especificamente ao prognóstico (métricas PHM). As publicações que utilizaram estas métricas no conjunto de dados N°1 do repositório de prognóstico da NASA são listadas na Tabela 5.10.

A avaliação de desempenho do estudo de caso é realizada comparando as RUL estimadas com as RUL reais fornecidas no arquivo de texto “rul-FD001.txt”. Através das equações apresentadas no Apêndice A obterem-se os valores das métricas de prognóstico utilizadas para realizar uma comparação quantitativa. As métricas obtidas são comparadas com os resultados de outros trabalhos que usaram o mesmo repositório de dados e os resultados são apresentados na Tabela 5.11.

Tabela 5.10- Publicações e as métricas de desempenho usadas.

Categoria	Métrica	Publicações
Exatidão	PHM08	Siegel (2009), Ramasso (2014a) e Ramasso (2014b)
	FPR, FNR	Ramasso e Gouriveau (2010), e Ramasso e Gouriveau (2013), Ramasso et al. (2013), Ramasso (2014a) e Ramasso (2014b)
	MSE	Liao e Sun (2011), Ramasso (2014a) e Ramasso (2014b)
	MAPE	Javed et al. (2012), Liu <i>et al.</i> (2013), Javed <i>et al.</i> (2013), Ramasso (2014a) e Ramasso (2014b)
	MAE	Ramasso (2014a) e Ramasso (2014b)
Precisão	ME	Gouriveau e Zerhouni (2012), Liu <i>et al.</i> (2013), Javed <i>et al.</i> (2013)
	MAD	Gouriveau e Zerhouni (2012)
Prognóstico	PH	Wang (2010), Peng <i>et al.</i> (2012b)
	α - λ	Wang (2010), Peng <i>et al.</i> (2012b)
	RA	Wang (2010), Peng <i>et al.</i> (2012b), Li <i>et al.</i> (2013b)
	CV	Wang (2010), Peng <i>et al.</i> (2012b), Li <i>et al.</i> (2013b)
	AB	Li <i>et al.</i> (2013b)

Observando os resultados da Tabela 5.11 nota-se que a abordagem ESN-ABC apresenta em termos da métrica PHM08 um melhor desempenho do que a ESN clássica desenvolvida. Segundo esta métrica pode-se dizer que a abordagem superou o desempenho de uma ESN clássica desenvolvida também neste trabalho. Peng *et al.* (2012a) apresentou uma abordagem baseada em uma rede ESN clássica e obteve como resultado de prognóstico RMSE = 210,63, nesta tese o resultado obtido por uma ESN clássica é RMSE = 39,47. Uma segunda abordagem proposta por Peng *et al.* (2012a) consiste em uma ESN treinada com Filtro de Kalman, esta deu como resultado RMSE = 63,46, enquanto que a abordagem desenvolvida nesta tese ESN-ABC obteve uma RMSE = 37,62. Comparando estes resultados numéricos comprova-se que a abordagem proposta supera os resultados alcançados pelos métodos de Peng *et al.* (2012a). O resultado unicamente realiza a comparação utilizando uma única métrica de prognóstico, a RMSE.

Comparando através da métrica MAPE o desempenho da abordagem ESN-ABC obteve menor desempenho em relação aos resultados obtidos por Liu *et al.* (2013), Ramasso (2014a) e Ramasso (2014b). A análise com o uso da métrica MAE mostra que o desempenho da abordagem ESN-ABC é menor do que as apresentadas por Siegel (2009), Ramasso (2014a) e Ramasso (2014b). Depois da análise comparativa, em termos gerais verificou-se que os resultados da abordagem ESN-ABC são satisfatórios.

Tabela 5.11- Comparação dos resultados através de métricas de prognóstico.

	EXATIDÃO					PRECISÃO	
	PHM08	FPR	RMSE	MAPE	MAE	ME	MAD
ESN-ABC	7634	14	37,62	39,45	28,79	21,39	22,57
ESN	9988	10	39,47	63,94	31,46	24,05	24,29
Peng <i>et al.</i> (2012a) ESN+FK	---	---	63,46	---	---	---	---
Peng <i>et al.</i> (2012a) ESN Clássica	---	---	210,63	---	---	---	---
Siegel (2009)	---	---	---	---	[3,18]	---	---
Liu <i>et al.</i> (2013)	---	---	---	9	---	---	---
Ramasso <i>et al.</i> (2013)	---	66	---	---	---	---	---
Ramasso (2014a) e Ramasso (2014b)	---	56	176	20	10	---	---

5.7 SÍNTESE DO CAPÍTULO

O estudo de caso foi realizado usando dados de monitoramento de condição do repositório de prognóstico da NASA. O conjunto de dados de treinamento são dados de funcionamento até a falha, significa que na coleta do último dado o equipamento apresentou uma falha funcional. No conjunto de dados de teste é fornecida a RUL no instante da coleta do último dado. Na implementação computacional foi usada a ferramenta computacional *Matlab*. Inicialmente mostra-se os resultados de uma ESN clássica e depois a abordagem ESN-ABC. Os resultados obtidos são comparados também com os obtidos por outros autores em diferentes publicações que utilizaram o mesmo conjunto de dados com outros algoritmos. São realizadas comparações quantitativas através de métricas de prognóstico. Os resultados obtidos mostram que a abordagem híbrida ESN-ABC supera os resultados alcançados pela ESN clássica. Comparando através da métrica MSE os resultados obtidos pela ESN proposta por Peng *et al.* (2012a), tanto a ESN clássica, quanto a ESN-ABC desenvolvidas neste trabalho obtiveram melhores resultados.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo são apresentadas as conclusões da tese como resposta às hipóteses apresentadas. Na segunda seção são listadas as propostas para trabalhos futuros.

6.1 CONCLUSÕES

Esta tese apresentou uma abordagem para prognóstico de RUL baseado em redes com estados de eco e no algoritmo ABC. A abordagem chamada ESN-ABC é um método para otimizando dos parâmetros e dos pesos do reservatório de dinâmicas de uma ESN através do algoritmo ABC. Esta abordagem cria uma ESN com topologia fixa, apenas com as conexões básicas. A busca dos parâmetros se baseia no método clássico de gerar bom reservatório que foi proposta por Jaeger (2001) e Jaeger (2002b), e a busca se concentra apenas na quantidade de neurônios no reservatório, no raio espectral, e nas escalas e deslocamento de entrada e saída. A conectividade da matriz de pesos do reservatório foi definida nesta abordagem no valor de 20%, gerando um reservatório esparso.

O método ESN-ABC combina o algoritmo ABC com a ESN e busca simultaneamente pelos melhores valores dos parâmetros e dos pesos da camada intermediária, a matriz é reescalada por um raio espectral, desta forma se garante a propriedade de estado de eco. As vantagens da abordagem apresentada são que o projeto da ESN não depende da experiência do analista, já que ele permite o ajuste automático combinando diferentes parâmetros e pesos, mostrando bons resultados nas análises do conjunto de dados do repositório de prognóstico da NASA.

O algoritmo ABC é utilizado como método de otimização dos parâmetros e pesos do reservatório. Os algoritmos desenvolvidos utilizam o processamento serial, demandando um tempo considerável, depende do espaço de busca e da quantidade de iterações do algoritmo. No processo experimental manteve-se fixo o número de unidades do reservatório, o algoritmo ABC conseguiu ajustar os outros cinco parâmetros da ESN de forma satisfatória.

A comparação quantitativa através de métricas de prognóstico foi realizada com os resultados de uma rede ESN apresentada por Peng *et al.* (2012a). Peng *et al.* (2012a) através do treinamento clássico obteve um valor de RMSE=210,629, e com treinamento através do filtro de Kalman um RMSE=63,4565. Os resultados nesta tese para treinamento clássico é

RMSE=37,61 e com o algoritmo ESN-ABC um RMSE = 39,47, mostrando que o algoritmo ABC consegue ajustar os parâmetros e pesos da ESN para fornecer um bom desempenho.

A ferramenta computacional desenvolvida, com algumas adaptações, poderá ser aplicada em prognóstico de máquinas reais. Para isto é necessário contar com um repositório de dados de funcionamento até a falha, incluindo dados de monitoração de condição e dados de falha, agrupando-se os dados por máquinas do mesmo tipo. Separando o conjunto de dados em dados de treinamento e teste pode-se aplicar a abordagem proposta.

Em geral a abordagem desenvolvida conseguiu ajustar os parâmetros e pesos de forma eficiente no conjunto de dados, os resultados obtidos foram melhores comparado com a ESN clássica e com a ESN desenvolvida por Peng *et al.* (2012a), todos para o mesmo conjunto de dados. A abordagem não mostrou resultados precisos em todas as turbinas do conjunto de dados de teste, precisando incluir outros parâmetros para serem ajustados ou dados da topologia da ESN como a função de ativação ou os laços de realimentação entre as camadas.

6.2 TRABALHOS FUTUROS

Embora a estrutura desta tese é definida com o intuito de superar os desafios importantes e as limitações dos modelos atuais para otimização do reservatório de dinâmicas de uma ESN e sua aplicação no prognóstico de falhas, existe ainda alguns problemas que necessitam ser considerados. Alguns desafios foram identificados, para cada desafio os tópicos das perspectivas futuras são descritos a seguir:

1. Otimizar a topologia e outros parâmetros da ESN: A abordagem desenvolvida apenas otimiza seis parâmetros da rede ESN, e seria interessante otimizar os outros parâmetros e, ainda, a topologia da rede, parâmetros como a função de ativação e as conexões entre as diferentes camadas.

2. Utilizar o método proposto em outras técnicas de RC: Este método foi aplicado a uma ESN, sugerindo-se a aplicação deste método de otimização de parâmetros e pesos às outras técnicas baseadas em computação com reservatório como são Máquinas de Estado Líquido (LSM – *Liquid State Machines*) (Maass *et al.*, 2002; Natschlagler *et al.*, 2002), e o *Backpropagation Decorrelation* (Steil, 2004).

3. Utilizar outros métodos de otimização dos pesos do Reservatório: O método proposto utiliza algoritmos de colônia de abelhas artificiais para realizar a busca simultânea de

melhores valores dos parâmetros e dos pesos da camada intermediária da ESN. Sugere-se a aplicação de outras técnicas de otimização como, por exemplo, algoritmos genéticos, otimização por enxame de partículas, e outros algoritmos bioinspirados sejam investigados.

4. Investigar o desempenho da abordagem com outros conjuntos de dados: O desempenho do método ESN-ABC foi avaliado para o conjunto de dados de degradação de turbinas turbofan do repositório de prognóstico da NASA. Seria interessante investigar a capacidade da abordagem para outros conjuntos de dados, podendo ser do repositório de prognóstico da NASA (ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository), dados de degradação de rolamentos, de carga descarga de baterias, de desgaste de ferramentas de máquinas de Controle Numérico Computadorizado (CNC – *Computerized Numerical Control*).

REFERÊNCIAS BIBLIOGRÁFICAS

- Abbas, M. (2010). "System level health assessment of complex engineered processes." Unpublished doctoral dissertation, Georgia Institute of Technology.
- Abdulrasool, A.S. e Abbas, S.M. (2013). "Reservoir Computing: Size and Connectivity Optimization using the Worm Algorithm." *International Journal of Computer Applications*, v. 69, n. 4, p.18–22.
- Al-Salah, T., Zein-Sabatto, S. e Bodruzzaman, M. (2012). "Decision fusion software system for turbine engine fault diagnostics." In *Southeastcon, 2012 proceedings of IEEE*, v. 1, n. 1, p. 1–6.
- Altay, N. e Green III, W.G., (2006). "OR/MS research in disaster operations management." *European Journal of Operational Research*, v. 1, n. 175, p. 475–493.
- Amaya, E.J., Alvares, A.J. e Gudwin, R. (2009). "An Expert System for Fault Diagnostics in Condition Based Maintenance." *20th International Congress of Mechanical Engineering (COBEM2009)*, Gramado, RS, Brasil.
- Amaya, E.J. e Alvares, A.J. (2010). "SIMPREGAL: An expert system for real-time fault diagnosis of hydrogenerators machinery." *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, v. 1, no. 1, pp.1–8.
- Amaya, E.J. e Alvares, A.J. (2012). "Expert system for power generation fault diagnosis using hierarchical meta-rules." *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*, vol. 1, no. 1, pp.1–8.
- Andre, D., Nuhic, A., Soczka-Guth, T. e Sauer, D.U. (2013). "Comparative study of a structured neural network and an extended Kalman filter for state of health determination of lithium-ion batteries in hybrid electricvehicles." *Engineering Applications of Artificial Intelligence*, v. 26, n. 3, p. 951–961.
- Antonelo, E. A., Schrauwen, B. e Stroobandt, D. (2008). "Event detection and localization for small mobile robots using reservoir computing." *Neural Networks*, v. 21, n. 6, p. 862–871.
- Arizmendi, C., Sierra, D.A., Vellido, A. e Romero, E. (2011). "Brain tumour classification using Gaussian decomposition and neural networks." *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, vol. 1, no. 1, p.5645,5648.

- Babinec, S. e Pospichal, J. (2006). “Merging echo state and feedforward neural networks for time series forecasting.” Em Proceedings of the 16th International Conference on Neural Networks (ICANN), vol. 1, n. 4131, p. 367–375.
- Bailey, C., Chunyan, Y., Hua, L., Musallam, M. e Johnson, C.M. (2010). “Current status of prognostics techniques and application to power electronics.” Integrated Power Electronics Systems (CIPS), 2010 6th International Conference on, v. 1, n. 1, p. 1–6.
- Banks, J. e Merenich, J. (2007). “Cost Benefit Analysis for Asset Health Management Technology.” Proceedings of the IEEE Reliability and Maintainability Symposium (RAMS), v. 1, n. 1, p. 95 – 100.
- Baraldi, P., Mangili, F. e Zio, E. (2012). “A Kalman Filter-Based Ensemble Approach With Application to Turbine Creep Prognostics.” Reliability, IEEE Transactions on, v. 61, n. 4, p. 966–977.
- Berenji, H. R. (2006). “Case-Based Reasoning for Fault Diagnosis and Prognosis.” 2006 IEEE International Conference on Fuzzy Systems, v. 1, n. 1, p. 1316–1321.
- Berenji, H.R. e Yan, W. (2006). “Wavelet Neural Networks for Fault Diagnosis and Prognosis.” Fuzzy Systems, IEEE International Conference on, v. 1, n. 1, p. 1334–1339.
- Biagetti, T. e Sciubba, E. (2004). “Automatic diagnostics and prognostics of energy conversion processes via Knowledge-Based Systems.” Energy, v. 29, n. 1, p. 12–15.
- Bin, Z., Liang, T., DeCastro, J. e Goebel, K. (2010). "A verification methodology for prognostic algorithms," AUTOTESTCON, 2010 IEEE , v. 1, n. 1, p.1,8.
- Bin, L., Yibin, L. e Xuewen, R. (2011). “Comparison of Echo State Network and Extreme Learning Machine on Nonlinear Prediction.” Journal of Computational Information Systems, v. 7, n. 6, p. 1863–1870.
- Bishop, C.M. (1995). “Neural Networks for Pattern Recognition.” Oxford University Press.
- Bishop, C.M. (2006). “Pattern recognition and machine learning.” In Information Science and Statistics. Springer.
- Bizarria, C.O. (2009). “Prognóstico de falhas no atuador do leme da aeronave EMBRAER-190.” 98f. Dissertação (Mestrado Profissionalizante em Engenharia Aeronáutica) – Programa de Pós-Graduação em Engenharia Aeronáutica e Mecânica, Instituto Tecnológico de Aeronáutica, São José dos Campos.

- Boccatto, L. (2013). “Novas propostas e aplicações de redes neurais com estados de eco.” Tese de doutorado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP, Brasil.
- Boedecker, J., Obst, O., Mayer, N.M. e Asada, M. (2009b). “Studies on Reservoir Initialization and Dynamics Shaping in Echo State Networks.” Proceedings of the 17th European Symposium On Artificial Neural Networks (ESANN).
- Braga, A.P., Carvalho, L.F. e Ludermir, T.B. (2007). “Redes Neurais Artificiais: Teoria e Aplicações.” 2a edição, Livros Técnicos e Científicos.
- Brotherton, T., Jahns, G., Jacobs, J. e Wroblewski, D. (2000). “Prognosis of faults in gas turbine engines.” Aerospace Conference Proceedings. IEEE, v. 6, n. 1, p.163-171.
- Bush, K.A. e Anderson, C.W. (2005). “Modeling reward functions for incomplete state representations via echo state networks.” Em Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), v. 5, n. 1, p. 2995–3000.
- Bush, K.A., e Anderson, C.W. (2006). “Exploiting iso-error pathways in the N, k-plane to improve echo state network performance.” Em Proceedings of the Conference of Neural Information Processing Systems (NIPS).
- Butani, R.C., Gajjar, B.D. e Thakker, R.A. (2011). “Performance evaluation of Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) Algorithm.” International Conference on Advanced Computing, Communication and Networks.
- Butcher, J.B., Verstraeten, D., Schrauwen, B., Day, C.R. e Haycock, P.W. (2010). “Extending reservoir computing with random static projections: a hybrid between extreme learning and RC.” Em Proceedings of ESANN - European Symposium on Artificial Neural Networks, v. 1, n. 1, p. 303–308.
- Butcher, J.B., Verstraeten, D., Schrauwen, B., Day, C.R. e Haycock, P.W. (2013). “Reservoir computing and extreme learning machines for non-linear time-series data analysis.” Neural Networks, v. 38, n. 1, p. 76–89.
- Byington, C.S., Roemer, M.J. e Kalgren, P.W. (2005). “Verification and Validation of Diagnostic/Prognostic Algorithms.” Machinery Failure Prevention Technology (MFPT) Conference: Virginia Beach, VA.
- Carpenter, G.A. e Grossberg, S. (1987). “A massively parallel architecture for a selforganizing neural pattern recognition machine.” Computer Vision, Graphics, and Image Processing, v. 37, n. 1, p. 54–115.

- Carr, M.J. e Wang, W. (2011). “An approximate algorithm for prognostic modelling using condition monitoring information.” *European Journal of Operational Research*, v. 211, n. 1, p. 90–96.
- Cazamine, S. e Sneyd, J. A. (1991). “model of collective nectar source selection by honey bees.” *Journal of Theoretical Biology*, v. 149, n. 4, p. 547–571.
- Celaya, J.R., Saha, B., Wysocki, P.F (2008). “Prognostics for Electronics Components of Avionics Systems.” *IEEE Aerospace Conference Proceedings*.
- Chang-Yu, W., Tsair-Fwu, L., Chun-Hsiung, F. e Jyh-Horng, C. (2012). “Fuzzy Logic-Based Prognostic Score for Outcome Prediction in Esophageal Cancer.” *Information Technology in Biomedicine, IEEE Transactions on*, v. 16, n. 6, p.1224–1230.
- Chaochao, C., Bin, Z., Vachtsevanos, G., Orchard, M. (2011). “Machine Condition Prediction Based on Adaptive Neuro-Fuzzy and High-Order Particle Filtering.” *Industrial Electronics, IEEE Transactions on*, v. 58, n. 9, p. 4353–4364.
- Chaochao, C. e Pecht, M. (2012). “Prognostics of lithium-ion batteries using model-based and data-driven methods.” *Prognostics and System Health Management (PHM), IEEE Conference on*, v. 1, n. 1, p. 1–6.
- Chen, C., Vachtsevanos, G. e Orchard, M.E. (2012). “Machine remaining useful life prediction: An integrated adaptive neuro-fuzzy and high-order particle filtering approach.” *Mechanical Systems and Signal Processing*, v. 28, n. 1, p. 597–607.
- Cheng, S., Tom, K., Thomas, L., Pecht, M. (2010). “A Wireless Sensor System for Prognostics and Health Management.” *Sensors Journal, IEEE*, v. 10, n. 4, p. 856–862.
- Ciarelli, P.M., Oliveira, E. e Salles, E. (2012). “An incremental neural network with a reduced architecture.” *Neural Networks*, v. 35, n. 1, p. 70–81.
- Coble, J. e Hines, J. (2008). “Prognostic algorithm categorization with PHM challenge application.” In *IEEE int. conf. on prognostics and health management*.
- Coble, J. (2010). “Merging data sources to predict remaining useful life - an automated method to identify prognostic parameters.” Unpublished doctoral dissertation, University of Tennessee, Knoxville.
- Coble, J. e Hines, W. (2011). “Applying the general path model to estimation of remaining useful life.” *International Journal of Prognostics and Health Management*, v. 2, n. 1, p. 1–13.
- Compare, M. e Zio, E. (2014). “Predictive Maintenance by Risk Sensitive Particle Filtering.” *Reliability, IEEE Transactions on*, v. 63, n. 1, p. 134–143.

- Cui, L.R., Loh, H.T. e Xie, M., (2004). “Sequential inspection strategy for multiple systems under availability requirement.” *European Journal of Operational Research*, v. 155, n. 1, p. 170–177.
- Cybenko, G. (1989). “Approximation by superpositions of a sigmoidal function.” *Mathematics of Control Signals and Systems*, v. 2, n. 1, p. 303–314.
- Daigle, M. e Goebel, K. (2010). “Model-based prognostics under limited sensing.” *Aerospace Conference, IEEE*, v. 1, n. 1, p. 1–12.
- Daigle, M., Saha, B. e Goebel, K. (2012). “A comparison of filter-based approaches for model-based prognostics,” *Aerospace Conference, IEEE*, v. 1, n. 1, p. 1–10.
- Darogheh, N., Meskin, N. e Khorasani, K. (2014). “A novel particle filter parameter prediction scheme for failure prognosis.” *American Control Conference (ACC)*, v. 1, n. 1, p. 1735–1742.
- Das, S., Hall, R., Herzog, S., Harrison, G. e Bodkin, M. (2011). “Essential steps in prognostic health management.” *Prognostics and Health Management (PHM), IEEE Conference on*, v. 1, n. 1, p. 1–9.
- Daubechies, I. (1990). “The Wavelet Transform, Time-Frequency Localization and Signal Analysis.” *IEEE Transactions on Information Theory*, v. 36, n. 1, p. 961–1005.
- Deihimi, A. e Showkati, H. (2012). “Application of echo state networks in short-term electric load forecasting.” *Energy*, v. 39, n. 1, p. 327–340.
- Dempster, A.P., Laird, N.M. e Rubin, D.B. (1977). “Maximum Likelihood from Incomplete Data via the EM Algorithm.” *Journal of the Royal Statistical Society*, v. 39, n. 1, p. 1–38.
- Dong, M. e He, D., (2007a). “Hidden semi-Markov model-based methodology for multisensory equipment health diagnosis and prognosis.” *European Journal of Operational Research*, v. 178, n. 3, p. 858–878.
- Dos Santos, E.P. e Von Zuben, F.J. (2000). “Efficient second-order learning algorithms for discrete-time recurrent neural networks.” *Em Recurrent neural networks: design and applications*, v. 1, n. 1, p. 47–75.
- Doya, K. (1993). “Universality of fully connected recurrent neural networks,” Dept. of Biology, UCSD, Tech. Rep.
- Duda, R.O., Hart, P.E. e Stork, D.G. (2001). “Pattern classification.” (2a ed.) John Wiley & Sons.

- El-Koujok, M., Gouriveau, R. e Zerhouni, N. (2011). “Reducing arbitrary choices in model building for prognostics: An approach by applying parsimony principle on an evolving neuro-fuzzy system.” *Microelectronics Reliability*, v. 51, n. 2 p. 310–320.
- Elwany, A.H. e Gebraeel, N.Z., (2008). “Sensor-driven prognostic models for equipment replacement and spare parts inventory.” *IIE Transactions*, v. 40, n. 1, p. 629–639.
- Engel, S.J., Gilmartin, B.J., Bongort, K. e Hess, A. (2000). “Prognostics, the real issues involved with predicting life remaining.” *Aerospace Conference Proceedings, IEEE*, vol.6, no., pp.457-469 vol.6.
- Engel, P. e Heinen, M. (2011). “Incremental learning of multivariate gaussian mixture models.” *Advances in Artificial Intelligence–SBIA 2010*, v. 1, n. 1, p. 82–91.
- Ferreira, A. A., Ludermir, T.B., Aquino, R., Lira, M.M. e Neto, O.N. (2008). “Investigating the use of reservoir computing for forecasting the hourly wind speed in short-term.” In *International Joint Conference on Neural Networks - IJCNN 2008*, v. 1, n. 1, p. 1950–1957.
- Ferreira, A. A. e Ludermir, T.B. (2009). “Genetic algorithm for reservoir computing optimization.” *Neural Networks. IJCNN - International Joint Conference on*, v. 1, n. 1, p. 811–815.
- Ferreira, A. A. e Ludermir, T.B. (2010). “Evolutionary strategy for simultaneous optimization of parameters, topology and reservoir weights in echo state networks.” in *Neural Networks (IJCNN), The International Joint Conference on*, v. 1, n. 1, p. 1–7.
- Ferreira, A. A. e Ludermir, T.B. (2011). “Comparing evolutionary methods for reservoir computing pre-training.” in *Neural Networks (IJCNN), The International Joint Conference on*, v. 1, n. 1, p. 283–290.
- Ferreira, A. A. (2011). “Um Método para Design e Treinamento de Reservoir Computing Aplicado à Previsão de Séries Temporais.” Tese de doutorado, Universidade Federal de Pernambuco, Recife, PE, Brasil.
- Ferreira, A. A., Ludermir, T.B. e Aquino, R. (2013). “An approach to reservoir computing design and training.” *Expert Systems with Applications*, v. 40, n. 10, p. 4172–4182.
- Ferreiro, S., Arnaiz, A., Sierra, B. e Irigoien, I. (2012). “Application of Bayesian networks in prognostics for a new Integrated Vehicle Health Management concept.” *Expert Systems with Applications*, v. 39, n. 7, p. 6402–6418.
- Frederick, D., De Castro, J. e Litt, J. (2007). “User’s Guide for the Commercial Modular Aero-Propulsion System Simulation (CMAPSS).” NASA/ARL.

- Funahashi, K.-I. e Nakamura, Y. (1993). "Approximation of dynamical systems by continuous time recurrent neural networks." *Neural Networks*, v. 6, n. 1, p. 801–806.
- Gasperin, M., Juricic, Baskoski, P. e Jozef, V. (2011). "Model-based prognostics of gear health using stochastic dynamic models." *Mechanical Systems and Signal Processing*, v. 25, n. 1, p. 537–538.
- Gebraeel, N.Z. e Lawley, M.A. (2008). "A neural network degradation model for computing and updating residual life distributions." *IEEE Transaction on Automation Science and Engineering*, v. 5, n. 1, p. 154–163.
- Geramifard, O., Jian-Xin, X., Jun-Hon, Z., Xiang, L. (2012). "A Physically Segmented Hidden Markov Model Approach for Continuous Tool Condition Monitoring: Diagnostics and Prognostics." *Industrial Informatics, IEEE Transactions on*, v. 8, n. 4, p. 964–973.
- Ghasemi, A., Yacout, S. e Ouali, M.S. (2010). "Evaluating the reliability function and the mean residual life for equipment with unobservable states." *IEEE Transactions on Reliability*, v. 59, n. 1, p. 45–54.
- Ghavami, P. e Kapur, K. (2012). "Artificial neural network-enabled prognostics for patient health management." *Prognostics and Health Management (PHM), IEEE Conference on*, v. 1, n. 1, p. 1–8.
- Goebel, K., Saha, B. e Saxena, A. (2008a). "A comparison of three data-driven techniques for prognostics." *NASA Ames Research Center*.
- Goebel, K., Saha, B., Saxena, A., Celaya, J.R. e Christophersen, J.P. (2008b). "Prognostics in Battery Health Management." *Instrumentation & Measurement Magazine, IEEE*, v. 11, n. 4, p. 33–40.
- Goodman, D.L, Wood, S. e Turner, A. (2005). "Return-on-Investment (ROI) for Electronic Prognostics in MIL/AERO Systems." *Proceedings of the IEEE Systems Readiness Technology Conference (AUTOTESTCON)*, v. 1, n. 1, p. 73–75.
- Gouriveau, R. e Zerhouni, N. (2012). "Connexionist systems based long term prediction approaches for prognostics." *IEEE Trans. on Reliability*, v. 61, n. 1, p. 909–920.
- Gouriveau, R., Ramasso, E. e Zerhouni, N. (2013). "Strategies to face imbalanced and unlabelled data in PHM applications." *Chemical Engineering Transactions*, v. 33, n. 1, p. 115–120.
- Greitzer, F.L. e Pawlowski, R.A. (2002). "Embedded Prognostics Health Monitoring." *International Instrumentation Symposium, Embedded Health Monitoring Workshop*.

- Groer, P. (2000). "Analysis of Time to Failure with a Weibull Mode." Maintenance and Reliability Conference, MARCON.
- Hai, Q. e Lee, J. (2004), "Feature fusion and degradation using self-organizing map." Machine Learning and Applications, Proceedings. International Conference on, v. 1, n. 1, p. 107–114.
- Hassan, R., Cohanin, B., De Weck, O. e Venter, G. (2005). "A Comparison of Particle Swarm Optimization And The Genetic Algorithm." 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, v. 1, n. 1, p. 1–13.
- Haykin, S. (1996). "Adaptive filter theory." (3a ed.) NJ: Prentice Hall.
- Haykin, S. (1998). "Neural networks: A comprehensive foundation." (2a ed.) Prentice Hall.
- He, D., Wu, S., Banerjee, P. e Bechhoefer, E. (2006). "Probabilistic Model Based Algorithms for Prognostics." IEEE Aerospace Conference, Big Sky, MT.
- Hebb, D.O. (1949). "The Organization of Behavior." New York: Wiley & Sons.
- Heimes, F.O. (2008). "Recurrent neural networks for remaining useful life estimation." Prognostics and Health Management (PHM). International Conference on, v. 1, n. 1, p. 1–6.
- Heng, A., Zhang, S. Tan, A. e Mathew, J. (2009). "Rotating machinery prognostics: State of the art, challenges and opportunities." Mechanical Systems and Signal Processing, v. 23, n. 3, p. 724–739.
- Hines, J.W., Garvey, D., Seibert, R., Usynin, A. e Arndt, S.A. (2008). "Technical Review of On-Line Monitoring Techniques for Performance Assessment (NUREG/CR-6895) Vol. 2 Theoretical Issues." Published May, 2008.
- Hopfield, J.J. (1982). "Neural networks and physical systems with emergent collective computational properties." Proceedings of the National Academy of Sciences of the USA, v. 79, n. 8, p. 2554–2558.
- Hornik, K., Stinchcombe, M. e White, H. (1989). "Multilayer feedforward networks are universal approximators." Neural Networks, v. 2, n. 5, p. 359–366.
- Hossain, M.S. e El-Shafie, A. (2014). "Evolutionary techniques versus swarm intelligences: application in reservoir release optimization." Neural Computing and Applications, v. 24, n. 7–9, p. 1583–1594.

- Hsieh, S., Sun, T., Liu, C. e Tsai, S.J. (2009). “Efficient Population Utilization Strategy for Particle Swarm Optimizer.” *IEEE Transactions, Man and Cybernetics*, v. 30, n. 1, p. 444–456.
- Hu, C., Youn, B.D. e Wang, P. (2011). “Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life.” *Prognostics and Health Management (PHM), IEEE Conference*, v. 1, n. 1, p. 1–10.
- Hu, C., Youn, B., Wang, P. e Yoon, J. (2012). “Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life.” *Reliability Engineering and System Safety*, v. 103, n. 1, p. 120 – 135.
- Huang, G.-B., Zhu, Q.-Y. e Siew, C.-K. (2004). “Extreme learning machine: a new learning scheme of feedforward neural networks.” *Em Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2004)*, v. 1, n. 1, p. 985–990.
- Huang, G.-B., Zhu, Q.-Y. e Siew, C.-K. (2006). “Extreme learning machine: theory and applications.” *Neurocomputing*, v. 70, n. 1, p. 489–501.
- Huiguo, Z., Rui, K. e Pecht, M. (2009). “A hybrid prognostics and health management approach for condition-based maintenance.” *Industrial Engineering and Engineering Management. IEEM 2009. IEEE International Conference on*, v. 1, n. 1, p. 1165–1169.
- Ishii, K., van der Zant, T., Becanovic, V. e Ploger, P. (2004). “Identification of motion with echo state network.” In *OCEANS 2004 MTS/IEEE – TECHNOOCEAN 2004 Conference*, v. 3, n. 1, p. 1205–1210.
- Ishibashi, R. e Nascimento Junior, C. (2013). “GFRBSPHM: A genetic fuzzy rule-based system for PHM with improved interpretability.” In *IEEE conference on prognostics and health management (PHM)*, v. 1, n. 1, p. 1–7.
- ISO 13381 (2004). “Condition monitoring and diagnostics of machines – Prognostics – Part 1: General guidelines.” *International Organization for Standardization*.
- Jaeger, H (2001). “The Echo state approach to analyzing and training recurrent neural networks.” *Technical Report GMD Report 148, German National Research Center for Information Technology*, URL <http://minds.jacobs-university.de/sites/default/files/uploads/papers/EchoStatesTechRep.pdf>.
- Jaeger, H (2002a). “Short-term memory in echo state networks.” *Technical report, GDM 152, German National Resource Center for Information Technology*.
- Jaeger, H (2002b). “A tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the echo state network approach.” *Technical Report GMD Report 159*,

German National Research Center for Information Technology, URL <http://minds.jacobs-university.de/sites/default/files/uploads/papers/ESNTutorialRev.pdf>.

- Jaeger, H. e Haas, H. (2004). “Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication.” *Science*, 304(5667): pp. 78–80, URL www.faculty.jacobs-university.de/hjaeger/pubs/ESNScience04.pdf.
- Jaeger, H (2007). “Discovering multiscale dynamical features with hierarchical echo state networks.” Technical Report No. 9, Jacobs University Bremen.
- Jardine, A.K.S., Lin, D. e Banjevic, D. (2006). “A review on machinery diagnostics and prognostics implementing condition-based maintenance.” *Mechanical Systems and Signal Processing*, v. 20, n. 1, p. 1483–1510.
- Javed, K., Gouriveau, R., Zerhouni, N., Zemouri, R. e Li, X. (2012). “Robust, reliable and applicable tool wear monitoring and prognostic: Approach based on an improved-extreme learning machine.” in *Proc. IEEE Conf. Prognostics Health Manage.*, Denver, CO, USA, v. 1, n. 1, p. 1–9.
- Javed, K., Gouriveau, R. e Zerhouni, N. (2013). “Novel failure prognostics approach with dynamic thresholds for machine degradation.” *Industrial Electronics Society, IECON - 39th Annual Conference of the IEEE*, v. 1, n. 1, p. 4404–4409.
- Javed, K., Gouriveau, R. e Zerhouni, N. (2014). “SW-ELM: A summation wavelet extreme learning machine algorithm with a priori parameter initialization.” *Neurocomputing*, v. 123, n. 1, p. 299–307.
- Jianzhong, S., Hongfu, Z., Haibin, Y. e Pecht, M. (2010). “Study of ensemble learning-based fusion prognostics.” In *Prognostics and health management conference, 2010*, v. 1, n. 1, p. 1–7.
- Jouin, M., Gouriveau, R., Hissel, D., Péra, M. e Zerhouni, N. (2013). “Prognostics and Health Management of PEMFC – State of the art and remaining challenges.” *International Journal of Hydrogen Energy*, v. 38, n. 35, p. 15307–15317.
- Kacprzyński, G.J., Liberson, A., Palladino, A., Roemer, M.J., Hess, A.J. e Begin, M. (2004). “Metrics and Development Tools for Prognostic Algorithms.” *Proceedings of the IEEE Aerospace Conference*, v. 1, n. 1, p. 3809 – 3815.
- Karaboga, D. (2005). “An Idea based on honey bee swarm for numerical optimization.” *Reporte técnico*.

- Karaboga, D. e Akay, B. (2009). "A comparative study of Artificial Bee Colony algorithm." *Applied Mathematics and Computation*, v. 214, n. 1, p. 108–132.
- Katipamula, S. e Brambley, M.R. (2005). "Methods for fault detection, diagnostics, and prognostics for building systems - a review." part II. *HVAC and R Research*, v. 11, n. 2, p. 169–187.
- Ke-Xu, Z., Hao-Dong, M., Hong-Zheng, F. e Da-Wei, Y. (2011). "Study of prognostics for spacecraft based-on particle swarm optimized neural network." *Prognostics and System Health Management Conference (PHM-Shenzhen)*, v. 1, n. 1, p. 1–5.
- Kim, C, Weston, R.H., Hodgson, A. e Lee, K. (2002). "The complementary use of IDEF and UML modelling approaches." In: *Computers in Industry*, v. 50, n. 1, p. 35 – 56.
- Kim, E.Y, Tan, A., Yang B.-S. e Kosse, V. (2007). "Experimental Study on Condition Monitoring of Low Speed Bearings:Time domain Analysis. 5th Australasian Congress on Applied Mechanics, ACAM2007. Brisbane, Australia.
- Kim, K.O. e Kuo, W., (2009). "Optimal burn-in for maximizing reliability of repairable non-series systems." *European Journal of Operational Research*, v. 193, n. 1, p. 140–151.
- Kohonen, T. (2000). "Self-organizing maps." (3a ed.) Springer.
- Kothamasu, R., Huang, S.H. e VerDuin, W.H. (2006). "System health monitoring and prognostics: a review of current paradigms and practices." *Int J Adv Manuf Technol*, v. 28, n. 1, p. 1012–1024.
- Kumar, S., Torres, M., Chan, Y.C. e Pecht, M. (2008). "A hybrid prognostics methodology for electronic products." *Neural Networks. IJCNN. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, v. 1, n. 1, p. 3479–3485.
- Kuznetsov, Y., Kuznetsov, L. e Marsden, J. (1998). "Elements of applied bifurcation theory." (2a ed.) Springer-Verlag.
- Lall, P., Gupta, P. e Panchagade, D. (2010). "Self-organized mapping of failure modes in portable electronics subjected to drop and shock." *Electronic Components and Technology Conference (ECTC), 2010 Proceedings 60th*, v. 1, n. 1, p. pp.1195–1208.
- Laslett, O.W., Mills, A.R., Zaidan, M.A. e Harrison, R.F. (2014). "Fusing an ensemble of diverse prognostic life predictions." *Aerospace Conference, 2014 IEEE*, v. 1, n. 1, p. 1–10.
- Lee, J., Ni, J., Djurdjanovic, D., Qiu, H. e Liao, H. (2006). "Intelligent prognostics tools and e-maintenance." *Computers in industry*, v. 57, n. 1, p. 476–489.

- Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L. e Siegel, D. (2014). “Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications.” *Mechanical Systems and Signal Processing*, v. 42, n. 1–2, p. 314–334.
- Lei, Y., He, Z., Zi, Y. e Hu, Q. (2007). “Fault diagnosis of rotating machinery based on multiple ANFIS combination with GAs.” *Mechanical Systems and Signal Processing*, v. 21, n. 1, p. 2280–2294.
- Li, D., Wang, W., e Ismail, F. (2013a). “Enhanced fuzzy-filtered neural networks for material fatigue prognosis.” *Applied Soft Computing*, v. 13, n. 1, p. 283–291.
- Li, X., Qian, J. e Wang, G. (2013b). “Fault prognostic based on hybrid method of state judgment and regression.” *Advances in Mechanical Engineering*, v. 149, n. 1, p. 1–10.
- Liang, E., Rodriguez, R.J. e Hussein, A.A. (1988). “Prognostics/diagnostics of mechanical equipment by neural network.” *Neural Networks*, v. 1, n. 1, p. 33.
- Liang, T., DeCastro, J., Kacprzyński, G. e Goebel, K. (2010). “Vachtsevanos, G., "Filtering and prediction techniques for model-based prognosis and uncertainty management.” *Prognostics and Health Management Conference*, v. 1, n. 1, p. 1–10.
- Liao, L. e Kottig, F. (2014). “Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction.” *Reliability, IEEE Transactions on*, v. 63, n. 1, p. 191–207.
- Liao, H. e Sun, J. (2011). “Nonparametric and semiparametric sensor recovery in multichannel condition monitoring systems.” *IEEE Transactions on Automation Science and Engineering*, v. 8, n. 4, p. 744–753.
- Lin, Y., Chen, M. e Zhou, D. (2013). “Online probabilistic operational safety assessment of multi-mode engineering systems using Bayesian methods.” *Reliability Engineering and System Safety*, v. 119, n. 1, p. 150–157.
- Line, J.K. e Clements, N.S. (2006). “Prognostics Usefulness Criteria.” *Proceedings of the IEEE Aerospace Conference*.
- Liu, J., Saxena, A. e Goebel, K. (2010). “An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries.” *Annual conference of the Prognostics and Health Management Society*.
- Liu, K., Gebraeel, N. Z. e Shi, J. (2013). “A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis.” *IEEE Trans. on Automation Science and Engineering*.

- Long, B., Xian, W., Jiang, L. e Liu Z. (2013). “An improved autoregressive model by particle swarm optimization for prognostics of lithium-ion batteries.” *Microelectronics Reliability*, v. 53, n. 6, p. 821–831.
- Luenberger, D. G. (2003). “Linear and nonlinear programming.” (2a ed.) Springer.
- Lukosevicius, M. e Jaeger, H. (2009). “Reservoir computing approaches to recurrent neural network training.” *Computer Science Review*, v. 3, n. 1, p. 127–149.
- Lukosevicius, M. (2012). “A practical guide to applying echo state networks.” volume 7700 of *Lecture Notes in Computer Science*, v. 1, n. 1, p. 659–686. Springer Berlin Heidelberg, 2 edition.
- Maass, W., Natschlager, T. e Markram, H. (2002). “Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, v. 14, n. 11, p. 2531–2560.
- Majidian, A. e Saidi, M. (2007). “Comparison of fuzzy logic and neural network in life prediction of boiler tubes,” *Int. J. Fatigue*, v. 29, n. 3, p. 489–498.
- Malhi, A., Ruqiang, Y. e Gao, R.X. (2011). “Prognosis of Defect Propagation Based on Recurrent Neural Networks.” *Instrumentation and Measurement, IEEE Transactions on*, v. 60, n. 3, p. 703–711.
- McCulloch, W. e Pitts, W. (1943). “A logical calculus of the ideas immanent in nervous activity.” *Bulletin of Mathematical Biophysics*, v. 5, n. 4, p. 115–133.
- Medjaher, K., Tobon-Mejia, D.A. e Zerhouni, N. (2012). “Remaining Useful Life Estimation of Critical Components with Application to Bearings.” *Reliability, IEEE Transactions on*, v. 61, n. 2, p. 292–302.
- Minsky, M.L. e Papert, S.A. (1969). “Perceptrons: an Introduction to Computational Geometry.” MIT Press.
- Moghaddass, R. e Zuo, M.J. (2012). “A parameter estimation method for a condition-monitored device under multi-state deterioration.” *Reliability Engineering & System Safety*, v. 106, n. 1, p. 94–103.
- Moghaddass, R. e Zuo, M.J. (2014). “An integrated framework for online diagnostic and prognostic health monitoring using a multistate deterioration process.” *Reliability Engineering & System Safety*, v. 124, n. 1, p. 92–104.
- Morando, S., Jemei, S., Gouriveau, R., Zerhouni, N. e Hissel, D. (2013). “Fuel Cells prognostics using echo state network.” *Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE*, v. 1, n. 1, p. 1632–1637.

- Morando, S., Jemei, S., Gouriveau, R., Zerhouni, N. e Hissel, D. (2014). “ANOVA Method Applied to PEMFC Ageing Forecasting Using an Echo State Network.” 11th International Conference on Modeling and Simulation of Electric Machines, Converters and Systems (ElectrIMACS 2014), v. 1, n. 1, p. 652–657
- Mosallam, A., Medjaher, K. e Zerhouni, N. (2013). “Bayesian approach for remaining useful life prediction.” Chemical Engineering Transactions, v. 33, n. 1, p. 139–144.
- Murphy, K. (2002). “Dynamic Bayesian networks: representation, Inference and learning.” Ph.D.Thesis, University of California, Berkley.
- Natschlager, T., Maass, W. e Markram, H. (2002). “The ‘liquid computer’: A novel strategy for real-time computing on time series.” Special Issue on Foundations of Information Processing of TELEMATIK, v. 8, n. 1, p. 39–43.
- Oentaryo, R.J., Meng, J.E., Linn, S., Lianyin, Z. e Xiang, L. (2011). “Bayesian ART-based fuzzy inference system: A new approach to prognosis of machining processes.” Prognostics and Health Management (PHM), IEEE Conference on, v. 1, n. 1, p. 1–10.
- Olivares, B.E., CerdaMunoz, M.A, Orchard, M.E. e Silva, J.F. (2013), "Particle-Filtering-Based Prognosis Framework for Energy Storage Devices with a Statistical Characterization of State-of-Health Regeneration Phenomena." Instrumentation and Measurement, IEEE Transactions on, v. 62, n. 2, p. 364–376.
- Ongenaes, F., Looy, S.V., Verstraeten, D., Verplancke, T., Dominique, B., Turck, F., Dhaene, T., Schrauwen, B. e Decruyenaere, J. (2013). “Time series classification for the prediction of dialysis in critically ill patients using echo statenetworks.” Engineering Applications of Artificial Intelligence, v. 26, n. 3, p. 984–996.
- Orchard, M., Wu, B.Q., Vachtsevanos, G. (2005). “A particle filtering framework or failure prognosis.” Proceedings of WTC2005 World Tribology Congress, Washington, D.C.,USA.
- Orchard, M., Vachtsevanos, G. (2009). “A Particle filtering approach for on-line fault diagnosis and failure prognosis.” Transactions of the Institute of Measurement and Control, v. 31, n. 3–4, p. 221–246.
- Panda, S., Padhy, N.P. (2007). “Comparison of Particle Swarm Optimization and Genetic Algorithm for TCSC-based Controller Design.” International Journal of Computer Science & Engineering, v. 1, n. 1, p. 41.
- Pandit,S. e Wu, S. (1983). “Time Series and System Analysis with Applications,” John Wiley & Sons, New York.

- Papakostas, N., Papachatzakis, P., Xanthakis, V., Mourtzis, D., Chryssolouris, G., (2010). "An approach to operational aircraft maintenance planning." *Decision Support Systems*, v. 48, n. 1, p. 604–612.
- Parker, B.E., Nigro, T.M., Carley, M.P., Barron, R.L., Ward, D.G., Poor, H.V., Rock, D. e DuBois, T.A. (1993). "Helicopter gearbox diagnostics and prognostics using vibration signature analysis." in: *Proceedings of the SPIE - The International Society for Optical Engineering*, v. 1965, n. 1, p. 531–542.
- Pecht, M. (2008). "Prognostics and Health Management of Electronics." John Wiley, New Jersey.
- Pecht, M. e Jaai, R. (2010). "A prognostics and health management roadmap for information and electronics-rich system." *Microelectronics Reliability*, v. 50, n. 1, p. 317–323.
- Peel, L. (2008). "Data driven prognostics using a Kalman filter ensemble of neural network models." In *Int. conf. on prognostics and health management*.
- Peng, Y., Dong, M. e Zuo, M.J. (2010). "Current status of machine prognostics in condition-based maintenance: A review." *International Journal of Advanced Manufacturing Technology*, v. 50, n. 1–4, p. 297–313.
- Peng, Y. e Dong, M. (2011). "A prognosis method using age-dependent hidden semi-Markov model for equipment health prediction." *Mechanical Systems and Signal Processing*, v. 25, n. 1, p. 237–252.
- Peng, Y., Wang, H., Wang, J., Liu, D. e Peng, X. (2012a). "A modified echo state network based remaining useful life estimation approach." *Prognostics and Health Management (PHM), IEEE Conference on*, v. 1, n. 1, p. 1–7.
- Peng, Y., Xu, Y., Liu, D. e Peng, X. (2012b). "Sensor selection with grey correlation analysis for remaining useful life evaluation." In *Annual conference of the PHM society*.
- Pinto, R. C., Engel, P. M. e Heinen, M. R. (2011). "Echo State Incremental Gaussian Mixture Network for Spatio-Temporal Pattern Processing." In: *CSBC 2011 - VIII ENIA, Natal. Anais do VIII ENIA*.
- Pla, A., López, B., Gay, P. e Pous, C. (2013). "eXiT*CBR.v2: Distributed case-based reasoning tool for medical prognosis, *Decision Support Systems*, v. 54, n. 3, p. 1499–1510.
- Prechelt. L. (1994). "Proben1 - a set of neural network benchmark problems and benchmarking rules." Technical report, 21/94, Fakultät für Informatik, Universität Karlsruhe, Germany.

- Ramasso, E. (2009). "Contribution of belief functions to hidden Markov models with an application to fault diagnosis." In *Machine learning for signal processing*.
- Ramasso, E. e Gouriveau, R. (2010). "Prognostics in switching systems: Evidential Markovian classification of real-time neuro-fuzzy predictions." In *IEEE prognostics and health management conference*.
- Ramasso, E. e Denoeux, T. (2013). Making use of partial knowledge about hidden states in hidden Markov models: an approach based on belief functions. *IEEE Transactions on Fuzzy Systems*.
- Ramasso, E. e Gouriveau, R. (2013). "RUL estimation by classification of predictions: an approach based on a neuro-fuzzy system and theory of belief functions." *IEEE Transactions on Reliability*.
- Ramasso, E., Rombaut, M. e Zerhouni, N. (2013). "Joint prediction of continuous and discrete states in time-series based on belief functions." *IEEE Trans. Cybern.*, v. 43, n. 1, p. 37–50.
- Ramasso, E. (2014a). "Investigating computational geometry for failure prognostics." *Int. Journal on Prognostics and Health Management*.
- Ramasso, E. (2014b). "Investigating computational geometry for failure prognostics in presence of imprecise health indicator: Results and comparisons on C-MAPSS datasets." In *European conf. on prognostics and health management*.
- Ramasso, E. e Saxena, A. (2014). Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets," *International Journal of Prognostics and Health Management*, v. 5, n. 2, p. 1–15.
- Riad, A., Elminir, H. e Elattar, H. (2010). "Evaluation of neural networks in the subject of prognostics as compared to linear regression model." *International Journal of Engineering & Technology*, v. 10, n. 1, p. 52–58.
- Richter, H. (2012). "Engine models and simulation tools." In *Advanced control of turbofan engines*, pp. 19–33. Springer New York.
- Rosenblatt, F. (1958). "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological Review*, v. 65, n. 6, p. 386–408.
- Ross, T.J. (2004). "Fuzzy Logic with Engineering Applications." Wiley, New York.
- Rumelhart, D.E., Hinton, G.E. e Williams, R.J. (1986). "Learning representations by back-propagating errors." *Nature*, v. 323, n. 1, p. 533–536.

- Saha, B., Celaya, J.R. e Goebel, K. (2009a). "Towards Prognostics for Electronics Components." IEEE Aerospace Conference Proceedings.
- Saha, B., Goebel, K., Christophersen, J. (2009b). "Comparison of prognostic algorithms for estimating remaining useful life of batteries." Transactions of the Institute of Measurement & Control, v. 31, n. 3–4, p. 293–308.
- Saikiran, P.V.R., VijayaramKumar, S., Vijayakumar, P.S., Varakhedi, V. e Upendranath, V. (2013). "Application of Kalman Filter to prognostic method for estimating the RUL of a bridge rectifier." Emerging Trends in Communication, Control, Signal Processing & Computing Applications (C2SPCA), International Conference on, v. 1, n. 1, p. 1–9.
- Sarkar, S., Jin, X. e Ray, A. (2011). "Data-driven fault detection in aircraft engines with noisy sensor measurements." Journal of Engineering for Gas Turbines and Power, 133.
- Saxena, A. e Goebel, K. (2008). "C-MAPSS Data Set." NASA Ames Prognostics Data Repository, [<http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/>], NASA Ames, Moffett Field, CA.
- Saxena, A., Goebel, K., Simon, D. e Eklund, N. (2008a). "Damage propagation modeling for aircraft engine run-to-failure simulation." In Proceedings of the 2008 International Conference on Prognostics and Health Management, p. 1–9. International Conference On Prognostics And Health Management, Denver, CO, Oct 06-09.
- Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S. e Schwabacher, M. (2008b). "Metrics for Evaluating Performance of Prognostic Techniques." International Conference on Prognostics and Health Management, Denver, CO.
- Saxena, A., Celaya, J., Saha, B., Saha, S. e Goebel, K. (2009). "Evaluating Algorithm Performance Metrics Tailored for Prognostics." Proceedings of the IEEE Aerospace Conference, v. 1, n. 1, p. 1–13.
- Saxena, A., Celaya, J., Saha, B., Saha, S. e Goebel, K. (2010). "Metrics for offline evaluation of prognostic performance." International Journal of Prognostics and Health Management.
- Schafer, A.M. e Zimmermann, H.G. (2007). "Recurrent neural networks are universal approximators." International Journal of Neural Systems, v. 17, n. 5, p. 253–263.
- Schrauwen, B., Verstraeten, D. e Campenhout, J.V. (2007). "An overview of reservoir computing: theory, applications and implementations." In Proceedings of the 15th European Symposium on Artificial Neural Networks, v. 1, n. 1, p. 471–482.
- Schwabacher, M (2005). "A survey of data-driven prognostics." AIAA InfoTech Aerospace.

- Serapião, A. (2009). “Fundamentos de otimização por inteligência de enxames: uma visão geral.” *Revista Controle & Automacao*, v. 20, n. 3, p. 271–304.
- Sergio, A.T., Ludermir, T.B. (2012). “PSO for reservoir computing optimization.” *Proceeding ICANN'12 Proceedings of the 22nd international conference on Artificial Neural Networks and Machine Learning - Volume Part I*, v. 1, n. 1, p. 685–692.
- Sergio, A.T. (2013). “Reservoir Computing Optimization with PSO – Otimização de Reservoir Computing com PSO.” Master Thesis. Centro de Informática, Universidade Federal de Pernambuco.
- Sergio, A.T. e Ludermir, T.B. (2014). “Reservoir Computing optimization with a hybrid method.” *Neural Networks (IJCNN), International Joint Conference on*, v. 1, n. 1, p. 2653–2660.
- Serir, L., Ramasso, E. e Zerhouni, N. (2012). “An evidential evolving multimodeling approach for systems behavior prediction.” In *Annual conference of the PHM society*.
- Shi, Z. e Han, M. (2007). “Support vector echo-state machine for chaotic time-series prediction.” *IEEE Transactions on Neural Networks*, v. 18, n. 2, p. 359–372.
- Shimanek, L. (2003). “Battery Prognostics.” *Empfasis: A publication of the National Electronics Manufacturing Center of Excellence*, Disponível em: <<http://www.empf.org/html/empfasis/dec2003.pdf>> Acesso em 20 de julho de 2015.
- Si, X.S., Wang, W., Hu, C.H. e Zhou, D.H. (2011). “Remaining useful life estimation - a review on the statistical data driven approaches.” *European Journal of Operational Research*, v. 213, n. 1, p. 1–14.
- Siegel, D. (2009). “Evaluation of health assessment techniques for rotating machinery.” Unpublished master’s thesis, Division of Research and Advanced Studies of the University of Cincinnati.
- Sikorska, J.Z., Hodkiewicz, M. e Ma, L. (2011). “Prognostic modelling options for remaining useful life estimation by industry.” *Mechanical Systems and Signal Processing*, v. 25, n. 5, p. 1803–1836.
- Siqueira, H., Bocato, L., Attux, R. e Lyra, C. (2012). “Echo State Networks and Extreme Learning Machines: A Comparative Study on Seasonal Streamflow Series Prediction.” *Neural Information Processing, Lecture Notes in Computer Science*, v. 7664, n. 1, p. 491–500.
- Sloukia, F., El Aroussi, M., Medromi, H. e Wahbi, M. (2013). “Bearings prognostic using Mixture of Gaussians Hidden Markov Model and Support Vector Machine.” *Computer*

- Systems and Applications (AICCSA), ACS International Conference on, v. 1, n. 1, p. 1–4.
- Son, K. L., Fouladirad, M. e Barros, A. (2012). “Remaining useful life estimation on the non-homogenous gamma with noise deterioration based on gibbs filtering: A case study.” In IEEE int. conf. on prognostics and health management.
- Son, K. L., Fouladirad, M., Barros, A., Levrat, E. e Iung, B. (2013). “Remaining useful life estimation based on stochastic deterioration models: A comparative study.” *Reliability Engineering and System Safety*, v. 112, n. 1, p. 165 –175.
- Soualhi, A, Medjaher, K. e Zerhouni, N. (2014). “Bearing Health Monitoring Based on Hilbert--Huang Transform, Support Vector Machine, and Regression.” *Instrumentation and Measurement, IEEE Transactions on*, v. 1, n. 99, p. 1–10.
- Sreejith, B., Verma, A.K. e Srividya, A. (2008). “Fault diagnosis of rolling element bearing using time-domain features and neural networks.” *Proceedings of Third IEEE International Conference on Industrial and Information Systems ICIIS*. Kharagpur, India.
- Steil, J.J. (2004). “Backpropagation-decorrelation: online recurrent learning with $O(N)$ complexity.” In *Proc. IJCNN*.
- Sun, F., Hu, X., Zou, Y. e Li, S. (2011). “Adaptive unscented Kalman filtering for state of charge estimation of a lithium-ion battery for electric vehicles.” *Energy*, Volume 36, Issue 5, pp. 3531–3540.
- Sun, J., Zuo, H., Wang, W. e Pecht, M. (2012). “Application of a state space modeling technique to system prognostics based on a health index for condition-based maintenance.” *Mechanical Systems and Signal Processing*, v. 28, n. 1, p. 585 – 596.
- Suykens, J., Vandewalle, J. e De Moor, B. (1996). “Artificial neural networks for modeling and control of non-linear system.” Springer.
- Swanson, D.C., Spencer, J.M. e Arzoumanian, S.H. (2000). “Prognostic Modelling of Crack Growth in a Tensioned Steel Band.” *Mechanical Systems and Signal Processing*, v. 14, n. 5, p. 789–803.
- Tamilselvan, P. e Wang, P. (2013). “Failure diagnosis using deep belief learning based health state classification.” *Reliability Engineering and System Safety*, v. 115, n. 0, p. 124 – 135.

- Tereshko, V. (2000). "Reaction-diffusion model of a honeybee colony foraging behavior." *Parallel Problem Solving from Nature VI. Lecture Notes in Computer Science*, v. 1917, n. 1, p. 807–816.
- Tereshko, V. e Loengarov, A. (2005). "Collective decision-making in honey bee foraging dynamics." *Journal of Computing and Information Systems*, v. 9, n. 3, p. 1–7.
- Theodoridis, S. e koutroumbas, K. (2006). "Pattern recognition." San Diego, CA, Academic Press.
- Tian Z. (2009). "An artificial neural network approach for remaining useful life prediction of equipments subject to condition monitoring." *Reliability, Maintainability and Safety, 2009. ICRMS 2009. 8th International Conference on*, v. 1, n. 1, p. 143–148.
- Tian, Z. e Zuo, M.J. (2009). "Health condition prognostics of gears using a recurrent neural network approach." in: *Proceedings of the Reliability and Maintainability Symposium, Fort Worth, TX, USA*.
- Tian, Z., Wong, L. e Safaei, N. (2010). "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.
- Tian, X., Cao, Y.P. e Chen, S. (2011). "Process fault prognosis using a fuzzy-adaptive unscented Kalman predictor." *International Journal of Adaptive Control and Signal Processing*, v. 25, n. 9, p. 813–830.
- Tobon-Mejia, D.A, Medjaher, K., Zerhouni, N. e Tripot, G. (2011). "Hidden Markov Models for failure diagnostic and prognostic." *Prognostics and System Health Management Conference (PHM-Shenzhen)*, v. 1, n. 1, p. 1–8.
- ToolboxABC (2015). MATLAB code v2 of the basic ABC algorithm. Disponível em: <<http://mf.erciyes.edu.tr/abc>>. Acesso em: 04 março 2015.
- ToolboxESN (2015). Simple and very simple Matlab toolbox for Echo State Networks. Disponível em: <<http://reservoir-computing.org>>. Acesso em: 04 março 2015.
- Trunk, G.V. (1979). "A problem of dimensionality: a simple example." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 1, n. 1, p. 306–307.
- Turanoglu, E., Ozceylan, E. e Kiran, M.S. (2011). "Particle Swarm Optimization and Artificial Bee Colony Approaches to Optimize of Single Input-Output Fuzzy Membership Functions." *Proceedings of the 41st International Conference on Computers & Industrial Engineering*.

- Vachtsevanos, G. (2003). "Performance Metrics for Fault Prognosis of Complex Systems." Proceedings of the IEEE Systems Readiness Technology Conference (AUTOTESTCON), v. 1, n. 1, p. 341 – 345.
- Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A. e Wu, B. (2006). "Intelligent Fault Diagnosis and Prognosis for Engineering Systems." 1st ed. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Verstraeten, D., Schrauwen, B., D'Haene, M. e Stroobandt, D. (2007). "An experimental unification of reservoir computing methods." *Neural Networks*, v. 20, n. 1, p. 391–403.
- Wang, P. e Vachtsevanos, G. (2001). "Fault prognostics using dynamic wavelet neural networks Artificial Intelligence for Engineering Design." *Analysis and Manufacturing*, v. 15, n. 1, p. 349–365.
- Wang, W.Q., Golnaraghi, M.F. e Ismail, F. (2004). "Prognosis of machine health condition using neuro-fuzzy systems." *Mechanical Systems and Signal Processing*, v. 18, n. 4, p. 813–831.
- Wang, W. (2007). "A prognosis model for wear prediction based oil-based monitoring." *Journal of the Operational Research Society*, v. 58, n. 1, p. 887–893.
- Wang, W., Zhang, W., (2008). "An asset residual life prediction model based on expert judgments." *European Journal of Operational Research*, v. 188, n. 1, p. 496–505.
- Wang, T., Yu, J., Siegel, D. e Lee, J. (2008). "A similarity based prognostics approach for remaining useful life estimation of engineered systems." In *Int. conf. on prognostics and health management*, v. 1, n. 1, p. 1–6.
- Wang, L., Chu, J. e Mao, W., (2009). "A condition-based replacement and spare provisioning policy for deteriorating systems with uncertain deterioration to failure." *European Journal of Operational Research*, v. 194, n. 1, p. 184–205.
- Wang, T. (2010). "Trajectory Similarity Based Prediction for Remaining Useful Life Estimation." PhD Thesis, Department of Industrial Engineering, University Of Cincinnati.
- Wang, P., Youn, B. e Hu, C. (2012). "A generic probabilistic framework for structural health prognostics and uncertainty management." *Mechanical Systems and Signal Processing*, v. 28, n. 1, p. 622–637.

- Weiming, W., Bing, L., Min, L. e Houjun, W. (2014). “Prognostics of Lithium-Ion Batteries Based on the Verhulst Model, Particle Swarm Optimization and Particle Filter.” *Instrumentation and Measurement, IEEE Transactions on*, v. 63, n. 1, p. 2–17.
- Werbos, P.J. (1974). “Beyond regression: New tools for prediction and analysis in the behavioral sciences.” Tese de doutorado, Harvard University.
- Werbos, P.J. (1990). “Backpropagation through time: what it does and how to do it.” *Em Proceedings of the IEEE*, v. 78, n. 1, p. 1550–1560.
- Williams, R.J. e Zipser, D. (1989). “A learning algorithm for continually running fully recurrent neural networks.” *Neural Computation*, v. 1, n. 1, p. 270–280.
- Xiangjun, D. e Tongmin, J. (2012). “Reliability prediction based on degradation measure distribution and wavelet neural network.” *Prognostics and System Health Management (PHM), IEEE Conference on*, v. 1, n. 1, p. 1–5.
- Xiaochuang, T., Zili, W., Jian, M. e Huanzhen, F. (2012). “Study on fault detection using wavelet packet and SOM neural network.” *Prognostics and System Health Management (PHM), IEEE Conference on*, v. 1, n. 1, p. 1–5.
- Xi, Z., Jing, R., Wang, P. e Hu, C. (2013). A copula based sampling method for data-driven prognostics and health management. In *ASME 2013 international design engineering technical conferences and computers and information in engineering conference*.
- Xin, W., Xiao-yun, C., Ke-ju, X., Rong-min, Z., Qing-jun, P. e Lin, L. (2012). “Ice-coated AC transmission lines tension prognosis with autoregressive model. 3rd International Conference on System Science, Engineering Design and Manufacturing Informatization, v. 2, n. 1, p. 168–170.
- Xu, J., Wang, Y. e Xu, L. (2014). “PHM-Oriented Integrated Fusion Prognostics for Aircraft Engines Based on Sensor Data.” *Sensors Journal, IEEE*, v. 14, n. 4, p. 1124–1132.
- Xue, Y., Williams, D. e Qiu, H. (2011). Classification with imperfect labels for fault prediction. In *Proceedings of the first international workshop on data mining for service and maintenance*, v. 1, n. 1, p. 12–16.
- Ying, P., Ming, D. e Ming, J.Z. (2010). “Current status of machine prognostics in condition-based maintenance: a review.” *Int J AdvManuf Technol*, v. 50, n. 1, p. 297–313.
- Yinjiao, X., Ma, E.W.M., Tsui, K.L. e Pecht, M. (2012). “A case study on battery life prediction using particle filtering.” *Prognostics and System Health Management (PHM), IEEE Conference on*, v. 1, n. 1, p. 1–6.

- Yonglin, P., Baad, M.J., Yulei, J. e Yisheng, C. (2014). "Application of artificial neural network and multiple linear regression models for predicting survival time of patients with non-small cell cancer using multiple prognostic factors including FDG-PET measurements." *Neural Networks (IJCNN), International Joint Conference on*, v. 1, n. 1, p. 225–230.
- Yu, J. (2013). "A nonlinear probabilistic method and contribution analysis for machine condition monitoring." *Mechanical Systems and Signal Processing*, v. 37, n. 1, p. 293–314.
- Zedda, M. e Singh, R. (2002). "Gas turbine engine and sensor fault diagnosis using optimization techniques." *Journal of Propulsion and Power*, v. 18, n. 5, p. 1019–1025.
- Zein-Sabatto, S., Bodruzzaman, J. e Mikhail, M. (2013). "Statistical approach to online prognostics of turbine engine components." In *Southeastcon, 2013 proceedings of IEEE*, v. 1, n. 1, p. 1–6.
- Zhang, Y.X. e Randall, R.B. (2009). "Rolling element bearing fault diagnosis based on the combination of genetic algorithms and fast kurtogram." *Mechanical Systems and Signal Processing*, v. 23, n. 1, p. 1509–1517.
- Zhang, Y., Zhao, X., Liu, W., Zhang, J., Jia, Y. e Feng, T. (2011). "Research on gearbox wearing prognosis based on Gamma-State Space Model." *Reliability, Maintainability and Safety (ICRMS), 9th International Conference on*, v. 1, n. 1, p. 279–283.
- Zhang, Q., Hua, C. e Xu, G. (2014). "A mixture Weibull proportional hazard model for mechanical system failure prediction utilising lifetime and monitoring data." *Mechanical Systems and Signal Processing*, v. 43, n. 1–2, p. 103–112.
- Zhao, D. e Willett. (2011). "Comparison of data reduction techniques based on SVM classifier and SVR performance." In *Proc. SPIE, signal and data processing of small targets*, v. 8137, n. 1, p. 1–15.
- Zhigang, T. e Zuo, M.J. (2009). "Health condition prognostics of gears using a recurrent neural network approach." *Reliability and Maintainability Symposium, RAMS. Annual*, v. 1, n. 1, p. 460-465.
- Zhou, Y., Men, Z., Chen, X. e Wu, Z. (2012). "Helicopter engine performance prediction based on cascade-forward process neural network." *Prognostics and Health Management (PHM), IEEE Conference on*, v. 1, n. 1, p. 1–5.

- Zhu, X.L., Beauregard, G.T. e Wyse, L.L. (2007). "Real-time signal estimation from modified short-time Fourier transform magnitude spectra." *IEEE Transactions on Audio Speech and Language Processing*, v. 15, n. 1, p. 1645–1653.
- Zhuang, Z.Y., Churilov, L., Burstein, F. e Sikaris, K. (2009). "Combining data mining and case-based reasoning for intelligent decision support for pathology ordering by general practitioners." *European Journal of Operational Research*, v. 195, n. 3, p. 662–675.
- Zio, E. e Maio, F. (2010). "A fuzzy similiarity-based Method for Failure Detection and Recovery Time Estimation." *International Journal of Performability Engineering*, v. 6, n. 5, p. 407–424.

APÊNDICES

APÊNDICE A – MÉTRICAS DE PROGNÓSTICO

Neste apêndice apresentam-se as métricas que podem ser usadas na avaliação de desempenho de algoritmos de prognóstico, bem como realizar uma comparação quantitativa entre diferentes métodos.

A.1 INTRODUÇÃO

Na comparação quantitativa dos resultados de diferentes métodos de prognóstico, torna-se necessária a definição de métricas que permitam avaliar o desempenho dos modelos para diferentes sistemas, regimes de operação, modos de falha, etc. Indicadores para monitoração, detecção de falhas e diagnóstico de sistemas estão estabelecidas (Hines *et al.*, 2008). Entre estes indicadores tem-se a precisão, robustez, detectabilidade de falhas, medidas de incertezas, tempo de detecção de falha e taxa de alarmes falhas.

Entretanto, os indicadores de prognóstico são menos conhecidos do que as métricas de diagnóstico. As pesquisas em indicadores de desempenho de modelos de prognóstico se focam em três áreas: desempenho do algoritmo, rapidez computacional e indicadores econômicos. Obviamente é desejável ter algoritmos de prognóstico que realizem estimações precisas e exatas da RUL. Porém, pela natureza de prognóstico, essas estimações são realizadas em tempo real, o que significa que o algoritmo deve ter um tempo computacional curto.

Nos sistemas de prognóstico é particularmente importante dispor de aquisição de dados em tempo real que permita a estimação da RUL durante o uso. Na literatura são encontrados indicadores que caracterizam o custo computacional de algoritmos de prognóstico, entre eles estão a complexidade (Byington *et al.*, 2005), especificação (Vachtsevanos, 2003) e tempo de CPU (Saxena *et al.*, 2008b). Indicadores econômicos como Retorno sobre Investimento (ROI – *Return on Investment*) são considerados como desempenho do modelo na análise custo benefício (Goodman *et al.*, 2005, He *et al.*, 2006, Saxena *et al.*, 2008, Wood e Goodman, 2006, e Banks e Merenich, 2007). Em geral essas análises mostram que um sistema de prognóstico diminui os custos de manutenção enquanto aumenta a disponibilidade e melhora a segurança.

O estudo de indicadores de desempenho de algoritmos de prognóstico é mais interessante na área de pesquisa e nesse contexto foram objetos de estudos nos últimos anos (Line e Clements, 2006, Vachtsevanos, 2003, Saxena *et al.*, 2008b, Banks e Merenich, 2007, Kacprzyński *et al.*, 2004, Saxena *et al.*, 2009). Os resultados de cada estudo forneceram diferentes definições de indicadores e alguns concorrentes, porém os indicadores estão entre duas categorias: precisão e exatidão. As medições tradicionais de erro não são consideradas nas medidas de níveis de precisão e exatidão. As métricas mais usadas em prognóstico foram propostas por Vachtsevanos *et al.* (2006) e Saxena *et al.* (2010). A seguir descreve-se algumas das principais métricas mais usadas.

Para avaliar as predições, define-se o erro de predição dado pela Equação A.1. Este erro será utilizado na maioria das métricas de prognóstico.

$$\varepsilon = \hat{t}_{RUL} - t_{RUL} \quad (\text{A.1})$$

Onde: ε é o erro de predição, \hat{t}_{RUL} é o resultado produzido pelo algoritmo de prognóstico, e t_{RUL} é o valor verdadeiro de RUL fornecido no conjunto de dados de teste.

A.2 MÉTRICAS DE EXATIDÃO

A.2.1 Exatidão

Um valor de exatidão perto de zero significa que as predições não são boas, enquanto uma exatidão próximo de 1 corresponde a boas predições. A exatidão é descrita através da Equação A.2, que foi apresentada por Vachtsevanos *et al.* (2006).

$$E = \frac{1}{M} \sum_{m=1}^M e^{-\frac{|\varepsilon_m|^\alpha}{t_{RUL}}} \quad (\text{A.2})$$

Onde: $\alpha \in [1,2]$, M é o número total de máquinas, t_{RUL} é o valor real da RUL da máquina m , e ε_m é o erro de predição para a máquina m .

A.2.7 PHM08

Para avaliar os resultados da competição PHM08 (realizada na primeira conferência de PHM no ano 2008) foi definida a métrica mostrada na Equação A.3. A função de pontuação para os dados é exponencial e assimétrica ao redor do RUL real para penalizar fortemente estimações atrasadas de RUL e valorizar as RULs antecipadas (Saxena *et al.*, 2008a).

$$PHM08 = \begin{cases} \sum_{m=1}^M e^{-\left(\frac{\varepsilon_m}{a_1}\right)} - 1 & \text{para } \varepsilon < 0 \\ \sum_{m=1}^M e^{\left(\frac{\varepsilon_m}{a_2}\right)} - 1 & \text{para } \varepsilon \geq 0 \end{cases} \quad (\text{A.3})$$

Onde: a_1 e a_2 são parâmetros para controlar a preferência assimétrica. Para comparação com outros algoritmos baseados no repositório de prognóstico pode-se utilizar os valores de $a_1 = 10$ e $a_2 = 13$ (p. ex., $I = [-10, 13]$). ε_m é o erro de predição para a máquina m .

A.2.8 MSE, MAE e RMSE

Indicadores de desempenho bastante usados em prognóstico de falhas são o Erro Quadrático Médio (MSE – *Mean Square Error*) representado pela Equação A.4, o Erro Absoluto Médio (MAE – *Mean Absolute Error*) descrito pela Equação A.5, e a raiz quadrada do erro quadrático médio (RMSE – *Root Mean Square Error*) cujo valor é calculado com a formula mostrada na Equação 3.8.

$$MSE = \frac{1}{M} \sum_{m=1}^M \varepsilon_m^2 \quad (\text{A.4})$$

$$MAE = \frac{1}{M} \sum_{m=1}^M |\varepsilon_m| \quad (\text{A.5})$$

Onde: ε_m representa o erro de predição da máquina m , e M é o número total de máquinas.

A.2.8 Taxa Falsos Positivos e Falsos Negativos (FPR e FNR)

Os erros de predição podem ser apresentados através de gráficos de histogramas. Para uma melhor avaliação são considerados os cálculos de falsos positivos e falsos negativos. Para o cálculo define-se os limiares de falsos positivos e falsos negativos como mostrado na Figura A.1. No estudo de caso apresentado neste trabalho o intervalo de predições corretas foi $I = [-10, 13]$, e fora desse intervalo são consideradas predições antecipadas ou atrasadas. Este intervalo é bastante usado para calcular o índice de comparação (Saxena *et al.*, 2008a), o qual deve ser o menor possível. No contexto de PHM, geralmente é desejado ter estimações de RUL antecipadas mais do que as atrasadas, pois o motivo principal é conhecer as falhas antecipadamente e evitar falhas.

Os falsos positivos (FP – *False Positive*) e falsos negativos (FN – *False Negative*) são calculadas pelas Equação A.6 e Equação A.7 respectivamente. Os limiares de falso positivo e negativo são definidos pelo usuário.

$$FP = \begin{cases} 1 & \varepsilon > t_{FP} \\ 0 & \text{Outro} \end{cases} \quad (\text{A.6})$$

$$FN = \begin{cases} 1 & \varepsilon < t_{FN} \\ 0 & \text{Outro} \end{cases} \quad (\text{A.7})$$

Onde: FP é o falso positivo, FN é o falso negativo, t_{FP} é o limiar de falso positivo, t_{FN} é o limiar de falso negativo, e ε representa o erro de predição.

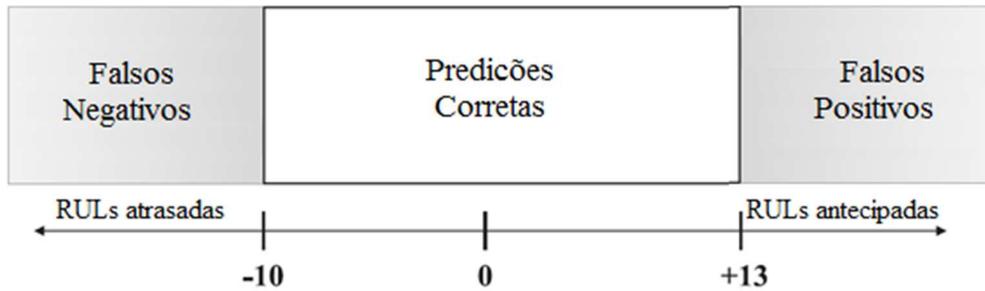


Figura A.1- Limiar de falsos positivos e negativos.

A.2.3 Porcentagem do Erro Médio Absoluto (MAPE)

A Porcentagem do Erro Médio Absoluto (MAPE – *Mean Absolute Percentage Error*) é ilustrada na Equação A.8, que quantifica o erro médio em porcentagem.

$$MAPE = \frac{1}{M} \sum_{m=1}^M \left| \frac{100\varepsilon_m}{t_{RUL}} \right| \quad (\text{A.8})$$

Onde: M é o número total de máquinas, t_{RUL} é o valor real da RUL da máquina m , e ε é o erro de predição.

A.3 MÉTRICAS DE PRECISÃO

A.2.2 Erro Médio (ME)

O erro médio (ME – *Mean Error*) é apresentado na Equação A.9.

$$\bar{\varepsilon} = \frac{1}{M} \sum_{m=1}^M \varepsilon_m \quad (\text{A.9})$$

Onde: M é o número total de máquinas, e ε é o erro de predição.

A.2.2 Desvio Médio Absoluto (MAD)

O desvio médio absoluto (MAD – *Mean Absolute Deviation*) é uma medida de dispersão do erro de predição, e é apresentado na Equação A.10.

$$\bar{\varepsilon} = \frac{1}{M} \sum_{m=1}^M |\varepsilon_m - \bar{m}|, \quad \bar{m} = \text{mediana}(\varepsilon_m) \quad (\text{A.10})$$

Onde: M é o número total de máquinas, e ε é o erro de predição.

A.2.2 Precisão

Essa métrica descrita na Equação A.11 foi apresentado por Saxena *et al.* (2010) e quantifica a dispersão do erro de prognóstico ao redor da média.

$$P = \sqrt{\frac{\sum_{m=1}^M (\varepsilon_m - \bar{\varepsilon})^2}{M}} \quad (\text{A.11})$$

Onde: M é o número total de máquinas, e ε é o erro de predição, e $\bar{\varepsilon}$ é o erro médio.

A.4 MÉTRICAS DE PROGNÓSTICO

A.2.4 Horizonte de prognóstico (PH)

O Horizonte de Prognóstico (PH – *Prognostic Horizon*) é definido pela Equação A.12 e Equação A.13 como a primeira predição de RUL que satisfaça o critério do limite α .

$$PH = t_E - t_{i_\alpha} \quad (\text{A.12})$$

$$t_{i_\alpha} = \min \{ t_i \mid t_i \in [t_P, t_{EoUP}], r_i^* - \alpha.t_E \leq r_i \leq r_i^* + \alpha.t_E \} \quad (\text{A.13})$$

Onde: i_α é o índice da primeira predição de RUL que satisfaz o critério do limite α . Para um modelo de predição de RUL que forneça o ponto de estimação de RUL r_i s cada tempo de amostragem t_i , o critério de limite α avalia se r_i está dentro do limite α da RUL real r_i^* ($\equiv t_E - t_i$). Na Figura A.2 mostra-se um exemplo de horizonte de prognóstico, onde as três linhas retas paralelas são $0,8^*(\text{RUL real})$, RUL real , e $1,2^*(\text{RUL real})$, a curva de RUL estimada intercepta à primeira linha reta no ciclo 100 aproximadamente, começando neste ponto a medição do PH até o último ciclo de RUL real.

A.2.5 Medida de desempenho (α - λ)

Esta medida mostrada na Equação A.14 é definida como uma métrica binária que avalia se a exatidão de predição em um determinado tempo t_λ está dentro do limite α . O tempo t_λ é uma fração de tempo entre t_P e t_{RUL} . O limite α é expressado como uma porcentagem do valor atual de RUL $r(i_\lambda)$ no instante t_λ .

$$P_{\alpha-\lambda} = \begin{cases} 1 & \text{se } \hat{t}_{RUL} \in [(1-\alpha)t_{RUL}, (1+\alpha)t_{RUL}] \\ 0 & \text{se } \textit{Outro} \end{cases} \quad (\text{A.14})$$

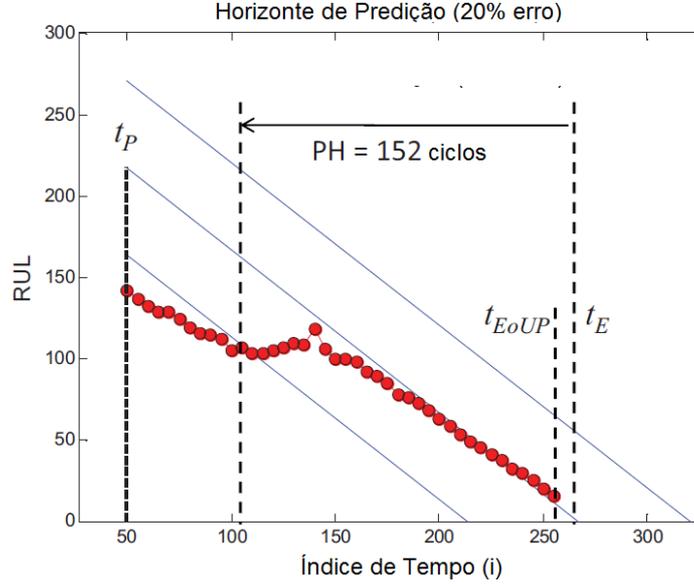


Figura A.2- Horizonte de Predição.

A.2.6 Medida de exatidão relativa (ER)

Saxena *et al.* (2010) apresentaram e formularam a métrica da Equação A.15. Essa métrica permite avaliar a exatidão da RUL estimada no instante de tempo $t = \lambda_c \cdot T \forall \lambda_c \in [0,1]$.

$$ER = 1 - \frac{|RUL_{real}(t) - RUL(t)|}{RUL_{real}(t)} \quad (\text{A.15})$$

A.2.7 Convergência (CV)

A convergência é uma meta métrica definida para quantificar a taxa na qual qualquer métrica (M) como a exatidão ou a precisão melhora com o tempo. Definido pela Equação A.16 como a distância entre a origem e o centróide de uma área sob uma curva para uma métrica. Basicamente é uma medida de taxa de convergência.

$$C_M = \sqrt{(X_C - t_P)^2 + Y_C^2} \quad (\text{A.16})$$

Onde: C_M é a distância euclidiana entre o centro de massa (X_C, Y_C) e $(t_P, 0)$, $M(i)$ é uma métrica de precisão ou exatidão não negativa de predição como o valor do tempo variável, e (X_C, Y_C) é o centro de uma área sob uma curva $M(i)$ entre t_P e t_{EoUP} definidos pela Equação A.17.

$$X_C = \frac{\frac{1}{2} \sum_{i=P}^{EoUP} (t_{i+1}^2 - t_i^2) M(i)}{\sum_{i=P}^{EoUP} (t_{i+1} - t_i) M(i)}, \quad Y_C = \frac{\frac{1}{2} \sum_{i=P}^{EoUP} (t_{i+1} - t_i) M(i)^2}{\sum_{i=P}^{EoUP} (t_{i+1} - t_i) M(i)} \quad (\text{A.17})$$

Onde: C_M é a distância euclidiana entre o centro de massa (X_C, Y_C) e $(t_P, 0)$, $M(i)$ é uma

APÊNDICE B – IMPLEMENTAÇÃO COMPUTACIONAL

Neste apêndice apresenta-se a implementação computacional desenvolvida em *Matlab*. Através desta ferramenta é possível acompanhar o processo de aquisição de dados, seleção do conjunto de dados. Também pode ser realizada a otimização dos parâmetros do reservatório, treinamento dos pesos da ESN e visualizar os gráficos de estimação de RUL. Os algoritmos desenvolvidos são baseados na ToolboxESN (2015) e ToolboxABC (2015), os códigos fontes dos algoritmos desenvolvidos nesta tese encontra-se disponível no link: <https://sourceforge.net/projects/esn-abc/>, no mesmo link tem um vídeo tutorial do funcionamento dos algoritmos. A seguir são apresentados os principais passos implementados no desenvolvimento dos algoritmos nesta tese.

B.1 AQUISIÇÃO DE DADOS

A aquisição de dados dos quatro conjuntos de dados foi desenvolvida com o intuito de analisar os dados de cada conjunto, para este fim dispõe-se da interface mostrada na Figura B.1. Nesta interface é possível escolher o conjunto de dados a analisar podendo ser 1, 2, 3 ou 4. Também é possível selecionar que tipo de gráfico se deseja, tem-se as tendências, os histogramas e os estados de operação.

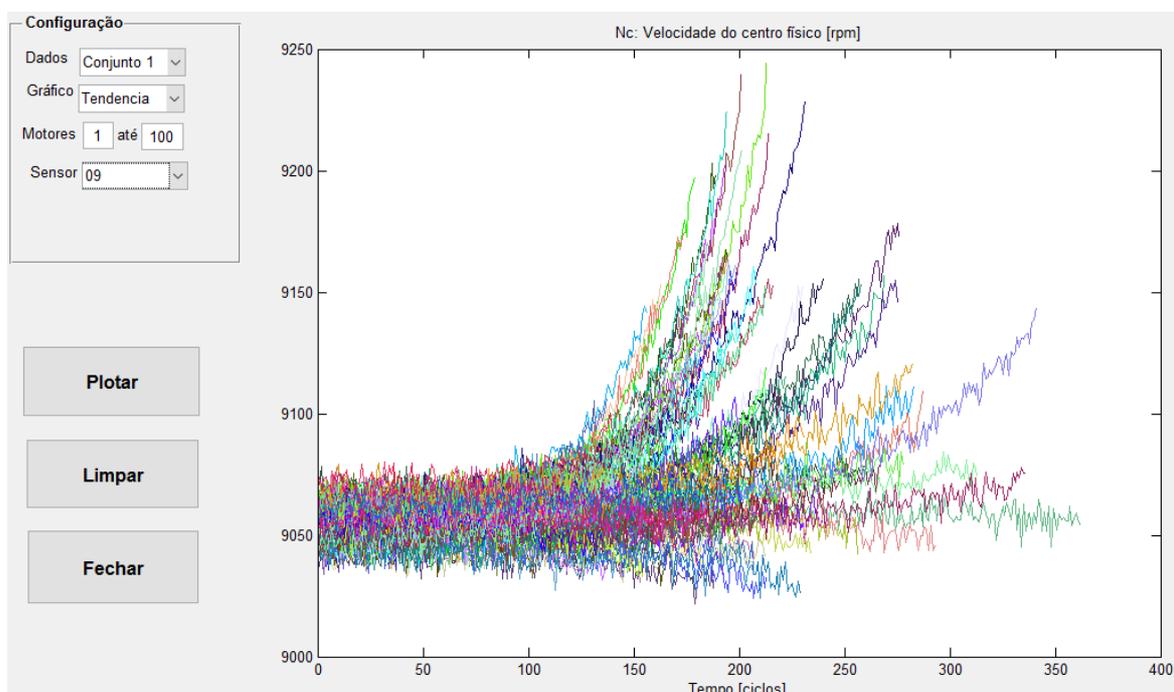


Figura B.1- Análise de tendências das variáveis monitoradas.

Na análise de tendências é possível selecionar a faixa dos motores turbofan, e o valor máximo depende da quantidade de casos (turbinas) que possui cada conjunto de dados. Para a faixa de turbinas selecionadas, escolher um sensor para visualizar como o seu valor varia no transcurso do tempo (ciclos de operação) e analisar as suas tendências. Esta ferramenta é necessária para definir as variáveis a serem utilizadas como entradas da rede ESN, foram escolhidas as variáveis que apresentavam tendência de degradação a cada ciclo.

A análise de ocorrência de falha através do histograma como mostrado na Figura B.2 permite conhecer a distribuição de RUL das turbinas dos diferentes conjuntos de dados. A quantidade de modos de operação das turbinas é possível visualizar na Figura B.3. Se um conjunto de dados possui mais de um modo de operação, as variáveis de condição de operação devem ser consideradas no modelo de prognóstico de falhas.

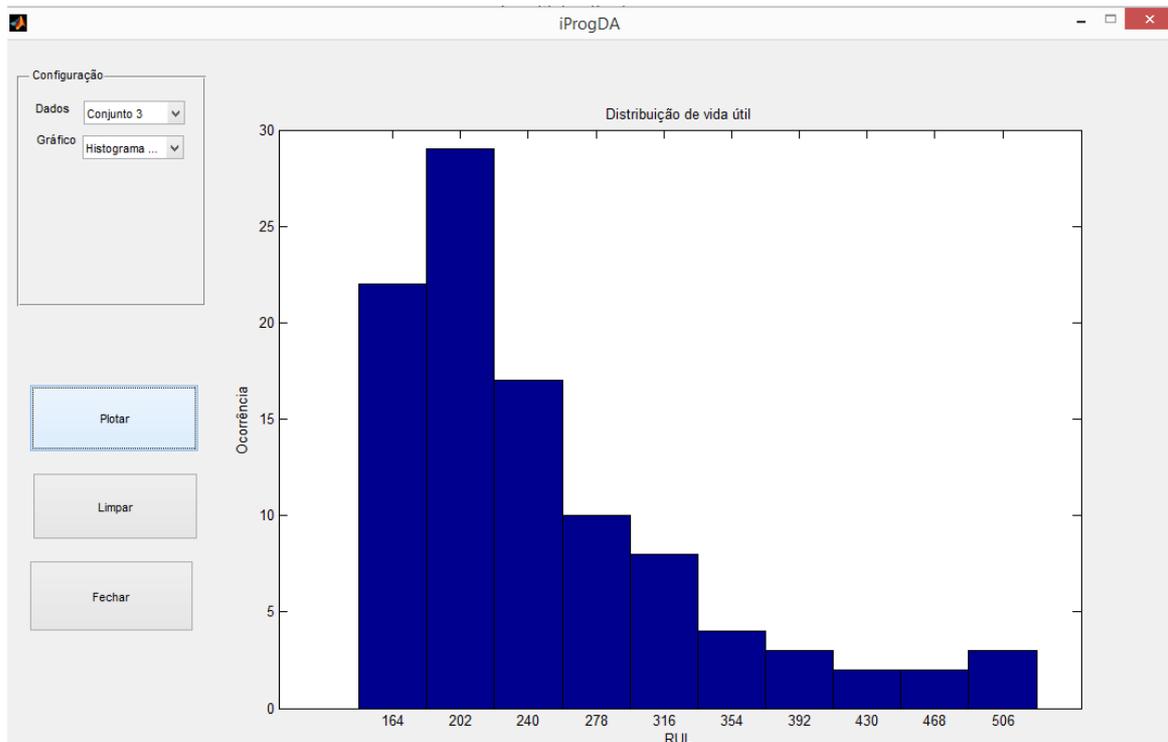


Figura B.2- Análise da ocorrência de RUL.

B.2 TREINAMENTO CLÁSSICO DE UMA ESN

A ferramenta para treinamento de uma ESN através do método clássico é mostrada na Figura B.4. Nesta interface escolheu-se a opção ESN clássica, podendo-se alterar os principais parâmetros do reservatório de uma ESN. Clicando no comando “executar treinamento”,

inicia-se o processo de treinamento, fornecendo na saída um arquivo com os parâmetros e pesos da rede treinada.

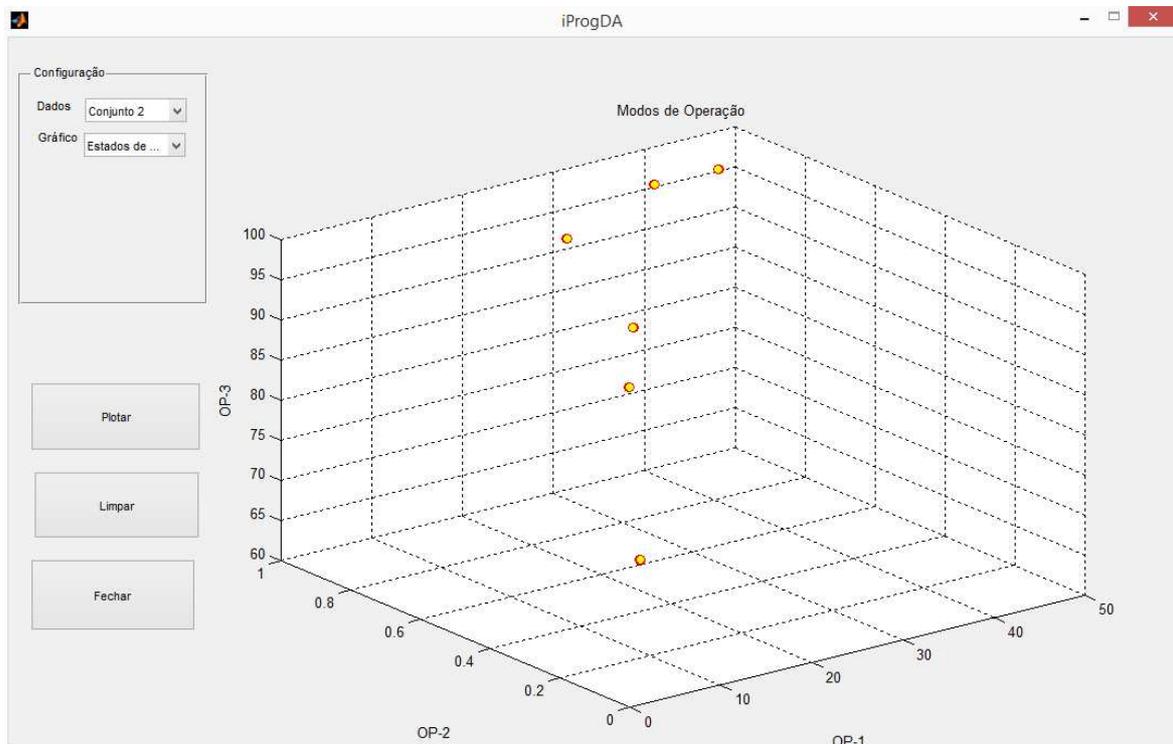


Figura B.3- Análise dos modos de operação.

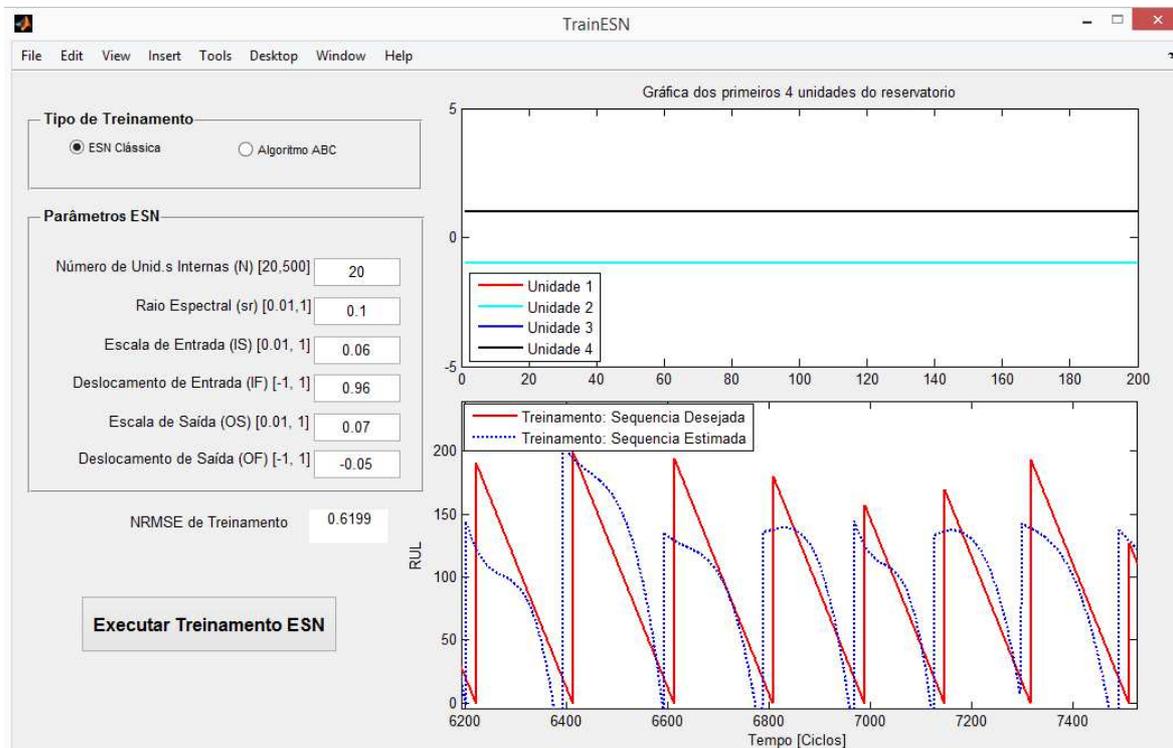


Figura B.4- Treinamento de ESN através do método clássico.

B.2.1 Seleção do Conjunto de dados

O código que permite a seleção do conjunto de dados a usar no processo de validação é apresentado na Figura B.5, podendo-se escolher um dos quatro conjuntos e os sensores que foram selecionados que são as entradas da rede.

```
%% Carregando o conjunto de dados
cnj = 1;
switch cnj
case 1
    trn = load('Database\train_FD001.txt');
    tst = load('Database\test_FD001.txt');
    rul = load('Database\RUL_FD001.txt');
case 2
    trn = load('Database\train_FD002.txt');
    tst = load('Database\test_FD002.txt');
    rul = load('Database\RUL_FD002.txt');
case 3
    trn = load('Database\train_FD003.txt');
    tst = load('Database\test_FD003.txt');
    rul = load('Database\RUL_FD003.txt');
case 4
    trn = load('Database\train_FD004.txt');
    tst = load('Database\test_FD004.txt');
    rul = load('Database\RUL_FD004.txt');
end
sensors = 5 + [2 3 4 7 8 9 11 12 13 14 15 17 20 21]; % Sensores selecionados
sequenceLength = length(rul); % Numero de dados de treinamento
nInputUnits = length(sensors); % Nro de nós de entrada da ESN
nOutputUnits = 1; % Nro de nós de saída da ESN
```

Figura B.5- Aquisição e seleção do conjunto de dados.

B.2.2 Parametrização e Treinamento

Os parâmetros da ESN a serem alterados durante o treinamento são mostrados na Figura B.6. Através do ajuste dos parâmetros é possível encontrar uma ESN com bom desempenho. Com os parâmetros definidos o próximo passo é a criação da ESN, este procedimento é mostrado na Figura B.7.

```
%% ===== PARAMETRIZAÇÃO DA ESN =====
nInternalUnits = 100; % Nro Neuronios do Reservatório
sr = 0.01; % Raio Espectral
iscale = 0.06*ones(nInputUnits,1); % Escala de Entrada
ishift = 0.96*ones(nInputUnits,1); % Escala de Entrada
oscale = 0.007; % Escala de Saída
oshift = -0.05; % Deslocamento de Saída
%=====
```

Figura B.6- Parametrização da ESN.

```

%% Gerar a rede ESN
esn = generate_esn(nInputUnits, nInternalUnits, nOutputUnits, ...
    'spectralRadius',sr,'inputScaling', iscale,'inputShift', ishift, ...
    'teacherScaling',oscale,'teacherShift',oshift,'feedbackScaling', 0, ...
    'type', 'plain_esn');

```

Figura B.7- Criação da rede ESN.

Com a rede criada procede-se a etapa de treinamento através da função “*train_esn*” como mostrado na Figura B.8. A rede treinada com todos seus parâmetros e pesos pode ser armazenada para o uso posterior na etapa de teste.

```

%% Treinar a ESN
nForgetPoints = 20 ; % Desprezando os primeiros "nForgetPoints" pontos
[trainedEsn stateMatrix] = train_esn(trainInputSequence, ...
    trainOutputSequence, esn, nForgetPoints) ;

%% Salvar a ESN Treinada
save_esn(trainedEsn, 'esn_prognostic');

```

Figura B.8- Treinamento e armazenamento da ESN

B.3 OTIMIZAÇÃO ABORDAGEM HÍBRIDA ESN-ABC

A abordagem ESB-ABC consiste no ajuste de pesos de uma rede ESN através do algoritmo ABC, os parâmetros necessários da ESN e do ABC através de interface gráfica para o funcionamento do algoritmo são mostrados na Figura B.9. No experimento realizado o valor de N definiu-se como constante para cada experimento.

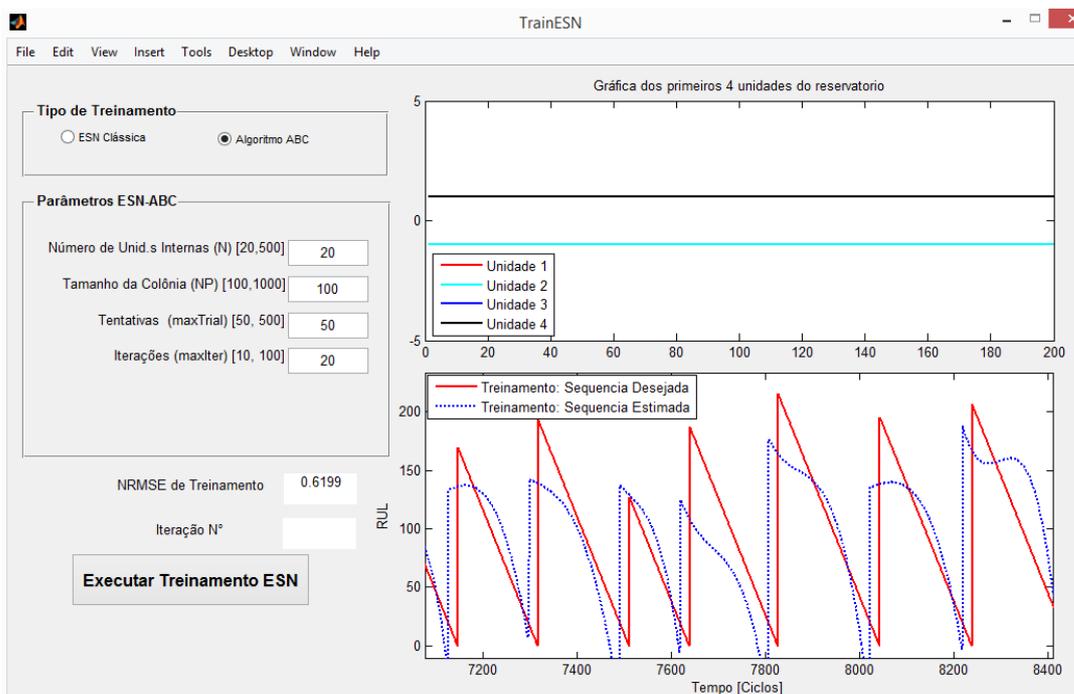


Figura B.9- Treinamento de ESN através do algoritmo ABC.

A busca pelas soluções ótimas é realizada através da função de aptidão mostrada na Figura B.10. Neste processo é executado o treinamento e teste para cada fonte de alimento do algoritmo ABC obtendo no final o valor da função de aptidão que depende do MSE do treinamento e o MSE teste.

```

function ObjVal = abcAptidao(alim,varargin)
global trainInputSequence trainOutputSequence testInputSequence testOutputSequence
na = size(alim,1); % Alim [naxnp]: na- nro de alimentos, e np- número de parametros
% ObjVal = [r is if os of]
for i = 1:na % Alim representa o conjunto de solucoes (S)
    sr = alim(i,1); % Captura a raio espectral de (S)
    viscale = alim(i,2); % Captura a escala de entrada de (S)
    vishift = alim(i,3); % Captura o deslocamento de entrada de (S)
    oscale = alim(i,4); % Captura a escala de saída de (S)
    oshift = alim(i,5); % Captura o deslocamento de saída de (S)
    Wf = alim(i,6:nInternalUnits^2+5); % Captura o vetor de pesos de (S)
    W = vec2mat(Wf,nInternalUnits); % Converte vetor em matriz
    iscale = viscale*ones(nInputUnits,1); % IS x nro de entradas
    ishift = vishift*ones(nInputUnits,1); % IF x nro de entradas
    esn = generate_esn(nInputUnits, nInternalUnits, nOutputUnits, ...
        'spectralRadius',sr,'inputScaling', iscale,'inputShift', ishift, ...
        'teacherScaling',oscale,'teacherShift',oshift,'feedbackScaling', 0, ...
        'type', 'plain_esn');
    esn.internalWeights = esn.spectralRadius * W; % W reescalada pelo sr
    %% Treinar a ESN
    nForgetPoints = 20 ; % Desprezamos os primeiros "nForgetPoints" pontos
    [trainedEsn stateMatrix] = train_esn(trainInputSequence, trainOutputSequence, esn)
    %% Calcular a saída da ESN Treinada para os dados de Treinamento e Teste
    nForgetPoints = 0 ; % Desprezamos os primeiros "nForgetPoints" pontos
    predictedTrainOutput = test_esn(trainInputSequence, trainedEsn, nForgetPoints);
    predictedTestOutput = test_esn(testInputSequence, trainedEsn, nForgetPoints);
    %% Calcular o erro NRMSE de treinamento e teste
    trainError = compute_NRMSE(predictedTrainOutput, trainOutputSequence);
    testError = compute_NRMSE(predictedTestOutput, testOutputSequence);
    %% Funcao de Aptidao a ser minimizada
    ObjVal(i) = trainError + abs(trainError-testError); % Valor de fAptidao
end

```

Figura B.10- Função de aptidão da abordagem ESN-ABC.

B.4 RESULTADOS DE PROGNÓSTICO

O prognóstico de RUL é realizado através da interface gráfica mostrado na Figura B.11, usando a rede treinada e fornecendo o conjunto de dados de teste. Esta ferramenta carrega o arquivo de configuração de parâmetros e pesos gerados na etapa de treinamento. Os resultados de prognóstico da implementação computacional são: a sequência real e estimada dos dados de teste, a RUL estimada, RUL real e o erro de predição, e finalmente os valores das métricas de prognóstico.

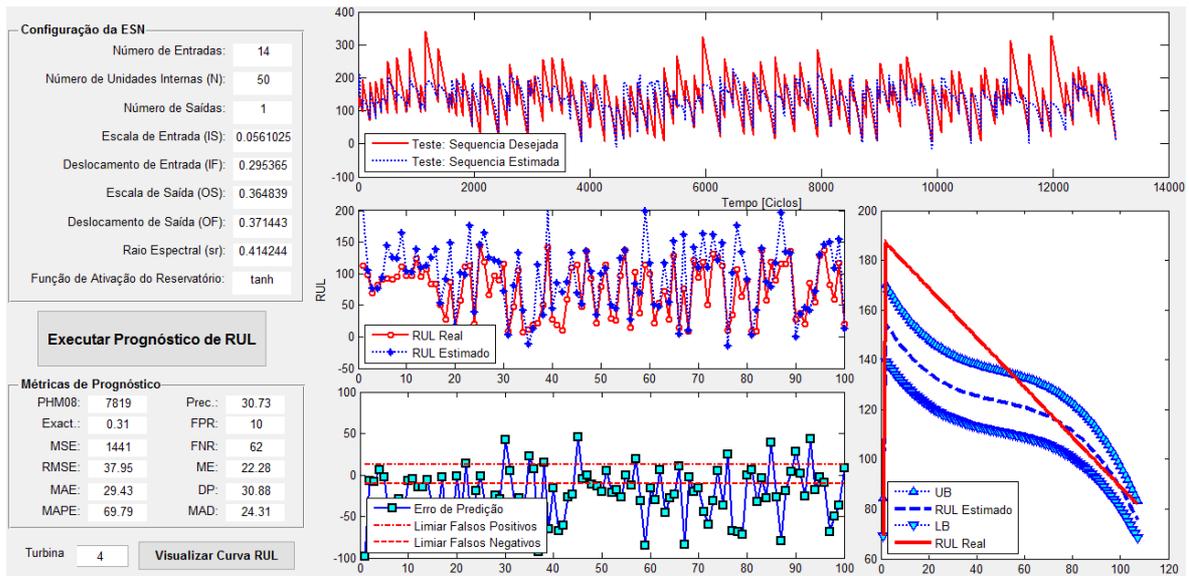


Figura B.11- Prognóstico de RUL.

APÊNDICE C – PUBLICAÇÕES REALIZADAS E SUBMETIDAS

Neste apêndice apresenta-se a relação de trabalhos publicados e o trabalho submetido a *journal* internacional, resultados obtidos desta tese de doutorado.

C.1 TRABALHOS PUBLICADOS

C.1.1 Manutenção Baseada em Condição

- Amaya, E. J.; Álvares, A. e Gudwin, R. (2009), An Expert System for Fault Diagnostics in Condition Based Maintenance, 20th International Congress of Mechanical Engineering (COBEM), Gramado-RS, Brasil.
- Amaya, E. J.; Álvares, A. e Gudwin, R. (2009), Open System Architecture for Condition Based Maintenance Applied to Hydroelectric Power Plant, The 8th Latin-American Congress on Electricity Generation and Transmission (CLAGTEE), Ubatuba-SP, Brasil.
- Amaya, E. J.; Álvares, A. (2010), The Application of Expert Knowledge to Implement an Intelligent Maintenance System, Asset Management Conference (ICOMS), June 21-25, Adelaide, South Australia, Australia.

C.1.2 Diagnóstico de Falhas

- Amaya, E. J.; Álvares, A. e Tonaco, R.P. (2009), Diagnóstico de Falhas de Turbinas Hidráulicas usando Sistemas Especialistas, 9º Congreso Iberoamericano de Ingeniería Mecánica (CIBIM), Las Palmas de Gran Canaria, España.
- Amaya, E. J.; Álvares, A. (2010), Detecção e Isolação de Falhas Baseado na Técnica de Agrupamento Possibilístico, Congresso Nacional de Engenharia Mecânica (VI CONEM), 18-21 agosto, Campina Grande, Paraíba, Brasil.
- Amaya, E. J.; Álvares, A. (2010), "SIMPREGAL: An expert system for real-time fault diagnosis of hydrogenerators machinery," *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, vol., no., pp.1,8, 13-16 Sept. 2010.
- Amaya, E. J.; Álvares, A. (2012), Detecção Falhas em Unidades Geradoras Hidráulicas usando Instrumentação Inteligente, Congresso Nacional de Engenharia Mecânica (VII CONEM), 31 julho-03 agosto, São Luiz, Maranhão, Brasil.

- Amaya, E. J.; Álvares, A.J., "Expert system for power generation fault diagnosis using hierarchical meta-rules," *Emerging Technologies & Factory Automation (ETFA)*, 2012 *IEEE 17th Conference on* , vol., no., pp.1,8, 17-21 Sept. 2012.

C.1.3 Prognóstico de Falhas

- Amaya, E. J.; Álvares, A. (2011), A data-driven approach for predicting the remaining useful life of hydroelectric equipments, 21st International Congress of Mechanical Engineering (COBEM). October 24-28, Natal, RN, Brazil.

C.2 TRABALHO SUBMETIDO EM JOURNAL INTERNACIONAL

Amaya, E. J.; Álvares, A.J., "Failure Prognostic based on Echo State Networks Optimized by Artificial Bee Colony," Submitted to: *International Journal of Prognostics and Health Management (IJPHM)*, 2015.