

**UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE FÍSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA AMBIENTAL**

**PREENCHIMENTO DE FALHAS DE DADOS
MICROMETEOROLÓGICOS UTILIZANDO
TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL**

Thiago Meirelles Ventura

Prof.^a Dr.^a Claudia Aparecida Martins

Cuiabá-MT, Março de 2012

UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE FÍSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA AMBIENTAL

**PREENCHIMENTO DE FALHAS DE DADOS
MICROMETEOROLÓGICOS UTILIZANDO
TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL**

Thiago Meirelles Ventura

Dissertação apresentada ao Programa de Pós-Graduação em Física Ambiental da Universidade Federal de Mato Grosso, como parte dos requisitos para obtenção do título de Mestre em Física Ambiental.

Prof.^a Dr.^a Claudia Aparecida Martins

Cuiabá-MT, Março de 2012

UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE FÍSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA AMBIENTAL

FOLHA DE APROVAÇÃO

TÍTULO: PREENCHIMENTO DE FALHAS DE DADOS MICROMETEOROLÓGICOS UTILIZANDO TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

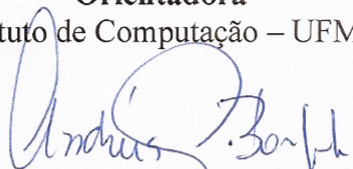
AUTOR: THIAGO MEIRELLES VENTURA

Dissertação de Mestrado defendida e aprovada em 09 de março de 2012, pela comissão julgadora:



Prof. Dra. Claudia Aparecida Martins
Orientadora

Instituto de Computação – UFMT



Prof. Dra. Andréia Gentil Bonfante
Examinadora Interna

Instituto de Computação – UFMT



Prof. Dr. Josiel Maimone de Figueiredo
Examinador Interno

Instituto de Computação – UFMT



Prof. Dra. Maria Carolina Monard
Examinadora Externa

Instituto de Ciências Matemáticas e de Computação– USP

V468p

Ventura, Thiago Meirelles

Preenchimento de Falhas de Dados Micrometeorológicos
Utilizando Técnicas de Inteligência Artificial. / Thiago Meirelles
Ventura. Cuiabá: UFMT, 2012.

73 fls.

Dissertação - Mestrado em Física Ambiental - UFMT.

Orientadora: Profa. Dra. Claudia Aparecida Martins

1. Preenchimento de Falhas. 2. Dados Micrometeorológicos.
3. Redes Neurais Artificiais. 4. Algoritmos Genéticos.
5. Seleção de Atributos. I. Título.

CDU 53:504

Dedicatória

Aos meus pais, Muriacy e Virgínia, que sempre deram prioridade aos meus estudos.

À minha esposa, Rosângela, que sempre está ao meu lado.

À Deus por ter me proporcionado todas as graças em minha vida.

Agradecimentos

- Aos meus pais eu agradeço por todo carinho e atenção que sempre me deram, por todas as lições durante a minha vida e por todo apoio que me dão nos estudos.
- A minha esposa Rosângela por caminhar junto comigo, tanto nos bons momentos quanto nos momentos difíceis. Muito obrigado por me dar forças em seguir em frente nesses novos desafios e por tudo que tem feito por mim.
- Aos meus irmãos que servem de exemplo para mim.
- Ao meu amigo Adriano Crestani Campos pela ajuda e paciência nesse período do mestrado.
- A professora Claudia Aparecida Martins pela paciência e compreensão durante a orientação desta dissertação. Aprendi muito nesse processo e espero continuar aprendendo nos próximos projetos.
- Ao professor Josiel Maimone de Figueiredo por me indicar esse mestrado e pela ajuda neste projeto e em todos os outros anteriores.
- Aos membros do grupo de pesquisa de Tratamento de Bases Ambientais (TABA) pelo esforço coletivo, possibilitando que este trabalho fosse possível.
- Ao professor Nelcileo Virgilio de Souza Araujo pela atenção durante a graduação e pelo incentivo durante a pós-graduação.
- Ao professor José de Souza Nogueira que, junto à sua esposa professora Marta Cristina de Jesus Albuquerque Nogueira, fazem do PPGFA um ótimo programa de pós-graduação. Tenho orgulho de ter feito o meu mestrado neste programa.
- Ao meu colega Allan Gonçalves de Oliveira pela grande ajuda nas disciplinas do mestrado, na escrita dos artigos e no desenvolvimento desse trabalho.
- A todos os meus colegas de mestrado que, juntos, enfrentamos dificuldades e desafios durante essa fase importante em nossas vidas.
- E por fim, a todos os outros professores, técnicos e alunos do PPGFA que, direta ou indiretamente, ajudaram em meus estudos.

Epígrafe

Quando você entende as leis da física, tudo é possível.
Sheldon Cooper

Sumário

Dedicatória	v
Agradecimentos	vi
Epígrafe	vii
RESUMO	1
ABSTRACT	2
1 Introdução	3
1.1 Problemática	3
1.2 Justificativa	4
1.3 Objetivo geral	5
1.4 Objetivos específicos	5
1.5 Contextualização do trabalho	6
1.6 Organização do trabalho	8
2 Fundamentação Teórica	10
2.1 Dados micrometeorológicos	10
2.2 Métodos para preenchimento de falhas	12
2.3 Redes neurais artificiais	14
2.3.1 Estrutura de uma RNA	14
2.3.2 Trabalhos relacionados com RNA	16
2.4 Algoritmos genéticos	17
2.4.1 Funcionamento de um algoritmo genético	18
2.4.2 Trabalhos relacionados com AG	21
3 Materiais e Métodos	23
3.1 A base de dados	23
3.2 Metodologia proposta	25
3.2.1 Preparação dos dados	26
3.2.2 Determinando a configuração da RNA com AG	29
3.2.3 Treinamento da RNA	32

3.2.4	Preenchimento dos valores ausentes	33
3.3	Ambiente de programação	34
3.4	Avaliação de desempenho	35
4	Resultados e Discussão	36
4.1	Resultados obtidos	36
4.1.1	Resultado do preenchimento de falhas para temperatura . . .	37
4.1.2	Resultado do preenchimento de falhas para saldo de radiação .	39
4.1.3	Resultado do preenchimento de falhas para fotoperíodo	41
4.1.4	Resultado do preenchimento de falhas para umidade	42
4.1.5	Resultado do preenchimento de falhas para insolação	44
4.2	Análise estatística dos resultados	46
5	Considerações Finais	48
	Bibliografia	51
A	Código do Algoritmo Genético	60
B	Código da Rede Neural Artificial	71

Lista de Tabelas

1	Descrição dos atributos na base de dados	24
2	Amostra de dados obtidos pelos equipamentos micrometeorológicos .	28
3	Número de exemplos sinalizados como inválidos para a fase de testes e número de exemplos reservados para a fase de treinamento em cada simulação realizada	29
4	Resultados encontrados pelo AG nos testes realizados da temperatura (T)	37
5	Resultados encontrados pelo AG nos testes realizados do saldo de radiação (Qn)	39
6	Resultados encontrados pelo AG nos testes realizados do fotoperíodo (N)	41
7	Resultados encontrados pelo AG nos testes realizados da umidade (UR)	42
8	Resultados encontrados pelo AG nos testes realizados da insolação (INSOL)	44
9	Erro médio absoluto para cada simulação realizada	46
10	Valores do desvio padrão para cada simulação realizada	46

Lista de Figuras

1	Sistema EnvRP	8
2	Pluviômetro após sofrer ação ambiental (DIAS, 2007)	11
3	Estrutura de uma rede neural artificial (HAYKIN, 2000)	15
4	Ciclo de vida de um algoritmo genético	18
5	Exemplo de um indivíduo de 10 bits armazenando 3 parâmetros	18
6	Processo do operador <i>crossover</i> no algoritmo genético	19
7	Processo do operador mutação no algoritmo genético	20
8	Etapas da metodologia proposta para preenchimento de falhas	25
9	Diagrama da sequência de passos necessários para realizar o preenchimento de falhas	27
10	Representação do indivíduo no AG	30
11	Ilustração do preenchimento de um valor ausente utilizando a RNA treinada	33
12	Ambiente de programação do Matlab	34
13	Valores dos coeficientes de correlação para temperatura	38
14	Valores comparativos entre os valores esperados e os valores obtidos pela RNA nos dados de temperatura	38
15	Valores dos coeficientes de correlação para saldo de radiação	40
16	Valores comparativos entre os valores esperados e os valores obtidos pela RNA nos dados de saldo de radiação	40
17	Valores dos coeficientes de correlação para fotoperíodo	41
18	Valores comparativos entre os valores esperados e os valores obtidos pela RNA nos dados de fotoperíodo	42
19	Valores dos coeficientes de correlação para umidade	43
20	Valores comparativos entre os valores esperados e os valores obtidos pela RNA nos dados de umidade	43
21	Valores dos coeficientes de correlação para insolação	45
22	Valores comparativos entre os valores esperados e os valores obtidos pela RNA nos dados de insolação	45
23	Erro médio absoluto de todas as simulações realizadas	47

RESUMO

VENTURA, T. M. Preenchimento de falhas de dados micrometeorológicos utilizando técnicas de inteligência artificial. 2012. 73f. Dissertação (Mestrado em Física Ambiental), Instituto de Física, Universidade Federal de Mato Grosso, Cuiabá, 2102.

Dados reais relacionados a fenômenos climatológicos são obtidos, geralmente, utilizando equipamentos como sensores. A captação destes dados está sujeita a falhas e influências externas que geram dados inválidos no conjunto de informações geradas. Técnicas matemáticas e computacionais são comumente usadas com o objetivo de preencher essas falhas nos dados. Este trabalho apresenta uma abordagem computacional que utiliza técnica de redes neurais, combinada com algoritmos genéticos, aplicada a dados reais com o objetivo de preencher falhas (valores ausentes) em séries de temperatura, saldo de radiação, fotoperíodo, umidade e insolação em uma região de cerrado no estado de Mato Grosso. Para avaliar os resultados, foram analisados o coeficiente de correlação entre os dados reais e os dados estimados, o erro médio absoluto e o desvio padrão dos dados, e foi constatado um bom resultado nessas análises.

Palavras-chave: preenchimento de falhas; dados micrometeorológicos; redes neurais artificiais; algoritmos genéticos; seleção de atributos.

ABSTRACT

VENTURA, T. M. Gap filling of micrometeorological data using techniques of artificial intelligence. 2012. 73f. Dissertation (Master in Environmental Physics), Institute of Physics, Federal University of Mato Grosso, Cuiabá, 2102.

Real data related to climatological phenomena are usually obtained using equipment like sensors. The capture of this data is subject to failures and external influences that generate invalid data in the set of generated information. Mathematical and computational techniques are commonly used in order to fill these data gaps. This work presents a computational approach that uses neural networks technique, combined with genetic algorithms applied to real data with the objective of filling the gaps in a series of temperature, radiation balance, photoperiod, humidity and insolation in an area of Mato Grosso's Cerrado. To evaluate the results, the correlation coefficient of the real data and the estimated data, the mean absolute error and the standard deviation of the data were analyzed, and a good result was observed in these analysis.

Keywords: gap filling; micrometeorological data; artificial neural networks; genetic algorithms; selection of attributes.

Capítulo 1

Introdução

1.1 Problemática

Em empresas privadas, órgãos governamentais ou instituições acadêmicas, os dados têm um grande valor, porque são os dados que representam toda informação dentro de tais unidades, possibilitando que decisões sejam tomadas baseadas na análise desses dados, produtos sejam criados ou modelagens de fenômenos físicos sejam representados. Por essas razões que às vezes os dados de uma unidade são considerados como um patrimônio da organização.

Dados relacionados à observação de fenômenos reais geralmente são denominados dados científicos (PFALTZ, 2007). Este tipo de dado pode ser, por exemplo, o histograma de uma imagem médica, dados oriundos de satélites ou resultados de uma simulação. Normalmente, esses dados têm uma maior complexidade em sua estrutura, e seus valores podem representar fenômenos do ambiente real gerando uma grande quantidade de dados/informação.

Para compreender fenômenos climáticos no sentido de ajudar em áreas como agricultura e planejamento urbano, é necessário o estudo das variáveis climáticas envolvidas, muitas vezes representadas por dados científicos. Nessa área de estudo, os dados geralmente são obtidos por diversos equipamentos, como os sensores, que fazem a leitura de uma ou mais variáveis durante o seu funcionamento, os quais, normalmente, ficam ligados durante 24 horas por dia, gerando uma grande quantidade de dados.

Além da grande quantidade de dados, uma característica que dificulta a análise dos dados científicos é que os sensores são instalados em estações micrometeorológicas que estão ao ar livre e, conseqüentemente, sujeitos às condições naturais do meio ambiente. A ação de animais próximos aos equipamentos e fortes mudanças no tempo podem atrapalhar as leituras dos sensores, causando falhas no armazenamento dos dados. Além do mais, os sensores são aparelhos eletrônicos e estão sujeitos a falhas.

Sendo assim, as condições naturais do meio ambiente e problemas técnicos podem interferir na leitura e armazenamento dos dados observados.

Quando a leitura das variáveis climáticas não é feita de maneira automática, é necessário que um observador anote as medidas captadas pelos equipamentos. Entretanto, problemas diversos podem ocorrer impedindo que o observador realize alguma leitura dos equipamentos em determinado horário do dia, provocando falha na leitura dos dados.

Assim, apesar de ser possível obter uma grande quantidade de dados, coletados por diversos sensores e/ou pesquisadores, durante um longo período de tempo, é provável que alguns dos dados coletados sejam inválidos (dados com valores incorretos ou ausentes), ocorridos no processo de leitura e/ou armazenamento. Isso gera dificuldades para análise e modelagem desses dados, pois os resultados serão obtidos baseados em séries de dados que não foram armazenadas de forma precisa.

No contexto do Programa de Pós-Graduação em Física Ambiental (PPGFA) da Universidade Federal de Mato Grosso (UFMT), grande parte do trabalho pesquisado é baseado em dados ambientais coletados a partir de sensores. Nos trabalhos desenvolvidos, é comum a presença de falhas em dados micrometeorológicos, sendo necessário a utilização de métodos para preencher valores ausentes gerados por falhas nas séries de dados.

Portanto, é necessário o desenvolvimento e aplicação de métodos que possam facilitar o preenchimento dessas falhas nos dados de estações micrometeorológicas visando analisar e modelar dados que representam fenômenos do mundo real.

1.2 Justificativa

Em um cenário ideal, os dados obtidos de sensores deveriam ser lidos e armazenados com exatidão. Entretanto, deve-se ter o cuidado de pré-processar os dados coletados que serão utilizados a fim de localizar dados incorretos ou nulos, antes de processar, modelar e analisar dados que representam fenômenos reais.

Em um processo de análise de dados, quando dados inválidos ou com valores ausentes são encontrados, deve-se decidir se os mesmos serão removidos completamente ou se será utilizado algum método para preencher os valores inválidos. Normalmente, um procedimento bastante utilizado é a média dos valores nos dados, das leituras que foram obtidas antes e depois da falha, substituindo o valor inválido ou faltante pela média obtida. Contudo, esse é um método simples que pode apresentar características como: a série de dados já não irá representar completamente a realidade e, também, a possibilidade de, caso uma série contínua de dados conter erros, não ser possível

a utilização deste método, já que são necessárias leituras próximas para preencher a falha.

Para tentar minimizar essas características indesejáveis, e estimar um valor mais próximo do valor real, uma alternativa é a utilização de técnicas computacionais para auxiliar na estimativa de valores no preenchimento de falhas. Estão sendo bastante utilizadas algumas técnicas da área de Inteligência Artificial (IA) que visa aprender o comportamento padrão de uma série de dados. Segundo Sellitto (2002), IA é uma área de conhecimento que oferece modelos de apoio à decisão e ao controle com base em fatos reais e conhecimentos empíricos e teóricos, mesmo que apoiados em dados incompletos. Algumas técnicas bastante utilizadas nesse contexto são, por exemplo, as Redes Neurais Artificiais (RNA) e os Algoritmos Genéticos (AG).

RNAs tem sido aplicadas em vários campos da engenharia e modelagem ambiental, por exemplo. RNAs tem a vantagem de usar funções não lineares que possam estimar a saída, a partir de alguns valores de entrada depois de treinada usando uma amostra de dados.

Dessa forma, utilizar métodos que possam automatizar o processo de preenchimento de falhas de maneira que represente valores mais próximos possível do valor real, é um desafio tanto para as pesquisas do PPGFA quanto da comunidade científica em geral. O desenvolvimento de uma ferramenta capaz de preencher falhas em séries de dados, resultará em uma maior consistência nos dados, permitindo interpretações e modelagens mais precisas e, conseqüentemente, auxiliando na análise dos resultados importantes para a comunidade.

1.3 Objetivo geral

O objetivo deste trabalho consiste no preenchimento de falhas em dados obtidos em estações micrometeorológicas, visando análise de fenômenos ambientais, utilizando técnicas de redes neurais em conjunto com algoritmos genéticos.

1.4 Objetivos específicos

Para alcançar o objetivo geral deste trabalho, foi necessário definir e atingir os seguintes objetivos específicos:

- Obter, estudar e pré-processar os dados coletados por equipamentos de estações micrometeorológicas.

- Investigar a técnica de algoritmos genéticos aplicada a dados ambientais e na otimização de redes neurais.
- Investigar a técnica de redes neurais artificiais aplicada a variáveis climatológicas.
- Especificar e desenvolver um algoritmo genético para definir parâmetros de redes neurais artificiais.
- Definir e implementar uma rede neural artificial para estimar valores no preenchimento de falhas.
- Realizar a integração entre o algoritmo genético e a rede neural artificial.
- Avaliar o desempenho dos resultados obtidos.

1.5 Contextualização do trabalho

Este trabalho, vinculado ao Programa de Pós-Graduação em Física Ambiental, se insere no contexto maior do Grupo de Pesquisa denominado Tratamento de Bases Ambientais (TABA), que se dedica a aplicar técnicas computacionais para tratamento, armazenamento e análise de dados ambientais (OLIVEIRA et al., 2010).

O Grupo de Pesquisa TABA se dedica atualmente no desenvolvimento de um ambiente para análise e armazenamento inteligente de dados de séries temporais aplicados a variáveis micrometeorológicas, utilizando técnicas de inteligência artificial, banco de dados, engenharia de software e estatística, entre outras. O ambiente que está em desenvolvimento, tem como base a investigação e a elaboração de uma arquitetura baseada em componentes de software buscando flexibilidade no desenvolvimento e aperfeiçoamento do ambiente.

Esse ambiente em desenvolvimento é denominado de Environmental Resource Planning (EnvRP) e utiliza conceitos relacionados com *Enterprise Resource Planning* (ERP), no qual um banco de dados robusto é utilizado de forma que as informações sejam todas inter-relacionadas e bem mapeadas, buscando-se eficiência no armazenamento e recuperação dos dados. O ERP possui característica de um sistema integrado e uma arquitetura aberta, viabilizando a operação com diversos sistemas operacionais, banco de dados e plataformas de hardware (JESUS; OLIVEIRA, 2007).

Dessa forma, no EnvRP haverá uma centralização e integração das informações em uma única base de dados. Isso irá permitir que os dados sejam analisados de forma

mais adequada, pois o tratamento é uniformizado para todas as manipulações, permitindo que seja feito melhor controle de qualidade dos resultados das análises (OLIVEIRA, 2011).

O sistema EnvRP (Figura 1) tem a finalidade de facilitar a extração de informações referentes a dados ambientais. Para tanto, foram planejados módulos dentro do sistema geral envolvendo técnicas computacionais de inteligência artificial, banco de dados e estatística. Esses módulos são descritos a seguir.

A Obtenção dos dados

1. Coleta de dados: A obtenção dos dados a serem utilizados pelo sistema é feita por sensores localizados em estações micrometeorológicas. Cada sensor tem as suas próprias características e pode gerar tipos diferentes de dados a cada leitura.
2. Dados digitais (.DAT): Os sensores podem obter dados a cada fração de segundo dependendo de sua configuração. Em todo caso, esses dados são enviados para outro equipamento que é responsável pelo armazenamento em campo. De tempo em tempo, os dados armazenados nesse equipamento são transformados em arquivos digitais (extensão .DAT) os quais os sistemas operacionais tradicionais possuem a capacidade de ler.

B Armazenamento e consulta dos dados

3. Importação no SGBD: Mesmo tendo os dados em forma digital a partir dos arquivos .DAT, ainda não é viável realizar consultas para extrair informações desses dados. Assim, se faz necessário uma importação em um sistema de gerenciamento de banco de dados (SGBD).
4. Consulta rápida e fácil: Este procedimento pode ser realizado desde o momento que os dados já estão no banco de dados ou depois que o tratamento dos dados seja executado. O objetivo aqui é que o usuário possa recuperar de maneira fácil e rápida as informações contidas no SGBD através de filtros de consulta.
5. Exportação dos dados: Caso o usuário queira trabalhar os dados em outro ambiente, esse procedimento pode ser executado e os dados serão transformados para um outro formato de arquivo.

C Tratamento dos dados

6. Limpeza dos dados: Trata de uniformizar os dados, ou seja, fazer com que as variáveis do mesmo tipo tenham a mesma unidade. Também é de responsabilidade desta etapa identificar os dados inválidos.
7. Preenchimento de falhas: Com a etapa anterior finalizada, o sistema identifica quais dados devem ser substituídos. Esta etapa tem como objetivo utilizar técnicas da inteligência artificial e estatística, entre outras, para preencher valores inválidos.
8. Detecção de padrões: Com a quantidade de dados coletados fica difícil a tarefa de observar relações entre as informações. Na etapa de detecção de padrões o próprio sistema tentará descobrir padrões dos dados.

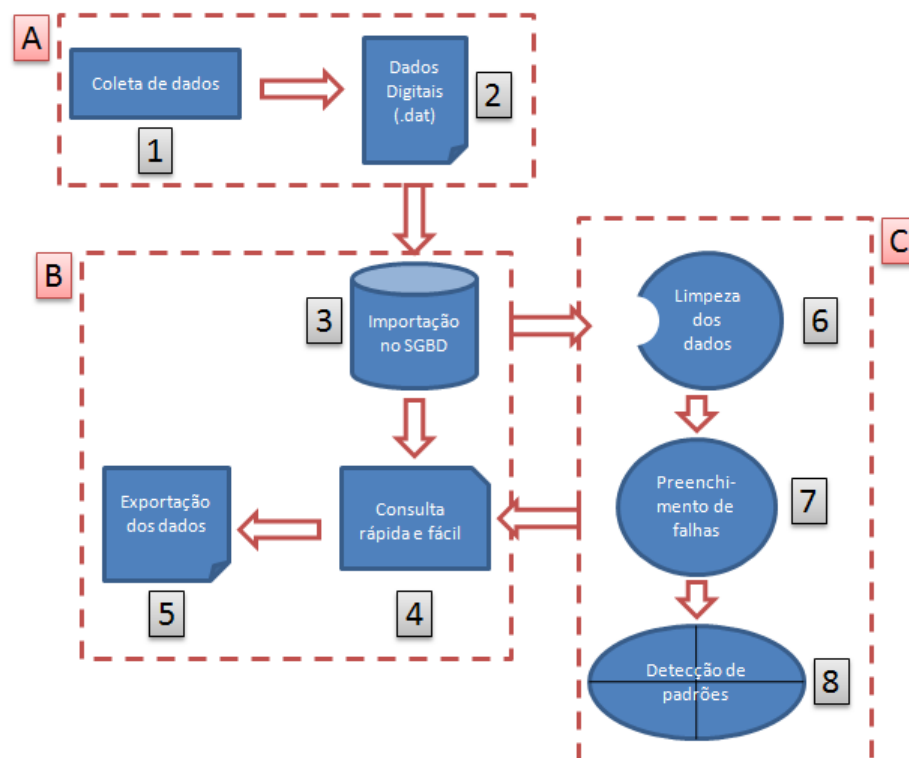


Figura 1: Sistema EnvRP

Este trabalho está situado no módulo C, mais especificamente no item 7, ou seja, na etapa de preenchimento de falhas. Mais detalhes do módulo A e do módulo B podem ser consultados em Oliveira et al. (2011).

1.6 Organização do trabalho

Este trabalho está organizado da seguinte forma:

- **Capítulo 2:** Este capítulo apresenta os conceitos básicos necessários para o desenvolvimento deste trabalho. São apresentadas informações sobre os dados de estações micrometeorológicas, conceitos sobre rede neural artificial e sobre os algoritmos genéticos.
- **Capítulo 3:** Descreve os dados micrometeorológicos utilizados no preenchimento de falhas, a metodologia proposta, o ambiente de programação utilizado na implementação do trabalho e a análise estatística dos resultados encontrados.
- **Capítulo 4:** Neste capítulo são mostrados os resultados obtidos no preenchimento de falhas de diversas simulações por meio de coeficientes de correlação entre os dados reais e os dados estimados, erro médio absoluto e desvio padrão.
- **Capítulo 5:** Apresenta as conclusões e as possibilidades de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

As condições climáticas do meio ambiente estão adquirindo cada vez mais importância no cenário mundial, principalmente porque no início deste novo milênio, as emissões decorrentes das atividades humanas cada vez mais comprometem a vida neste planeta (HEWITT, 2003).

Diversas áreas de pesquisa estudam as condições climáticas, uma delas é a meteorologia. A meteorologia é o ramo da ciência que lida com a atmosfera terrestre e os processos físicos que ocorrem nele (AHRENS, 2004). Por ser uma área extensa de estudo ela pode ser dividida em sub-áreas, como a macro, meso ou a micrometeorologia.

A macrometeorologia é o estudo dos fenômenos de macroescala. Os movimentos de macroescala tem uma escala de comprimento horizontal da ordem de 1000 km ou maior, limitado pela circunferência da Terra, e escala de tempo de um dia ou mais. Na vertical, os movimentos de macroescala podem se estender por toda a troposfera (BUSK, 2008). Já a mesometeorologia se refere ao estudo dos sistemas atmosféricos de dimensões intermédias entre as dos fenômenos observados às escalas “local” e “sinóptica” (ATKINSON, 1981). E a micrometeorologia tem como principal objeto de estudo o ambiente de vida da humanidade (FOKEN, 2008).

Dados micrometeorológicos, área de interesse deste trabalho, são obtidos usando equipamentos que estão em campo realizando leituras do ambiente.

2.1 Dados micrometeorológicos

O funcionamento de equipamentos responsáveis por medir elementos meteorológicos passam por sensores e os dados são armazenados em sistemas de aquisição de dados, chamados de *Datalogger* (NEVES, 2011).

Entretanto, como comentado, nem sempre os dados são armazenados com sucesso. É normal encontrar falhas nos dados coletados por sensores e estas falhas podem sig-

nificar a ausência de um valor ou o armazenamento incorreto de um valor. No caso de um valor incorreto ser gravado, a identificação da falha é mais difícil, já que para isso é necessário um conhecimento maior sobre a variável climática que o valor representa para que seja possível perceber que o valor está incoerente com a situação que se encontra a série de dados.

As falhas nos dados podem acontecer por diversos fatores. Em Dias (2007) são apresentados alguns destes fatores, como deterioração por fatores ambientais ou desligamento dos equipamentos por falta de energia. Na Figura 2 é apresentado um equipamento sendo afetado por uma ação da natureza no qual vários insetos entraram dentro do equipamento prejudicando o seu funcionamento.



Figura 2: Pluviômetro após sofrer ação ambiental (DIAS, 2007)

Como dito anteriormente, uma das consequências quando os equipamentos sofrem alguns destes problemas é o erro no armazenamento dos dados. Nos casos em que há ausência de valores em uma série de dados deve-se fazer uma escolha entre ignorar os períodos com dados incorretos ou aplicar alguma técnica para preencher os valores com falhas.

Alguns trabalhos mostram o tratamento de dados ausentes. No trabalho de Capistrano (2007) foi determinado um valor mínimo e máximo para o dado que estava sendo trabalhado e, assim, retirou-se os dados que estavam fora desse limite. O método utilizado para preencher os dados retirados foi fazer uma repetição do dado anterior ao retirado. É importante ressaltar que esse tipo de tratamento só foi possível ser utilizado porque essa operação não influenciava no estudo do respectivo trabalho.

Em Palú (2008) é observado que há vários valores ausentes na série de dados pesquisada, assim como dados incoerentes em comparação aos dados obtidos em períodos

próximos. Neste trabalho as falhas encontradas tiveram os seus registros ignorados, trabalhando apenas com os dados válidos.

No trabalho realizado por Gallon (2005) foi constatado a ausência de dados em diversos períodos por causa de falhas nos equipamentos. Em alguns casos havia falhas englobando um mês inteiro. Uma parte dos dados em diversos meses teve que ser descartada, dificultando a análise dos dados.

O número de dados das séries no trabalho de Mariano (2005) foi limitado devido às falhas encontradas por causa das condições de campo e de manutenção dos equipamentos. Nesse trabalho, essas falhas podem ser de algumas horas ou de vários dias, o que impossibilitou que quatro anos não pudessem ser analisados por causa das falhas contínuas. Quando o período tinha poucas falhas, foi utilizado a média entre o dado anterior e posterior à falha para preencher o valor ausente. No entanto é observado que esse método é apenas aceitável quando não há uma grande sequência de falhas na série de dados.

Como pode ser visto, é comum a ocorrência de falhas em séries de dados micro-meteorológicos. Pode ser observado também que a solução utilizada para resolver esse problema é normalmente o descarte dos registros com falhas ou o preenchimento com técnicas simples, como a repetição ou média de dados próximos. Entretanto, a utilização de métodos simples prejudica o desempenho do trabalho porque os dados deixam de representar com precisão as condições do ambiente no momento em que foram feitas as leituras. Por outro lado, há métodos mais complexos que preenchem as falhas de maneira mais precisa.

2.2 Métodos para preenchimento de falhas

Um dos métodos utilizado para preenchimento de falhas é a imputação múltipla (*Multiple Imputation, MI*). MI é uma técnica de Monte Carlo em que o valor ausente é substituído por $m > 1$ valores simulados, onde m é tipicamente pequeno, como valores entre 3 e 10 (SCHAFER, 1999). Fazer as operações do método de MI não é uma tarefa fácil, então foram criados diversos sistemas para automatizar esse processo. Em Orton e Ipsitz (2001) é comentado sobre vários sistemas para facilitar o uso deste método, como: Solas, Sas, S-Plus, Mice, dentre outros.

Inicialmente, o MI foi proposto para preencher falhas em questionários de pesquisa, mas atualmente é também utilizado em outros tipos de séries de dados. No trabalho de Hui (2004) foi usado esse método para preencher falhas em medidas do equipamento de *eddy covariance*, que faz leituras sobre a troca de gás carbônico, vapor d'água e calor entre uma superfície vegetada e a atmosfera. Já em Nunes, Kluck e

Fachel (2009) foi utilizado o método para dados epidemiológicos.

Em Falge et al. (2001) são comparados outros métodos de preenchimento de falhas, como a variação média diurna (*mean diurnal variation, MDV*) e o *look-up tables*. No método MDV, o valor ausente é substituído pela média calculado a partir dos valores obtidos nos dias adjacentes (KATO et al., 2006), sendo utilizado em dados meteorológicos como em Hu et al. (2009) e em Alavi, Warland e Berg (2006).

A abordagem do método *look-up tables* consiste em criar uma tabela com os valores de fluxos, com base nos valores correspondentes dos parâmetros externos (por exemplo, temperatura do solo, precipitação, umidade do solo, etc.), e preencher as falhas de acordo com essa tabela. A determinação dos parâmetros relevantes e seus valores críticos é um passo crucial para essa técnica ser bem sucedida (MISHUROV; KIELY, 2011). Pesquisas como as de Rodrigues, Pita e Mateus (2005), Wilson e Baldocchi (2001) e Shao et al. (2011) utilizaram esse método.

No trabalho de Moffat et al. (2007) foram comparadas algumas variações de técnicas de preenchimento de falhas em um mesmo conjunto de dados. Dentre elas estavam técnicas do tipo de regressão não-linear (RNL) e de *dual unscented Kalman filter* (UKF). As RNL são baseadas em parâmetros de equações não-lineares que expressam (semi-)relações empíricas entre o fluxo pesquisado e as variáveis ambientais, como temperatura e luz. O UKF é um método de estimativa recursivo de predição e correção através do qual os dados de medição são usados para atualizar continuamente o modelo (GOVE; HOLLINGER, 2006).

Técnicas computacionais também estão sendo utilizadas no preenchimento de falhas, como, por exemplo, as técnicas de inteligência artificial, redes neurais artificiais, algoritmos genéticos e lógica fuzzy.

Em Ooba et al. (2006) foi desenvolvido uma combinação das técnicas de algoritmos genéticos e redes neurais artificiais para preencher falhas em dados de fluxo de carbono. Foi feito também uma comparação dos resultados de sua aplicação com um método convencional que usa equações com parâmetros que foram determinados por meio de regressão não-linear. Os resultados mostraram que a combinação das técnicas utilizadas teve uma maior precisão comparado ao método convencional testado.

Este trabalho utiliza uma metodologia semelhante a que foi proposta por Ooba et al. (2006). No entanto, é importante salientar que a maioria das implementações das técnicas de preenchimento de falhas são orientadas para o tipo de dado específico em relação ao problema que se deseja resolver. Isso acontece porque esses métodos necessitam ter o conhecimento a priori do comportamento da variável que será ajustada para ter um bom desempenho. Levando em consideração esse fator, neste trabalho a metodologia utilizada para preenchimento de falhas de dados micrometeorológicos

trabalha, independentemente, da variável climática a ser corrigida. A técnica utilizada nesta metodologia são as redes neurais artificiais em conjunto com algoritmos genéticos.

2.3 Redes neurais artificiais

Segundo Russell e Norvig (2004), um neurônio é uma célula no cérebro cuja principal função é coletar, processar e disseminar sinais elétricos. Acredita-se que a capacidade de processamento de informações do cérebro emerge principalmente de redes de tais neurônios. Baseado nesse conceito, surgiu a ideia da técnica de redes neurais artificiais (RNA), que é uma das áreas de pesquisa de inteligência artificial (IA).

A RNA é composta por unidades de processamento chamadas neurônios, cada um deles gera um sinal que é uma função genérica não-linear da soma dos sinais recebidos pela unidade. Fontes de sinais recebidos são especificadas por meio de conexões. Neurônios recebem os sinais provenientes de fontes idênticas, compreendendo um grupo de neurônios denominada camada. Cada sinal recebido por um neurônio é pré-processado por meio de um parâmetro de ponderação multiplicativo chamado de peso. A definição dos valores dos pesos da rede, de modo a otimizar um critério, depende das saídas da rede, e esse processo é chamado de treinamento da rede (AGARWAL, 1997).

As RNAs têm sido utilizadas com sucesso para modelar relações envolvendo séries temporais complexas (HAYKIN, 2000). RNAs podem ser consideradas como uma poderosa técnica de reconhecimento de padrões em análises de dados ambientais e climatológicos em relação aos processos biológicos (BELGRANO; MALMGREN; LINDAHL, 2001). A maior vantagem das RNAs sobre os métodos convencionais é que não requerem informação detalhada sobre os processos físicos do sistema a ser modelado, sendo este descrito explicitamente na forma matemática (SUDHEER; GOSAIN; RAMASASTRI, 2003).

2.3.1 Estrutura de uma RNA

Na Figura 3 é ilustrada a estrutura de uma RNA. Os sinais de entradas $x_1, \dots, x_m$ são definidos pelos valores contidos na base de dados. Esses valores servirão de insumo para calcular o valor do sinal de saída y_k da rede neural, ou seja, a resposta ao problema que a RNA foi treinada para solucionar. Os pesos $w_{k1}, \dots, w_{km}$ de uma RNA são utilizados no cálculo da função de ativação. Segundo Moayed e Shell (2010), os pesos serão ajustados baseados no erro entre o valor estimado e o valor atual da saída durante a fase de treinamento.

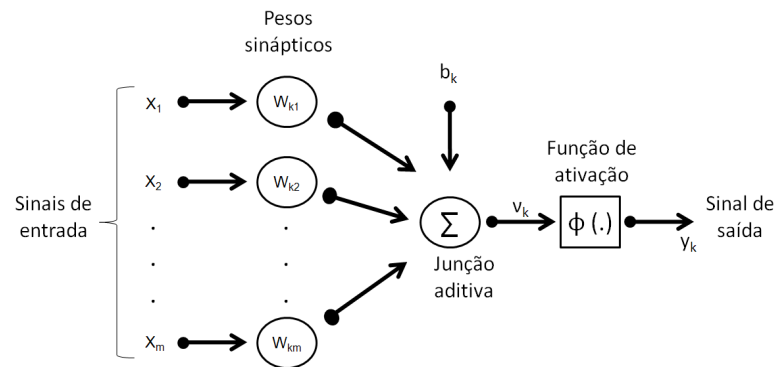


Figura 3: Estrutura de uma rede neural artificial (HAYKIN, 2000)

A junção aditiva possui o propósito de somar todos os sinais de entrada ponderados pelos pesos sinápticos de cada neurônio. O valor de b_k funciona aumentando ou diminuindo a influência do valor da entrada para a ativação do neurônio; já a função de ativação funciona restringindo a amplitude de saída de determinado neurônio e adicionando não-linearidade ao modelo (ZANETTI et al., 2008).

Uma RNA pode ter várias camadas ocultas, que são as camadas posicionadas entre a camada de entrada e a camada de saída. Em cada uma delas pode ter vários neurônios. O número certo de camadas ocultas e neurônios varia de problema para problema. Entretanto, esses valores podem ser baseados no Teorema de Kolmogorov-Nielsen, apresentado por Kovács (1996) onde:

"Dada uma função contínua arbitrária $f : [0, 1]^n \rightarrow R^m$, $f(x) = y$, existe sempre, para f , uma implementação exata com uma rede neural de três camadas, sendo a camada de entrada um vetor de dimensão n , a camada oculta composta por $2n + 1$ neurônios, e a camada de saída com m neurônios representando as m componentes do vetor y ".

Além dos componentes apresentados, faz parte também de uma rede neural o algoritmo de treinamento. Para Fausett (1994), o algoritmo de treinamento consiste dos procedimentos que modificam os pesos em uma rede neural, ou seja, é o algoritmo de treinamento que vai determinar a busca no espaço de estados possíveis para a rede neural encontrar uma solução para o problema.

Normalmente a fase de treinamento da RNA consiste em utilizar cerca de 70% dos dados para formar um conjunto de dados, chamado de conjunto de treinamento. Este conjunto é utilizado inúmeras vezes, no qual a cada utilização os pesos são atualizados até que o erro gerado alcance um nível aceitável para este conjunto de dados. O restante dos dados, cerca de 30%, servirá para verificar o desempenho do treinamento realizado.

2.3.2 Trabalhos relacionados com RNA

RNAs são utilizadas nas mais diversas áreas da ciência. Em Fausett (1994) são mostrados casos do seu uso em processamento de sinais, controle maquinário, reconhecimento de padrões, reconhecimento de fala, área médica, área de negócio, dentre outros. É comum a utilização da técnica de RNA em pesquisas na área ambiental, sendo elas para previsão, estimativa ou preenchimento de falhas.

Em Panella, Rizzi e Martinelli (2003) foram estudadas RNAs para melhorar a precisão em previsões de sequência de dados ambientais, se concentrando em séries com comportamento caótico, como o nível de ozônio e ruído, o que torna mais difícil a previsão. Utilizaram a rede neural baseada no modelo de mistura Gaussiano, com o algoritmo de treinamento denominado SHEM, que segundo Holtgrewe, Sanders e Schulz (2010) é um algoritmo muito rápido. Foram obtidos bons resultados, mostrando-se melhor que outros métodos de previsão para esse tipo de dado.

Em Tsukahara, Jensen e Caramori (2010) RNAs foram utilizadas para o preenchimento de falhas em séries horárias de dados meteorológicos do estado do Paraná-Brasil. Foi utilizada uma RNA do tipo *Multi Layer Perceptron*, com algoritmo de aprendizagem *Resilient*, com uma única camada oculta com quarenta e quatro neurônios. O trabalho utilizava os dados de onze estações meteorológicas, no qual as falhas de uma estação eram preenchidas com base nos valores das outras dez estações. A RNA conseguiu gerar bons resultados, mas a dependência da existência de outras estações meteorológicas é um ponto negativo.

No trabalho de Deswal e Pal (2008) é utilizado uma RNA para estimar a evaporação de um local. Foram utilizados dados de temperatura do ar, velocidade do vento, horas de sol por dia, umidade relativa e, para avaliar os resultados obtidos, a própria evaporação. Os resultados foram comparados com métodos de regressão linear e regressão linear múltipla, e o uso de RNA foi considerado uma melhor alternativa em relação aos outros métodos.

Já em Jain, Nayak e Sudheer (2008), foi aplicada a técnica de RNA também para estimar valores de evapotranspiração. Há uma dificuldade na medição de evapotranspiração porque requer um maior conhecimento de várias variáveis climáticas e o uso de equipamentos especiais. Neste trabalho essa dificuldade foi contornada pelo uso de uma RNA, com a qual foi possível estimar com precisão a evapotranspiração de um local mesmo utilizando poucas variáveis climáticas.

Na Seção 2.3.1 foi visto que a estrutura de uma RNA possui diversos parâmetros, podendo ser configurada de várias formas diferentes. E para cada conjunto de dados há uma melhor configuração da RNA para resolver o problema que lhe foi proposto. Entretanto, achar essa melhor configuração não é uma tarefa fácil. Nos trabalhos cita-

dos, provavelmente houve a necessidade de testar várias combinações de parâmetros até achar uma configuração aceitável para resolver cada problema, exigindo muito esforço. Neste trabalho é proposta a aplicação da técnica de RNA para preenchimento de falhas em diversos tipos de dados. Para tanto, foi utilizada a técnica de algoritmos genéticos para estimar os parâmetros de uma RNA, criando uma abordagem para que a configuração da RNA seja feita de forma automatizada.

2.4 Algoritmos genéticos

Algoritmos Genéticos (AGs), inicialmente propostos por John Holland em 1975, são baseados no princípio da seleção natural de Charles Darwin, fundamentando-se na afirmação de que, os indivíduos mais adaptados têm maior chance de sobreviver e gerar descendentes (LACERDA; CARVALHO, 1999). AGs são métodos estocásticos de busca de um ótimo global, no qual os indivíduos mais aptos tendem a sobressair (ASSUMPCÃO et al., 2011).

Segundo Goldberg (1989), os AGs apresentam algumas características que os tornam diferentes dos métodos tradicionais de busca e otimização. Em Correia (2003) são detalhadas algumas destas diferenças:

- Não trabalham diretamente com o domínio do problema, mas com representações dos seus elementos;
- Executam a busca em um conjunto de candidatos (população) e não apenas em um;
- Não tem conhecimento específico do problema, utilizam apenas uma função que avalia se um indivíduo é bom ou não;
- Utilizam basicamente regras probabilísticas.

Uma população é um grupo de indivíduos, no qual cada indivíduo tem um valor adaptativo associado, que é propagado em uma nova geração utilizando o princípio Darwiniano de reprodução e sobrevivência (KOZA, 1995).

De acordo com Michalewicz (1994), o princípio básico dos AGs consiste em fazer evoluir um conjunto de soluções iniciais (população) para uma solução ótima. O AG inicia selecionando aleatoriamente certo número de indivíduos, que são possíveis soluções para o problema, dentro do espaço de busca. Estes são avaliados em relação à capacidade de resolver o problema, e essa capacidade é expressa numericamente pela avaliação do indivíduo, por meio de uma função objetivo. Com base na aptidão dos indivíduos, uma nova população é formada fazendo uso de operadores probabilísticos de

seleção. No final, o indivíduo que está mais apto a resolver o problema é considerado como uma solução, dentro de um erro aceitável.

2.4.1 Funcionamento de um algoritmo genético

Na Figura 4 é mostrado o ciclo de vida de um algoritmo genético. Primeiramente, uma população de indivíduos, que é um conjunto de possíveis soluções, é criada aleatoriamente. O indivíduo, segundo Sheikh, Raghuwanshi e Jaiswal (2008), é um conjunto de parâmetros da solução codificado em uma cadeia de número, representando um ponto no espaço da busca. Normalmente, os indivíduos são codificados em uma cadeia binária, para facilitar as operações genéticas.

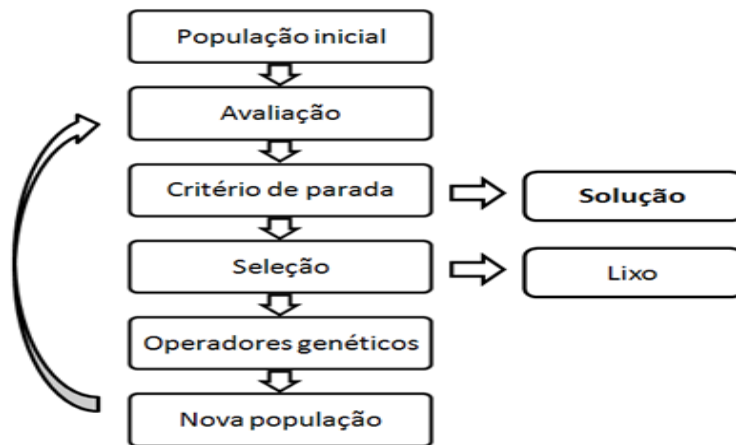


Figura 4: Ciclo de vida de um algoritmo genético

Na Figura 5 pode ser visualizado um exemplo de indivíduo armazenando os valores de três parâmetros em sua forma binária, no qual o primeiro parâmetro é 110, representando o valor 6, o segundo parâmetro é 1000, representando o valor 8, e o terceiro parâmetro é 101, ou seja, 5 no valor decimal.

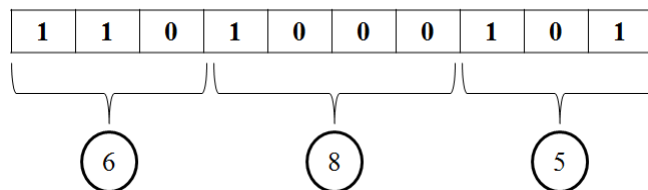


Figura 5: Exemplo de um indivíduo de 10 bits armazenando 3 parâmetros

Depois que a população foi criada, cada indivíduo será avaliado pela função objetivo. A função objetivo é responsável por descrever o quão bom é um indivíduo em cada geração (TANG; LEE, 2006), ou seja, é esta função que vai informar ao algo-

ritmo genético se aquela solução representada pelo indivíduo é uma boa solução para o problema em questão.

Quando todos os indivíduos forem avaliados, é possível verificar se a execução do algoritmo genético pode parar. Critérios de parada: i) número máximo de gerações foi alcançado ou ii) um ou mais indivíduos da população representem uma solução válida para o problema (possua um erro mínimo aceitável). Se nenhum dos critérios de parada foi atingido, alguns indivíduos na população são selecionados e é iniciada uma nova geração desta população.

Durante a seleção dos indivíduos pode ser realizado um procedimento de elitismo, para não permitir que indivíduos melhores sejam expulsos da população em favor de indivíduos piores (LAUMANN; ZITZLER; THIELE, 2000). O elitismo é responsável por propagar propositalmente uma parte dos melhores indivíduos para a próxima geração, conservando assim bons materiais genéticos.

Os indivíduos que não foram selecionados para a próxima geração são descartados da população. Os restantes são utilizados em operadores genéticos para gerar novos indivíduos. Os operadores genéticos utilizados são *crossover* e mutação.

Zhang, Gao e Lou (2007) descrevem o funcionamento do *crossover* da seguinte forma: duas partes de materiais genéticos são escolhidas aleatoriamente de dois indivíduos (pais), depois, dois novos indivíduos (filhos) são gerados trocando essas partes sorteadas. No final, são obtidos dois novos indivíduos, sendo um com a parte inicial do primeiro pai e a parte final do segundo pai, e o outro com a parte final do primeiro pai e a parte inicial do segundo pai. Esse processo é mostrado na Figura 6.

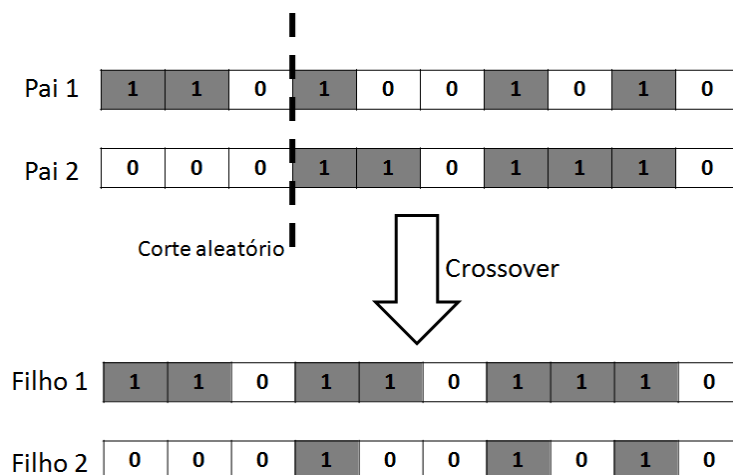


Figura 6: Processo do operador *crossover* no algoritmo genético

Por fim, o operador de mutação, segundo Zhou, Cheng e Zuo (2008), pode criar um novo indivíduo alterando um de seus materiais genéticos, conforme é mostrado na

Figura 7, que no caso, o bit com valor 0 foi alterado para o valor 1. No entanto, a fim de preservar a estabilidade do algoritmo, a probabilidade de mutação deve ser pequena.

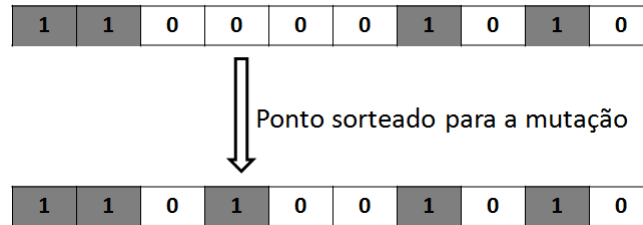


Figura 7: Processo do operador mutação no algoritmo genético

Após todos esses passos, uma nova geração da população foi formada, a qual será novamente avaliada, reiniciando o ciclo do algoritmo genético. Essa sequência continuará até que um dos critérios de parada seja alcançado.

O Pseudocódigo 1 representa a lógica do funcionamento de um AG.

Pseudocódigo 1: Lógica de um AG

- 1: Gerar população aleatória
 - 2: **Enquanto** não for atingido o critério de parada
 - 3: **Para** cada indivíduo da população
 - 4: Decodificar o indivíduo
 - 5: Avaliar o indivíduo com a função objetivo
 - 6: Armazenar a sua avaliação
 - 7: **Fim Para**
 - 8: **Se** nenhum indivíduo atingiu o erro mínimo **Então**
 - 9: Obtém uma nova geração da população
 - 10: Aplicar elitismo
 - 11: Aplicar crossover
 - 12: Aplicar mutação
 - 13: **Fim Se**
 - 14: **Fim Enquanto**
 - 15: Apresentar os indivíduos mais aptos (soluções para o problema)
-

Na primeira linha é gerada uma população de indivíduos, no qual cada indivíduo é codificado pelos parâmetros, dependendo do problema a ser resolvido. Em seguida, é dado início a um ciclo de operações que só irá parar quando um resultado satisfatório for alcançado. Estas operações consistem em avaliar cada indivíduo para verificar se os parâmetros que o mesmo armazena fornecem uma boa solução para o problema.

Na linha 3 é iniciado uma iteração entre cada indivíduo da população. Como os parâmetros dos indivíduos estão codificados, é necessário uma decodificação para trans-

formar novamente os parâmetros em valores possíveis de serem calculados e comparados (linha 4). Com os parâmetros decodificados, é possível aplicar a função objetivo para avaliar o indivíduo (linha 5). Em seguida, a avaliação gerada pela função objetivo é armazenada para que futuramente possa ser comparada com os erros dos outros indivíduos (linha 6).

Na linha 8 é verificado se não houve nenhum caso em que um indivíduo atingiu o erro mínimo aceitável pelo problema. Se isso aconteceu, uma nova geração é gerada com base nos operadores genéticos (linhas 9 a 12).

Com uma nova geração, teoricamente mais evoluída, o processo é realizado novamente voltando para a linha 2, reiniciando o ciclo até que um ou mais indivíduos estejam aptos para serem as soluções do problema ou seja atingido o número máximo de ciclos. Quando isso acontecer serão apresentados tais indivíduos (linha 15).

2.4.2 Trabalhos relacionados com AG

Assim como as RNAs, os algoritmos genéticos também são utilizados em diversas áreas. O trabalho de Forrest (1993) exemplifica algumas áreas em que o AG foi utilizado, como, por exemplo, otimização de função, problemas de ordenação, programação automática e modelos para processos dinâmicos.

É possível utilizar o AG em conjunto com outras técnicas, como as redes neurais artificiais. Em Hervás, Algar e Silva (2000) é mostrado que o uso de AG e RNA se mostrou útil fazendo correções de variação de temperatura. Nesse caso, o AG determinava o número de entradas da RNA obtendo um resultado melhor do que comparado com um método paramétrico não-linear.

Os AGs podem determinar toda a estrutura de uma RNA, como também apenas uma parte de sua configuração. Um exemplo desse caso é o trabalho de Lin, Chiu e Tsai (2006), no qual o algoritmo genético era responsável apenas por determinar os melhores pesos na configuração da rede neural. Os outros parâmetros eram determinados pelo conhecimento prévio do autor ou por meio de testes.

Em Tarca, Grandjean e Larachi (2002) também foi utilizada a combinação de AG e RNA na área da química. Nesse trabalho, as combinações das variáveis em questão eram inúmeras, tornando muito demorado a resolução do problema utilizando apenas RNA. Então foi utilizada a combinação com o AG para acelerar a busca por uma solução desse problema combinatório, diminuindo o processamento que durava semanas para apenas algumas horas. O AG foi responsável por determinar a melhor configuração das camadas da RNA.

Schaffer, Whitley e Eshelman (1992) realizaram um estudo sobre os principais pontos na combinação de AG e RNA, no qual foi visto que essa combinação pode fazer

com que a RNA auxilie o desenvolvimento do AG, assim como o AG pode auxiliar a criação da RNA. No caso do AG dando apoio à RNA, podem ser definidas quais as entradas serão utilizadas na RNA, bem como a definição da sua própria arquitetura. Já no caso da RNA apoiando o AG, normalmente são definidos os indivíduos iniciais da população. O trabalho conclui que o uso de AG para auxiliar uma RNA é uma área promissora no sentido de ter um retorno rápido de resultados em relação ao esforço aplicado e, por isso, tem sido aplicada com sucesso para vários problemas do mundo real.

Com base nesses trabalhos percebe-se que a combinação entre algoritmo genético e rede neural artificial é possível e está sendo aplicada com sucesso. Assim, neste trabalho, com objetivo de preencher dados com falhas, o AG será utilizado para determinar os principais parâmetros da RNA. Além disso, poderá determinar também quais entradas disponíveis serão utilizadas na RNA, já que neste caso há a possibilidade de algumas entradas prejudicarem o desempenho da RNA. Mais detalhes sobre o uso dessas técnicas será visto na Seção 3.2.2.

Capítulo 3

Materiais e Métodos

Como foi dito anteriormente, uma série de dados obtida de estações micrometeorológicas pode conter vários tipos de erros. Estes erros podem ser ocasionados por problemas nos instrumentos presentes nas estações, condições ambientais ou até mesmo falha humana. Essa situação pode comprometer análise de fenômenos que se baseiam nesses dados, sendo necessário corrigir, de alguma forma, esses erros.

Este trabalho tem o intuito de utilizar uma metodologia computacional com o objetivo de substituir dados inválidos por valores estimados utilizando técnicas de inteligência artificial. De forma geral, a metodologia proposta consiste em obter dados coletados de estações micrometeorológicas; definir e aplicar o algoritmo genético para configurar uma rede neural, selecionando os principais atributos que serão utilizados no cálculo do preenchimento de falhas; treinar a RNA; executar a RNA para estimar o valor que substituirá o dado inválido; e analisar os resultados encontrados.

3.1 A base de dados

Os dados utilizados neste trabalho foram coletados na Estação Agroclimatológica Padre Ricardo Remetter, localizada em uma fazenda experimental (latitude: $-15,78^{\circ}$; longitude: $-56,07^{\circ}$; altitude: 140 metros) da Universidade Federal de Mato Grosso (UFMT). A fazenda está situada no município de Santo Antônio de Leverger na Baixada Cuiabana, Mato Grosso, Brasil, distante aproximadamente 28 quilômetros da capital Cuiabá.

Esta estação fornece dados sobre temperatura do ar, umidade relativa, precipitação, velocidade do vento, radiação global, dentre outras variáveis climáticas. Os dados que esta estação gerou foram utilizados em diversos trabalhos de pesquisas do Programa de Pós-Graduação em Física Ambiental da UFMT (NOVAIS, 2011; SEIXAS, 2009; NETO, 2007; FONSECA, 2008).

O período que os dados foram coletados corresponde à 1^o de janeiro à 31 de dezembro de 2006, totalizando 365 dias. Para cada dia foi realizada a leitura dos equipamentos da estação agroclimatológica, armazenando ao todo 40 medidas representando várias variáveis climáticas.

Na Tabela 1 são descritos os atributos que foram selecionados para análise, num primeiro momento. Na tabela são mostrados o nome do atributo, a sua descrição e o intervalo dos valores armazenados para cada atributo, normalmente um número inteiro ou decimal. No entanto, nem todos os atributos são necessários para serem utilizados na execução da metodologia proposta. Houve uma necessidade de realizar um pré-processamento nesses dados antes de executá-la, como apresentado na Seção 3.2.1.

Tabela 1: Descrição dos atributos na base de dados

Atributo	Descrição	Intervalo dos Valores
Dia	Dia juliano	1 - 365
P12	Pressão atmosférica do ar na altura 12	9371 - 10072
P18	Pressão atmosférica do ar na altura 18	9872 - 10028
P24	Pressão atmosférica do ar na altura 24	9885 - 10039
P	Pressão atmosférica do ar média	974,73 - 1004,27
Tem12	Temperatura com bulbo seco na altura 12	14,2 - 31,8
Tem18	Temperatura com bulbo seco na altura 18	19,9 - 38,6
Tem24	Temperatura com bulbo seco na altura 24	17,2 - 32,2
Tbu12	Temperatura com bulbo úmido na altura 12	11,0 - 26,7
Tbu18	Temperatura com bulbo úmido na altura 18	13,7 - 27,8
Tbu24	Temperatura com bulbo úmido na altura 24	13,4 - 27,4
TX	Temperatura do ar máxima	20,8 - 38,8
TN	Temperatura do ar mínima	11,1 - 26,1
T	Temperatura com bulbo seco média	17,24 - 30,88
Tbu	Temperatura com bulbo úmido média	13,02 - 27,0
UR12	Umidade relativa do ar na altura 12	36 - 100
UR18	Umidade relativa do ar na altura 18	15 - 99
UR24	Umidade relativa do ar na altura 24	37 - 100
UR	Umidade relativa do ar média	35,75 - 99
Prec	Precipitação	0 - 75,5
Evapo	Evaporação	2 - 147
INSOL	Insolação	0,1 - 118
V12	Velocidade do vento na altura 12	4 - 9
V18	Velocidade do vento na altura 18	4 - 9
V24	Velocidade do vento na altura 24	2 - 7
V	Velocidade do vento média	4 - 8
Qn	Saldo de radiação	2,09 - 14,18
N	Fotoperíodo	10,95 - 12,83

Os instrumentos instalados na área de estudo deste trabalho são (CARRILHO, 2011):

- Saldo radiômetro NR LITE (Kipp e Zonen Delft, Inc., The Netherlands)

- Piranômetros LI-200X-L (Campbell Scientific, Inc., USA)
- Sensores quantum LI50 190SB-L (Campbell Scientific, Inc., USA)
- Termohigrômetros HMP 45AC (Vaisala, Inc., Helsinki, Finland)
- Anemômetros de conchas 03101-L Wind Sentry Anemometer (RM Young, Inc., USA)
- Anemômetro de concha e medidor da direção do vento 03002-L Wind Sentry Set (RM Young, inc., USA)
- Placa de fluxo de calor HFP01 (Rukseflux, Inc., The Netherlands)
- Reflectômetro CS616 Soil Moisture Sensor (Campbell Scientific, Inc., USA)
- Radiômetro Infravermelho de Precisão SI-111 (Precision Infrared Radiometer - Campbell Scientific, Inc., USA)
- Pluviômetro de báscula TR-525M Rainfall Sensor (Texas Eletronics, Inc., USA)
- Dataloggers CR 1000 (Campbell Scientific, Inc., USA)
- Placa multiplexadora AM16/32^a-ST-SW (Campbell Scientific, Inc., USA)

3.2 Metodologia proposta

Neste trabalho, a metodologia proposta para preenchimento de falhas em dados micro-meteorológicos leva em consideração os valores mensurados por outros sensores no mesmo momento que ocorreu a falha. Ou seja, para preencher a falha de um sensor que mede o saldo de radiação, por exemplo, serão considerados os valores mensurados da temperatura, insolação e velocidade do vento, naquele mesmo instante.

Para tanto, a partir dos dados coletados pela estação micrometeorológica, serão identificadas quais variáveis climáticas estão relacionadas com o sensor que falhou em sua leitura e, posteriormente, será calculado o valor ausente, com base nessas variáveis climáticas. Na Figura 8 são mostradas as etapas da metodologia proposta.

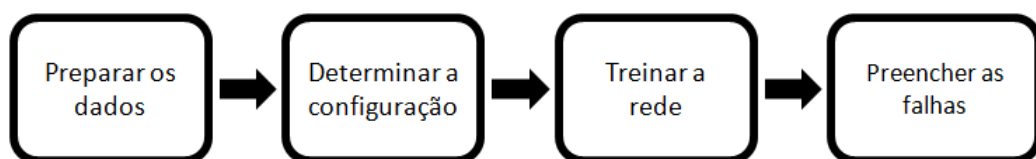


Figura 8: Etapas da metodologia proposta para preenchimento de falhas

Inicialmente, os dados devem ser pré-processados para que seja possível distinguir entre o sensor que deve ser restaurado com os sensores que ajudarão nesse processo.

O segundo passo consiste em determinar os valores dos parâmetros de treinamento que serão utilizados na RNA, utilizando AG, e verificar quais sensores estão relacionados com o sensor que sofreu a falha.

Após a especificação dos parâmetros da rede, a terceira fase realiza o treinamento da RNA, que tem o objetivo de preencher as falhas do sensor escolhido. Com a RNA treinada, será possível utilizá-la para estimar os valores dos dados com falhas, para que possam ser preenchidos.

Para ilustrar esse procedimento, o funcionamento da metodologia proposta pode ser visto de forma mais detalhada na Figura 9. Nela são mostrados os principais passos utilizados no preenchimento de falhas de dados micrometeorológicos.

Primeiramente alguns dados sem falhas são utilizados pelo AG com o intuito de fazer com que o sistema entenda os padrões dos dados. Assim, o AG irá determinar a melhor configuração para a RNA que preencherá as falhas. Durante o processamento do AG, serão criadas várias RNAs de acordo com as configurações armazenadas nos indivíduos gerados pelo próprio AG.

Cada RNA será avaliada utilizando os dados inseridos no AG para verificar se tal configuração é boa para o preenchimento de falhas. Isso é feito comparando o resultado estimado pela RNA com o valor real medido pelos sensores, por isso a importância de iniciar o AG com uma série de dados sem falhas. No final da execução do AG um indivíduo será selecionado como o mais apto da população, isso significa que o seu conteúdo representa a melhor configuração da RNA para o preenchimento de falhas destes dados.

A RNA criada de acordo com as configurações do indivíduo mais apto, será utilizada para o preenchimento de falhas da série de dados. Depois, é possível executá-la tendo como entrada os dados que possuem falhas em seus registros. No final da execução da RNA todos os dados inválidos serão substituídos por valores estimados pela RNA, preenchendo todas as falhas.

Nas seções a seguir, serão apresentados detalhes de cada fase dessa metodologia, a preparação dos dados, determinação da configuração da RNA, treinamento da RNA e, finalmente, o preenchimento de falhas.

3.2.1 Preparação dos dados

A etapa de preparação dos dados é responsável por pré-processar a base de dados de maneira que se possa trabalhá-los de maneira adequada. É necessário escolher quais atributos da base serão utilizados, já que o processamento demanda muito esforço,

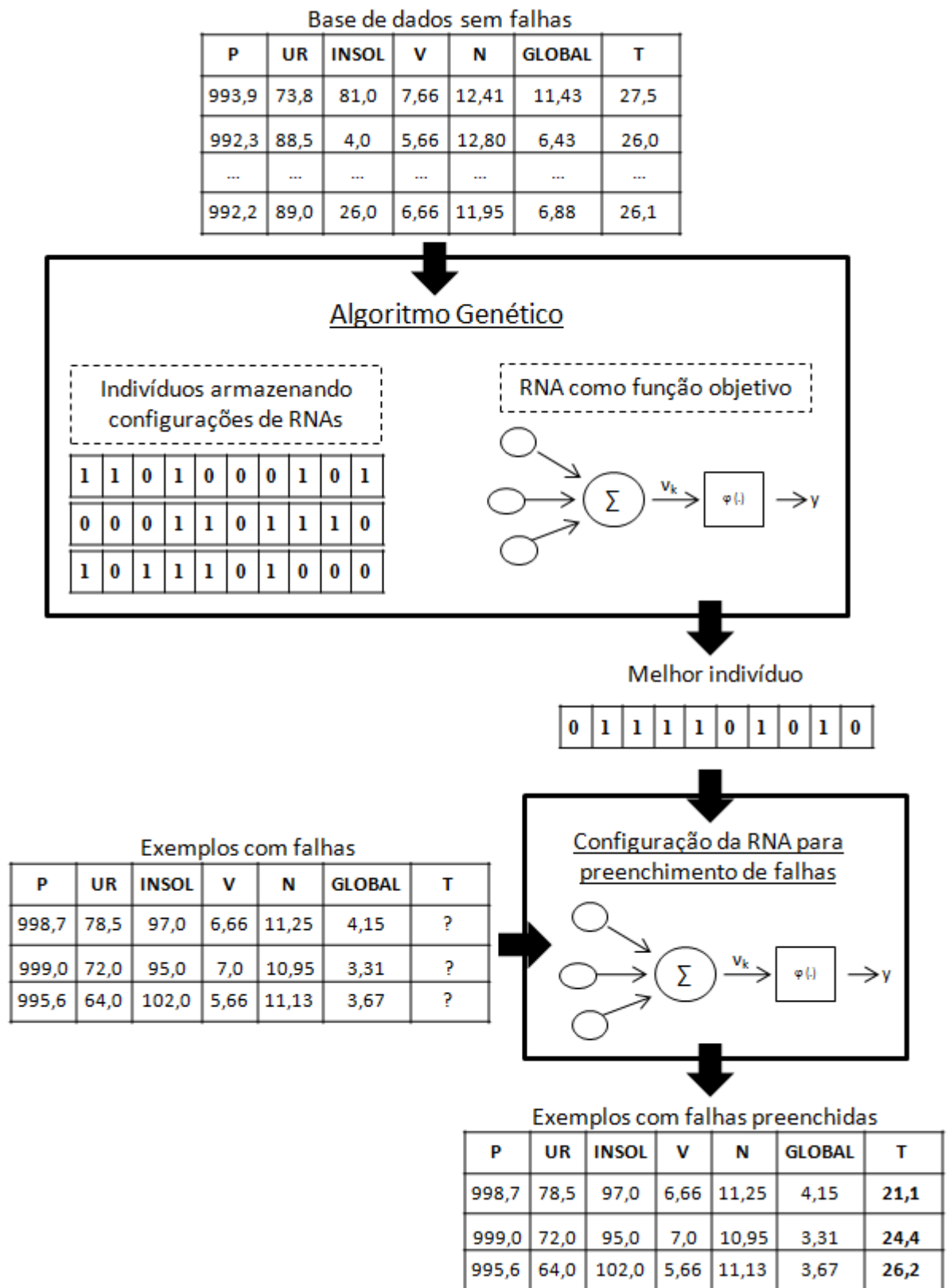


Figura 9: Diagrama da sequência de passos necessários para realizar o preenchimento de falhas

e deixar dados irrelevantes na base poderia aumentar o tempo de processamento. A escolha dos atributos geralmente é realizada por um especialista capaz de identificar quais variáveis climáticas estão relacionadas.

Um ponto importante nesta etapa é verificar se todos os valores das séries de dados estão no formato numérico. Tanto o AG quanto a RNA trabalham melhor com dados numéricos, então se houver algum atributo na série de dados que apresente valores no formato alfabético, estes devem ser codificados em números antes de inserí-los na arquitetura e decodificados depois dos resultados finais, retornando ao seu formato original.

Na Tabela 2 é apresentada uma amostra de como ficou a base de dados após esse pré-processamento. Assim, o conjunto de dados utilizado nesse trabalho contém dados que representam um ano com 365 dias (exemplos) e os seguintes atributos: as pressão atmosférica (P), temperatura média (T), umidade relativa do ar (UR), insolação (IN-SOL), velocidade média do vento (V), fotoperíodo (N) e saldo de radiação (Qn). No total permaneceram sete atributos.

Tabela 2: Amostra de dados obtidos pelos equipamentos micrometeorológicos

P	T	UR	INSOL	V	N	Qn
991,9	27,7	79,8	85	8	11,99	10,42
992,3	26,9	84,8	63	7,66	11,98	9,16
992,0	26,5	86,5	51	7,33	11,96	8,35
992,2	26,1	89,0	26	6,66	11,95	6,88
993,0	24,7	98,3	7	5,66	11,93	5,81
992,5	25,8	87,0	33	6,66	11,92	7,29
994,3	26,0	86,0	40	7	11,90	7,58

O próximo passo é dividir a base de dados em um conjunto de treinamento e um conjunto de teste. O conjunto de treinamento serve para o AG e a RNA aprenderem os padrões das séries de dados e, com isso, encontrar um modelo que possa preencher as falhas quando necessário. Já o conjunto de teste servirá para verificar se a RNA aprendeu corretamente os padrões das séries de dados, validando sua configuração.

Para análise dos resultados foram realizadas várias simulações diferentes, mais especificamente cinco maneiras diferentes. O objetivo foi verificar o desempenho da metodologia proposta quando houvesse poucos dados que apresentassem falhas e, de forma crescente, quando houvesse várias falhas na série de dados.

Na primeira simulação apenas 5% dos dados foram aleatoriamente selecionados como dados inválidos. Isso quer dizer que esses 5% foram utilizados na fase de testes, no qual o sistema não tinha conhecimento sobre qual o valor correto para o respectivo sensor. Os outros 95% serviram para a fase de treinamento, ou seja, dados que são usados para aprender os padrões presentes nas séries de dados. A mesma abordagem

foi utilizada nas demais simulações, para 10%, 20%, 30% e, por fim, 40% representados como dados inválidos. Na Tabela 3 são apresentadas as informações das cinco simulações.

Tabela 3: Número de exemplos sinalizados como inválidos para a fase de testes e número de exemplos reservados para a fase de treinamento em cada simulação realizada

Simulação	% Falhas	Quantidade de exemplos	
		com Falhas	para Treinamento
1	5	18	347
2	10	36	329
3	20	73	292
4	30	109	256
5	40	146	219

3.2.2 Determinando a configuração da RNA com AG

Nesta etapa é determinada a configuração da RNA responsável pelo preenchimento de valores em dados com falhas. O algoritmo genético será utilizado para determinar quais sensores serão utilizados para preencher as falhas (seleção de atributos) e determinar os parâmetros para o treinamento da RNA.

É possível que na entrada de dados existam diversos tipos de sensores diferentes, cada um com milhares de medidas de uma variável climática. É interessante que o AG selecione previamente quais sensores estão relacionados com o sensor que ocasionou o erro a fim de evitar duas situações:

- Dados de um sensor, que não tenha relação direta com o sensor a ser ajustado, gere falsos padrões entre os dados, causando um baixo desempenho no preenchimento dos valores com falha.
- Aumento do volume de dados no processamento da RNA e, assim, aumento do tempo de processamento da mesma.

Com relação aos parâmetros da RNA, o AG irá testar vários valores diferentes com a intenção de obter a melhor combinação dos parâmetros de acordo com os dados utilizados no treinamento. Neste trabalho, o AG determinará os seguintes parâmetros da RNA:

- Função de ativação da camada oculta e da camada de saída: função responsável por fornecer o valor de saída de um neurônio para a próxima camada.
- Algoritmo de treinamento: procedimento que escolhe o algoritmo que determina como a rede neural deve ser treinada.

- Taxa de aprendizagem: parâmetro que determina a velocidade do aprendizado da rede neural, sendo que um valor muito alto pode causar oscilações durante o treinamento e dificultar a aprendizagem da RNA.
- Taxa de *momentum*: parâmetro que também está relacionado com a velocidade do aprendizado, que considera as mudanças nas interações anteriores da busca para diminuir a instabilidade no treinamento.

Um ponto importante em um algoritmo genético é a forma que os indivíduos de sua população são estruturados. Como visto na Seção 2.4.1, os indivíduos armazenam em sua estrutura parâmetros que representam uma possível solução para o problema.

O número de bits influencia no número de combinações possíveis que o AG pode gerar. Quanto maior o número de combinações, maior são as chances de ter um indivíduo com a solução ideal para o problema a ser solucionado. Entretanto, o número de combinações possíveis é proporcional ao tempo de processamento para achar uma boa solução. Por isso é necessário haver um balanceamento do número de bits que cada parâmetro terá armazenado nos indivíduos.

Neste trabalho, os indivíduos do AG foram codificados por 18 bits, ilustrados na Figura 10, na qual representam:

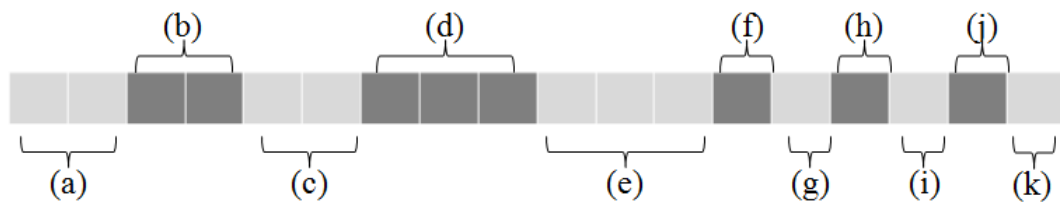


Figura 10: Representação do indivíduo no AG

(a) e (b): Função de ativação da camada oculta **(a)** e função de ativação da camada de saída **(b)**.

Neste caso, dois bits foram reservados para cada camada, possibilitando que quatro tipos de função de ativação possam ser escolhidas para cada uma. As quatro funções de ativação selecionadas foram escolhidas por serem utilizadas com frequência na literatura. As funções de ativação utilizadas são (MATHWORKS, 2011):

- *Linear transfer function* (bits 10)

$$\text{purelin}(n) = n \quad (1)$$

- *Log sigmoid transfer function* (bits 01)

$$\text{logsig}(n) = \frac{1}{1 + e^{-n}} \quad (2)$$

- *Hyperbolic tangent sigmoid transfer function* (bits 00)

$$\text{tansig}(n) = \frac{2}{(1 + e^{-2 \times n})} - 1 \quad (3)$$

- *Saturating linear transfer function* (bits 11)

$$\text{satlin}(n) = \begin{cases} 0, & \text{se } n \leq 0 \\ n, & \text{se } 0 \leq n \leq 1 \\ 1, & \text{se } 1 \leq n \end{cases} \quad (4)$$

(c): Algoritmo de treinamento.

Também foram reservados dois bits para representar o algoritmo de treinamento, podendo então ser selecionados quatro algoritmos de treinamento. As quatro opções de algoritmo de treinamento utilizados neste trabalho foram selecionados por já se saber que tais algoritmos são eficazes com estes tipos de dados. Os algoritmos de treinamento utilizados são (MATHWORKS, 2011):

- *Gradient descent backpropagation* (bits 10): atualiza os valores dos pesos e do limiar baseado na direção negativa do gradiente. Uma limitação é que o resultado pode convergir para um ponto de mínimo local.
- *Levenberg-Marquardt backpropagation* (bits 01): atualiza os valores dos pesos e do limiar de acordo com o algoritmo de Levenberg-Marquardt (LM). Segundo (LOURAKIS, 2005), o algoritmo de LM pode ser pensado como uma combinação mais acentuada do método de Gauss-Newton. A vantagem desse algoritmo de treinamento é que ele é muito rápido, apesar de consumir muita memória.
- *Resilient backpropagation* (bits 00): atualiza os valores dos pesos e do limiar com base no algoritmo do próprio nome. Segundo Chen e Su (2010), é um algoritmo de primeira ordem e seu tempo e escalas de requisito de memória mudam linearmente com o número de parâmetros, o que é bom se estiver trabalhando com muitos parâmetros.
- *Gradient descent with momentum and adaptive learning rate backpropagation* (bits 11): além de levar em consideração o gradiente para atualizar os valores dos pesos e do limiar, considera também a taxa de *momentum*.

(d) e (e): Taxa de aprendizado **(d)** e taxa de *momentum* **(e)**.

Para essas taxas foram reservados três bits, possibilitando que os seus valores variem de 0,1 a 0,8.

(f) - (k): Sensores.

Para cada sensor foi reservado um bit indicando se o mesmo deve ou não ser utilizado no preenchimento de falhas. Por exemplo, se o sensor de temperatura estiver sendo sinalizado pelo bit **(h)**, e este bit estiver com o valor 1, significa que o sensor de temperatura deve ser utilizado. Se estiver sinalizado como 0, significa que este sensor deve ser ignorado.

Um detalhe importante é que quando o EnvRP for implementado utilizando esta metodologia, será possível fazer com que tanto as funções de ativações quanto os algoritmos de treinamento sejam parametrizados. Ou seja, haverá a possibilidade de que a cada execução do sistema possam ser testados tipos diferentes de funções de ativação ou algoritmos de treinamento, havendo uma maior liberdade para a sua escolha. Outros parâmetros também poderão ser modificados a cada execução, como, por exemplo, o número de ciclos máximo do AG ou de iterações da RNA, que atualmente estão fixos em 100 e 400, respectivamente.

Outra questão importante de um algoritmo genético é a sua função objetivo. Nesse caso essa função é calculada da seguinte forma. Primeiramente uma RNA é criada utilizando os parâmetros definidos no indivíduo a ser avaliado. Posteriormente a RNA é treinada e seu desempenho é avaliado. Quanto melhor o desempenho da RNA, mais apto é o indivíduo avaliado.

Quando a população de indivíduos atinge um erro aceitável, o AG deixa de ser executado e o resultado é apresentado mostrando as melhores combinações de sensores e parâmetros da rede neural, que poderá preencher as falhas encontradas.

3.2.3 Treinamento da RNA

A RNA é treinada de acordo com a configuração obtida pelo AG. Logo, será o AG que especifica qual função de ativação da camada oculta, função de ativação da camada de saída, algoritmo de treinamento, taxa de aprendizagem e taxa de *momentum* que serão atribuídas à arquitetura da RNA.

No caso do número de neurônios da RNA, foi utilizado o teorema de Kolmogorov-Nielsen, citado na Seção 2.3.1. O n representa o número de neurônios na camada de entrada. Assim, a RNA utilizada nas simulações deste trabalho tem no máximo $n = 6$ (número de sensores), e contém no máximo $2n + 1 = 13$ neurônios na camada oculta.

Esse valor pode ser menor, já que o número de entradas pode ser reduzido caso seja detectado que não há necessidade do uso de algum sensor.

A RNA então tem todos os parâmetros configurados e, assim, é possível executá-la iniciando o treinamento. A RNA terá como entrada o conjunto de dados separados para o treinamento e, depois de vários ciclos de processamento, o resultado da RNA identificará o quão apta está a estrutura para resolver o problema proposto de preenchimento de falha. Com a rede treinada, ela está pronta para que o preenchimento de dados com falhas seja iniciado.

3.2.4 Preenchimento dos valores ausentes

O processo nesta fase consiste em informar à RNA quais os valores dos outros sensores no mesmo momento que ocorreu a falha em questão, como mostrada na Figura 11.

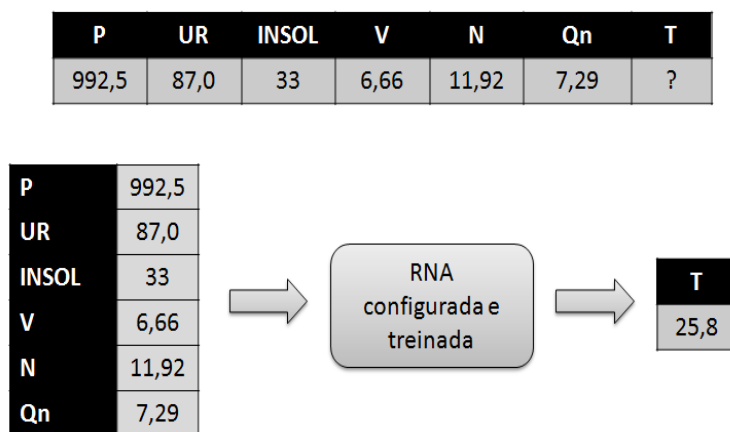


Figura 11: Ilustração do preenchimento de um valor ausente utilizando a RNA treinada

No exemplo da Figura 11, a medida do sensor de temperatura não foi armazenada com sucesso, causando uma falha. Para resolver isso, todas as outras medidas, dos outros sensores, no mesmo momento que houve a falha foram enviadas para a RNA treinada. Esta, por sua vez, com base nos valores informados, consegue estimar o valor da temperatura que deveria ser armazenado e, assim, preenchendo o valor ausente.

Esse procedimento foi utilizado para preencher uma única falha. Mas a partir do momento que a RNA está treinada, ela pode ser executada em quantas falhas forem necessárias, desde que a falha seja da mesma série de dados. Caso a falha seja de outra série de dados, a RNA deve ser novamente treinada para que aprenda os padrões dessa nova série de dados.

Vale ressaltar que, para preencher uma falha não são utilizadas medidas dos outros sensores em um momento diferente de quando ocorreu a falha, porque são utilizados apenas valores do mesmo momento que ocorreu a falha. Isso significa que mesmo se

houver uma sequência de falhas consecutivas, como por exemplo 5 dias seguidos de falhas, a metodologia proposta neste trabalho conseguiria tratar esse problema. Esse é um ponto que se diferencia de outras metodologias, que normalmente utilizam dados próximos à falha para substituir o valor inválido, impossibilitando que tenha uma boa precisão em falhas sequenciais.

3.3 Ambiente de programação

A implementação tanto do algoritmo genético quanto da rede neural artificial foi desenvolvida utilizando o ambiente de programação do software Matlab (MATLAB, 2011), utilizando a versão R2010a. Na Figura 12 é mostrada a sua interface.

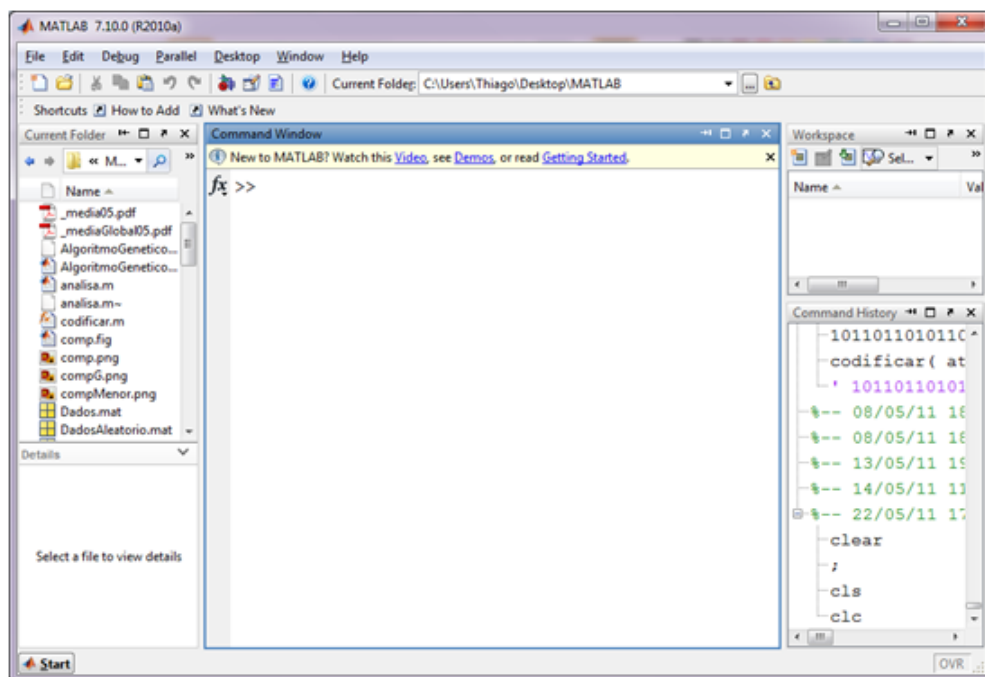


Figura 12: Ambiente de programação do Matlab

O Matlab é um ambiente que contém uma linguagem de alto desempenho, principalmente para as operações matemáticas que precisam de um processamento maior. Ele foi planejado para obter resultados rápidos para as operações mais complexas, que é o caso dos procedimentos realizados com os algoritmos genéticos e as redes neurais artificiais.

Tanto o algoritmo genético quanto a rede neural artificial foram programados nesse ambiente, sendo que a RNA utilizou um componente chamado Neural Network Toolbox. A Neural Network Toolbox provê ferramentas para projeto, implementação, visualização e simulação de uma rede neural (BEALE; HAGAN; DEMUTH, 2011).

Agregando os fundamentos de algoritmo genético e redes neurais pesquisados, no ambiente de programação Matlab, foi possível desenvolver a metodologia proposta para o preenchimento de falhas em dados micrometeorológicos.

No Apêndice A pode ser visto o código fonte da implementação do algoritmo genético, assim como suas funções auxiliares para codificar e decodificar um indivíduo. No Apêndice B é mostrado o código fonte da rede neural artificial.

3.4 Avaliação de desempenho

A validação estatística da estimativa do modelo foi baseada nos erros individuais (e_i) de cada estimativa, mostrado na Eq. (5), onde $P_i (i = 1, 2, \dots, n)$ são os valores estimados e $O_i (i = 1, 2, \dots, n)$ são os valores reais (WILLMOTT; MATSUURA, 2005).

$$e_i = P_i - O_i \quad (5)$$

Com o erro individual de cada estimativa, é calculado o desempenho do modelo usando do *Mean Absolute Error* (MAE), mostrado na Eq. (6), no qual a diferença da média pode ser alternativamente descrita pela raiz quadrada média do erro (WILLMOTT, 1982). Segundo Willmott, Matsuura e Robeson (2009), essa é a melhor forma de avaliação para modelos ambientais, devido principalmente a presença de *outliers*, ou seja, dados discordantes comparados ao restante da série de dados (PIRES A. M., 1999), e dados com desvio de normalidade.

$$MAE = \left[n^{-1} \sum_{i=1}^{i=n} |e_i|^2 \right] \quad (6)$$

Além da avaliação pelo erro médio absoluto, foi obtido o desvio padrão de cada simulação para assegurar se o preenchimento de falhas teve uma boa precisão. Segundo Assumpção et al. (2010), o desvio padrão é a maneira mais comum de medir a variação em um conjunto de amostras, quanto menor o desvio padrão, menor é o erro entre elas. Em Santos et al. (2007) é mostrado como deve ser feito o cálculo do desvio padrão (σ), representado na Eq. (7).

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (K_i - \bar{K})^2}{n - 1}} \quad (7)$$

Onde K_i os valores estimados e $\bar{K}$ o valor esperado dado pela Eq. (8).

$$\bar{K} = \sum_{i=1}^n \frac{K_i}{n} \quad (8)$$

Capítulo 4

Resultados e Discussão

Para analisar o desempenho da metodologia proposta foram realizadas diversas simulações de preenchimento de falhas sobre os dados apresentados na Seção 3.1, para os valores de cada sensor. Como citado na metodologia, primeiramente houve uma preparação dos dados, resultando em uma base semelhante à mostrada na Tabela 2, depois houve o processamento do AG e, posteriormente, o processamento da RNA para preencher as falhas na série dos dados.

Nas seções a seguir são detalhados os resultados do AG, da RNA e da análise estatística para as simulações realizadas dos sensores de temperatura, saldo de radiação, fotoperíodo, umidade e insolação.

4.1 Resultados obtidos

Para cada simulação usando os sensores, são mostrados os melhores resultados encontrados pelo AG, no qual:

- f_1 : função de ativação utilizada para a camada oculta.
- f_2 : função de ativação utilizada para a camada de saída.
- *train*: algoritmo de treinamento.
- *lr*: taxa de aprendizado.
- *mc*: taxa de *momentum*.
- *s*: simboliza o uso dos sensores.

As funções de ativações disponíveis foram a Saturating linear transfer function (satlin), Linear transfer function (purelin), Hyperbolic tangent sigmoid transfer function (tansig) e Log sigmoid transfer function (logsig). Os algoritmos de treinamento

possíveis foram o Resilient backpropagation (*trainrp*), Gradient descent backpropagation (*traingd*), Levenberg-Marquardt backpropagation (*trainlm*) e Gradient descent with momentum and adaptive learning rate backpropagation (*traingdx*).

O s simboliza os sensores, no qual s_t , s_s , s_f , s_u , s_i , s_v e s_p representam, respectivamente: temperatura, saldo de radiação, fotoperíodo, umidade relativa do ar, insolação, velocidade média do vento e pressão atmosférica, e informam se o sensor foi utilizado na entrada da RNA.

Nos preenchimento de falhas realizados, houve um tempo de processamento que durou entre uma a quatro horas para cada simulação.

4.1.1 Resultado do preenchimento de falhas para temperatura

Na Tabela 4 é mostrado o resultado do AG para as simulações de temperatura.

Tabela 4: Resultados encontrados pelo AG nos testes realizados da temperatura (T)

% Falhas	f_1	f_2	$train$	lr	mc	s_p	s_u	s_i	s_v	s_f	s_s
5	tansig	tansig	trainrp	0,5	0,1		s	s		s	s
10	satlin	tansig	traingd	0,2	0,4	s	s	s	s	s	s
20	tansig	tansig	trainrp	0,2	0,7		s	s	s	s	s
30	logsig	purelin	trainrp	0,1	0,2		s			s	s
40	tansig	purelin	trainrp	0,1	0,1		s	s		s	s

Por exemplo, na simulação com 5% de dados com falhas, a arquitetura da RNA é definida com as funções de ativação Hyperbolic tangent sigmoid transfer function (*tansig*) na camada oculta e na camada de saída, o algoritmo de treinamento Resilient backpropagation (*trainrp*), uma taxa de aprendizado 0,5, uma taxa de momentum 0,1 e tendo como entrada as medições de umidade relativa do ar (s_u), insolação (s_i), fotoperíodo (s_f) e saldo de radiação (s_s).

É interessante observar que com 10% de falhas, foi utilizado como entrada os valores de todos os sensores e com 30% foram necessários apenas três sensores: umidade, fotoperíodo e saldo de radiação. O sensor de pressão atmosférica foi utilizado apenas na simulação de 10% de falhas.

Os coeficientes de correlação entre os valores medidos e os valores estimados pela RNA para os dados de temperatura são apresentados na Figura 13. Os valores mostram que há forte correlação entre os valores medidos e estimados.

Como era previsto, a maior correlação encontrada foi para o teste com 5% de falhas e o pior para o teste com 40% de falhas, já que no primeiro caso há um maior número de exemplos para o treinamento, enquanto no último caso há um menor número de exemplos para treinamento e um grande número de exemplos para o preenchimento de falhas.

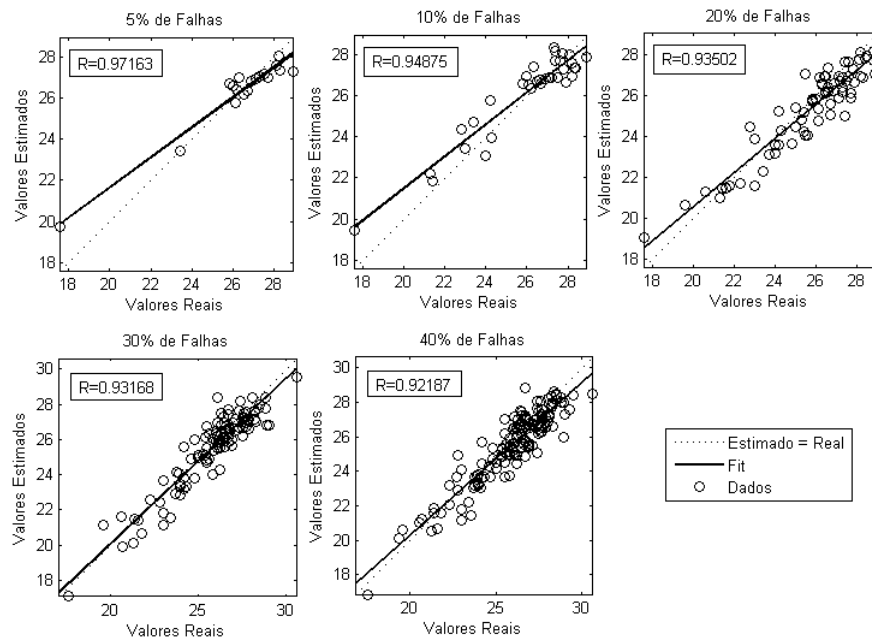


Figura 13: Valores dos coeficientes de correlação para temperatura

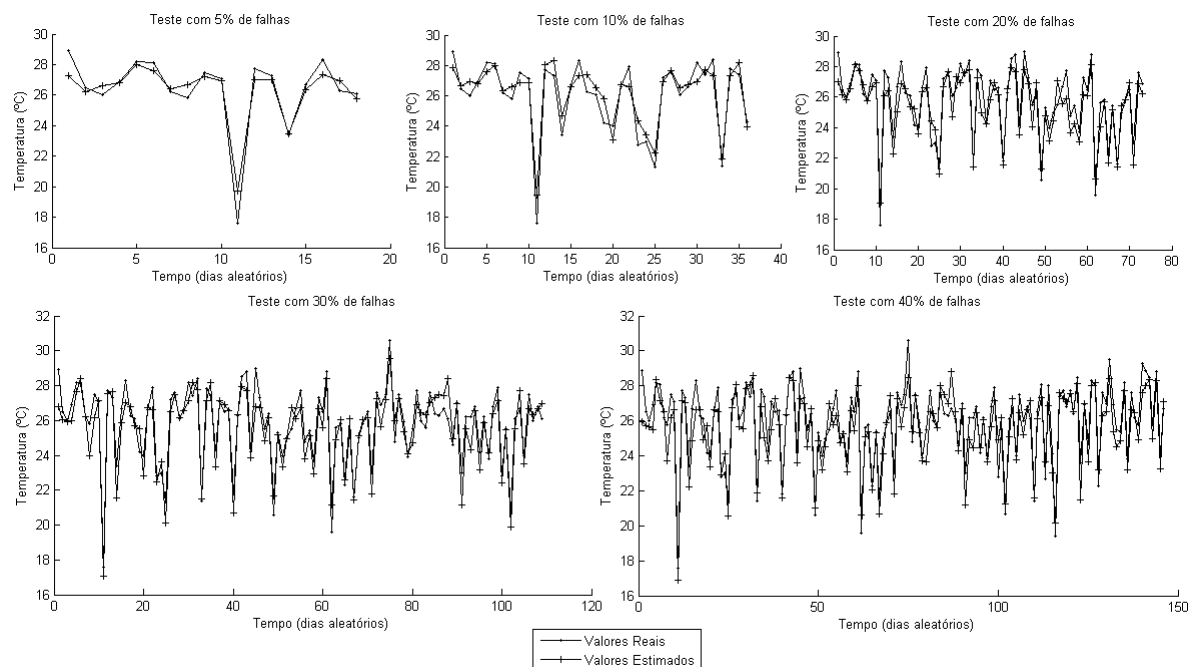


Figura 14: Valores comparativos entre os valores esperados e os valores obtidos pela RNA nos dados de temperatura

A simulação com 5% teve correlação de 0,97, já a simulação com 10% teve um resultado de aproximadamente 0,95. As simulação de 20% e 30% tiveram um resultado semelhante em 0,93. E de 40%, com a menor correlação, 0,92, porém, relativamente próximos aos anteriores.

O gráfico mostrado na Figura 14 faz um comparativo entre o valor medido pelo equipamento de temperatura (sem falhas) com os valores obtidos usando a metodologia proposta neste trabalho. Pelo gráfico pode ser observado que com apenas 5% de falhas na série de dados, o comportamento obtido pela metodologia aplicada é muito eficiente. Com o aumento da quantidade de falhas na série de dados essa eficiência diminui. É importante observar que mesmo com 40% dos dados faltantes, a metodologia teve um desempenho satisfatório no preenchimento das falhas.

4.1.2 Resultado do preenchimento de falhas para saldo de radiação

Para os testes com os dados de saldo de radiação, são mostrados os melhores resultados encontrados pelo AG para cada teste realizado, na Tabela 5.

Tabela 5: Resultados encontrados pelo AG nos testes realizados do saldo de radiação (Qn)

% Falhas	f_1	f_2	$train$	lr	mc	s_p	s_t	s_u	s_i	s_v	s_f
5	tansig	tansig	traingd	0,3	0,8		s		s		s
10	tansig	purelin	trainrp	0,8	0,5		s		s		s
20	satlin	purelin	trainlm	0,1	0,1	s	s	s	s		s
30	satlin	purelin	trainrp	0,6	0,2	s	s	s	s		s
40	satlin	purelin	trainlm	0,1	0,1	s	s	s	s		s

Pode ser observado que o sensor de velocidade do vento (s_v) não foi utilizado em nenhuma simulação de preenchimento de falhas no saldo de radiação. Isso pode acontecer porque, provavelmente, não foi encontrado relações entre esses sensores. Logo, significa que tais sensores não são necessários para preencher as falhas.

Para esses testes de preenchimento de falhas, os coeficientes de correlação entre os valores medidos e os valores estimados pela RNA são apresentados na Figura 15. Para este sensor o resultado foi melhor que o anterior no preenchimento de falhas.

Pode ser observado que em todas as simulações houve uma correlação maior que 0,99. No gráfico mostrado na Figura 16 há uma comparação do valor mensurado pelo equipamento em campo relativo ao saldo de radiação com os valores estimados.

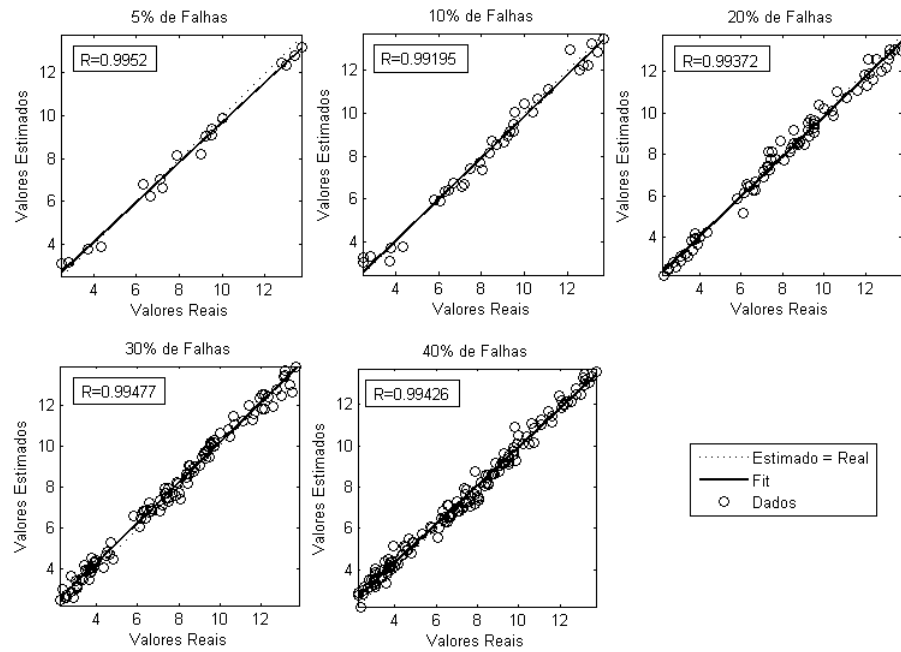


Figura 15: Valores dos coeficientes de correlação para saldo de radiação

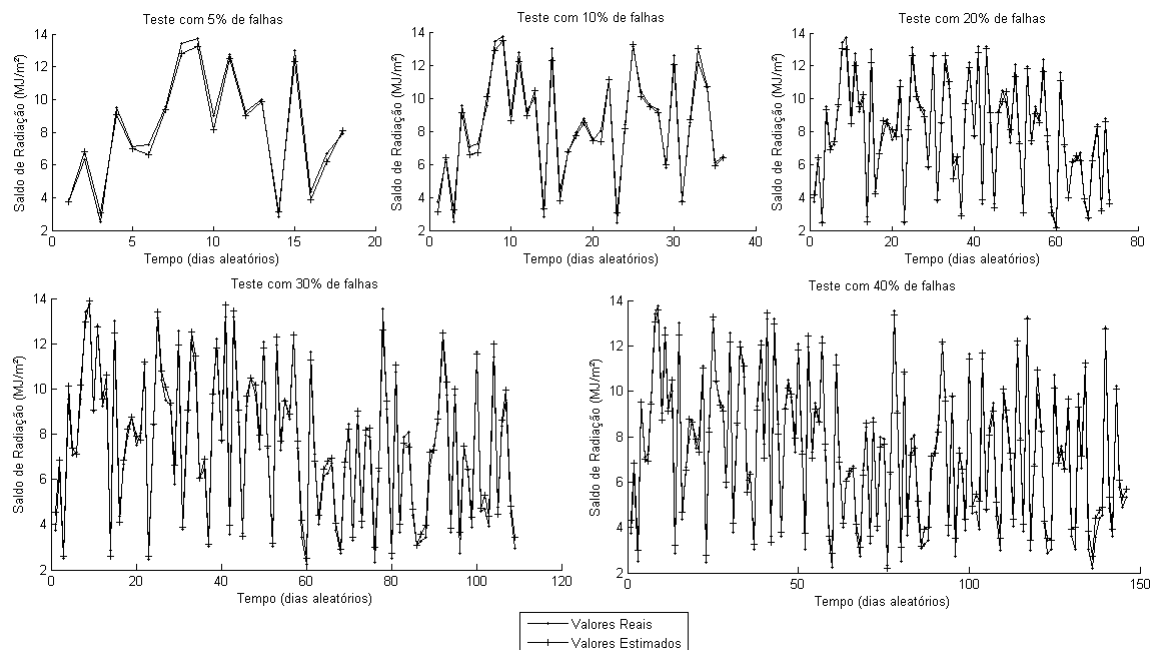


Figura 16: Valores comparativos entre os valores esperados e os valores obtidos pela RNA nos dados de saldo de radiação

4.1.3 Resultado do preenchimento de falhas para fotoperíodo

Na Tabela 6 são mostrados os melhores resultados encontrados pelo AG para cada teste realizado com os dados do fotoperíodo.

Tabela 6: Resultados encontrados pelo AG nos testes realizados do fotoperíodo (N)

% Falhas	f_1	f_2	$train$	lr	mc	s_p	s_t	s_u	s_i	s_v	s_s
5	tansig	tansig	trainrp	0,5	0,1		s	s			s
10	satlin	tansig	trainlm	0,3	0,2				s		s
20	satlin	purelin	trainrp	0,6	0,2			s	s		
30	tansig	purelin	trainrp	0,4	0,5		s	s	s		
40	satlin	purelin	trainrp	0,6	0,2	s	s	s	s		s

No resultado deste AG, em duas simulações foram utilizados apenas dois sensores para a entrada da RNA. Na simulação de 10% foram selecionados os sensores de insolação (s_i) e saldo de radiação (s_s), e na de 20% os sensores de umidade (s_u) e insolação (s_i).

No caso do preenchimento de falhas para o fotoperíodo, são apresentados na Figura 17 os coeficientes de correlação entre os valores medidos e os valores estimados.

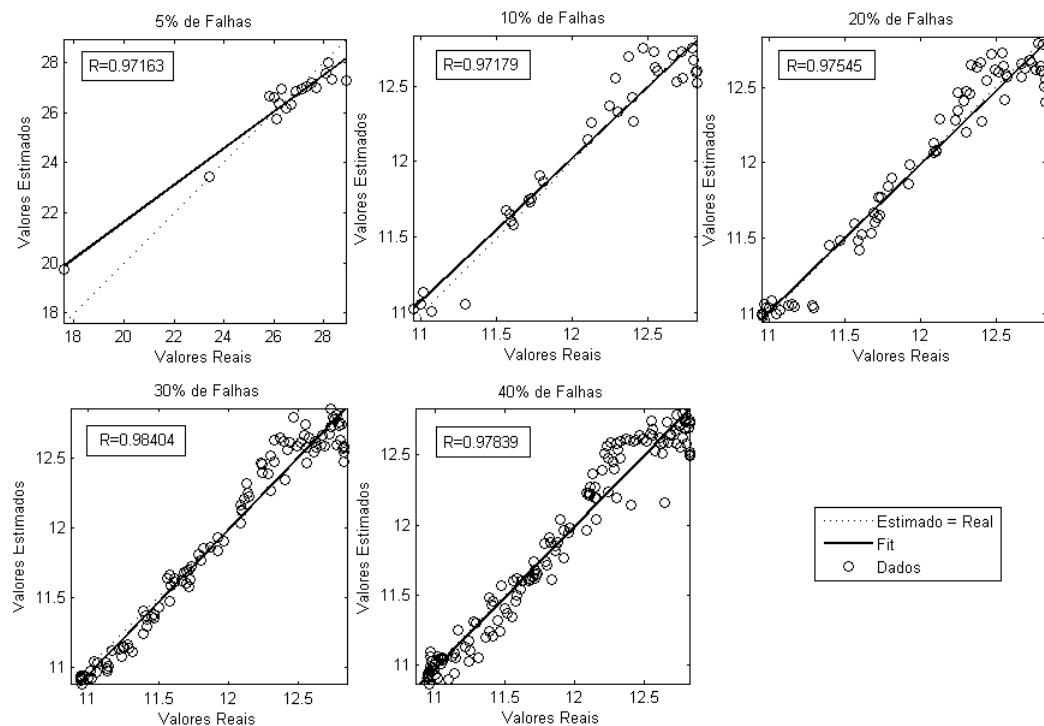


Figura 17: Valores dos coeficientes de correlação para fotoperíodo

Nas correlações do fotoperíodo é possível notar que houve uma regularidade na eficácia do preenchimento de falhas, sendo que os valores dos coeficientes de correlação ficou entre 0,971 e 0,984.

Na Figura 18 é mostrada a comparação entre os valores mensurados e os valores estimados.

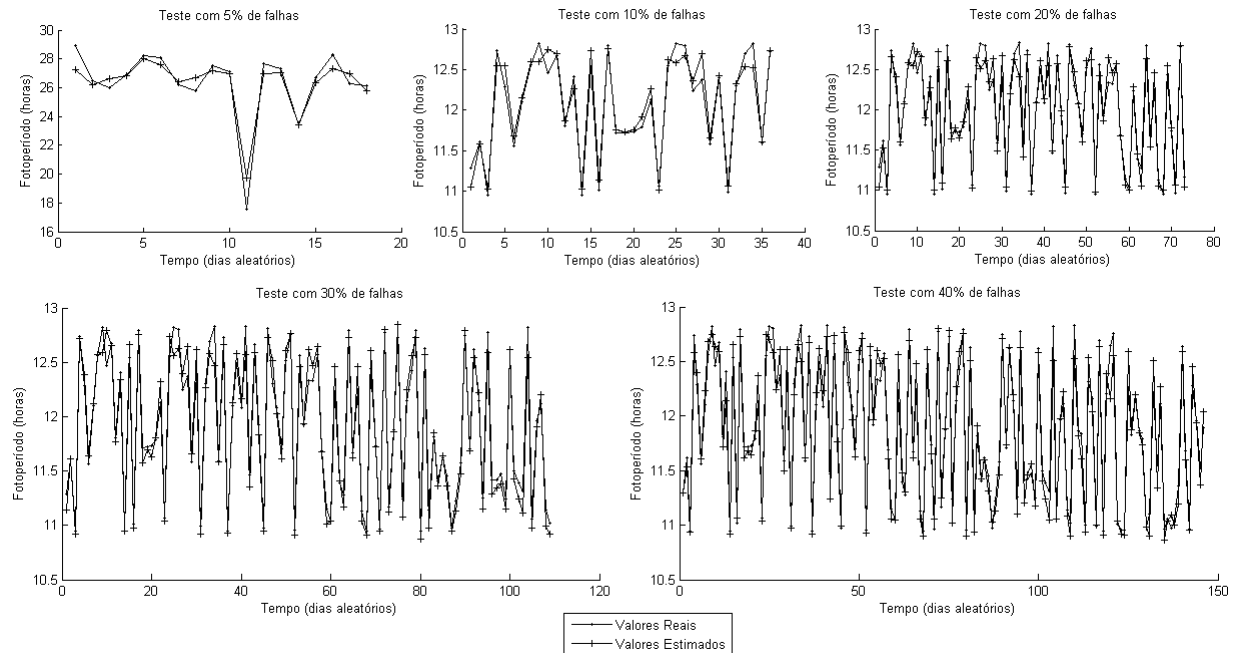


Figura 18: Valores comparativos entre os valores esperados e os valores obtidos pela RNA nos dados de fotoperíodo

4.1.4 Resultado do preenchimento de falhas para umidade

Na Tabela 7 são mostrados os melhores resultados encontrados pelo AG para cada teste realizado com os dados de umidade.

Tabela 7: Resultados encontrados pelo AG nos testes realizados da umidade (UR)											
% Falhas	f_1	f_2	$train$	lr	mc	s_p	s_t	s_i	s_v	s_f	s_s
5	satlin	purelin	trainrp	0,1	0,1			s	s	s	s
10	satlin	purelin	trainrp	0,3	0,1		s	s	s	s	s
20	tansig	tansig	trainrp	0,7	0,4		s	s	s	s	s
30	satlin	tansig	traingdx	0,4	0,6		s	s	s	s	s
40	satlin	purelin	trainrp	0,1	0,1		s	s	s	s	s

O sensor de pressão atmosférica (s_p) não foi utilizado em nenhuma simulação. Em compensação, praticamente todos os outros sensores foram utilizados em todas as simulações, com exceção do sensor de temperatura (s_t) que não foi utilizado na simulação de 5% de falhas.

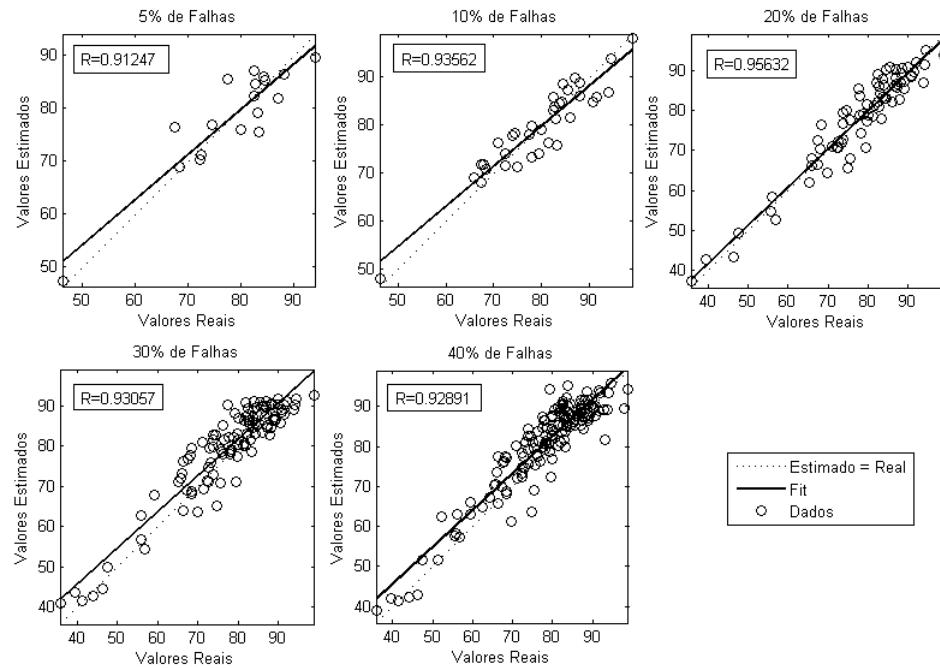


Figura 19: Valores dos coeficientes de correlação para umidade

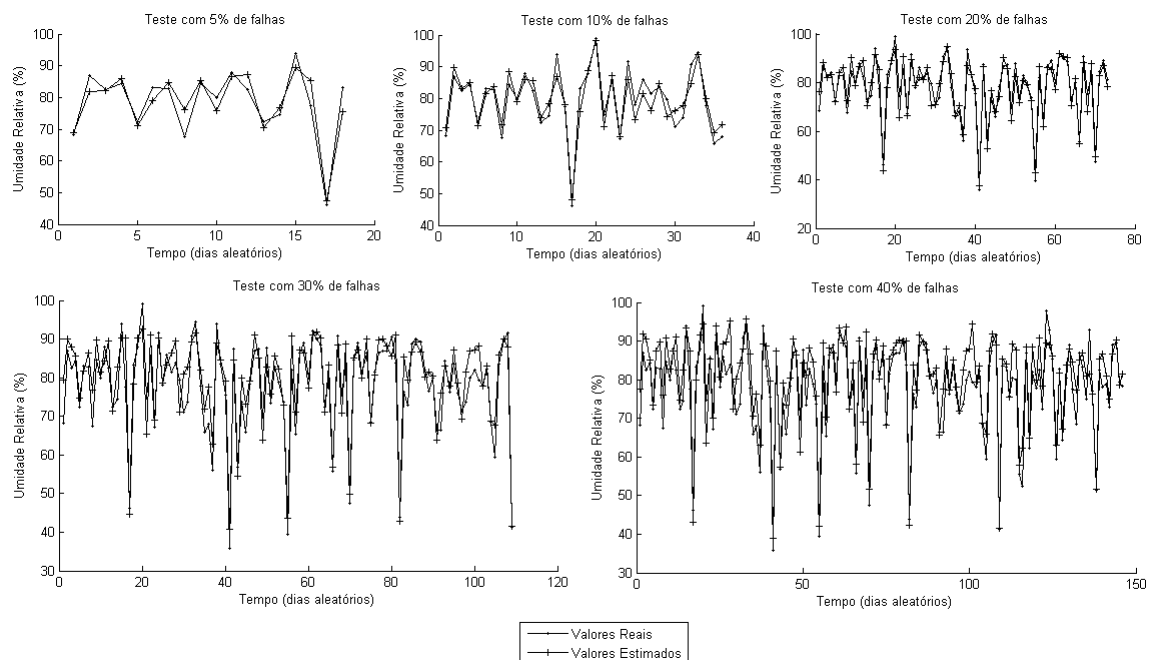


Figura 20: Valores comparativos entre os valores esperados e os valores obtidos pela RNA nos dados de umidade

Na Figura 19 é mostrado que no sensor de umidade também houve um bom desempenho no preenchimento das falhas.

O menor coeficiente de correlação dos dados de umidade foi de 0,91, e por incrível que parece na simulação com menos falhas (5%). Os valores dos coeficientes de correlação variaram de 0,912 a 0,956. É possível observar na Figura 20 a comparação entre os valores mensurados e os valores estimados dos dados de umidade.

4.1.5 Resultado do preenchimento de falhas para insolação

Por fim, na Tabela 8 são mostrados os resultados do AG com relação aos dados de insolação.

Tabela 8: Resultados encontrados pelo AG nos testes realizados da insolação (INSOL)

% Falhas	f_1	f_2	$train$	lr	mc	s_p	s_t	s_u	s_v	s_f	s_s
5	logsig	tansig	trainrp	0,5	0,7		s		s	s	s
10	satlin	tansig	trainrp	0,6	0,5	s		s	s	s	s
20	tansig	tansig	trainrp	0,7	0,4		s	s		s	s
30	satlin	purelin	trainrp	0,3	0,7		s	s	s	s	s
40	satlin	purelin	traingd	0,2	0,6		s	s	s	s	s

O sensor de pressão atmosférica (s_p) mais uma vez foi pouco utilizado (apenas na simulação de 10% de falhas). É importante ressaltar que a não seleção de um sensor não significa que a variável climática representada por este sensor não está relacionada com os dados do sensor que está sendo preenchido, significa que, neste caso, para os dados utilizados, a metodologia não precisou de seus valores para estimar o valor ausente.

Da mesma forma que nos outros sensores, na simulação do sensor de insolação houve bons resultados como pode ser visto na Figura 21.

Assim como nos dados referentes à temperatura, houve uma maior correlação no teste com 5% de falhas, diminuindo a eficácia com o aumento das falhas. A comparação entre os valores mensurados e os valores estimados dos dados de insolação podem ser vistos na Figura 22.

Pode ser observado que em séries de dados com poucas falhas (5%) ou em séries de dados com muitas falhas (40%) a metodologia utilizada conseguiu atingir os objetivos propostos. Considerando todas as simulações apresentadas, os valores de coeficiente de correlação variaram de 0,912 a 0,995.

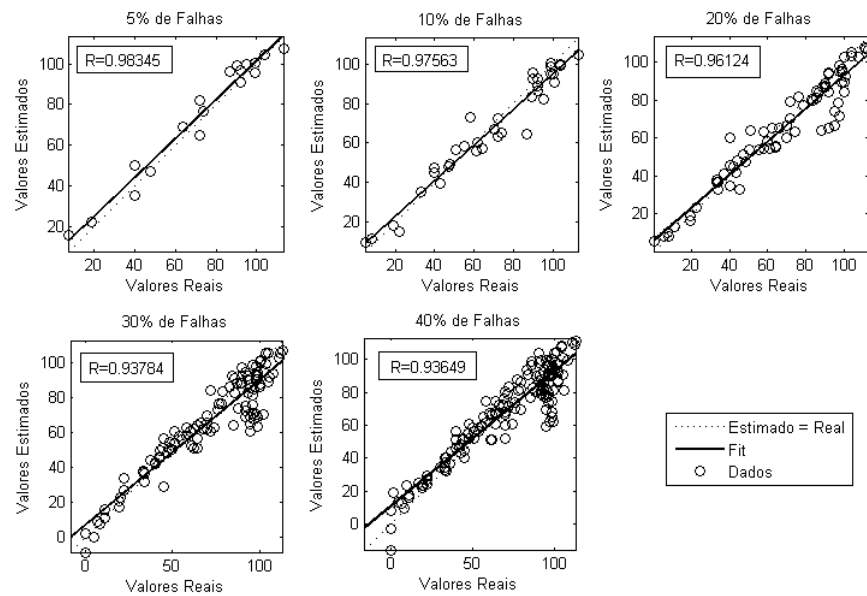


Figura 21: Valores dos coeficientes de correlação para insolação

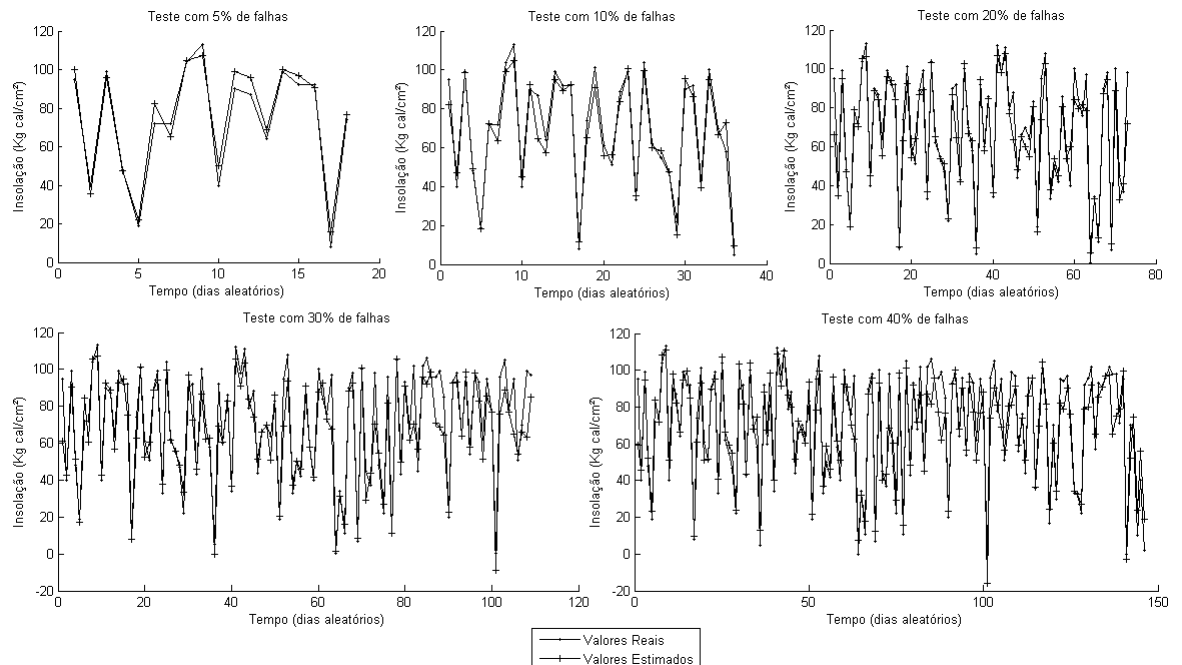


Figura 22: Valores comparativos entre os valores esperados e os valores obtidos pela RNA nos dados de insolação

4.2 Análise estatística dos resultados

Como mencionado na Seção 3.4, foi feito o cálculo do Erro Médio Absoluto (*Mean Absolute Error*) para as estimativas realizadas. Na Tabela 9 é mostrado o valor do MAE para cada simulação.

Tabela 9: Erro médio absoluto para cada simulação realizada

Falhas	MAE				
	Temperatura	Saldo de radiação	Fotoperíodo	Umidade	Insolação
5%	0,57	0,39	0,57	3,33	5,05
10%	0,68	0,34	0,11	2,94	5,26
20%	0,77	0,30	0,11	2,92	6,20
30%	0,68	0,33	0,09	3,88	8,00
40%	0,77	0,33	0,11	4,12	8,25

Os valores do MAE indicam resultados satisfatórios para os testes realizados. É importante lembrar que o valor do MAE está relacionado à unidade da variável envolvida. No caso da temperatura, podemos dizer que o erro da estimativa está em 0,57 °C para mais ou para menos no momento de preencher as falhas na simulação de 5%, o que é um valor baixo considerando que o intervalo de valores da temperatura nestes dados é de 17,24 a 30,88 °C. A mesma ideia pode ser aplicada para as outras simulações.

Outro ponto que deve ser observado é que nas simulações de 40% há pouca diferença no valor do MAE, o que significa que a metodologia proposta tem um bom desempenho inclusive com grandes quantidades de falhas. Na Figura 23 são mostrados os valores do erro médio absoluto das simulações realizadas de cada sensor.

Também foi analisado o desvio padrão dos dados para cada simulação, mostrado na Tabela 10.

Tabela 10: Valores do desvio padrão para cada simulação realizada

Falhas	σ				
	Temperatura	Saldo de radiação	Fotoperíodo	Umidade	Insolação
5%	1,90	3,32	1,90	9,94	29,49
10%	2,01	3,13	0,60	9,09	27,55
20%	2,10	3,19	0,64	12,58	27,55
30%	2,19	3,21	0,67	11,87	26,97
40%	2,13	3,06	0,66	11,85	27,07

Nos testes realizados, a diferença entre o valor real (medido em campo) e o valor estimado raramente é maior do que o desvio padrão calculado para cada simulação, mostrando que a estimativa gerada pela metodologia está tendo um bom desempenho no preenchimento de falhas. Além disso, com base nos resultados encontrados nos

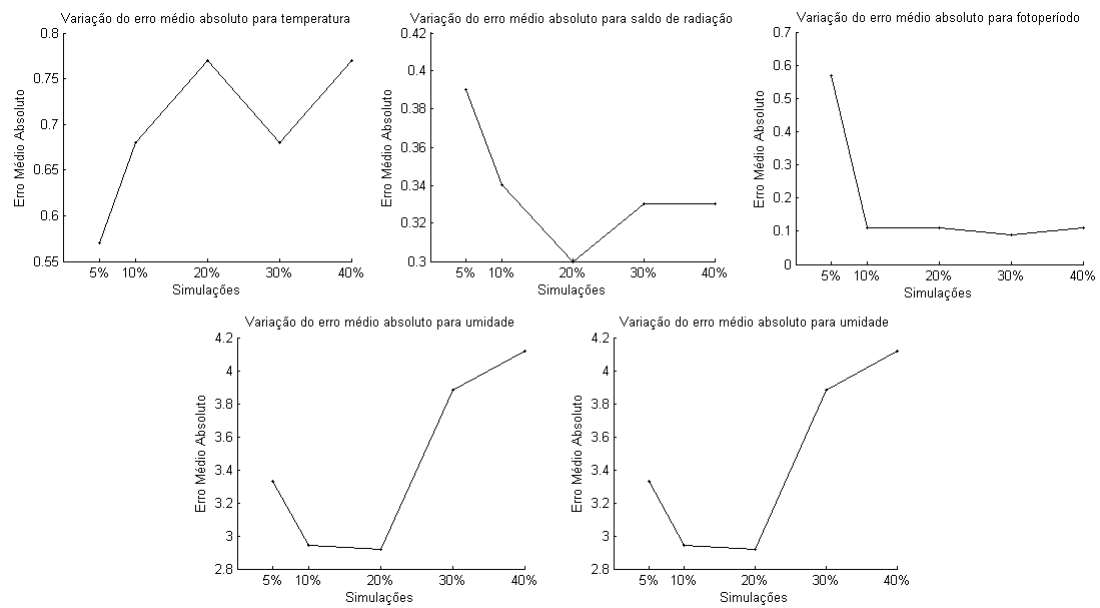


Figura 23: Erro médio absoluto de todas as simulações realizadas

coeficientes de correlação e nos MAEs para as variáveis climáticas utilizadas, pode ser dito que a eficácia da metodologia proposta foi validada.

Os resultados também estão sendo validados por especialistas do domínio e trabalhos comparativos com outras técnicas estão sendo analisadas.

Capítulo 5

Considerações Finais

A proposta deste trabalho consistiu em apresentar uma metodologia computacional usando técnicas de inteligência artificial para preenchimento de valores com falhas em séries de dados de estações micrometeorológicas. Os valores utilizados no preenchimento das falhas são estimados, baseando-se nos valores de outras séries de dados mensurados, por outros sensores obtidos no mesmo local e momento da falha detectada.

A integração de duas técnicas computacionais possibilitou o desenvolvimento da metodologia capaz de realizar o preenchimento de falhas. Foram utilizadas as técnicas de algoritmos genéticos e redes neurais artificiais. A técnica de RNAs é responsável por aprender e estimar os valores para o preenchimento de falhas, a partir de uma definição e configuração prévia da RNA. Considerando que o AG é uma técnica utilizada em problemas de busca e otimização, capaz de testar várias configurações diferentes em um tempo muito menor do que se fosse testado manualmente, o AG foi utilizado para encontrar a melhor configuração da RNA.

Além do AG determinar quais os parâmetros da configuração da RNA, ele também avalia quais sensores disponíveis são mais indicados para servir de entrada para o preenchimento de falhas. É comum em séries de dados micrometeorológicos haver vários sensores diferentes, entretanto, se for utilizar todos os sensores disponíveis a metodologia levará mais tempo para fazer o seu processamento. Além do mais, pode acontecer que um sensor represente uma variável climática que tenha um comportamento totalmente diferente da variável climática que se está corrigindo, isso fará com que a precisão da estimativa seja afetada. Assim, o AG é responsável também por selecionar apenas os sensores que poderão ser úteis para o preenchimento de falhas, reduzindo o tempo de processamento e aumentando a precisão da estimativa.

Para testar a eficácia da metodologia proposta, foram realizadas várias simulações de preenchimento de falhas. O resultado de cinco sensores foram apresentados, re-

presentando as variáveis climáticas de temperatura, saldo de radiação, fotoperíodo, umidade e insolação. Para cada sensor foram realizados cinco testes diferentes, no qual a quantidade de valores com falhas dentro da série de dados aumentava. O primeiro teste continha 5% dos dados com falhas, o segundo 10% de falhas, o terceiro com 20%, o quarto com 30% e o último teste com 40% de falhas. Ao todo, foram feitas 25 simulações de preenchimento de falhas, considerando os cinco sensores.

Os resultados de cada teste foram avaliados usando o coeficiente de correlação, o erro médio absoluto e o desvio padrão. No teste com a menor quantidade de falhas foi obtido uma correlação de 0,97 para a temperatura, 0,99 para o saldo de radiação, 0,97 para o fotoperíodo, 0,91 para a umidade e 0,98 para a insolação. O erro médio absoluto neste teste resultou em 0,57 °C para a temperatura, 0,39 MJ/m² para o saldo de radiação, 0,57 horas para o fotoperíodo, 3,33% para a umidade e 5,05 K $gcal/cm^2$ para a insolação. E o desvio padrão para os dados de temperatura variou de 1,9 a 2,19, para o saldo de radiação variou de 3,06 a 3,32, de 0,6 a 1,9 para o fotoperíodo, de 9,09 a 12,58 para a umidade e para a insolação variou de 26,97 a 29,49. Em todos os casos os valores estimados raramente excederam esses limites. Os resultados mostram que a proposta apresentou bons resultados para corrigir séries de dados com pequeno número de erros, mas também foi eficiente quando a série de dados tinha uma grande parte de falhas.

É importante dizer que, diferente dos cinco sensores apresentados, outros dois sensores não tiveram resultados satisfatórios. Os sensores que representavam as variáveis climáticas de pressão atmosférica e velocidade do vento, obtiveram coeficientes de correlação menores que 0,8. Para esses casos pode ser dito que os sensores disponíveis para o preenchimento de falhas não eram suficientes para se fazer uma boa estimativa.

Neste trabalho os testes foram realizados corrigindo as falhas das variáveis climáticas apresentadas. Uma vantagem é que é possível aplicar essa metodologia para qualquer sensor da área de micrometeorologia. O importante é que a série de dados a ser corrigida tenha medidas de outros sensores para que a RNA seja capaz de fazer as estimativas com base nessas outras medidas.

Uma outra vantagem desta metodologia é que ela pode ter uma boa precisão nos resultados mesmo que as falhas estejam em sequência, ou seja, as falhas nas séries de dados ocorram durante horas, dias ou até mesmo meses. Essa vantagem é devido ao fato de ela aproveitar medidas de outros sensores, e não do mesmo sensor em um momento próximo. Entretanto, é necessário haver dados válidos em um outro momento na série de dados. A quantidade mínima de dados válidos presentes na série de dados é difícil de ser determinada. Porém, quanto mais exemplos válidos, coerentemente, maiores são as chances da metodologia gerar um bom resultado.

A proposta deste trabalho foi alcançada, no qual analisando os resultados obtidos entende-se que é possível preencher valores ausentes de dados reais micrometeorológicos. Como perspectivas futuras, está em desenvolvimento um sistema computacional maior que utilize a abordagem proposta neste trabalho, como um dos módulos do sistema, no qual iria se ter uma automatização e, conseqüentemente, uma facilidade maior de preencher as falhas em bases de dados micrometeorológicos. A ideia é que o usuário pudesse inserir os dados e indicar qual sensor deseja corrigir. O sistema faria todos os passos seguintes automaticamente. Nessa mesma linha espera-se finalizar todos os módulos citados do EnvRP, tornando possível a centralização dos dados e facilitando a análise dos mesmos.

É desejado também um aprofundamento no estudo do comportamento das variáveis meteorológicas para se criar uma nova abordagem em que o preenchimento de falhas possa ser mais eficaz, levando em consideração também outros fatores que não sejam apenas as medições de outros sensores no mesmo momento. Conhecendo melhor como as variáveis climáticas se relacionam, será possível desenvolver outras formas que possam identificar essas relações e, assim, calcular com maior precisão, valores para preencher as falhas.

Seria interessante, também, a utilização de métodos para detectar *outliers* ou valores absurdos para uma determinada variável climática na série de dados. Dessa forma não somente os dados ausentes seriam preenchidos, mas todos os dados inválidos, inclusive aqueles não percebidos pelo usuário.

Referências Bibliográficas

AGARWAL, M. A systematic classification of neural-network-based control. *IEEE Control Systems Magazine*, v. 17, p. 75–93, 1997. Citado na página 14.

AHRENS, C. D. *Essentials of Meteorology*. [S.l.]: Brooks Cole, 2004. 496 p. Citado na página 10.

ALAVI, N.; WARLAND, J. S.; BERG, A. A. Filling gaps in evapotranspiration measurements for water budget studies: Evaluation of a kalman filtering approach. *Agricultural and Forest Meteorology*, v. 141, n. 1, p. 57–66, dez. 2006. Disponível em: <<http://www.sciencedirect.com/science/article/B6V8W-4M7K9JB-2/2-/06ad4bba8ec003189de1bb82a9f9af20>>. Citado na página 13.

ASSUMPÇÃO, M. E.; CHRISTOFORO, A. L.; PANZERA, T. H.; BRITO, J. N.; LAMIM-FILHO, P. C. M. Emprego de algoritmos genéticos na determinação das dimensões Ótimas de uma torre treliçada. In: *XI Congresso Nacional de Engenharia Mecânica, Metalúrgica e Industrial*. Porto Alegre, RS: ABEMEC, 2011. p. 1–7. Citado na página 17.

ASSUMPÇÃO, M. H. M. T.; FREITAS, K. H. G.; SOUZA, F. S.; FATIBELLO-FILHO, O. Construção e adaptação de materiais alternativos em titulação ácido-base. *Eclética Química*, Universidade Estadual Paulista Júlio de Mesquita Filho, Brasil, v. 35, n. 4, p. 133–138, 2010. ISSN 0100-4670. Citado na página 35.

ATKINSON, B. *Meso-scale atmospheric circulations*. Academic Press, 1981. ISBN 9780120659609. Disponível em: <<http://books.google.com.au/books?id=gStRAAAAMAAJ>>. Citado na página 10.

BEALE, M. H.; HAGAN, M. T.; DEMUTH, H. B. *Neural Network Toolbox Getting Started Guide*. Natick, MA, USA: MathWorks, 2011. 408 p. Citado na página 34.

BELGRANO, A.; MALMGREN, B. A.; LINDAHL, O. Application of artificial neural networks (ann) to primary production time-series data. *Journal of Plankton Research*.

arch, v. 23, n. 6, p. 651–658, 2001. Disponível em: <<http://plankt.oxfordjournals.org/content/23/6/651.abstract>>. Citado na página 14.

BUSK, D. *Solução Giltt Bidimensional em Geometria Cartesiana: Simulação da Dispersão de Poluentes na Atmosfera*. 236 p. Dissertação (Tese) — UFRS, 2008. Citado na página 10.

CAPISTRANO, V. B. *Análise de Séries Temporais de Variáveis Microclimatológicas Medidas em Sinop-MT Utilizando a Teoria da Complexidade*. 62 p. Dissertação (Dissertação) — UFMT, 2007. Citado na página 11.

CARRILHO, S. B. P. *Avaliação de Modelos para Estimativa da Radiação de Onda Longa Atmosférica no Cerrado Mato-Grossense*. 124 p. Dissertação (Dissertação) — UFMT, 2011. Citado na página 24.

CHEN, C. S.; SU, S. L. Resilient back-propagation neural network for approximation 2-d gdop. *Proceedings of the International MultiConference of Enginneers and Computer Scientists*, IMECS, v. 2, p. 900–904, 2010. Citado na página 31.

CORREIA, M. Algoritmos genéticos. *Revista Dosalgarves*, Escola Superior de Gestão, Hotelaria e Turismo da Universidade do Algarve, v. 12, p. 36–43, 2003. Citado na página 17.

DESWAL, S.; PAL, M. Artificial neural network based modeling of evaporation losses in reservoirs. *International Journal of Mathematical Physical and Engineering Sciences*, v. 39, n. 2, p. 177–181, 2008. Citado na página 16.

DIAS, C. A. A. *Procedimentos de Medição e Aquisição de Dados de uma Torre Micro-meteorológica em Sinop-MT*. 89 p. Dissertação (Dissertação) — UFMT, 2007. Citado nas páginas xi e 11.

FALGE, E.; BALDOCCHI, D.; OLSON, R.; ANTHONI, P.; AUBINET, M.; BERNHOFFER, C.; BURBA, G.; CEULEMANS, R.; CLEMENT, R.; DOLMAN, H.; AL. et. Gap filling strategies for defensible annual sums of net ecosystem exchange. *Agricultural and Forest Meteorology*, v. 107, n. 1, p. 43–69, 2001. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0168192300002252>>. Citado na página 13.

FAUSETT, L. (Ed.). *Fundamentals of neural networks: architectures, algorithms, and applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1994. ISBN 0-13-334186-0. Citado nas páginas 15 e 16.

FOKEN, T. *Micrometeorology*. [S.l.]: Springer Heidelberg, 2008. 308 p. Citado na página 10.

FONSECA, M. *Análise Harmônica do Regime de Precipitação em Duas Localidades da Baixada Cuiabana*. 66 p. Dissertação (Dissertação) — UFMT, 2008. Citado na página 23.

FORREST, S. Genetic algorithms: principles of natural selection applied to computation. *Science*, v. 261, n. 5123, p. 872–878, ago. 1993. Disponível em: <<http://dx.doi.org/10.1126/science.8346439>>. Citado na página 21.

GALLON, M. M. P. *Um Estudo Sobre a Dinâmica de Sistemas Complexos a partir de Séries Temporais de Dados Microclimatológicos para uma Floresta de Transição no Noroeste de Mato Grosso*. 130 p. Dissertação (Dissertação) — UFMT, 2005. Citado na página 12.

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. New York: Addison-wesley, 1989. Citado na página 17.

GOVE, J. H.; HOLLINGER, D. Y. Application of a dual unscented kalman filter for simultaneous state and parameter estimation in problems of surface-atmosphere exchange. *Journal of Geophysical Research*, v. 111, p. 1–21, 2006. Disponível em: <<http://www.agu.org/pubs/crossref/2006/2005JD006021.shtml>>. Citado na página 13.

HAYKIN, S. S. *Redes Neurais - Principios E Pratica*. [S.l.]: Bookman, 2000. Citado nas páginas xi, 14, e 15.

HERVÁS, C.; ALGAR, J. A.; SILVA, M. Correction of temperature variations in kinetic-based determinations by use of pruning computational neural networks in conjunction with genetic algorithms. *Journal of Chemical Information and Computer Sciences*, v. 40, n. 3, p. 724–731, 2000. Citado na página 21.

HEWITT, C. N. *Handbook of atmospheric science - principles and applications*. [S.l.]: Wiley-Blackwell, 2003. 600 p. Citado na página 10.

HOLTGREWE, M.; SANDERS, P.; SCHULZ, C. Engineering a scalable high quality graph partitioner. In: *24th IEEE International Parallal and Distributed Processing Symposium*. [S.l.: s.n.], 2010. p. 1–12. Citado na página 16.

HU, Z.; YU, G.; ZHOU, Y.; SUN, X.; LI, Y.; SHI, P.; WANG, Y.; SONG, X.; ZHENG, Z.; ZHANG, L. Partitioning of evapotranspiration and its controls in four grassland ecosystems: Application of a two-source model. *Agricultural and Forest Meteorology*, v. 149, n. 9, p. 1410–1420, set. 2009. ISSN 01681923. Disponível em: <<http://dx.doi.org/10.1016/j.agrformet.2009.03.014>>. Citado na página 13.

HUI, D. Gap-filling missing data in eddy covariance measurements using multiple imputation (mi) for annual estimations. *Agricultural and Forest Meteorology*, v. 121, n. 1–2, p. 93–111, 2004. Disponível em: <[http://linkinghub.elsevier.com/retrieve/pii-S0168192303001588](http://linkinghub.elsevier.com/retrieve/pii/S0168192303001588)>. Citado na página 12.

JAIN, S. K.; NAYAK, P. C.; SUDHEER, K. P. Models for estimating evapotranspiration using artificial neural networks, and their physical interpretation. *Hydrological Processes*, v. 2234, n. February, p. 2225–2234, 2008. Disponível em: <<http://dx.doi.org/10.1002/hyp.6819>>. Citado na página 16.

JESUS, R. G.; OLIVEIRA, M. O. F. Implantação de sistemas erp: Tecnologia e pessoas na implantação do sap r/3. *Revista de Gestão da Tecnologia e Sistemas de Informação*, TECSI FEA USP, UFES, p. 315 – 330, 2007. Citado na página 6.

KATO, T.; TANG, Y.; GU, S.; HIROTA, M.; DU, M.; LI, Y.; ZHAO, X. Temperature and biomass influences on interannual changes in CO₂ exchange in an alpine meadow on the Qinghai-Tibetan plateau. *Global Change Biology*, Blackwell Publishing, v. 12, n. 7, p. 1285–1298, jul. 2006. ISSN 1354-1013. Disponível em: <<http://dx.doi.org/10.1111/j.1365-2486.2006.01153.x>>. Citado na página 13.

KOVÁCS, Z. L. *Redes Neurais Artificiais: Fundamentos e Aplicações*. [S.l.]: Collegium Cognitio, 1996. Citado na página 15.

KOZA, J. R. Survey of genetic algorithms and genetic programming. In: *In Proceedings of the Wescon 95 - Conference Record: Microelectronics, Communications Technology, Producing Quality Products, Mobile and Portable Power, Emerging Technologies*. [S.l.]: IEEE Press, 1995. p. 589–594. Citado na página 17.

LACERDA, E. G. M.; CARVALHO, A. C. P. L. *Introdução aos algoritmos genéticos*. Porto Alegre, RS: Ed. Universidade/UFRGS, 1999. 99-150 p. Citado na página 17.

LAUMANNS, M.; ZITZLER, E.; THIELE, L. A unified model for multi-objective evolutionary algorithms with elitism. In: *In Congress on Evolutionary Computation (CEC 2000)*. [S.l.]: IEEE Press, 2000. p. 46–53. Citado na página 19.

LIN, T. H.; CHIU, S. H.; TSAI, K. C. Supervised Feature Ranking Using a Genetic Algorithm Optimized Artificial Neural Network. *J. Chem. Inf. Model.*, Institute of Molecular Medicine & Department of Life Science, National Tsing Hua University, Hsinchu, Taiwan 30013, R.O.C., v. 46, n. 4, p. 1604–1614, jul. 2006. Disponível em: <<http://dx.doi.org/10.1021/ci0600354>>. Citado na página 21.

LOURAKIS, M. I. A. A brief description of the levenberg-marquardt algorithm implemented by levmar. *Matrix*, v. 3, p. 1–6, 2005. Disponível em: <<http://www.ics.forth.gr/~lourakis/>>. Citado na página 31.

MARIANO, R. T. G. *Análise Espectral de Séries Temporais de Variáveis Micro-Climatológicas em uma Área de Ecótono entre os Biomas Amazônia e Cerrado do Norte de Mato Grosso*. 100 p. Dissertação (Dissertação) — UFMT, 2005. Citado na página 12.

MATHWORKS. *Página de referência das funções da Neural Network Toolbox*. 2011. Disponível em: <<http://www.mathworks.com/help/toolbox/nnet/ref/f7-23438.html>>. Citado nas páginas 30 e 31.

MATLAB. *The Language Of Technical Computing*. [S.l.]: Disponível na URL:<http://www.mathworks.com/products/matlab>. Data de último acesso: 22/05/2011., 2011. Citado na página 34.

MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. New York: Springer-Verlag, 1994. Citado na página 17.

MISHUROV, M.; KIELY, G. Gap-filling techniques for the annual sums of nitrous oxide fluxes. *Agricultural and Forest Meteorology*, v. 151, p. 1763–1767, 2011. Citado na página 13.

MOAYED, F. A.; SHELL, R. L. *Application of artificial neural network models in occupational safety and health utilizing ordinal variables*. Oxford University Press, 2010. Disponível em: <<http://dx.doi.org/10.1093/annhyg/meq079>>. Citado na página 14.

MOFFAT, A. M.; PAPALE, D.; REICHSTEIN, M.; HOLLINGER, D. Y.; RICHARDSON, A. D.; BARR, A. G.; BECKSTEIN, C.; BRASWELL, B. H.; REICHSTEIN, M.; DESAI, A. R.; AL. et. Comprehensive comparison of gap-filling techniques for eddy covariance net carbon fluxes. *Agricultural and Forest Meteorology*, v. 147, n. 3–4, p. 209–232, 2007. Disponível em: <<http://www.sciencedirect.com/science/article/B6V8W-4PRRH4S-1/2/8f35f7f68e06ef1f6eb7ad5e3ce6c1df>>. Citado na página 13.

NETO, A. A. C. *Estudo Bioclimático no Campus da Universidade Federal de Mato Grosso*. 159 p. Dissertação (Dissertação) — UFMT, 2007. Citado na página 23.

NEVES, G. A. R. *Desenvolvimento de Estação Micrometeorológica com Armazenamento de Dados*. 61 p. Dissertação (Dissertação) — UFMT, 2011. Citado na página 10.

NOVAIS, J. W. Z. *Dinâmica Geotérmica do Solo de uma Floresta de Cambarás no Pantanal Mato-Grossense*. 54 p. Dissertação (Dissertação) — UFMT, 2011. Citado na página 23.

NUNES, L. N.; KLUCK, M. M.; FACHEL, J. M. G. Uso da imputação múltipla de dados faltantes: uma simulação utilizando dados epidemiológicos. *Cad. Saúde Pública*, v. 25, n. 2, p. 268–278, 2009. Disponível em: <<http://www.scielo.br/pdf/csp/v25n2-/05.pdf>>. Citado na página 13.

OLIVEIRA, A. G. *Arquitetura de dados para gerenciamento de informações micro-meteorológicas*. 86 p. Dissertação (Dissertação) — UFMT, 2011. Citado na página 7.

OLIVEIRA, A. G.; LIBERATTI, G.; RODRIGUES, R.; MARTINS, C. A.; FIGUEIREDO, J. M.; BONFANTE, A. G. Uma arquitetura de componentes para processamento de imagens ambientais. *Anais do III Simpósio de Geotecnologias no Pantanal*, Embrapa Informática Agropecuária, São José dos Campos, SP, p. 885 – 893, 2010. Citado na página 6.

OLIVEIRA, A. G.; VENTURA, T. M.; CURADO, L. F. A.; FIGUEIREDO, J. M.; MARTINS, C. A. Armazenamento e gerenciamento de dados ambientais. *Coletânea Física Ambiental I*, Baraúna, São Paulo, v. 1, p. 39–47, 2011. Citado na página 8.

Ooba, M.; Hirano, T.; Mogami, J.-I.; Hirata, R.; Fujinuma, Y. Comparisons of gap-filling methods for carbon flux dataset: A combination of a genetic algorithm and an artificial neural network. *Ecological Modelling*, v. 198, n. 3-4, p. 473 – 486, 2006. ISSN 0304-3800. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0304380006002766>>. Citado na página 13.

ORTON, N. J. H.; IPSITZ, S. R. L. Multiple imputation in practice : Comparison of software packages for regression models with missing variables. *American Statistician*, v. 55, n. 3, p. 244–254, 2001. Disponível em: <<http://www.jstor.org/stable/2685809>>. Citado na página 12.

PALÚ, A. E. *Determinação do Tempo de Defasagem Mais Adequado para Análise de Séries Temporais de Variáveis Microclimatológicas Medidas numa Floresta de Transição no Norte de Mato Grosso*. 48 p. Dissertação (Dissertação) — UFMT, 2008. Citado na página 11.

PANELLA, M.; RIZZI, A.; MARTINELLI, G. Refining accuracy of environmental data prediction by mog neural networks. *Neurocomputing*, v. 55, n. 3-4, p. 521–549, 2003. Citado na página 16.

PFALTZ, J. L. What constitutes a scientific database? In: *19th International Conference on Scientific and Statistical Database Management, SSDBM*. Banff, Canada: IEEE Computer Society, 2007. p. 1–10. Citado na página 3.

PIRES A. M., A. C. (Ed.). *A utilização da função de influência para melhorar o desempenho do método de reamostragem bootstrap*. Actas do VI Congresso Anual da SPE. [S.l.: s.n.], 1999. 91–99 p. Citado na página 35.

RODRIGUES, A.; PITA, G.; MATEUS, J. a. Turbulent fluxes of carbon dioxide and water vapour over an eucalyptus forest in portugal. *Silva Lusitana*, v. 13, n. 2, p. 169 – 180, 2005. Citado na página 13.

RUSSELL, S.; NORVIG, P. *INTELIGENCIA ARTIFICIAL*. CAMPUS, 2004. ISBN 9788535211771. Disponível em: <<http://books.google.com/books?id=wBMvAAAACAAJ>>. Citado na página 14.

SANTOS, V. F.; ARÊDES, A. F.; PEREIRA, M. W. G.; SOUZA, U. R.; ARÊDES, A. Stock cost and risk-prize: an economic analysis of the best period for arabic coffee commercialization. *Custos e Agronegócio*, v. 3, n. 2, p. 22–38, 2007. ISSN 1808-2882. Citado na página 35.

SCHAFER, J. L. Multiple imputation: a primer. *Stat Methods Med Res*, v. 8, p. 3–15, February 1999. Citado na página 12.

SCHAFFER, J. D.; WHITLEY, D.; ESHELMAN, L. J. Combinations of genetic algorithms and neural networks: a survey of the state of the art. In: _____. *Combinations of Genetic Algorithms and Neural Networks 1992 COGANN92 International Workshop on*. IEEE Comput. Soc. Press, 1992. p. 1–37. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=273950>>. Citado na página 21.

SEIXAS, G. B. *Determinação da Transpiração em Plantas de Nim Indiano (Azadirachta indica A. Juss) Utilizando Métodos de Estimativa de Fluxo de Seiva*. 71 p. Dissertação (Dissertação) — UFMT, 2009. Citado na página 23.

SELLITTO, M. A. Inteligência artificial: uma aplicação em uma indústria de processo contínuo. *Gestão e Produção*, scielo, v. 9, p. 363 – 376, 12 2002. ISSN 0104-530X. Citado na página 5.

SHAO, C.; CHEN, J.; LI, L.; TENNEY, G.; XU, W.; XU, J. Role of net radiation on energy balance closure in heterogeneous grasslands. *Biogeosciences Discussions*, v. 8, n. 2, p. 2001–2033, 2011. Disponível em: <<http://www.biogeosciences-discuss.net/8-/2001/2011/>>. Citado na página 13.

SHEIKH, R. H.; RAGHUWANSHI, M.; JAISWAL, A. N. Genetic algorithm based clustering: A survey. *Emerging Trends in Engineering and Technology, International Conference on*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 314–319, 2008. Disponível em: <<http://www.computer.org/portal/web/csdl/doi/10.1109/ICETET.2008.48>>. Citado na página 18.

SUDHEER, K. P.; GOSAIN, A. K.; RAMASASTRI, K. S. Estimating actual evapotranspiration from limited climatic data using neural computing technique. *Journal of Irrigation and Drainage Engineering-asce*, v. 129, p. 214–218, 2003. Citado na página 14.

TANG, P.; LEE, G. An adaptive fitness function for evolutionary algorithms using heuristics and prediction. *Proc. of the World Automation Congress*, ISSCI, Budapest, p. 1–6, 2006. Citado na página 18.

TARCA, L. A.; GRANDJEAN, B. P. A.; LARACHI, F. Integrated genetic algorithm-artificial neural network strategy for modeling important multiphase-flow characteristics. *Industrial & Engineering Chemistry Research*, v. 41, p. 2543–2551, 2002. Citado na página 21.

TSUKAHARA, R. Y.; JENSEN, T.; CARAMORI, P. H. Utilização de redes neurais artificiais para preenchimento de falhas em séries horárias de dados meteorológicos. *Congresso Brasileiro de Meteorologia*, p. 1–5, 2010. Citado na página 16.

WILLMOTT, C.; MATSUURA, K. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, v. 30, p. 79–82, 2005. Disponível em: <<http://www.int-res.com/abstracts/cr/v30/n1/p79-82/>>. Citado na página 35.

WILLMOTT, C. J. Some Comments on the Evaluation of Model Performance. *Bulletin of the American Meteorological Society*, v. 63, p. 1309–1369, 1982. Disponível em: <[http://dx.doi.org/10.1175/1520-0477\(1982\)063](http://dx.doi.org/10.1175/1520-0477(1982)063)>. Citado na página 35.

WILLMOTT, C. J.; MATSUURA, K.; ROBESON, S. M. Ambiguities inherent in sums-of-squares-based error statistics. *Atmospheric Environment*, Elsevier Ltd, v. 43, n. 3, p. 749–752, 2009. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1352231008009564>>. Citado na página 35.

WILSON, K.; BALDOCCHI, D. Comparing independent estimates of carbon dioxide exchange over 5 years at a deciduous forest in the southeastern united states. *Journal of Geophysical Research-Atmospheres*, Wilson, KBNOAA, Atmospher Turbulence &

Diffus Div, 151 Hilgard Hall, Oak Ridge, TN 37831 USANOAA, At, v. 106, n. D24, p. 34167–34178, 2001. ISSN 0747-7309. Citado na página 13.

ZANETTI, S. S.; SOUSA, E. F.; CARVALHO, D. F. de; BERNARDO, S. Estimação da evapotranspiração de referência no estado do rio de janeiro usando redes neurais artificiais. p. 174–180, 2008. Citado na página 15.

ZHANG, M.; GAO, X.; LOU, W. A new crossover operator in genetic programming for object classification. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 37, p. 1332–1343, 2007. Citado na página 19.

ZHOU, Y.; CHENG, C.; ZUO, J. Dynamic mutation and recombination using self-selecting crossover method for genetic algorithms. In: *Proceedings of the 2008 Fourth International Conference on Natural Computation - Volume 01*. Washington, DC, USA: IEEE Computer Society, 2008. (ICNC '08), p. 586–592. ISBN 978-0-7695-3304-9. Disponível em: <<http://dx.doi.org/10.1109/ICNC.2008.844>>. Citado na página 19.

Apêndice A

Código do Algoritmo Genético

Principal.m

```
1  NUMERO_MAX_GERACAO = 100;
2  ERRO_MAXIMO_VALIDO = 0.04;
3  TAMANHO_POPULACAO = 40;
4  TAMANHO_INDIVIDUO = 18;
5  PORCENTAGEM_MUTACAO = 0.1;
6  mErroIndividuo = 100;
7
8  menorErro = 999999;
9  melhorSolucao = [];
10
11 % Gerando uma populacao aleatoria
12 populacao = [];
13 for i = 1 : TAMANHO_POPULACAO
14     individuo = '';
15     for j = 1 : TAMANHO_INDIVIDUO
16         if ( rand(1) > 0.5 )
17             bit = '0';
18         else
19             bit = '1';
20         end
21         individuo = [individuo , bit];
22     end
23     populacao(i,:) = individuo;
24 end
25
```

```

26 geracoesGeradas = 0;
27 erroGeracao = [];
28 while( geracoesGeradas < NUMERO_MAX_GERACAO )
29
30     geracoesGeradas = geracoesGeradas + 1;
31     ok = 0;
32     somatorioDeErro = 0;
33     avaliacoes = [];
34
35     for i = 1 : TAMANHO_POPULACAO
36
37         % Decodificando o individuo
38         [func1 func2 algT lr mc s1 s2 s3 s4 s5 s6
39          ] = decodificar(populacao(i,:));
40
41         % Calculando o erro deste individuo
42         [erroIndividuo , IW, LW, b] = rnIDA(func1 ,
43         func2 ,algT ,lr ,mc,s1 ,s2 ,s3 ,s4 ,s5 ,s6 );
44
45         if (erroIndividuo < mErroIndividuo)
46             % Guardar a configuracao caso
47             seja
48             % o melhor individuo ate o
49             momento
50             mErroIndividuo = erroIndividuo;
51             mIW = IW;
52             mLW = LW;
53             mb = b;
54             nfunc1 = func1;
55             nfunc2 = func2;
56             malgT = algT;
57             mlr = lr;
58             mmc = mc;
59             ms1 = s1;
60             ms2 = s2;
61             ms3 = s3;
62             ms4 = s4;

```

```

59         ms5 = s5;
60         ms6 = s6;
61         end
62
63         avaliacoes(i) = erroIndividuo;
64
65         somatorioDeErro = somatorioDeErro +
66             erroIndividuo;
67     end
68     erroGeracao(geracoesGeradas) = (somatorioDeErro /
69         TAMANHO_POPULACAO)
70
71     % Verificando se atingiu o erro minimo
72     if( erroGeracao(geracoesGeradas) <=
73         ERRO_MAXIMO_VALIDO )
74         [erros , indices] = sort(avaliacoes);
75         menorErro = erros(1);
76         melhorSolucao = populacao(indices(1) , :);
77         break;
78     end
79
80     % Se nao achou o erro minimo, criar uma nova
81     geracao
82
83     % Ordena o vetor de acordo com as melhores
84     avaliacoes
85     [erros , indices] = sort(avaliacoes);
86
87     % elitismo
88     pais = [];
89     novaGeracao = [];
90     countNovaGeracao = 1;
91     menorErro = erros(1);
92     melhorSolucao = populacao(indices(1) , :);

```

```

91     for i = 1 : 2 : TAMANHO_POPULACAO
92
93         % recuperando os pais
94         pai = populacao(indices(i), :);
95         mae = populacao(indices(i+1), :);
96
97         % deixa os melhores pais na proxima geracao
98         novaGeracao(countNovaGeracao, :) = pai;
99         if( countNovaGeracao == TAMANHO_POPULACAO
100             )
101             break;
102         else
103             countNovaGeracao =
104                 countNovaGeracao + 1;
105         end
106
107         novaGeracao(countNovaGeracao, :) = mae;
108         if( countNovaGeracao == TAMANHO_POPULACAO
109             )
110             break;
111         else
112             countNovaGeracao =
113                 countNovaGeracao + 1;
114         end
115
116         % Realizando cruzamentos entre os pais
117         pontoDeCorte = round( rem( ( rand(1,1) *
118             100 ), TAMANHO_INDIVIDUO-1 ) ) + 1;
119         filho1 = [ pai(1 : pontoDeCorte) mae(
120             pontoDeCorte+1 : length(mae)) ];
121         filho2 = [ mae(1 : pontoDeCorte) pai(
122             pontoDeCorte+1 : length(pai)) ];
123
124         % Adicionando os filhos
125         novaGeracao(countNovaGeracao, :) = filho1
126             ;

```

```

119         if( countNovaGeracao == TAMANHO_POPULACAO
120             )
121             break;
122         else
123             countNovaGeracao =
124                 countNovaGeracao + 1;
125         end
126
127         novaGeracao(countNovaGeracao, :) = filho2
128             ;
129         if( countNovaGeracao == TAMANHO_POPULACAO
130             )
131             break;
132         else
133             countNovaGeracao =
134                 countNovaGeracao + 1;
135         end
136     end
137
138     populacao = novaGeracao;
139
140     % realizando a mutacao
141     for i = 1 : TAMANHO_POPULACAO
142
143         % verificar se esse individuo deve sofrer
144         % mutacao
145         if( rand(1,1) < PORCENTAGEM_MUTACAO )
146
147             bit = round( rem( ( rand(1,1) *
148                 100 ), TAMANHO_INDIVIDUO-1 ) )
149                 + 1;
150
151             % invertendo o bit
152             if( populacao(i,bit) == '0' )
153                 populacao(i,bit) = '1';
154             else
155                 populacao(i,bit) = '0';
156             end
157         end
158     end

```

```

148                                     end
149                                 end
150                            end
151                        end
152
153    % Fim do AG, mostrar o melhor resultado
154
155    % Mostrando os individuos mais evolucionados
156    for i = 1 : TAMANHO_POPULACAO
157        [func1 func2 treinamento lr mc s1 s2 s3 s4 s5 s6]
            = decodificar(populacao(1,:));
158    end
159
160    % Numeros de geracoes necessarias
161    geracoesGeradas;
162
163    % A melhor solucao
164    [func1 func2 algT lr mc s1 s2 s3 s4 s5 s6] = decodificar(
        melhorSolucao)
165
166    % O valor do erro do melhor individuo
167    menorErro

```

Codificar.m

```

1  function [ binario ] = codificar(ativacao1 ,ativacao2 ,
    treinamento ,lr ,mc,s1 ,s2 ,s3 ,s4 , s5 , s6)
2
3      binAtivacao1=' ';
4      binAtivacao2=' ';
5      binTreinamento=' ';
6      binLr=' ';
7      binMc=' ';
8      bS1=' ';
9      bS2=' ';
10     bS3=' ';
11     bS4=' ';
12     bS5=' ';

```

```

13      bS6='';
14
15      %formando a cadeia de binário correspondente ao
16      indivíduo de acordo com os parâmetros
17      determinados
18
19      if (strcmp(ativacao1, 'purelin'))
20          binAtivacao1=['0','0'];
21      else
22          if (strcmp(ativacao1, 'logsig'))
23              binAtivacao1=['0','1'];
24          end
25      end
26
27      if (strcmp(ativacao1, 'tansig'))
28          binAtivacao1=['1','0'];
29      else
30          if (strcmp(ativacao1, 'satlin'))
31              binAtivacao1=['1','1'];
32          end
33      end
34
35      if (strcmp(ativacao2, 'purelin'))
36          binAtivacao2=['0','0'];
37      else
38          if (strcmp(ativacao2, 'logsig'))
39              binAtivacao2=['0','1'];
40          end
41      end
42
43      if (strcmp(ativacao2, 'tansig'))
44          binAtivacao2=['1','0'];
45      else
46          if (strcmp(ativacao2, 'satlin'))
47              binAtivacao2=['1','1'];
48          end
49      end

```

```

48     if (strcmp(treinamento , 'traingd'))
49         binTreinamento=['0','0'];
50     else
51         if (strcmp(treinamento , 'traingdm'))
52             binTreinamento=['0','1'];
53         end
54     end
55
56     if (strcmp(treinamento , 'traingdx'))
57         binTreinamento=['1','0'];
58     else
59         if (strcmp(treinamento , 'trainlm'))
60             binTreinamento=['1','1'];
61         end
62     end
63
64     binLr = dec2bin( ((lr - 0.1) * 10), 3);
65     binMc = dec2bin( ((mc - 0.1) * 10), 3);
66
67     if (s1 == 1)
68         bS1 = '1';
69     else
70         bS1 = '0';
71     end
72
73     if (s2 == 1)
74         bS2 = '1';
75     else
76         bS2 = '0';
77     end
78
79     if (s3 == 1)
80         bS3 = '1';
81     else
82         bS3 = '0';
83     end
84

```

```

85         if (s4 == 1)
86             bS4 = '1';
87         else
88             bS4 = '0';
89         end
90
91         if (s5 == 1)
92             bS5 = '1';
93         else
94             bS5 = '0';
95         end
96
97         if (s6 == 1)
98             bS6 = '1';
99         else
100             bS6 = '0';
101         end
102
103         %Composicao da cadeia de binário
104         binario =[binAtivacao1 ,binAtivacao2 ,
                   binTreinamento ,binLr ,binMc ,bS1 ,bS2 ,bS3 ,bS4 ,bS5
                   ,bS6];

```

Decodificar.m

```

1  function [ativacao1 ,ativacao2 ,treinamento ,lr ,mc,s1 ,s2 ,s3 ,
2      s4 , s5 , s6] = decodificar(binario)
3
4      % Decodificando o individuo
5
6      % 1 – Extraindo o binario de cada parte dos
7      %      parametros
8      bAtivacao1 = binario(1:2);
9      bAtivacao2 = binario(3:4);
10     bTreinamento = binario(5:6);
11     bLr = binario(7:9);
12     bMc = binario(10:12);
13     bS1 = binario(13:13);

```

```

12     bS2 = binario(14:14);
13     bS3 = binario(15:15);
14     bS4 = binario(16:16);
15     bS5 = binario(17:17);
16     bS6 = binario(18:18);
17
18     if (bAtivacao1 == [48 48])
19         ativacao1='purelin';
20     else
21         if (bAtivacao1 == [48 49])
22             ativacao1='logsig';
23         else
24             if (bAtivacao1 == [49 48])
25                 ativacao1='tansig';
26             else
27                 ativacao1='satlin';
28             end
29         end
30     end
31
32     if (bAtivacao2 == [48 48])
33         ativacao2='purelin';
34     else
35         if (bAtivacao2 == [48 49])
36             ativacao2='logsig';
37         else
38             if (bAtivacao2 == [49 48])
39                 ativacao2='tansig';
40             else
41                 ativacao2='satlin';
42             end
43         end
44     end
45
46     if (bTreinamento == [48 48])
47         treinamento='traingd';
48     else

```

```
49         if (bTreinamento == [48 49])
50             treinamento='traingdx';
51         else
52             if (bTreinamento == [49 48])
53                 treinamento='trainlm';
54             else
55                 treinamento='trainrp';
56             end
57         end
58     end
59
60     lr = bin2dec(char(bLr)) / 10 + 0.1;
61     mc = bin2dec(char(bMc)) / 10 + 0.1;
62
63     s1 = bS1 == '1';
64     s2 = bS2 == '1';
65     s3 = bS3 == '1';
66     s4 = bS4 == '1';
67     s5 = bS5 == '1';
68     s6 = bS6 == '1';
```

Apêndice B

Código da Rede Neural Artificial

Rna.m

```
1 function [error , IW, LW, b] = rna(func1 , func2 , algT , lr ,  
    mc, s1 , s2 , s3 , s4 , s5 , s6)  
2  
3     warning off all;  
4     load 'simulacao.mat';  
5     entradas = [];  
6     entradasTeste = [];  
7     neuronio=0;  
8  
9     % Verificando quais sensores estao ativos  
10    if (s1)  
11        neuronio=neuronio+1;  
12        entradas = entradaRN(:,1);  
13        entradasTeste = entradaTeste(:,1);  
14    end  
15  
16    if (s2)  
17        neuronio=neuronio+1;  
18        entradas = [entradas entradaRN(:,2)];  
19        entradasTeste = [entradasTeste  
20                           entradaTeste(:,2)];  
21  
22    if (s3)  
23        neuronio=neuronio+1;
```

```

24         entradas = [entradas entradaRN(:,3)];
25         entradasTeste = [entradasTeste
                           entradaTeste(:,3)];
26     end
27
28     if (s4)
29         neuronio=neuronio+1;
30         entradas = [entradas entradaRN(:,4)];
31         entradasTeste = [entradasTeste
                           entradaTeste(:,4)];
32     end
33
34     if (s5)
35         neuronio=neuronio+1;
36         entradas = [entradas entradaRN(:,5)];
37         entradasTeste = [entradasTeste
                           entradaTeste(:,5)];
38     end
39
40     if (s6)
41         neuronio=neuronio+1;
42         entradas = [entradas entradaRN(:,6)];
43         entradasTeste = [entradasTeste
                           entradaTeste(:,6)];
44     end
45
46     targets = saidaRN;
47     targetsTeste = saidaTeste;
48
49     neuronio =(neuronio*2)+1;
50
51     %normalização
52     [pn,minp,maxp,tn,mint,maxt] = premnmx(entradas',
        targets');
53     [pnT,minp,maxp,tnT,mint,maxt] = premnmx(
        entradasTeste',targetsTeste');

```

```

54     net = newff( minmax(pn), [neuronio 1], {func1
        func2 }, algT );
55
56     % configurando a RNA
57     net.trainParam.showWindow = 0;
58     net.trainParam.epochs = 400;
59     net.trainParam.goal = 0.0001;
60     net.performFcn = 'mse';
61     IW = net.IW;
62     LW = net.LW;
63     b = net.b;
64     net.trainParam.lr = lr;
65     net.trainParam.mc = mc;
66     net.trainParam.lr_inc = 1.05;
67     net.trainParam.lr_dec = 0.7;
68     net.trainParam.max_perf_inc = 1.04;
69
70     % treinando a RNA
71     net = train( net,pn, tn);
72
73     % testando a RNA
74     saidaTeste = sim(net,pnT);
75
76     % obtendo o erro gerado pela RNA
77     e = tnT-saidaTeste;
78     error = mse(e)

```