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

Criação de um ambiente computacional para detecção de outliers e preenchimento de falhas em dados meteorológicos

Thiago Meirelles Ventura

Orientador: **Prof. Dr. Josiel Maimone de Figueiredo**

Coorientadora: **Profa. Dra. Marta Cristina de Jesus Albuquerque Nogueira**

Cuiabá - MT
Fevereiro/2015

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

Criação de um ambiente computacional para detecção de outliers e preenchimento de falhas em dados meteorológicos

Thiago Meirelles Ventura

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

Prof. Dr. Josiel Maimone de Figueiredo

Profa. Dra. Marta Cristina de Jesus Albuquerque Nogueira

Cuiabá, MT

Fevereiro/2015

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

FOLHA DE APROVAÇÃO

TÍTULO: **CRIAÇÃO DE UM AMBIENTE COMPUTACIONAL PARA DETECÇÃO DE OUTLIERS E PREENCHIMENTO DE FALHAS EM DADOS METEOROLÓGICOS**

AUTOR(A): **THIAGO MEIRELLES VENTURA**

Tese defendida e aprovada em 27 de fevereiro de 2015 pela comissão julgadora:

Prof. Dr. Josiel Maimone de Figueiredo
Orientador
Instituto de Computação – UFMT

Profa. Dra. Marta Cristina de Jesus Albuquerque Nogueira - Coorientadora
Faculdade de Arquitetura, Engenharia e Tecnologia - UFMT

Profa. Dra. Claudia Aparecida Martins
Examinadora Interna
Instituto de Computação – UFMT

Prof. Dr. Jonathan Willian Zangeski Novais
Examinador Externo
Universidade de Cuiabá - UNIC

Profa. Dra. Maria Camila Nardini Barioni
Examinadora Externa
Faculdade de Computação
Universidade Federal de Uberlândia - UFU

DEDICATÓRIA

Ao meus pais, Muriacy e Virgínia, à minha esposa, Rosangela, e ao meu filho, Leandro. Pelo apoio, incentivo e fazer, mesmo dos períodos mais difíceis, bons momentos.

Dados Internacionais de Catalogação na Fonte.

V468c Ventura, Thiago Meirelles.
Criação de um ambiente computacional para detecção de outliers e preenchimento de falhas em dados meteorológicos / Thiago Meirelles Ventura. -- 2015
96 f. : il. color. ; 30 cm.

Orientador: Josiel Maimone de Figueiredo.
Co-orientadora: Marta Cristina de Jesus Albuquerque Nogueira.
Tese (doutorado) - Universidade Federal de Mato Grosso, Instituto de Física, Programa de Pós-Graduação em Física Ambiental, Cuiabá, 2015.
Inclui bibliografia.

1. tratamento de dados. 2. dados ambientais. 3. inteligência artificial. 4. framework. I. Título.

Agradecimentos

- Aos meus pais eu agradeço pelo apoio que me deram em todos os momentos de minha vida.
- A minha esposa Rosangela pela sensibilidade, esforço, companhia e sacrifício. Tenho orgulho de ser o seu marido. Muito obrigado.
- Aos professores Josiel M. Figueiredo, Claudia A. Martins e Marta Cristina J. A. Nogueira pela paciência e ensinamentos.
- Ao professor José de Souza Nogueira por sempre estar disponível a ajudar e por fazer do PGFA um ótimo programa de pós-graduação.
- Ao professor Todor Ganchev pela orientação no doutorado sanduíche e a todos os meus novos amigos da Bulgária.
- Aos membros do TABA pelo esforço coletivo. Em especial ao Wagner, Ariane e Luy pela ajuda nos projetos.
- A Allan, Raphael, Henrique, César e Gracyeli pela grande ajuda em vários trabalhos deste doutorado.
- Aos colegas Thiago Rangel, Leone, Jonathan, Paula e Renan, pelas reuniões que ajudaram na conclusão desse doutorado.
- A CAPES pelo auxílio financeiro em meu doutorado sanduíche na Bulgária pelo PDSE.
- E, por fim, a todos os outros professores, técnicos e alunos do PGFA que, direta ou indiretamente, ajudaram em meus estudos.

Se você aproveitar o tempo a fim de melhorar-se, o tempo aproveitará você para realizar maravilhas.

André Luiz

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

AG	Algoritmos Genéticos 16
API	Application Programming Interface 40
AUC	*Area Under Curve* 29
BD	Banco de Dados 91
CRISP-DM	Cross Industry Standard Process for Data Mining 6
CSV	*Comma Separated Values* 43
EMA	Erro médio absoluto 29
FP	False Positive 29
HANTS	*Harmonic Analysis of Time Series* 15
HMM	Hidden Markov Model 16
IA	Inteligência Artificial 2
IM	Imputação Múltipla 15
IW	Interface Web 92
JSF	JavaServer Faces 93
MANNGA	*Method with Artificial Neural Network and Genetic Algorithm* ... 31
MB	Megabyte 10
MLP	*Multilayer Perceptron* 16

RESUMO

VENTURA, T. M. Criação de um Ambiente Computacional para Detecção de Outliers e Preenchimento de Falhas em Dados Meteorológicos. Cuiabá, 2015, 96f. Tese (Doutorado em Física Ambiental) - Instituto de Física, Universidade Federal de Mato Grosso.

Para realizar o estudo do meio ambiente é necessário efetuar análises em séries de dados meteorológicos. Entretanto, essas séries de dados podem conter erros, por causa de falhas eletrônicas, atividades de animais, ações de fenômenos climáticos, dentre outros fatores. Essas falhas podem ser dados ausentes ou a presença de *outliers*, causando uma dificuldade na análise dos dados. Portanto, é importante que os *outliers* nas séries de dados sejam detectados e as falhas preenchidas. Sendo assim, este trabalho apresenta um ambiente computacional para possibilitar o tratamento de dados ambientais. Para tanto, três novos métodos foram criados neste trabalho: um para preenchimento de falhas e dois para detecção de *outliers*. Além do mais, três outros métodos foram obtidos da literatura e implementados, em conjunto com os novos métodos, em um único *framework*. Esses métodos utilizam técnicas da área de inteligência artificial e da estatística, o que normalmente exige um estudo aprofundado para a aplicação dos mesmos. Entretanto, o *framework* desenvolvido viabiliza a aplicação desses métodos, exigindo apenas a configuração de alguns parâmetros. Com isso, o *framework* permite que sistemas sejam desenvolvidos com funcionalidades de preenchimento de falhas e detecção de *outliers*. Para demonstrar a aplicação dos métodos um sistema *web-based* foi desenvolvido integrado ao *framework*. Além disso, testes foram realizados para verificar o desempenho de cada método criado comparados aos obtidos da literatura. Acredita-se que a disponibilização desse ambiente melhorará a qualidade dos dados ambientais, auxiliando diversas pesquisas científicas.

Palavras-chaves: tratamento de dados, dados ambientais, inteligência artificial, *framework*.

ABSTRACT

VENTURA, T.M. Development of a Computational Environment for Outlier Detection and Gap Filling in Meteorological Data. Cuiabá, 2015, 96f. Thesis (Doctorate in Environmental Physics); Institute of Physics, Federal University of Mato Grosso.

In order to study the environment meteorological data series must be analyzed. However, these data series may contain errors, because of electronic failures, animal action or weather phenomena, among other factors. These failures can result in missing data or outliers, causing difficulties in the data analysis. Therefore, it is important to detect the outliers in the data series and fill in the missing data. This work presents a computational environment that will enable the correction of environmental data. In order to achieve this, three new methods were created in this work: one for gap filling and two for outlier detection. In addition, three other methods were obtained from other studies and were implemented together with the new methods in a single framework. These methods use techniques from the area of artificial intelligence and statistics, which often requires a deep study in order to apply them. However, the developed framework enables the application of these methods, only demanding the configuration of some parameters. Thus, the framework allows the development of applications with functionalities of gap filling and outlier detection. To demonstrate the applicability of these methods a web-based application was developed integrated with the framework. Besides, tests were carried out to verify the performance of each method created compared with those obtained from other studies. It is expected that this structure will increase the quality of data series, assisting in several scientific researches.

Keywords: data correction, environmental data, artificial intelligence, framework.

Capítulo 1

Introdução

As informações a respeito das condições climáticas do meio ambiente estão ganhando cada vez mais importância no cenário mundial. Isso acontece principalmente porque alguns especialistas defendem que agora as emissões de poluentes oriundas de atividades humanas têm potencial de colocar em risco a vida no planeta (HEWITT, 2003).

Dados relacionados à observação de fenômenos reais geralmente são denominados dados científicos (PFALTZ, 2007). Esses tipos de dados podem ser, por exemplo, oriundos de equipamentos instalados em campo, dados obtidos por satélites ou resultados de uma simulação. Normalmente, esses dados têm uma maior complexidade em sua estrutura e, por representar fenômenos reais, podem gerar uma grande quantidade de dados.

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. Nesse sentido, dados meteorológicos tem tido um papel importante nas pesquisas científicas. Baseando-se em dados meteorológicos, interpretações sobre fenômenos climáticos podem ser realizadas, permitindo que a comunidade científica compreenda diversas características do nosso planeta.

Atualmente, há uma popularização dos sensores para medições de dados meteorológicos. Com isso, uma grande quantidade de dados que representam variáveis climáticas estão sendo coletados em diversos locais. Entretanto, a manipulação dos dados e, principalmente, a extração de informações sobre esses dados, ainda é complexa.

Em pesquisas envolvendo dados meteorológicos, as séries de dados são, por exemplo, leituras da temperatura do ar no decorrer do tempo. Nessas medições, dados ausentes ou inválidos são problemas comuns devido, por exemplo,

à avaria ou desligamento de equipamentos, manutenção, calibração, limitações físicas ou fenômenos climáticos (HUI, 2004), ou seja, essas falhas podem ocorrer na leitura ou armazenamento dos dados, não sendo possível assegurar a integridade de todos os dados coletados. Em todo caso, a falha criada na série de dados prejudicará a interpretação dos dados, causando afirmações errôneas.

Quando se deseja tratar os dados meteorológicos, há duas opções a serem feitas: remover períodos inteiros de falhas em uma série de dados para permanecer apenas os períodos válidos ou utilizar um método para encontrar e tratar as falhas.

Na primeira opção, há a desvantagem da possibilidade de remover justamente os períodos que poderiam mostrar fenômenos importantes. Ao invés de se trabalhar com a totalidade dos dados, apenas uma parcela deles estariam sendo processados e analisados.

O segundo caso, ou seja, a correção dos dados, é considerado o ideal. Mas a correção normalmente é um procedimento complexo e que demanda muito tempo. Caso seja optado por utilizar um método mais simples, a presença de *outliers* pode não ser detectada e a precisão do preenchimento de falhas provavelmente será baixa. Então, é possível perceber que seria de grande utilidade ter uma maneira prática para detectar os dados inválidos, como os *outliers*, e tratar esses dados, com a realização de preenchimento de falhas.

1.1 Justificativa

Foi compreendido que há a necessidade de realizar um tratamento nos dados meteorológicos de forma eficaz, obtendo uma série de dados com maior integridade. Para realizar esses procedimentos, em uma grande quantidade de dados, alguns métodos estatísticos podem ser aplicados, como Regressão Linear, Média Móvel, *Hidden Markov Model* e Z-score, além de métodos computacionais, especialmente técnicas de Inteligência Artificial (IA), como Redes Neurais Artificiais, Algoritmos Genéticos.

Por outro lado, utilizar técnicas de tratamento de dados não é uma tarefa simples, exigindo conhecimentos avançados sobre o comportamento dos dados, técnicas computacionais ou métodos estatísticos. Além disso, para ter uma boa precisão no tratamento dos dados, os métodos normalmente devem ser adaptados para que o mesmo funcione com o tipo de dado utilizado, gerando um trabalho considerável.

Para contornar essa situação, é possível que sejam criados métodos que abstraem a complexidade da correção dos dados. Além do mais, pode ser de-

senvolvido um *framework* que agregue diversos métodos de tratamento de dados, simplificando todo o processo para aumentar a qualidade das séries de dados. Com o *framework*, torna-se viável o desenvolvimento de sistemas, simplificando ainda mais todas as operações de tratamento de dados meteorológicos.

1.2 Objetivo Geral

O objetivo deste trabalho é criar novos métodos de tratamento de dados meteorológicos e desenvolver um ambiente computacional que propicie a aplicação tanto dos novos métodos quanto dos já existentes na literatura.

1.3 Objetivos Específicos

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

- Entender e implementar métodos de detecção de *outliers* e preenchimento de falhas já existentes na literatura;
- Criar e implementar novos métodos de detecção de *outliers* e preenchimento de falhas;
- Criar uma estrutura que padronize a utilização de métodos de tratamento de dados ambientais;
- Encapsular todos os métodos implementados em um único *framework* obedecendo as regras da estrutura criada;
- Desenvolver uma aplicação integrada ao *framework*.

1.4 Organização do Trabalho

Este trabalho está organizado da seguinte forma:

- **Capítulo 2:** Este capítulo apresenta os conceitos principais sobre tratamento de dados meteorológicos. São apresentadas informações sobre mineração de dados ambientais, características dos dados meteorológicos, tratamento de dados e conceitos sobre técnicas computacionais e estatísticas.

- **Capítulo 3:** Comenta sobre os métodos existentes no *framework*, tanto os que foram criados quanto os que foram implementados da literatura. Além do mais, descreve como os testes de avaliação foram realizados.

- **Capítulo 4:** Neste capítulo são mostrados em detalhes os métodos criados e a estrutura do *framework*, como ele deve ser utilizado em outros sistemas e os resultados dos testes realizados com todos os métodos existentes no *framework*.

- **Capítulo 5:** No capítulo final, as considerações finais deste trabalho são realizadas, as contribuições são apresentadas, as publicações originadas deste trabalho são listadas e indicações para trabalhos futuros são feitas.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta os tópicos gerais sobre interpretação de fenômenos climáticos e tratamento de dados meteorológicos. O entendimento dos fenômenos climáticos possui grande importância para diversas áreas científicas, mas para que isso seja feito, a extração de informações em uma enorme quantidade de dados deve ser realizada. Para tanto, uma metodologia de mineração de dados pode ser empregada para auxiliar todo o processo, sendo que, primeiramente, um tratamento nos dados deve ser realizado.

Para facilitar o entendimento dos conceitos envolvidos, as próximas seções abordarão os aspectos sobre a metodologia para realizar a mineração de dados e, assim, possibilitar o entendimento dos fenômenos climáticos, as características dos dados meteorológicos (objeto de estudo desta área) e sobre as técnicas computacionais e estatísticas para realizar as devidas correções nos dados meteorológicos, garantindo a qualidade dos dados para que as análises sejam feitas corretamente.

2.1 Mineração de Dados Ambientais

O entendimento dos fenômenos climáticos auxiliam na tomada de decisão das organizações, na criação de medidas preventivas e em previsões variadas sobre o meio ambiente. Segundo Kozievitch (2005), essas informações podem ser aproveitadas em várias áreas, como na agricultura (para determinar a época ideal de colheita, previsão de geadas e granizo), energia (controle dos níveis de reservatório de usinas, informações para fontes alternativas de energia), construção civil (realização de construções mais confortáveis, observando a insolação e umidade dos locais), transporte (condições do tempo nas estradas), segurança (alertas sobre ventanias, inundações e ressacas), meio ambiente (acompanhamento da qualidade

do ar e monitoramento de queimadas), saúde (identificação de áreas alagadas) e turismo (verificação da previsão para feriados e épocas de férias). Então, é compreendido que há a necessidade de se entender como se comportam os fenômenos climáticos, objetivando modelar, prever e correlacionar tais fenômenos.

Para auxiliar na compreensão dos fenômenos climáticos, técnicas de mineração de dados podem ser aplicadas. A mineração de dados é a aplicação de algoritmos específicos para extração de padrões dos dados (FAYYAD et al., 1996), sendo extremamente útil para adquirir conhecimento em grandes bancos de dados.

Uma metodologia para executar a mineração de dados é a CRISP-DM (*Cross Industry Standard Process for Data Mining*), descrita em termos de um modelo de processo hierárquico, que consiste em um conjunto de tarefas separadas em quatro níveis de abstração (do geral para o específico): fase, tarefa genérica, tarefa especializada e instância de processo (CHAPMAN et al., 2000).

Na Figura 1 estão as etapas da metodologia para realização dessas tarefas. Segundo Chapman et al. (2000), o ciclo externo representa o processo contínuo da mineração de dados, mostrando que o processo não termina quando uma solução é encontrada, podendo servir como informação para que o sistema encontre informações ainda mais detalhadas. Em cada ciclo, seis etapas podem ser desempenhadas, sendo que a ordem de execução depende da tarefa que está sendo realizada, ou seja, não é rígida.

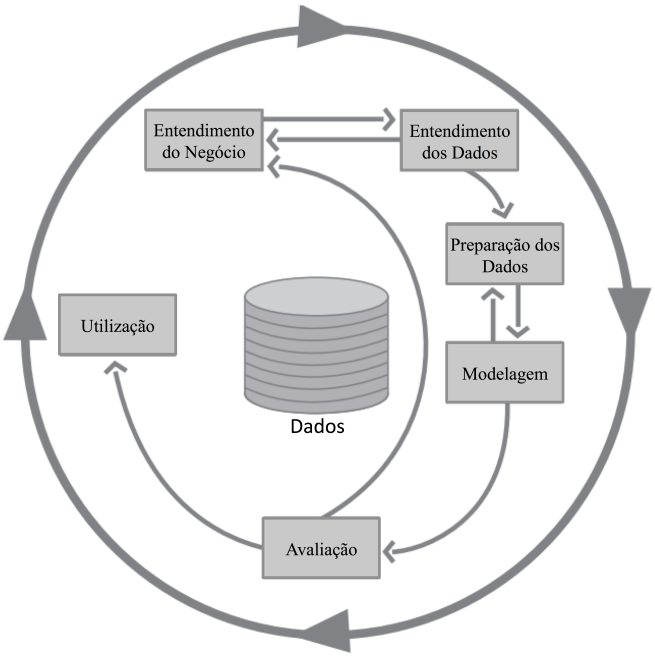

Figura 1: Modelo de referência das etapas do CRISP-DM (CHAPMAN et al., 2000)

Entendendo cada uma das etapas envolvidas, é possível aplicar a metodologia CRISP-DM para interpretações de fenômenos climáticos. Em Sferra e Corrêa (2003) é detalhado cada uma das seis etapas existentes na metodologia CRISP-DM:

- **Entendimento do Negócio:** visa o entendimento dos objetivos e requisitos do projeto, do ponto de vista do negócio. Baseado no conhecimento adquirido, o problema de mineração de dados é definido e um plano preliminar é projetado para alcançar os objetivos.
  Em se tratando de entendimento dos fenômenos climáticos, pode ser considerado que essa etapa é a análise preliminar de pesquisadores planejando um projeto de pesquisa e definindo as modelagens que serão aplicadas.

- **Entendimento dos Dados:** inicia com uma coleção de dados e prossegue com atividades que visam buscar familiaridade, identificar problemas de qualidade, descobrir os primeiros discernimentos nos dados ou detectar subconjuntos interessantes para formar hipóteses da informação escondida. Para a área ambiental, essa etapa contempla o processo de digitalização dos dados oriundos dos diversos tipos de equipamentos instalados em estações meteorológicas no qual, normalmente, funcionam 24 horas por dia, 7 dias por semana, durante anos consecutivos, gerando uma enorme quantidade de valores com diversos *gigabytes* ou até *terabytes* de dados. Também pode ser atribuído a esta etapa os procedimentos de calibragem dos equipamentos, além das primeiras análises pelos especialistas.

- **Preparação dos Dados:** cobre todas as atividades de construção do conjunto de dados final. As tarefas de preparação de dados são, provavelmente, desempenhadas várias vezes e sem qualquer ordem prescrita. Essas tarefas incluem a seleção de tabelas, registros e atributos, bem como a transformação e limpeza dos dados para as ferramentas de modelagem.
  No contexto deste trabalho, os dados meteorológicos devem ser uniformizados e sofrerem as aplicações de métodos complexos para encontrar e corrigir as falhas, resultando em uma base de dados de melhor qualidade. O trabalho descrito aqui está relacionado especificamente com esta etapa, visando selecionar e criar métodos próprios para o tratamento de dados meteorológicos.

- **Modelagem:** várias técnicas de modelagem são selecionadas e aplicadas, e seus parâmetros são ajustados para valores ótimos. Geralmente, existem

várias técnicas para o mesmo tipo de problema de mineração de dados. Algumas delas têm requisitos específicos na formação de dados. Portanto, retornar à fase de preparação de dados é frequentemente necessário.
Esta etapa tem forte relação com a área ambiental, já que a modelagem dos dados é uma das principais tarefas para o entendimento dos fenômenos climáticos. Então, as técnicas utilizadas para simular ou explicar determinados fenômenos serão realizadas nesta etapa.

- **Avaliação:** o modelo (ou modelos) construído na fase anterior é avaliado e os passos executados na sua construção são revistos, para se certificar que o modelo representa os objetivos do negócio. Seu principal objetivo é determinar se existe alguma questão de negócio importante que não foi suficientemente considerada. Nesta fase, uma decisão sobre o uso dos resultados de mineração de dados deverá ser obtida.
  O mesmo será realizado caso esteja utilizando dados meteorológicos, ou seja, serão feitas as análises dos resultados gerados pelas modelagens da etapa anterior.

- **Utilização ou Aplicação:** após a construção e avaliação do modelo (ou modelos), ele pode ser utilizado de duas formas: em uma, o analista pode recomendar ações a serem tomadas baseando-se, simplesmente, na visão do modelo e de seus resultados; na outra, o modelo pode ser aplicado a diferentes conjuntos de dados.
  Além disso, para o contexto ambiental, é muito importante a visualização dos dados ambientais e, posteriormente, a publicação dos resultados, seja ela por artigos científicos, relatórios técnicos ou em outras mídias para o público.

A mineração de dados com o CRISP-DM já foi aplicada em trabalhos da área ambiental, como em Thamada et al. (2013) e Meira e Rodrigues (2005) na utilização em sistemas de alertas contra doenças agrárias. Também já foi utilizada para análises de séries temporais, como em Dourado et al. (2013) e Boschi et al. (2009).

Como este trabalho está relacionado diretamente com a preparação dos dados, é necessário um maior entendimento sobre os dados meteorológicos. É importante conhecer as características desses dados e suas possíveis falhas, para que o tratamento desse tipo de dado possa ser realizado com exatidão.

2.2 Dados Meteorológicos

Para ter um entendimento melhor sobre fenômenos climáticos são necessários analisar dados meteorológicos. Para tanto, são utilizadas as estações meteorológicas para captar, armazenar e analisar dados desse contexto (JÚNIOR, 2008). Os equipamentos responsáveis por medir elementos meteorológicos passam por sensores e os dados são armazenados em sistemas de aquisição de dados (NEVES, 2011).

Normalmente, vários equipamentos são instalados para coletar e armazenar dados para análise, e cada equipamento é responsável por mensurar uma ou mais variáveis climáticas, como temperatura, umidade relativa do ar, radiação solar, entre outras. Só depois é possível realizar a análise dos dados e tirar conclusões sobre as variáveis climáticas estudadas.

Na Tabela 1 é apresentada uma amostra de dados meteorológicos representando os valores de tempo, pressão atmosférica (P), temperatura média (T), umidade relativa do ar (UR), insolação (INSOL), velocidade média do vento (u), fotoperíodo (N) e saldo de radiação (Qn). Os atributos presentes nas séries de dados dependerá dos equipamentos instalados nas estações meteorológicas e das configurações realizadas na importação dos dados.

Tabela 1: Amostra de dados obtidos por equipamentos meteorológicos

Dia	P	T	UR	INSOL	u	N	Qn
1	991,9	27,7	79,8	85	8	11,99	10,42
2	992,3	26,9	84,8	63	7,66	11,98	9,16
3	992,0	26,5	86,5	51	7,33	11,96	8,35
4	992,2	26,1	89,0	26	6,66	11,95	6,88
5	993,0	24,7	98,3	7	5,66	11,93	5,81
6	992,5	25,8	87,0	33	6,66	11,92	7,29
7	994,3	26,0	86,0	40	7	11,90	7,58

Um problema que surge é quando os dados coletados pelos equipamentos nas estações meteorológicas sofrem algum tipo de falha. Como todo aparelho eletrônico, os equipamentos de medições de dados meteorológicos estão sujeitos a falha.

As falhas nos dados meteorológicos 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. Essas falhas comprometem as análises realizadas com base nos dados das estações de meteorologia. Por isso, é importante que essas falhas sejam detectadas e corrigidas, para uma análise mais confiável.

As possíveis falhas nos dados meteorológicos são a presença de *outliers* e a ausência dos dados, sejam elas por falha do equipamento de medição, falha no equipamento de armazenamento, falha na comunicação dos dados entre a estação meteorológica e o seu destino final, corrupção no arquivo que detinha os dados ou outra ocorrência que causaria uma alteração no valor que estava representando uma variável climática em determinado momento.

Para que a análise dos dados não seja prejudicada, métodos de tratamento de dados meteorológicos devem ser aplicados para realizarem as correções das falhas. Os métodos devem ser escolhidos ou desenvolvidos considerando as características dos dados meteorológicos.

Uma característica importante desse tipo de dado é que eles são multivariados. Um conjunto de dados multivariado é aquele que tem muitas variáveis dependentes que podem ser correlacionadas entre si em diferentes graus (SANTOS, 2004). É possível aproveitar dessa característica para detectar padrões nos dados, auxiliando na estimativa de valores para preencher falhas ou detectar *outliers*.

Os dados meteorológicos geralmente são complexos e não lineares. Por exemplo, a evapotranspiração tem essa característica pois depende da interação entre vários elementos climatológicos, como temperatura, umidade, velocidade do vento e radiação (KUMAR et al., 2002). A complexidade e não linearidade de algumas variáveis climáticas impedem que técnicas lineares tenham um bom desempenho na correção.

Além do mais, um ponto importante sobre este tipo de dado é a frequência em que sua coleta é realizada. Isso varia de acordo com a aplicação, podendo ser a cada dia, hora, minuto ou até mesmo a cada milisegundo, dependendo do tipo de análise que deseja ser realizado. Isso implicará diretamente na quantidade de dados armazenados.

Menos de 1 *MB* de dados é necessário para armazenar os valores de uma única variável climática, que está sendo coletada a cada 15 minutos, ininterruptamente durante todo o dia, durante um ano. Entretanto, se a mesma variável climática for armazenada a cada 1 segundo, o espaço necessário para o mesmo tempo de coleta seria de aproximadamente 240 *MBs*. Para uma estação meteorológica com vários equipamentos, esse montante pode chegar facilmente a *gigabytes* de dados armazenados (considerando que cada valor armazenado necessite de 8 *bytes* - tamanho de uma variável do tipo *double*). Então, deve ser levado em consideração um provável rápido crescimento na quantidade de dados coletados.

A frequência que um dado é armazenado e o possível tamanho que a série

de dados pode possuir são características que podem tornar impraticável o uso de determinados métodos por causa do tempo de processamento necessário quando se tem uma grande quantidade de dados.

Outra característica sobre os dados meteorológicos é a importância do tempo em sua série de dados, ou seja, são considerados como dados temporais. Métodos específicos para séries temporais possuem grandes chances de obterem bons desempenhos ao serem utilizados para os dados coletados nas estações meteorológicas.

É possível concluir que as características dos dados meteorológicos interferem diretamente no método de tratamento de dados. Para um maior embasamento, nas próximas seções são mostrados os tipos de falhas em dados meteorológicos e os métodos normalmente aplicados para realizarem as respectivas correções.

2.2.1 Detecção de *Outliers*

Os *outliers*, ou anomalias, são dados que parecem desviar significativamente dos outros membros da amostra da qual faz parte (GRUBBS, 1969). Esses valores, que não são consistentes com o restante da série de dados, podem ser falhas ou ruídos provocados por erros de equipamentos, além de ocorrências reais em campo que afetam de maneira negativa os sensores dos equipamentos. Tais valores precisam ser removidos em uma etapa de pré-processamento ou, pelo menos, necessitam de uma análise mais criteriosa.

Devido à característica dos equipamentos de medições meteorológicas estarem sujeitos às condições ambientais, por estarem instalados em um ambiente externo, é comum a necessidade de utilizar métodos de detecção de *outliers* nos dados coletados. Um método simples para detectar *outliers* consiste em estabelecer um valor mínimo e máximo para os dados válidos e considerar como *outliers* os dados que saem desse limite. Provavelmente não será atingida uma boa eficácia ao utilizar um método simples como este. Por outro lado, há vários outros métodos de detecção de *outliers* que buscam uma maior precisão.

Em Chen et al. (2010) são citados as categorias dos métodos de detecção de *outliers*, como os métodos baseados em estatística, baseados na profundidade, baseados na distância, baseado na densidade e ainda os baseados em desvio. A diferença entre cada categoria está na forma que o método separa um dado comum dos dados considerados *outliers*.

Os métodos baseados em estatística assumem que os dados possuem uma distribuição normal e consideram que os dados fora dessa distribuição normal são

outliers (HODGE; AUSTIN, 2004). Esses métodos exigem um conhecimento da distribuição dos dados e das relações entre os parâmetros dessa distribuição (CHEN et al., 2010). Além disso, os métodos baseados em estatística univariados não aproveitam da característica multivariada dos dados meteorológicos.

Um conjunto dos métodos baseados em estatística são os métodos baseados em profundidade. Nesses métodos, cada objeto é representado como um ponto em um espaço $k - d$, no qual lhe é atribuído uma profundidade. Os objetos considerados *outliers* são os dados com profundidades menores (BREUNIG et al., 2000). Devido a complexidade computacional, estes métodos são inviáveis em grandes conjuntos de dados com mais de quatro dimensões (KNORR et al., 2000).

O conceito de representar um dado como um ponto em um espaço multidimensional é também utilizado em outros métodos. Segundo Chávez et al. (2001), o conjunto desses pontos é chamado de espaço métrico, no qual, com base em uma função de distância, é possível dizer quão similar é um objeto em relação a outro, possibilitando realizar buscas por similaridade. A função de distância, criada por um especialista na área, é calculada com base em um conjunto de características que são extraídas a partir do dado (CHÁVEZ et al., 2001). Uma função de distância tem seu uso mais efetivo quando ocorre especificamente sobre as características em que ela faz uma melhor discriminação (FIGUEIREDO J, 2005).

Nos métodos baseados na distância dos dados, os *outliers* são aqueles dados que se distanciam mais dos outros dados. Segundo Knorr et al. (2000), um objeto O em uma série de dados T é definido como um *outlier* se uma fração p de objetos em T encontra-se a uma distância maior que D de O. A dificuldade deste tipo de método está em criar a função distância e determinar qual tamanho de p e D. Geralmente esses métodos não são projetados para lidar com a maldição da dimensionalidade (AGGARWAL; YU, 2005), além de necessitar de um ajuste refinado que dificulta a definição dos parâmetros em aplicações práticas (CHEN et al., 2010).

A maldição da dimensionalidade ocorre devido à esparsidade dos dados nos espaços de alta dimensão, o que faz com que a distância medida pela função tenda a um valor semelhante para todos os elementos do conjunto de dados, isto é, a discriminção dos elementos tende a não ocorrer (FIGUEIREDO J, 2005; BRAUNER; MORDECHAI, 2000; BELLMAN, 1957).

Os métodos baseados em densidade seguem o conceito de agrupamento dos dados. Utilizando de uma medida criada para os dados que estão sendo

manipulados, os dados são agrupados em conjuntos e, segundo Murugavel e Punithavalli (2013), conjuntos grandes (de grande densidade) representam os dados normais, já os conjuntos pequenos (de pouca densidade) são considerados valores discrepantes (*outliers*). A desvantagem é que esses tipos de métodos são computacionalmente caros e conjuntos densos frequentemente possuem dados comuns e *outliers* ao mesmo tempo (GUHA et al., 1999).

De acordo com Arning et al. (1996), os métodos baseados em desvio podem ser considerados como um caso especial de agrupamento de dados, com a diferença que há apenas dois grupos: dados normais e os *outliers*. Nesse tipo de método, não há a necessidade de uma função de distância entre os elementos de dados, requerendo apenas uma função capaz de produzir um resultado de dissimilaridade entre os dados, ou seja, o valor aumenta a medida que os dados se tornam mais diferentes (LUXBURG, 2004). Em Zhang e Feng (2009) são apresentados três algoritmos desse tipo de método.

Em Arning et al. (1996), na tentativa de criar uma única função que funcionasse para todo tipo de dado, baseando-se em experiências com várias funções em vários conjuntos de dados, chegou-se a conclusão de que é muito difícil ter uma função universal que funcione bem para todos os conjuntos de dados. Logo, a modificação de um método é quase que essencial para que o mesmo funcione corretamente para os dados que estão sendo tratados.

Algo que pode facilitar esse processo é a escolha de um método que já foi avaliado para o tipo de dado que se deseja tratar. Existem técnicas de detecção de *outliers* para cada categoria mencionada (baseados em estatística, baseados na profundidade, baseados na distância, baseado na densidade e baseados em desvio), então a escolha da técnica que deverá ser utilizada deve levar em consideração não só os pontos positivos e negativos de determinada categoria, mas também o tipo de dado que está sendo tratado.

Além de enumerar os diferentes tipos de *outliers* e as diferentes aplicações que a detecção de *outliers* pode ser aplicada, em Chandola et al. (2007) são listadas várias técnicas encontradas na literatura. Algo semelhante pode ser visto em Zhang (2013), no qual alguns dos avanços de detecção de *outliers* são mostrados, inclusive para lidar com grandes conjuntos de dados, dados por *streaming* e dados complexos de várias dimensões.

Em Gupta et al. (2014) são mostradas diversas técnicas para vários casos diferentes em se tratando de dados temporais, como dados por *streaming*, dados distribuídos em múltiplos locais e dados espaço-temporais. É comentado ainda que a modelagem de dados temporais é uma tarefa desafiadora, devido à natureza

dinâmica e padrões evolutivos complexos nos dados.

Ainda sobre dados do tipo espaço-temporal, Sun e Genton (2012) descreve uma forma de detectar *outliers* utilizando o *boxplot*, envolvendo dados de temperatura da superfície do mar e de precipitação em diferentes locais.

Há também algoritmos de detecção de *outliers* para variáveis mais específicas. Em Weekley et al. (2010) foi utilizado técnicas de processamento de imagem e de agrupamento para detectar *outliers* em dados sobre o vento. Já em Wu et al. (2010) foi aperfeiçoado um algoritmo para que *outliers* em dados de precipitação pudessem ser encontrados, mesmo que hajam falhas em regiões vizinhas.

Outliers de dados de temperatura do solo foram detectados no trabalho de Sadik e Gruenwald (2010). Foram utilizadas médias diárias de 50 estações meteorológicas e um método próprio de detecção de *outliers* baseado em distância.

É possível observar que há inúmeras opções para escolha de um método de detecção de *outliers*. Entretanto, foi possível observar também que em cada método são necessárias alterações para que se adapte ao dado que está sendo processado, podendo obter assim bons resultados. Além das dificuldades metodológicas, há ainda a dificuldade em aplicar os métodos. A utilização dos métodos não é simples para aqueles que não estão familiarizados com as técnicas. Esse cenário também ocorre no caso de métodos para preenchimento de falhas.

2.2.2 Preenchimento de Falhas

Preencher falhas de dados meteorológicos ausentes consiste em estimar os valores, modelando o comportamento do fenômeno baseado em dados históricos. Entretanto, assim como nos métodos para detecção de *outliers*, aplicar métodos de preenchimento de falhas é uma operação complexa.

Por causa disso, é normal encontrar trabalhos que preferem excluir períodos inteiros de dados ou utilizar métodos simples de preenchimento de falhas, como a repetição ou média de dados próximos. Trabalhos como os de Soares et al. (2014), Palú (2008), Capistrano (2007), Gallon (2005) e Mariano (2005) mostram essa dificuldade.

Alguns métodos estatísticos são amplamente utilizados em dados de variáveis climáticas. Para dados de chuva, por exemplo, foram utilizados no trabalho de Oliveira et al. (2010) métodos da ponderação regional, regressão linear e regressão potencial. Em Chibana et al. (2005), além de utilizar alguns métodos estatísticos, também aproveitou de dados coletados próximos à estação em que houve a falha para realizar o tratamento dos dados. Metodologia semelhante foi

aplicada em Pinheiro et al. (2013) e Ferrari e Ozaki (2014).

Outros métodos estatísticos, como o Imputação Múltipla (IM), uma técnica de Monte Carlo, foi utilizado em Hui (2004) e Orton e Ipsitz (2001). Além do mais, em Falge et al. (2001) são comparados outros métodos de preenchimento de falhas, dentre eles a Variação Média Diurna (VMD), utilizado em Hu et al. (2009) e em Alavi et al. (2006), o *look-up tables*, utilizado em Mishurov e Kiely (2011) e em Shao et al. (2011), e a regressão linear, utilizado em Tardivo e Berti (2014).

Há métodos que podem atingir uma precisão ainda melhor no preenchimento de falhas. Em Ooba et al. (2006) foi mostrado que uma combinação de técnicas da área de Inteligência Artificial (redes neurais e algoritmos genéticos) teve melhores resultados do que um método mais convencional que usa equações com parâmetros que foram determinados por meio de regressão não linear.

Nos trabalhos de Tsukahara et al. (2010), Deswal e Pal (2008), Jain et al. (2008) e Lima (2010) também foram utilizadas técnicas da área de Inteligência Artificial para realizar estimativas em dados da área ambiental, como temperatura, umidade relativa, radiação solar, evapotranspiração e precipitação.

Para preenchimento de falhas em dados de precipitação, em Wanderley et al. (2012) foi feita interpolação com técnicas geoestatísticas considerando estações meteorológicas próximas, em Hasan e Croke (2013) utilizou-se de distribuição de *Poisson-gamma* (PG) e em Nascimento et al. (2009) foram feitas regressões lineares.

Dados com relação às medidas de *eddy covariance* também já foram tratados utilizando técnicas de preenchimento de falhas. Nos trabalhos de Noormets et al. (2007), Moffat et al. (2007), Serrano-Ortiz et al. (2009), Falge et al. (2001) e Dengel et al. (2013) foram utilizados métodos variados, como VMD, IM, *look-up tables*, regressão não linear, *unscented Kalman filter*, *Marginal Distribution Sampling*, e redes neurais artificiais para realizar o tratamento desses dados.

A evapotranspiração também necessita de preenchimento de falhas, principalmente no caso de estimativas por satélites que podem ser prejudicadas por causa da presença de nuvens. Em Xiong et al. (2010) e em Jia et al. (2009) são utilizados, respectivamente, *Penman-Monteith* e o algoritmo denominado *Harmonic Analysis of Time Series* (HANTS) para realizar o preenchimento desses dados.

Apesar de haver vários casos de preenchimento de falhas em dados meteorológicos, ainda existe a dificuldade de conhecimentos específicos sobre os dados para que seja possível alterar os métodos mais complexos a fim de obter a precisão

desejada. Como isso não é uma atividade trivial, seria de grande utilidade um método que se adaptasse automaticamente às características dos dados, e nisso as técnicas computacionais também podem ser úteis.

2.3 Técnicas Computacionais

As técnicas computacionais, como as da área de Inteligência Artificial, podem auxiliar na resolução de diversos problemas. Sobre o campo da Inteligência Artificial, Russell e Norvig (2004) diz que tenta compreender e construir entidades inteligentes, abrangendo uma enorme variedade de subcampos de estudo. 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.

Há várias técnicas na área de IA. Dentre elas, estão as Redes Neurais Artificiais (RNA) e os Algoritmos Genéticos (AG). Além das técnicas de IA, pode ser citado o modelo probabilístico de *Hidden Markov Model* (HMM) como uma técnica computacional. Essas técnicas foram utilizadas neste trabalho e, por tanto, são detalhadas nas próximas seções.

2.3.1 Redes Neurais Artificiais

As Redes Neurais Artificiais são técnicas computacionais cujo funcionamento baseia-se no contexto da estrutura neural de seres vivos e que realizam tarefas de computação adquirindo conhecimento através de experiência, construída por um processo de aprendizagem, tendo como principais vantagens as características de adaptabilidade, generalização e tolerância a falhas (HAYKIN, 2001).

Técnicas computacionais como RNA têm sido utilizadas com sucesso para modelar relações envolvendo séries temporais complexas (ZANETTI et al., 2007). A utilização de RNAs em problemas de modelagens complexas se deve em função de sua estrutura não linear e a capacidade de captar características mais complexas dos dados, o que nem sempre é possível com a utilização das técnicas estatísticas tradicionais (GALVÃO et al., 1999).

Uma RNA pode ter diferentes tipos de arquiteturas. Uma muito utilizada por sua eficiência é a perceptron de múltiplas camadas (MLP). Nesse tipo de rede o sinal de entrada se propaga para frente (*feedforward*), camada por camada, sendo em seguida retropropagado para a correção do erro (ajuste dos pesos sinápticos). Este procedimento é repetido durante várias iterações até a finalização

do treinamento (ZANETTI et al., 2008).

O funcionamento de uma rede neural, baseando-se na estrutura apresentada na Figura 2, é basicamente uma função dos sinais de entrada pelos seus respectivos pesos sinápticos (w_k). O bias (b_k) funciona aumentando ou diminuindo a influência do valor da entrada líquida 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).

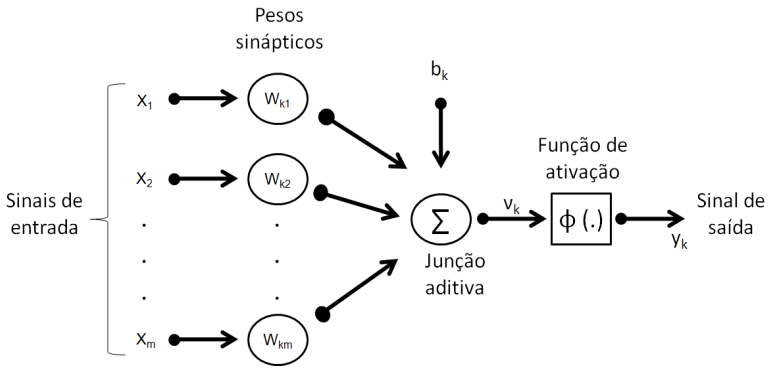

Figura 2: Estrutura básica de uma Rede Neural Artificial (HAYKIN, 2001)

Quando se trabalha com a técnica de RNA existe uma certa dificuldade em encontrar a melhor estrutura da rede que, geralmente, consiste em investigar todo um espaço de estados possíveis (NETO et al., 2005). Como a análise de todas as possibilidades é impraticável, é possível utilizar algum algoritmo de busca para encontrar uma solução satisfatória. Nesse contexto, técnicas de Algoritmos Genéticos podem ser utilizadas para auxiliar na definição da estrutura da rede, por ser um método de busca que tem como característica encontrar uma solução baseada no ótimo global (ASSUMPÇÃO et al., 2010).

2.3.2 Algoritmos Genéticos

Algoritmos Genéticos foram inicialmente propostos por John Holland em 1975, e 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).

Os AGs são amplamente utilizados para problemas de otimização, seu princípio básico consiste em fazer evoluir um conjunto de soluções candidatas iniciais (indivíduos), para uma solução ótima. Indivíduos, segundo Sheikh et al. (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.

Para que uma solução ótima seja alcançada, um ciclo é realizado, como apresentado na Figura 3. O ciclo inicia selecionando aleatoriamente certo número de indivíduos dentro do espaço de busca. Os indivíduos selecionados são avaliados em relação à capacidade de resolver o problema, e essa capacidade é expressa numericamente pela avaliação do indivíduo, através de uma função objetivo (MICHALEWICZ, 1994). Com base nessa informação, uma nova população é formada utilizando operadores probabilísticos de seleção, *crossover* (recombinação) e mutação.

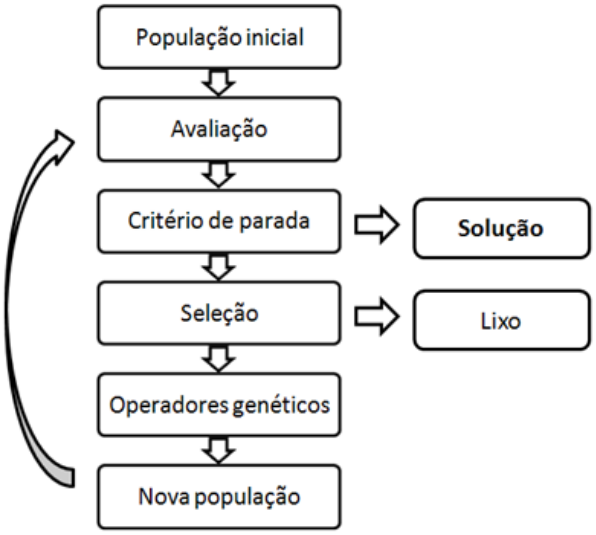

Figura 3: Ciclo de vida de um Algoritmo Genético

Alguns trabalhos já foram desenvolvidos com o objetivo de otimizar a estrutura de uma RNA por meio de AG. Em Neto et al. (2005) é possível observar a utilização de AG para a seleção de entradas da RNA para previsão de dados em séries temporais. Já em Ahmad et al. (2010), além de selecionar as melhores entradas, foi definido o número de neurônios na camada oculta de uma RNA para diagnóstico de câncer. Em Ooba et al. (2006) foi definido as entradas, as taxas de aprendizagem e de *momentum* e, ainda, os pesos iniciais das conexões para corrigir falhas em dados de fluxo de carbono. Logo, a aplicação de AGs podem resolver o problema elencado na Seção 2.3.1, para encontrar a melhor estrutura de uma RNA em relação a um determinado conjunto de dados.

2.3.3 *Hidden Markov Model*

A técnica *Hidden Markov Model* foi descrita em Rabiner (1989). O HMM é um modelo probabilístico temporal no qual o estado do processo é descrito por uma única variável aleatória discreta (RUSSELL; NORVIG, 2004). Segundo

Ghahramani (2001), HMM é uma ferramenta para representar a probabilidade de distribuições sobre sequências de observações.

O modelo de Markov pode ser representado como uma máquina de estados finito Young et al. (2006), como mostrado na Figura 4. As observações (o_k) utilizadas para o treinamento irão determinar uma sequência entre os estados (1 a 6) existentes, no qual cada transição entre um estado e outro ocorre de acordo com determinadas probabilidades (a_{ij} e b_k). Depois do modelo ser treinado, é possível verificar quão similar é uma nova observação com relação às observações utilizadas no treinamento e, assim, determinar se a nova observação pertence a este modelo, realizando uma classificação.

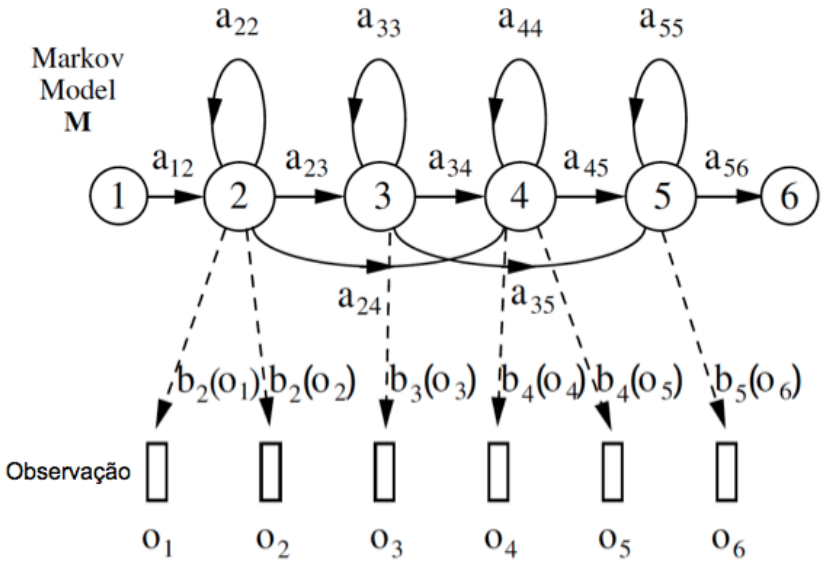

Figura 4: Modelo de Markov (YOUNG et al., 2006)

Como pode ser visto, o HMM usa estados, normalmente com misturas gaussianas, para modelar um tipo de problema e ser capaz de identificar novos tipos de dados depois de treinado. O treinamento pode utilizar diversos algoritmos, sendo o mais comum o *Baum–Welch* (BAUM et al., 1970) para estimar os melhores parâmetros para os modelos que estão sendo treinados (SHEN et al., 2011; MICHALEK; TIMMER, 1999).

O HMM é aplicado em praticamente todos os sistemas modernos de reconhecimento de voz (GALES; YOUNG, 2008). Também é aplicado fortemente na identificação automática de espécies, baseando-se no som emitido por elas, como em Chu e Blumstein (2011), Trifa et al. (2008), Bardeli et al. (2010), Ganchev et al. (2007), Ventura et al. (2014) e Oliveira et al. (2014).

Em Ghahramani (2001) é comentado que essa técnica é utilizada para

compressão de dados, reconhecimento de padrões e visão computacional. Ela também foi aplicada na área da biologia, como mostrado em Eddy (2004). Como pode ser visto, a aplicação do HMM pode ser realizada nas mais diversas áreas.

2.4 Técnicas Estatísticas

Além das técnicas computacionais citadas, técnicas estatísticas tradicionais de tratamento de séries temporais também são utilizadas. Na verdade, alguns métodos da Inteligência Artificial possuem bases diretamente ligadas aos métodos estatísticos. E, como os métodos da área de IA são amplamentes utilizados para o preenchimento de falhas e detecção de *outliers*, pode ser afirmado que métodos puramente estatísticos também podem ter bons índices de precisão nessas operações.

Em Biudes et al. (2009) foram utilizados modelos de média móvel, exponencial simples e exponencial duplo para preenchimento de falhas em valores de fluxo de seiva obtidos pelo método de balanço de calor no caule, em uma mangabeira sob irrigação e uma não irrigada. Outra alternativa estatística, utilizada em Tatsch et al. (2007), é o método MI, no qual foi usada para preencher falhas em medidas do aparelho de *eddy covariance* para dados de fluxo de energia.

Estudos também foram feitos na avaliação da variabilidade estatística de séries históricas. Em Rihbane et al. (2012), por exemplo, foi realizado o preenchimento de falhas injetando a aleatoriedade para dados de fluxo de CO_2.

Em Oliveira et al. (2010) é realizada uma comparação de diversos métodos estatísticos de preenchimento de falhas em dados de precipitação pluvial anual. O método Regressão Linear Múltipla (RLM) obteve os melhores resultados neste trabalho.

Na detecção de *outliers* os métodos estatísticos também são aplicados. Em Schiffler (1988), por exemplo, é aplicado o método *z-score* para determinar os dados que diferem significativamente dos outros em uma mesma série de dados.

2.5 Conclusão

Neste capítulo foram abordados aspectos relacionados com a mineração de dados ambientais para a interpretação de fenômenos climáticos, tendo como principal enfoque procedimentos e técnicas da etapa de preparação dos dados. Essa etapa consiste em tarefas para permitir que a análise dos dados possa ser feita sem problemas, assegurando a qualidade dos dados.

Para tanto, métodos para detectar dados inválidos e preencher valores ausentes devem ser utilizados. Diversos métodos já foram aplicados em trabalhos que manipulavam dados meteorológicos. Mas é importante ressaltar que em praticamente todos os trabalhos citados, as técnicas são aplicadas de forma totalmente dependente do tipo de dado utilizado, ou seja, a aplicação se torna bastante específica a fim de tratar uma determinada variável climática, sendo necessário tempo e um conhecimento profundo do domínio sobre os respectivos dados, para atingir as necessidades do mesmo.

Foi comentado as principais características dos dados meteorológicos para o tratamento de falhas, a saber: multivariados, não lineares, grande quantidade e temporais. Para uma melhor eficácia no tratamento desses dados, uma técnica deve suportar tais características, ou seja, utilizar das variáveis inter-relacionadas, tratar não linearidade, ser escalável e aproveitar da temporalidade dos dados. Na Tabela 2 é apresentado um resumo das características dos métodos comentados neste capítulo que serão aplicados na correção de dados meteorológicos.

Tabela 2: Comparativo das características entre as técnicas selecionadas.

	Características			
Técnica	Multivariado	Não Linear	Escalabilidade	Temporal
Média Móvel			✓	
Regressão Linear Múltipla	✓		✓	✓
Z-Score			✓	
Redes Neurais Artificiais	✓	✓		✓
Hidden Markov Model	✓	✓		✓
Algoritmos Genéticos	✓	✓		✓

De forma geral, há a dificuldade de uma técnica suportar todas as características desejadas. Média Móvel, RLM e Z-Score tem um rápido desempenho mesmo com grandes quantidades de dados, mas nenhuma suporta não linearidade e duas delas não aproveitam das características dos dados multivariados e temporais. Por outro lado, RNA, HMM e AG demandam mais processamento quando há mais dados envolvidos devido suas fases de treinamento. Contudo, mesmo as técnicas não possuindo todas as características desejadas, os trabalhos citados mostram que as técnicas da Tabela 2 podem atingir bons resultados com dados meteorológicos. Mais detalhes sobre a escolha e uso dos métodos poderão ser vistos no Capítulo 3.

O trabalho proposto incorpora diversas técnicas, tanto computacionais quanto da área da estatística, em um único produto. Com isso, é esperado bons resultados no preenchimento de falhas e na detecção de *outliers* em dados meteorológicos, aproveitando das características das técnicas selecionadas.

Capítulo 3

Materiais e Métodos

Os dados meteorológicos possuem diversas características que dificultam a deteção de dados inválidos, como os *outliers*, e o preenchimento de falhas. Por outro lado, diversos trabalhos demonstram a aplicação de métodos para realizar a deteção de *outliers* e o preenchimento de falhas. Entretanto, a aplicação desses métodos exigem conhecimentos específicos que nem sempre o pesquisador que deseja corrigir os seus dados possui.

Visando facilitar esse processo, este trabalho propõe um ambiente que viabiliza a aplicação de diversos métodos de tratamento de dados meteorológicos. Os métodos selecionados são comentados neste capítulo, tanto de preenchimento de falhas quanto de detecção de *outliers*, assim como a forma que eles foram avaliados.

3.1 Métodos Selecionados

Três métodos foram selecionados para realizar o preenchimento de falhas. Dois deles já estão definidos na literatura: Regressão Linear Múltipla e Média Móvel. O terceiro método, de desenvolvimento próprio, utiliza uma combinação de duas técnicas de IA (Redes Neurais Artificiais e Algoritmos Genéticos).

Para a detecção de *outliers* foram selecionados três métodos. Assim como no preenchimento de falhas, também foi criado um método que utiliza da combinação entre RNA e AG, mas desta vez para encontrar possíveis *outliers* nas séries de dados. Outra técnica computacional, o *Hidden Markov Model*, foi utilizada na construção de um novo método para detectar *outliers*. Por fim, o terceiro método para detectar *outliers*, chamado Z-Score, pode ser encontrado na literatura.

Todos os métodos selecionados foram incluídos em um mesmo ambiente.

O ambiente foi desenvolvido com a linguagem de programação Java. Esta linguagem é Orientada a Objetos (OO), consegue realizar processamento paralelo (*multi-threaded*) e tem a característica de ser portável, o que significa dizer que consegue ser executada nos principais sistemas operacionais, como Windows, Linux e MacOS. Estas características fazem do Java uma boa escolha para a implementação deste ambiente.

Os métodos retirados da literatura (Regressão Linear Múltipla, Média Móvel e Z-Score) são descritos nas próximas seções. O método criado para preencher falhas e os dois novos métodos para detectar *outliers* são descritos nas Seções 4.1.1, 4.1.2 e 4.1.3.

3.1.1 Regressão Linear Múltipla

Métodos de regressão são utilizados em diversos trabalhos de modelagem. A análise de regressão é realizada de forma a determinar as correlações entre duas ou mais variáveis que mantenham relações de causa-efeito, realizando previsões utilizando a relação (UYANıK; GÜLER, 2013). Esses tipos de métodos possibilitam que sejam estimados valores baseados em outras variáveis, o que combina com as características dos dados meteorológicos.

No caso univariado, também chamado de regressão linear simples (RLS), modelos de regressão são aqueles modelos que são limitados a uma única variável (HAASE, 2011). A Eq. (1) apresenta a RLS.

$$Y_i = \alpha + \beta X_i \tag{1}$$

No qual Y_i é a variável dependente, α é o coeficiente linear, β é o coeficiente angular e X_i a variável independente.

Algumas variáveis meteorológicas podem ser estimadas baseando-se em apenas uma variável meteorológica. Entretanto, normalmente, é necessário a relação com mais de uma variável para conseguir uma estimativa aceitável. Para esses casos pode ser utilizada a regressão linear múltipla (RLM).

Os mesmos conceitos analíticos podem ser aplicados para os casos em que mais de uma variável dependente deve ser analisada simultaneamente (HAASE, 2011). Para tanto, os valores de cada variável devem ser levados em consideração. A Eq. (2) apresenta a RLM.

$$P_x = a_0 + \sum_{i=1}^{n} a_i P_i \tag{2}$$

No qual P_x é a variável dependente, a_0 e a_i são coeficientes do modelo linear, n é o número de variáveis independentes e P_i são as variáveis independentes.

Para o preenchimento de falhas em dados meteorológicos, P_x seria a falha a ser preenchida e P_i são os valores das variáveis meteorológicas disponíveis para serem utilizadas no cálculo do preenchimento de falhas. Os valores de a_0 e a_i são calculados resolvendo a regressão para cada série de dados.

Esse método pode ser utilizado também quando estiverem disponíveis estações meteorológicas próximas à estação onde ocorreu a falha. Com isso, as variáveis auxiliares (P_i) seriam do mesmo tipo da falha. Por exemplo, se for necessário o preenchimento de falhas em precipitação da estação meteorológica A, seriam utilizados para P_i os valores de precipitação coletados nas estações B, C e D, caso elas estejam próximas à estação A.

Para estimar os valores de a_0 e a_i foi utilizado o Código 10 do Anexo I, desenvolvido por Sedgewick e Wayne (2014).

Com esse código, é possível obter os valores de a_0 e a_i acessando o vetor armazenado na variável *beta*. Então, para calcular o valor a ser preenchido da falha, é multiplicado cada dado com seu respectivo coeficiente ($a_i = beta[i]$), somando os resultados.

3.1.2 Média Móvel

Há uma facilidade na aplicação do método de média para preenchimento de falhas devido a sua simplicidade. Nesses casos, a Eq. (3) é utilizada, onde a média entre o valor anterior e posterior à falha é utilizado para preencher a falha.

$$x_i = \frac{x_{i-1} + x_{i+1}}{2} \tag{3}$$

Uma média simples tem poucos dados para realizar um preenchimento de falhas. Logo, a sua precisão pode ser baixa. Para tentar obter uma precisão um pouco maior é utilizado a Média Móvel (MM). A MM é calculada adicionando os n dados mais recentes e dividindo-os por n (GENÇAY, 1996), conforme a Eq. (4).

$$x_i = \frac{x_{i-1} + x_{i-2} + x_{i-3} + ... + x_{i-n}}{n} \tag{4}$$

Para o preenchimento de falhas em dados meteorológicos, além dos dados anteriores à falha, é possível utilizar também os dados posteriores à falha. Dessa forma, mais dados são agregados à estimativa, possibilitando uma precisão ainda

maior. A Eq. (5) apresenta a MM utilizando os dados anteriores e posteriores à falha.

$$x_i = \frac{\sum_{k=i-\frac{n}{2}}^{i-1} x_k + \sum_{k=i+1}^{i+\frac{n}{2}} x_k}{n} \tag{5}$$

Se for determinado que dez elementos serão utilizados para o cálculo de preenchimento de falhas, cinco valores anteriores à falha e cinco valores posteriores à falha serão utilizados para calcular uma média e, assim, preencher a falha.

Caso haja uma outra falha entre os elementos anteriores ou posteriores à falha que está sendo tratada no momento, esse valor não será considerado e apenas $n - 1$ dados serão utilizados para computar a média.

3.1.3 Z-Score

O método *Z-Score* não necessita dos valores de outras variáveis para realizar as estimativas, ou seja, esse método leva em consideração apenas o valor da própria variável que está sendo tratada.

Este método irá atribuir uma pontuação para cada dado avaliado. Quanto mais esta pontuação distancia de zero, maior a probabilidade desse dado ser um *outlier*. Esta pontuação é calculada por meio da Eq. (6).

$$z_i = \frac{x_i - \bar{x}}{s} \tag{6}$$

Então, para calcular a pontuação de cada dado, primeiramente é necessário calcular a média ($\bar{x}$) e o desvio padrão (s) da variável que está sendo tratada. Depois, é calculado a diferença entre o dado (x_i) e a média, e depois dividido pelo desvio padrão. O resultado da equação (z_i) é a pontuação que irá sugerir se aquele dado é um *outlier* ou não.

Após calcular a pontuação para todos os dados da variável que está sendo tratada, é possível ordená-los de acordo com sua pontuação. Para tanto, o módulo da pontuação é utilizado.

Nesse método é possível escolher que sejam indicados como *outliers* os N dados com maiores pontuações ou atribuir uma pontuação limite, no qual os dados com pontuações maiores que o limite serão considerados como *outliers*.

Pela característica do método, ele terá uma melhor precisão principalmente se a variável tratada tiver uma distribuição normal.

3.2 Testes de Avaliação

Para avaliar os métodos criados e implementados da literatura, diferentes séries de dados foram selecionadas, envolvendo um número variado de variáveis climáticas. Desta forma, é possível verificar se os métodos terão bom desempenho para diferentes variáveis, além de determinar qual método tem o melhor desempenho para cada variável.

Os testes foram executados em um computador HP Compaq Elite 8300 All-in-One, com processador intel i5 3470 de 3.2Ghz de 4 núcleos e com 4 GB DDR3 de memória RAM. O sistema operacional instalado é um Linux de distribuição Ubuntu 14.04.

Nas próximas seções estão descritos em detalhes os dados utilizados, como foram feitas as simulações de falhas e *outliers* nas séries de dados e como a análise estatística foi realizada.

3.2.1 Dados Utilizados

Os dados utilizados neste trabalho foram obtidos de três fontes de dados diferentes: Instituto Nacional de Meteorologia (INMET, 2014), satélite *Tropical Rainfall Measuring Mission* (TRMM) (TRMM, 2014) e Ameriflux (AMERIFLUX, 2015).

3.2.1.1 Dados do INMET

Cinco séries de dados foram obtidas do INMET, cada uma representando uma estação meteorológica automática. As estações foram selecionadas de Estados diferentes, visando analisar dados meteorológicos com características diferentes. Na Tabela 3 são mostradas as informações das estações selecionadas e no Anexo II há um exemplo dos dados dessa fonte.

Tabela 3: Informações das cinco estações meteorológicas do INMET selecionadas.

Estação	Cidade	Latitude	Longitude	Altitude (m)
1	Goiás - GO	-15.939729	-50.141433	513
2	Campina Verde - MG	-19.539210	-49.518133	559
3	Sorriso - MT	-12.555107	-55.722863	367
4	Diamante do Norte - PR	-22.639366	-52.890156	368
5	Campo Bom - RS	-29.674293	-51.064042	23

Todas as séries de dados dessa fonte de dados possuem três meses de observações, coletadas no período de novembro de 2014 a janeiro de 2015. A

leitura dessas séries de dados foi realizada de hora em hora, para cada uma das seis variáveis climáticas existentes: temperatura, umidade relativa do ar, ponto de orvalho, pressão atmosférica, radiação solar e velocidade do vento. A Tabela 4 mostra a unidade e o desvio padrão de cada variável climática.

Tabela 4: Unidade e desvio padrão das variáveis climáticas presentes nas séries de dados do INMET.

Variável	Unidade	σ
Temperatura	°C	3,93
Umidade relativa do ar	%	17,11
Ponto de orvalho	°C	2,10
Pressão atmosférica	hPa	2,4
Radiação solar	kJ/m^2	1187,65
Velocidade do vento	m/s	1,17

3.2.1.2 Dados do TRMM

Em Silva e Rocha (2013) é mostrado que dados estimados pelo satélite TRMM podem ser utilizados como uma ferramenta para auxiliar na obtenção de dados pluviométricos. Os dados do TRMM são separados por pontos, com uma resolução de 0,25 °, fornecendo dados diários de chuva (em mm) para esses pontos. A Tabela 5 mostra 10 pontos selecionados, sendo que cada ponto está distante aproximadamente 25 km entre um e outro, todos próximos à Cuiabá-MT.

Tabela 5: Informações dos dez pontos selecionados do TRMM.

Ponto	Latitude	Longitude	σ
p1	-15,375	-56,375	3,87
p2	-15,875	-56,125	4,00
p3	-15,875	-56,375	3,71
p4	-15,375	-56,125	3,39
p5	-15,125	-56,375	4,46
p6	-15,875	-56,625	4,80
p7	-15,375	-56,625	4,16
p8	-15,625	-56,125	3,56
p9	-15,625	-56,375	3,77
p10	-15,625	-56,625	3,96

Foram coletados 3 meses de dados, de novembro de 2010 a janeiro de 2011. Com isso, uma única série de dados foi elaborada contendo 10 atributos, representando os pontos selecionados, no qual cada registro apresenta a quantidade de chuva em um mesmo dia para cada ponto. No Anexo III há um exemplo dos dados dessa fonte.

3.2.1.3 Dados do Ameriflux

O AmeriFlux fornece medições contínuas de florestas, pastagens, áreas úmidas e terras agrícolas da América do Norte, América Central e América do Sul (BODEN et al., 2013). Para esse trabalho, os dados da estação *LBA Tapajos KM67 Mature Forest* foram utilizados. Essa estação está localizada na latitude -2,8566 e longitude -54,9589, com elevação de $88m$.

Os dados utilizados dessa estação foram coletados no ano de 2003, de 1° de janeiro à 31 de dezembro, de hora em hora. Ao todo são 23 atributos, armazenado informações sobre quando o dado foi coletado, concentrações de CO_2, radiação, condições do vento, temperatura, umidade, dentre outras informações. No Anexo IV há um exemplo dos dados dessa fonte.

As variáveis climáticas de temperatura do ar e umidade relativa do ar serão testadas com duas fontes de dados diferentes. Uma será esta fonte de dados (Ameriflux) contendo 23 atributos e com desvio padrão de 1,97 para temperatura e 1363,85 para umidade, e a segunda base é proveniente do INMET, com apenas 6 atributos, detalhada na Seção 3.2.1.1.

3.2.2 Simulações de Falhas e *Outliers*

Os métodos de preenchimento de falhas devem estimar um valor para preencher as falhas simuladas. Os valores estimados foram comparados com os valores reais e, desta forma, foi calculado a precisão de cada método de preenchimento de falhas. Para cada variável climática testada, 5%, 15%, 30% e 50% dos dados foram aleatoriamente removidos, causando falhas. Com essas proporções houve inclusive ocorrências de falhas em sequência, o que para alguns métodos pode ser um cenário difícil de tratar.

Para os métodos de detecção de *outliers*, 2% e 5% dos dados de cada variável climática testada foram aleatoriamente modificados. Essas modificações visam tornar os dados comuns em *outliers*. Os valores foram modificados em 30% e 50% para mais ou para menos. Os métodos de detecção de *outliers* devem identificar os dados modificados, provando a sua eficácia.

Para a base TRMM não houve testes com 2% de *outliers* devido a pouca quantidade de dados nas séries originadas dessa base (92 registros). Então os dados de precipitação foram testados com apenas dois cenários diferentes ao invés de quatro como as outras variáveis climáticas.

3.2.3 Avaliação Estatística

A avaliação estatística de cada método de preenchimento de falhas foi baseada nos erros individuais (e_i) de cada estimativa, mostrado na Eq. (7), onde $P_i(i = 1, 2, ..., n)$ são os valores estimados e $O_i(i = 1, 2, ..., n)$ são os valores reais (WILLMOTT; MATSUURA, 2005).

$$e_i = P_i - O_i \tag{7}$$

Com o erro individual de cada estimativa, é calculado o desempenho do modelo usando o Erro Médio Absoluto (EMA), mostrado na Eq. (8). Segundo Willmott et al. (2009), essa é a melhor forma de avaliação para modelos ambientais, devido principalmente a presença de *outliers* e dados com desvio de normalidade.

$$EMA = \frac{\sum_{i=1}^{i=n} |e|}{n} \tag{8}$$

Além disso, também foi calculado para os testes de preenchimento de falhas o coeficiente de correlação de Pearson (PEARSON, 1896), mostrado na Eq. (9).

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{9}$$

Para os métodos de detecção de *outliers*, os dados identificados como *outliers* foram comparados com os dados que realmente foram modificados, os que simulam os *outliers*. Com estas informações foi possível obter a precisão e a área abaixo da curva ROC de cada método de detecção de *outliers*.

A precisão é calculada conforme a Eq. (10), no qual são utilizados a quantidade de *outliers* detectados corretamente (*true positive* - TP) e a quantidade de dados indicados como *outliers* mas que na verdade eram dados comuns (*false positive* - FP).

$$precisão = \frac{TP}{TP + FP} \tag{10}$$

Os gráficos *Receiver Operating Characteristics* (ROC) são úteis para visualizar o desempenho de classificadores (FAWCETT, 2006). Esse tipo de gráfico leva em consideração a taxa de TP (eixo y) e a taxa de FP (eixo x). A Figura 5 mostra um exemplo de um gráfico ROC.

Baseando-se no gráfico ROC, é possível obter um valor escalar calculando a área abaixo da curva (do inglês, *Area Under Curve* - AUC). Com isso, é possível

obter um valor, de 0 a 1, que representa a performance do classificador. Na Figura 5, por exemplo, a área abaixo da curva de A tem um valor maior que a área abaixo da curva de B.

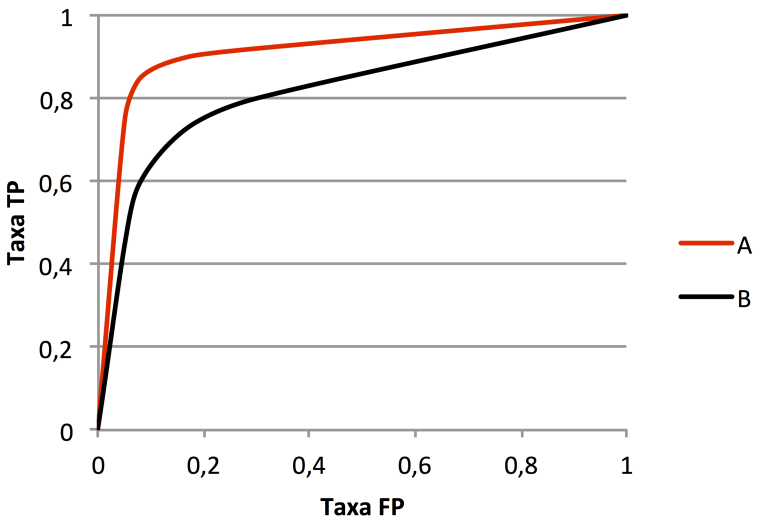

Figura 5: Exemplo de um gráfico ROC. É possível interpretar que houve um melhor desempenho do classificador A em relação ao classificador B, uma vez que as taxas de TP foram maiores para A do que para B.

3.3 Conclusão

No desenvolvimento do ambiente para tratamento de dados meteorológicos, seis métodos foram selecionados: três para preenchimento de falhas e três para detecção de *outliers*.

Dos seis métodos selecionados, três foram criados e desenvolvidos com técnicas da área de inteligência artificial. Os outros três métodos são puramente estatísticos, e já estão publicados na literatura. A adição desses métodos no ambiente visa facilitar a aplicação dos mesmos sem ter a necessidade de conhecimentos profundos da área de estatística.

Dependendo dos dados a serem tratados e de suas características, um método pode obter uma maior precisão do que outro. Tendo três métodos à disposição para cada tipo de operação, espera-se que pelo menos um deles esteja apto a atender às especificidades da série de dados que está sendo tratada.

Por fim, para avaliar o comportamento de cada método, diversos testes foram planejados. Os resultados desses testes, assim como a forma de utilizar os métodos, são apresentados no capítulo 4.

Capítulo 4

Apresentação e Análise dos Resultados

Para que os novos métodos, além dos selecionados da literatura, possam ser utilizados, os mesmos devem ser programados. Para tanto, neste trabalho foi desenvolvido um ambiente que tem como objetivo agregar métodos de tratamento de dados ambientais.

Além dos detalhamentos dos três métodos criados, nas próximas seções são apresentadas a arquitetura desenvolvida do ambiente, os resultados dos testes para avaliar os métodos selecionados, a análise estatística dos resultados obtidos e um teste da integração entre o *framework* criado e um novo sistema para tratamento de dados ambientais.

4.1 Criação de Novos Métodos

Três novos métodos foram criados, um para preenchimento de falhas e dois para detecção de *outliers*. Foram utilizadas três técnicas diferentes para criar os três métodos de tratamento de dados ambientais: Redes Neurais Artificiais, Algoritmos Genéticos e *Hidden Markov Model*. Dois dos novos métodos seguem uma mesma metodologia, integrando Redes Neurais Artificiais com os Algoritmos Genéticos, enquanto o terceiro método utiliza unicamente o *Hidden Markov Model*.

4.1.1 MANNGA para Preenchimento de Falhas

Este método, denominado *Method with Artificial Neural Network and Genetic Algorithm* (MANNGA), foi descrito inicialmente em Ventura (2012), e

realiza o cálculo para preenchimento de falhas levando 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, são considerados os valores mensurados da temperatura, radiação global e velocidade do vento, naquele mesmo instante.

Para tanto, a partir dos dados coletados pela estação meteorológica, são identificadas quais variáveis climáticas estão relacionadas com o sensor que falhou em sua leitura e, posteriormente, é calculado o valor ausente, com base nessas variáveis climáticas. Na Figura 6 estão ilustrados os procedimentos realizados pelo método.

Primeiramente, uma quantidade pré-determinada de dados sem falhas são escolhidos e utilizados pelo AG com o intuito de fazer com que o sistema entenda os padrões dos dados. Assim, o AG pode determinar a melhor configuração para a RNA que preencherá as falhas. Durante o processamento do AG, são criadas várias RNAs de acordo com as configurações armazenadas nos indivíduos gerados pelo próprio AG.

Cada RNA é avaliada utilizando os dados inseridos no AG para verificar se tal configuração tem potencial 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 é 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, que foi criada de acordo com as configurações do indivíduo mais apto, é 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 são substituídos por valores estimados pela RNA, preenchendo todas as falhas.

Todos esses processos podem ser divididos em quatro etapas. Nas seções a seguir serão apresentados os detalhes de cada fase deste método: preparação dos dados, determinação da configuração da RNA, treinamento da RNA e o preenchimento de falhas.

4.1.1.1 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

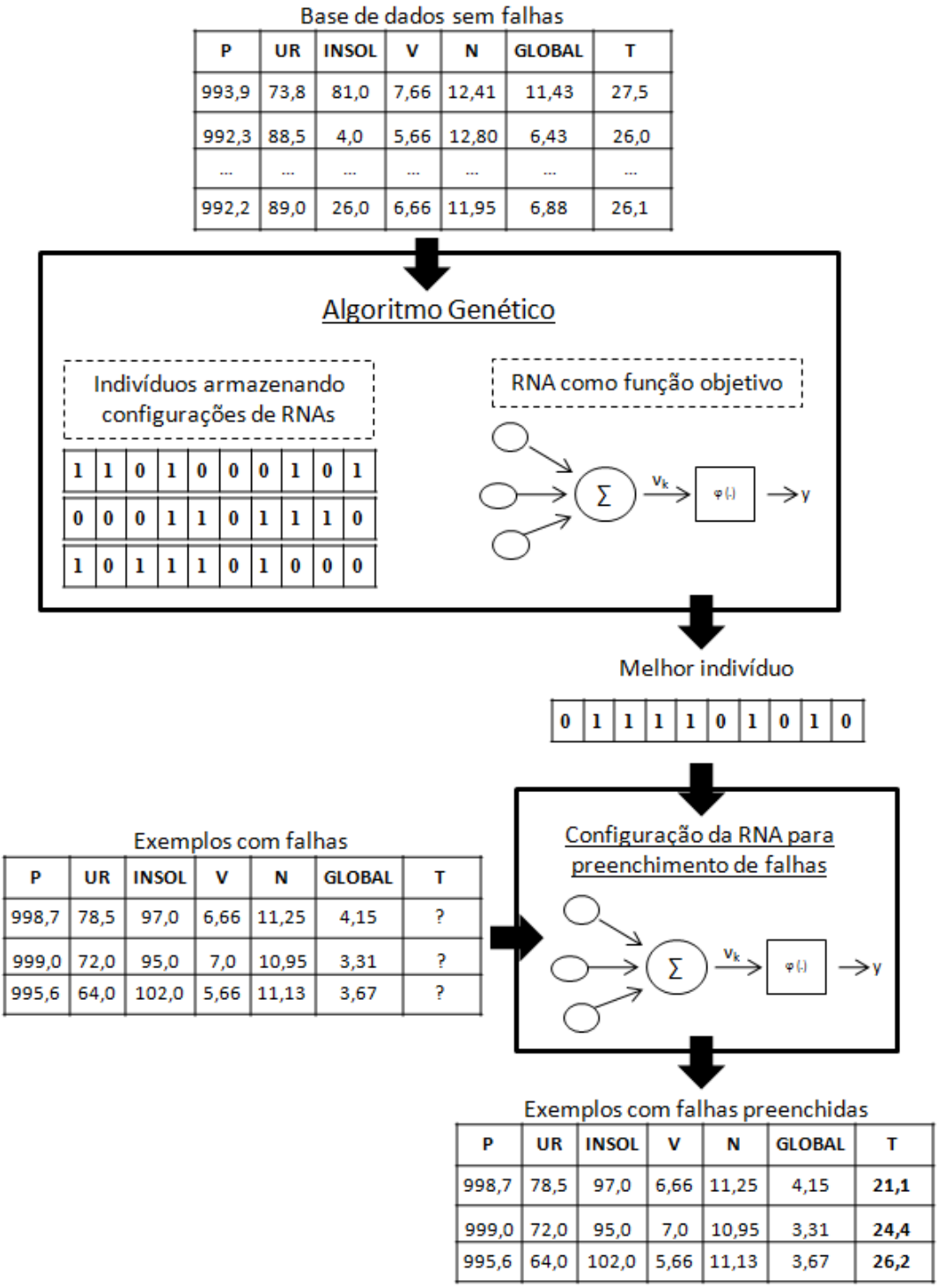

Figura 6: Diagrama da sequência de passos necessários para realizar o preenchimento de falhas com o MANNGA (VENTURA, 2012).

demanda muito esforço, 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.

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 serve para verificar se a RNA aprendeu corretamente os padrões das séries de dados, validando sua configuração.

4.1.1.2 Determinando a Configuração da RNA com AG

Nessa etapa é determinada a configuração da RNA responsável pelo preenchimento de valores em dados com falhas. O algoritmo genético é utilizado para determinar quais sensores são 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. Como visto na Seção 2.2.1 e 2.2.2, os métodos de detecção de *outliers* e de preenchimento de falhas podem sofrer com a maldição da dimensionalidade, então 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 tenham 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 realiza o teste de vários valores diferentes com a intenção de obter a melhor combinação dos parâmetros de acordo com os dados utilizados no treinamento. 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.

Neste método, os indivíduos do AG foram codificados por no mínimo 12 *bits*, no qual representam funções de ativação da camada oculta e da camada de saída (4 *bits*), algoritmo de treinamento (3 *bits*), taxa de aprendizagem (3 *bits*), taxa de *momentum* (2 *bits*). Além disso, 1 *bit* é adicionado para cada variável climática existente para dizer se a respectiva variável é ou não utilizada para realizar o cálculo de preenchimento de falhas.

A função objetivo do AG é calculada com base na criação de uma RNA utilizando os parâmetros definidos no indivíduo a ser avaliado. Essa RNA é treinada e seu desempenho é avaliado. Quanto melhor o desempenho da RNA, mais apto é o indivíduo avaliado do AG.

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 RNA, que poderá preencher as falhas encontradas.

4.1.1.3 Treinamento da RNA

A RNA é treinada de acordo com a configuração obtida pelo AG. Logo, é 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 Teorema de Kolmogorov-Nielsen, apresentato 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*".*

A RNA então tem todos os parâmetros configurados e, assim, é possível executá-la iniciando o treinamento. A RNA tem como entrada o conjunto de dados separados para o treinamento e, depois de vários ciclos de processamento, o resultado da RNA identifica o quão apta está a estrutura para resolver o problema proposto de preenchimento de falha.

4.1.1.4 Preenchimento dos Valores Ausentes

A RNA treinada na etapa anterior pode agora ser utilizada para realizar o preenchimento de falhas. Falha a falha, ela é utilizada recebendo como dados de entrada os valores obtidos pelos sensores no mesmo momento que houve a respectiva falha. Com base nos valores informados, a RNA consegue estimar o valor que deveria ser armazenado quando houve a falha, preenchendo o valor ausente.

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, por exemplo 10 leituras seguidas de falhas, o método conseguiria tratar esse problema. Esse é um ponto que se diferencia de outros métodos, que normalmente utilizam dados próximos à falha para substituir o valor inválido, impossibilitando que tenha uma boa precisão em falhas sequenciais.

4.1.2 MANNGA para Detecção de *Outliers*

Da mesma forma que o MANNGA para preenchimento de falhas, o MANNGA para detectar *outliers* também utiliza técnicas de Redes Neurais Artificiais e Algoritmos Genéticos. As RNAs têm sido utilizadas com sucesso para modelar relações envolvendo séries temporais complexas (HAYKIN, 2001) e a maior vantagem delas sobre os métodos convencionais é que elas não requerem informação detalhada sobre os processos físicos do sistema a ser modelado, sendo este descrito explicitamente na forma matemática (SUDHEER et al., 2003). Uma dificuldade é a definição da arquitetura da RNA, mas isso pode ser solicionado utilizando o AG, como já foi visto na Seção 4.1.1.2.

Aproveitando dessas características, neste novo método a RNA tem o objetivo de estimar um determinado valor de uma variável climática específica,

baseando-se nos valores das outras variáveis climáticas. Por exemplo, a RNA pode estimar o valor da temperatura do ar, às 10:00 horas, usando valores das outras variáveis climáticas coletadas no mesmo horário.

Até este momento, pouco se difere do método de preenchimento de falhas detalhado anteriormente que utilizam as mesmas técnicas de IA. Mas, neste método de detecção de *outliers*, esta ação de estimativa de valores pode ser executada para todos os momentos da base de dados. Desta forma, é possível obter um valor estimado para cada leitura da variável que está sendo tratada.

Considerando que os valores estimados são bem próximos aos valores reais lidos pelos sensores, devido ao desempenho da RNA e da forte relação entre as variáveis climáticas, a detecção de *outliers* neste método consiste em comparar os valores estimados com os valores reais lidos pelos sensores, sendo que quanto maior a diferença entre esses valores, maior a probabilidade do dado ser um possível *outlier*.

Em resumo, este método utiliza uma RNA, cujos parâmetros da arquitetura utilizou AG, para estimar valores de variáveis climáticas e, posteriormente, compará-las com os valores reais, verificando sua divergência com relação ao restante da série de dados. A diferença entre o dado obtido pelos sensores e o valor estimado pela RNA é a distância entre os dados. Quanto maior a distância, maior a possibilidade daquele dado ser um *outlier*.

A Eq. (11) apresenta a equação de distância para este método.

$$d = \frac{|r - e|}{r} \tag{11}$$

Onde d é a distância calculada, r é o valor do dado real, ou seja, o valor armazenado pelos sensores, e e é o valor estimado pelo método. Utilizando a Eq. (11) para todos os registros de uma variável, é possível estabelecer quais são os registros com maior possibilidade de ser um *outlier*, auxiliando na detecção dos mesmos.

4.1.3 ODHiMM

O método criado para detectar *outliers* utilizando a técnica *Hidden Markov Model*, denominado *Outlier Detection with Hidden Markov Model* (ODHiMM), é dividido em três etapas: preparação dos dados, treinamento dos modelos HMM e classificação dos dados.

Na preparação dos dados, conforme mostrado na Figura 7, são criados mais dois conjuntos de dados baseados na série original. A diferença entre os novos

conjuntos e a série original está no valor da variável que está sendo tratada. No caso da Figura 7, é o dado de temperatura (T).

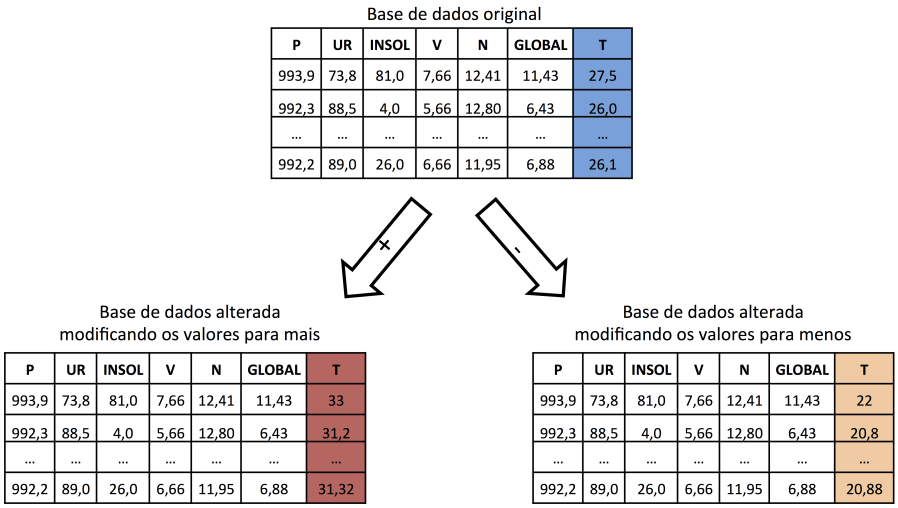

Figura 7: Preparação para treinamento dos três modelos: dados normais, *outliers* com valores superiores ao normal e *outliers* com valores inferiores ao normal.

No primeiro novo conjunto de dados, todos os valores de temperatura são acrescidos de acordo com uma porcentagem pré-determinada. Já no segundo conjunto, os valores são decrescidos com a mesma porcentagem. Com isso, o primeiro conjunto (original) demonstra o comportamento normal da variável. O segundo e o terceiro conjunto (dados modificados para mais e para menos, respectivamente) apresentam o comportamento da variável quando é registrado como um *outlier*.

O valor da porcentagem de modificação (no exemplo da Figura 7, 20%) pode ser configurado ao executar o método, dependendo da característica do dado que está havendo a detecção de *outlier*. Também pode ser configurado a quantidade de dados a ser treinado, podendo ser apenas uma parcela da série de dados ou sua totalidade, afetando diretamente o tempo de processamento mas possibilitando um melhor treinamento.

Depois da separação dos dados, é possível realizar a etapa de treinamento. Três modelos de HMM são criados, representando cada um dos conjuntos: um modelo para os dados normais e dois modelos para os *outliers*. O algoritmo de treinamento *Baum–Welch* é executado para que os modelos se adequem aos conjuntos de dados que foram atribuídos a cada um deles, como mostrado na Figura 8.

Várias iterações são realizadas para que os modelos aprendam o comportamento em cada cenário, sendo que a quantidade de iterações pode ser con-

figurada antes de executar o método. O número de iterações, assim como a quantidade de dados a ser treinado, influencia na qualidade do treinamento e no tempo de processamento. Logo, poderá ser testado valores diferentes para tentar encontrar uma configuração balanceada.

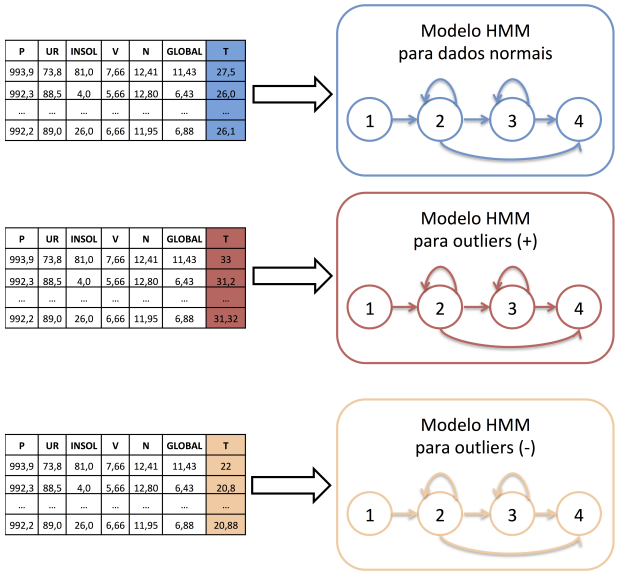

Figura 8: Treinamento dos três modelos criados no método ODHiMM.

Depois das iterações de treinamento os modelos estão aptos para realizar a terceira e última etapa do método: classificação dos dados. A classificação identifica quando um registro está em sua forma normal ou como um *outlier*, ilustrado na Figura 9.

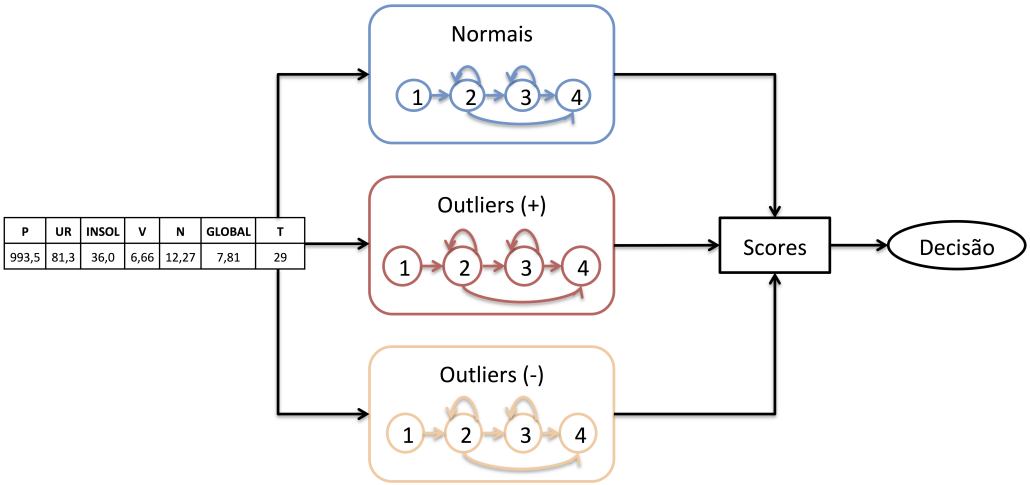

Figura 9: Forma de realizar a classificação de um registro da base de dados em dado normal ou *outlier*.

Quando um novo registro deve ser analisado, os valores das variáveis são transmitidas para os três modelos. Cada modelo verifica o registro e atribuir uma

pontuação (*score*) proporcional a probabilidade desse registro pertencer àquele modelo. Após a avaliação dos modelos, uma decisão é feita baseado nos resultados gerados, classificando o registro como um dado comum ou como um *outlier*.

4.2 Arquitetura do Ambiente

Para que as funcionalidades desejadas do ambiente pudessem ser implementadas, foi planejada uma arquitetura que contemplasse o requisito de facilidade de uso e de ser expansível, aceitando novos métodos no futuro.

Primeiramente, uma API (*Application Programming Interface* - Interface de Programação de Aplicações) foi criada para definir um conjunto de classes, funções e regras para a organização do ambiente de tratamento de dados meteorológicos.

Com a API criada, métodos foram implementados e agregados em um único *framework*, possibilitando que novos sistemas utilizem dessas funcionalidades para permitir que operações de tratamento de dados possam ser realizadas.

Na Figura 10 é mostrado o ambiente para a realização de tratamento dos dados, ilustrando a interação da API com os métodos já implementados e as aplicações que utilizam destas funcionalidades.

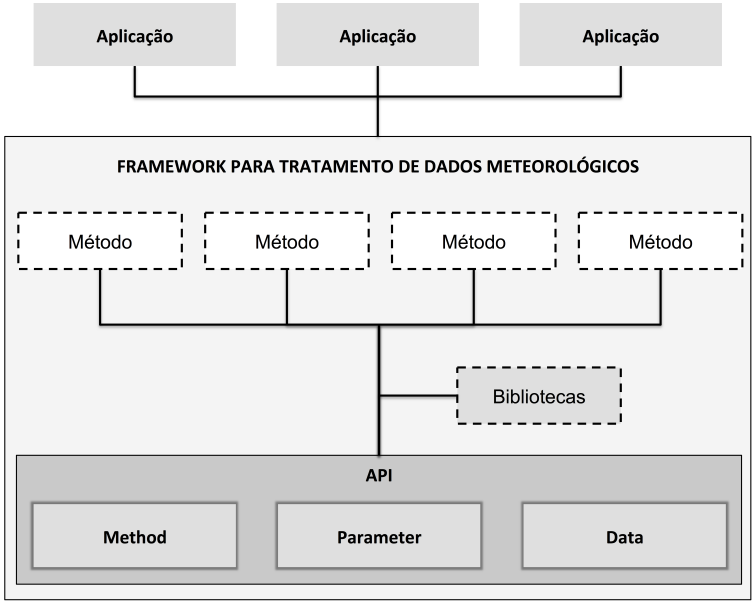

Figura 10: Visão geral do ambiente de tratamento de dados meteorológicos.

A API contém padrões, como o "Method", "Parameter" e "Data", que os métodos implementados no *framework* devem respeitar. Desta forma, são estabelecidas regras para que todos os métodos funcionem da mesma forma, facilitando

o entendimento e uso dos mesmos.

Em “Method”, são elencados as funções que um método de tratamento deve ter, como funcionalidades para selecionar os dados, realizar o treinamento e processar os dados. Em “Parameter”, itens do método que podem ser configurados estão definidos nessa entidade, como a quantidade de dados que serão utilizados nos testes ou número de iterações na etapa de treinamento. Já na entidade “Data”, é determinado como os dados podem ser carregados para que o *framework* consiga trabalhar com eles.

Alguns métodos podem utilizar de bibliotecas para desempenhar suas funcionalidades. Estão incorporados no *framework* as bibliotecas de Encog (2014), com algoritmos avançados de aprendizado de máquina, e de Francois (2014), implementando a técnica de *Hidden Markov Model*.

Uma vantagem dessa arquitetura é a facilidade de adicionar novos métodos para tratamento de dados meteorológicos. O único requisito são as regras que a API estabelece para a implementação do método, ou seja, um novo método deve aceitar o formato de dado já aceito pelos outros métodos, possibilitar que alguns parâmetros sejam modificados para um melhor funcionamento do método e implementar certas funções pré-determinadas para que o seu uso não seja diferente dos outros métodos já existentes.

Os seis métodos selecionados foram adicionados ao *framework*. Na próxima seção são descritos a forma que esses métodos implementaram a API e como eles devem ser utilizados.

4.3 Utilização do *Framework*

Os métodos para tratamento dos dados, como comentado na seção anterior, seguem uma API. As funções desta API devem ser utilizadas para que todo o processo de tratamento seja realizado. A Figura 11 ilustra o ciclo de vida do *framework* utilizando esta API, com o intuito de obter os dados corrigidos.

Primeiramente, deverá ser escolhido o método que se deseja utilizar para corrigir os dados, seja ele de preenchimento de falhas ou de detecção de *outliers*. Para tanto, deverá ser selecionado, dentre os métodos existentes, qual operação será executada. Depois, os parâmetros do método selecionado são configurados, a fim de assegurar uma melhor eficácia do desempenho do método com relação aos dados existentes. Os dados de treinamento são enviados e carregados no ambiente e, assim, o treinamento é iniciado para que o método possa entender os padrões contidos nos dados.

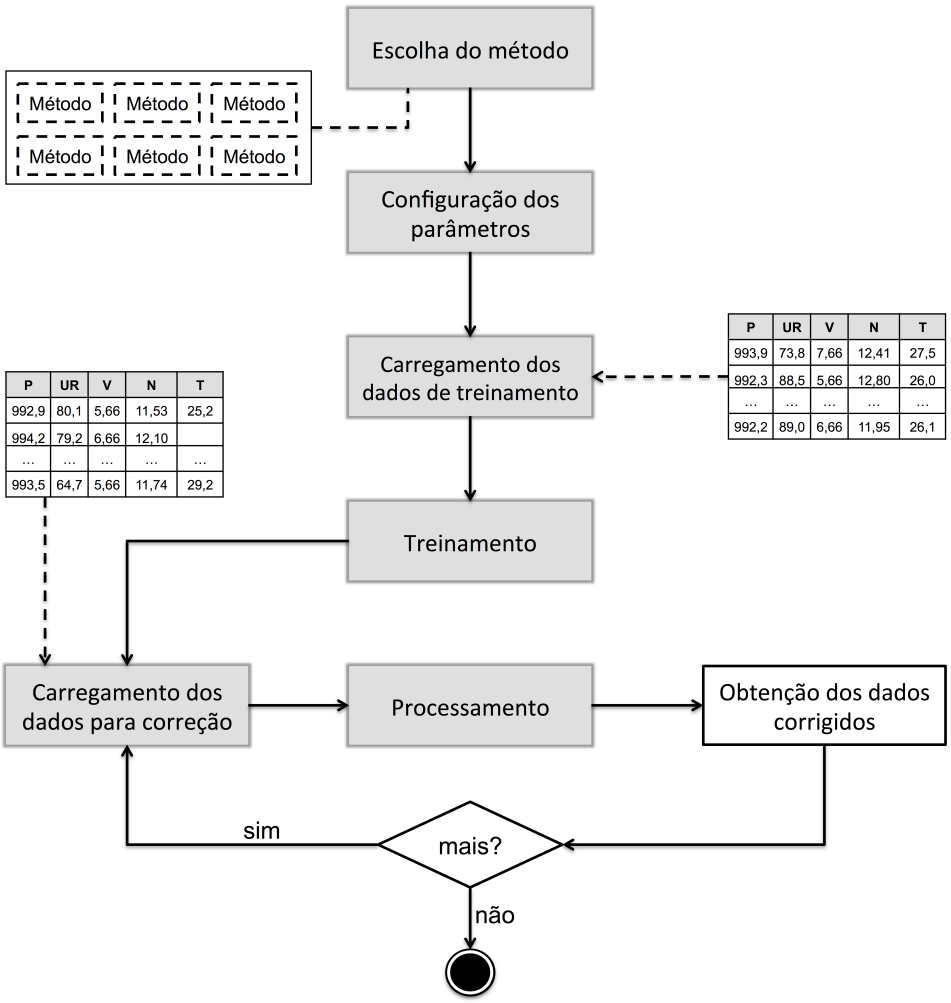

Figura 11: Ciclo de vida padrão do *framework* para realizar um tratamento nos dados.

Com a finalização do treinamento, os dados que realmente serão tratados devem ser enviados. Em seguida, os dados são processados e, depois, a solução com os dados corrigidos pode ser obtida. Como o método escolhido já está treinado, novos dados para correção podem ser enviados e corrigidos, sem a necessidade de haver um novo treinamento caso os dados a serem tratados tenham as mesmas características dos dados utilizados no treinamento.

O *framework*, como foi feito na linguagem Java, consiste em um único arquivo no formato JAR. Esse arquivo deve ser importado no projeto do sistema que está sendo desenvolvido para que as funcionalidades do *framework* possam ser executadas. Nas próximas seções são abordados em detalhes o uso do *framework*, principalmente dos comandos necessários para que cada método existente possa ser executado.

4.3.1 Manipulação dos Dados

Para todos os métodos será necessário que o *framework* receba os dados que devem ser tratados. Para isso, o *framework* está preparado para trabalhar com dois formatos diferentes de dados: matriz numérica e arquivo do tipo CSV (RFC4180, 2005).

Arquivos no formato CSV (*Comma Separated Values* - Valores Separados por Vírgula) têm sido utilizados para troca e conversão de dados entre vários programas de planilhas de dados (RFC4180, 2005). Por essa razão, é relevante que o *framework* suporte esse tipo de arquivo em suas funcionalidades.

É importante ressaltar que em todos os casos os dados devem ser numéricos do tipo *double*. Os métodos implementados não são preparados para trabalhar com tipos literais, então caso haja algum dado desse tipo, esse deverá ser convertido (codificado) para números e, depois do processamento, reconvertido (decodificado) para o seu formato original.

O Código 1 exemplifica o carregamento de duas séries de dados, cada uma com um formato diferente. Em *dados1*, será lido o arquivo *dadosmeteorologicos.csv*, no qual há 6 variáveis auxiliares e 1 variável para ser tratada (a terceira, no caso). Além disso, o valor *true* indica que o arquivo possui um cabeçalho na primeira linha.

Em *dados2*, foi criada uma matriz numérica com os dados representando os valores das variáveis (quatro variáveis com apenas 5 dados cada). É aconselhado utilizar esse formato quando os valores estão, por exemplo, em um banco de dados, tornando mais fácil a criação das matrizes numéricas. Com os dados carregados em um objeto da classe *Data*, a aplicação dos métodos pode ser realizada.

Código 1: Duas formas diferentes para carregar dados no *framework*: por arquivo ou informando os valores diretamente.

```
1 Data dados1 = new Data("dadosmeteorologicos.csv", 6, 1, 3, true);
2
3 double valores[][] = {{3.0,4.0,3.0,4.0},
4                       {5.0,6.0,2.0,3.0},
5                       {9.0,2.0,7.0,4.0},
6                       {1.0,7.0,0.0,3.0},
7                       {5.0,3.0,4.0,6.0}};
8 Data dados2 = new Data(valores, 5);
```

4.3.2 MANNGA para Preenchimento de Falhas

O método de preenchimento de falhas MANNGA possibilita que os itens responsáveis pelo comportamento de cada uma das técnicas sejam configurados a fim de tentar obter uma melhor precisão na utilização do método. No caso da RNA, é possível definir:

- Quantidade máxima de iterações em seu treinamento (épocas).
- Erro aceitável para parar o treinamento.
- Possíveis algoritmos de treinamento.
- Possíveis funções de ativação.

Para o AG, os parâmetros configuráveis são:

- Número máximo de gerações.
- Erro aceitável para parar de criar novas gerações.
- Tamanho da população de indivíduos.
- Porcentagem de mutação que os indivíduos sofrerão.

Além disso, é possível definir também a porcentagem de dados utilizados para realizar o treinamento. Para definir os parâmetros, a classe *ParametersManngа* deve ser utilizada, conforme apresentado no Código 2.

Código 2: Definição dos parâmetros no método de preenchimento de falhas MANNGA

```
1 ParametersMannga parameters = new ParametersMannga();
2 parameters.setEpoch(200);
3 parameters.setAnnError((double) 0.002);
4 parameters.setActivationFunction(ActivationFunctionsEnum.values());
5 parameters.setTrainingAlgorithm(TrainingAlgorithmsEnum.values());
6 parameters.setMaxNumberGeneration(60);
7 parameters.setErrorMaximumValid((double) 0.001);
8 parameters.setPopulationSize(50);
9 parameters.setPercentageOfMutation((float) 0.1);
```

As enumerações *ActivationFunctionsEnum* e *TrainingAlgorithmsEnum* já contém uma série de funções de ativação e algoritmos de treinamento por padrão. Além do mais, todos os parâmetros têm valores padrões, ou seja, só é necessário alterá-los caso acredite que o método funcionará melhor com os novos valores.

A classe responsável por realizar o preenchimento de falhas é a *DataGapFilling*. O Código 3 apresenta um exemplo do uso desta classe. O início da operação começa criando o objeto da classe *DataGapFilling*. Depois, os parâmetros são definidos e o treinamento é iniciado. Tanto os parâmetros quanto os dados necessários para o treinamento foram criados anteriormente. Os Códigos 1 e 2 são exemplos dessas criações.

Depois do treinamento, que dependendo da quantidade de dados pode demorar horas, é possível realizar o preenchimento das falhas. No Código 3 foi utilizado uma função que preenche todos os dados ausentes encontrados durante o treinamento. Mas há também uma outra função que é possível passar novos dados, permitindo uma separação das séries de dados entre dados de treinamento e dados de teste. Por fim, com os dados preenchidos, é possível ver o resultado por meio da classe *GapFillingResult*, além de visualizar o erro pela função *getError*.

Código 3: Utilização do método de preenchimento de falhas MANNGA

```
1 DataGapFilling gapFill = new DataGapFilling();
2 gapFill.setParameters(parameters);
3 gapFill.train(data);
4 gapFill.setTestData(testData);
5 GapFillingResult fillResult = gapFill.process();
6 float error = gapFill.getError();
```

Como pode ser visto, não é necessário conhecimentos na área de IA para utilizar esse método, mesmo utilizando internamente técnicas como a RNA e o AG. Além do mais, com poucas linhas de código é possível realizar uma operação complexa como a de preenchimento de falhas.

4.3.3 Regressão Linear Múltipla para Preenchimento de Falhas

Para o método de preenchimento de falhas utilizando a Regressão Linear Múltipla, a etapa de definição dos parâmetros não é necessária. O método se adapta aos dados selecionados não necessitando portanto nenhuma configuração adicional.

A única classe utilizada neste método é a *MLRGapFilling*. Com ela é possível passar os dados que serão processados, analisar os dados, preencher as falhas e obter os resultados, conforme mostrado no Código 4.

O método inicia já na análise dos dados, por meio do método *train*, no qual os dados são passados para o método. Depois de um processamento rápido, pode ser passado para o método os dados que deverão ser tratados.

Código 4: Utilização do método de preenchimento de falhas RLM

```
1 MLRGapFilling mlr = new MLRGapFilling();
2 mlr.train(data);
3 mlr.setTestData(testData);
4 double[] result = mlr.process();
5 float error = mlr.getError();
6 long time = mlr.getTimeTrainning();
```

Como este método foi feito para dados multivariados, os dados passados no comando *setTestData* consiste apenas nas variáveis auxiliares para o preenchimento das falhas. No comando *process* é retornado o resultado do preenchimento, ou seja, os dados tratados.

Ainda é possível obter o erro estimado desse método e o tempo de processamento, por meio dos comandos *getError* e *getTimeTrainning*, respectivamente.

O método de preenchimento de falhas utilizando a Regressão Linear Múltipla está preparado para trabalhar com qualquer quantidade de variáveis, sendo mais rápido e mais fácil de aplicar se comparado ao trabalho manual em planilhas eletrônicas.

4.3.4 Média Móvel para Preenchimento de Falhas

No método de preenchimento de falhas utilizando Média Móvel, os parâmetros que podem ser configurados são:

- Quantidade de itens para serem levados em consideração para calcular a média.
- Quantidade de itens para testar a operação.
- Número de variáveis contidas na série de dados.
- Coluna que deve ser tratada.

A quantidade de itens para calcular a média é o parâmetro que tem a maior relação com o comportamento dos dados. Dependendo da variável meteorológica, apenas poucos valores anteriores e posteriores à falha devem ser considerados, enquanto outras variáveis necessitam de uma quantidade muito maior.

A frequência que os dados são coletados pode auxiliar na tomada dessa decisão. Se um dado é armazenado a cada 5 minutos, uma quantidade maior de dados pode ser utilizada para o cálculo da média, enquanto que um dado armazenado 1 vez por dia, talvez seja melhor calcular com apenas 1 valor anterior e posterior à falha.

Como este método possui um processamento rápido, vários testes podem ser realizados para verificar qual é o melhor valor para este parâmetro e, assim, conseguir efetuar um preenchimento com uma maior precisão.

O Código 5 apresenta a definição de parâmetros e a execução de preenchimento de falhas com esse método. As primeiras linhas definem os quatro parâmetros do método: itens para calcular a média (10 antes e 10 depois da falha), itens para estimar o erro no teste da operação (100 falhas simuladas), quantidade de variáveis existentes no arquivo (5) e a coluna que deve ser tratada (segunda coluna).

Código 5: Utilização do método de preenchimento de falhas Média Móvel

```
1  ParametersAvg parameters = new ParametersAvg();
2  parameters.setValuesToUse(10);
3  parameters.setItensCalculateError(100);
4  parameters.setSizeOfFile(5);
5  parameters.setFillColumn(2);
6
7  DataGapFillingAvg avg = new DataGapFillingAvg(parameters);
8  avg.train(data);
9  double[] result = avg.process();
10 float error = avg.getError();
```

Depois de definido os parâmetros, o comando *train* pode ser chamado e, em seguida, o comando *process*. Dessa forma, os dados serão analisados e, nas falhas encontradas, tratados.

Assim como nos outros métodos já mencionados, o erro pode ser obtido. Para estimar o erro deste método, é feita uma simulação de falhas de acordo com a configuração do respectivo parâmetro. Dessa forma, é possível comparar o valor de um preenchimento de falhas com um valor real, verificando esta diferença e, assim, calculando o erro.

Esse método é univariado e não executa equações complexas nos dados. Logo, é de se esperar que o seu processamento seja mais rápido que os métodos multivariados e mais complexos.

4.3.5 MANNGA para Detecção de *Outliers*

O uso do método de detecção de *outliers* MANNGA é semelhante a maneira de utilização do método de preenchimento de falhas com as mesmas técnicas de IA, mostrado na Seção 4.3.2.

A classe para definir os parâmetros do método é a mesma (*Parameters-Manngа*), então os seus itens configuráveis também são os mesmos. A diferença

na utilização do método de detecção de *outliers* em comparação com o método de preenchimento de falhas vai ser no uso da classe *OutlierDetection*.

O Código 6 apresenta um exemplo do uso desta classe. Assim como nos outros métodos, a classe do método é iniciada, os parâmetros são definidos e o treinamento é realizado. Posteriormente, os dados podem ser processados e o erro do método pode ser obtido. Além disso, a obtenção dos resultado da detecção de *outliers* desse método pode ser feita de três formas diferentes, mostradas no Código 7.

Código 6: Utilização do método de detecção de *outliers* MANNGA

```
1 OutlierDetection outlierDetection = new OutlierDetection();
2 outlierDetection.train(data);
3 outlierDetection.process(parameters);
4 float error = outlierDetection.getError();
```

Código 7: Obtenção dos resultados do método de detecção de *outliers* MANNGA

```
1 double[] distances = outlierDetection.getDistanceResult();
2 int[] idxOutliers = outlierDetection.getOutlierResult(0.02);
3 int[] idxOutliers2 = outlierDetection.getOutlierResult(10);
```

A primeira opção é obter as distâncias de todos os registros processados pelo método. Desta forma, é possível realizar uma análise própria para definir quais são os reais *outliers* na série de dados.

Tanto a segunda quanto a terceira opção, recupera uma parte da série de dados que possuem as maiores distâncias calculadas pelo método, ou seja, serão retornados os dados que foram considerados como os mais prováveis *outliers*. A segunda opção obtém uma porcentagem dos dados (2% no exemplo) e a terceira opção uma quantidade específica (10 no exemplo).

Uma das dificuldades em detectar os *outliers* é definir uma regra que separa os dados normais dos dados anormais. Alguns métodos tentam classificar automaticamente esses dados e outros, como este método, especificam um *ranking* de probabilidade para elencar os possíveis *outliers*, deixando a decisão final para o especialista que está tratando os dados.

4.3.6 ODHiMM para Detecção de *Outliers*

Diferente do método anterior, este método tenta classificar os dados como *outliers* ou dados normais, não estabelecendo um *ranking* de possíveis *outliers*, mas sim, determinando-os.

Para tanto, os seguintes parâmetros devem ser configurados:

- Iterações de treinamento - Influencia no aprendizado do método. De maneira geral, quanto maior esse valor melhor é o treinamento do modelo e maior tempo de processamento é exigido.

- Porcentagem dos dados para o treinamento - Também atua no aprendizado do modelo, servindo de base para a compreensão dos padrões envolvidos na série de dados.

- Porcentagem mínima de alteração para simular os *outliers* - Para criar os modelos HMM que representam os *outliers*, os dados são alterados a fim de que os modelos aprendam as características dos dados quando eles distanciam do seu comportamento normal. O valor mínimo que os dados são alterados vai depender deste parâmetro.

- Porcentagem máxima de alteração para simular os *outliers* - Também vai informar o quanto o dado deve ser alterado para modelar os modelos HMM que representam *outliers*, mas este é específico para limitar o máximo do valor de alteração. Então, para cada registro, um valor será sorteado entre o mínimo e o máximo determinado neste e no parâmetro anterior para modificar o registro.

- Quantidade de modelos HMM para os dados válidos - Determina quantos modelos HMM são criados para representar os dados válidos. Se mais de 1 for determinado, o método separa em faixas de valores para que cada modelo possa aprender uma determinada faixa.

O Código 8 mostra a configuração dos parâmetros e o uso do método.

Código 8: Utilização do método de detecção de *outliers* HMM

```
1  ParametersHmm parameter = new ParametersHmm();
2  parameter.setIterations(500);
3  parameter.setPercentageOfTraining(70);
4  parameter.setMinPercentageOutlier(15);
5  parameter.setMinPercentageOutlier(25);
6  parameter.setQuantityHmmValid(1);
7
8  HmmOutlierDetection hod = new HmmOutlierDetection(parameter);
9
10 hod.train(data);
11 List<Long> idxOutliers = hod.process();
12 float error = hod.getError();
13 long time = hod.getTimeTrainning();
```

Neste exemplo, foram configuradas 500 iterações para o treinamento com 70% dos dados. A simulação de *outliers* iriam variar de 15% a 25%, para mais ou para menos, e apenas 1 modelo HMM será criado para representar os dados válidos. Todos esses valores foram definidos na classe *ParametersHmm*.

Depois de configurado os parâmetros, um objeto da classe *HmmOutlierDetection* foi criado e o treinamento iniciado (por meio do comando *train*). Com os modelos treinados, é possível chamar o comando *process* e, assim, obter os índices dos dados que foram classificados como *outliers*. Assim como nos outros métodos, também é possível recuperar o erro estimado do método e o tempo gasto para realizar o treinamento.

A classificação dos dados como *outliers* por este método vai depender diretamente dos valores configurados nos parâmetros que limitam o mínimo e máximo de porcentagem para simular os *outliers* nos modelos HMM. Logo, é possível que o especialista, conhecendo o comportamento dos seus dados, possa determinar com uma melhor exatidão quando um dado passa a ser um *outlier*.

4.3.7 Z-Score para Detecção de *Outliers*

Os parâmetros configuráveis do método Z-Score para detecção de *outliers* estão listados a seguir:

- Coluna que deve ser tratada para detectar os *outliers*.
- Número de variáveis contidas na série de dados.
- Quantidade de itens para testar a operação.

Com estes parâmetros é possível dizer como estão os dados que serão enviados, informando qual coluna deve ser tratada. Também é possível dizer quantos elementos devem ser considerados para analisar a operação, possibilitando o cálculo do erro.

O Código 9 apresenta a definição de parâmetros e a execução de preenchimento de falhas com este método. As primeiras linhas definem os três parâmetros do método: coluna que deve ser tratada (terceira), número de variáveis (sete) e itens para estimar o erro no teste da operação (250 *outliers* simulados).

Depois de definido os parâmetros, o comando *train* pode ser chamado e, em seguida, o comando *process*. Com isso, os dados serão analisados e todos os *scores*, ou seja, as pontuações de cada registro que representam sua normalidade com relação ao restante da série, são calculados. O resultado dos *scores* são

retornados para que possa ser feito uma análise própria. Além disso, nessa etapa é possível obter o erro estimado da operação com o comando *getError*.

Código 9: Utilização do método de detecção de *outliers* Z-Score

```
1  ParametersZscore parameters = new ParametersZscore();
2  parameters.setOutlierColumn(3);
3  parameters.setSizeOfFile(7);
4  parameters.setValuesToChange(250);
5
6  Zscore zs = new Zscore(parameters);
7  zs.train(data);
8  double[] scores = zs.process();
9  float error = zs.getError();
10
11 int[] idxOutliers1 = zs.getOutliers(n);
12 int[] idxOutliers2 = zs.getOutliers(score);
```

Para estimar o erro neste método, é feita uma simulação dos *outliers* alterando uma cópia da série de dados. n itens (definido por parâmetro) são modificados para mais ou para menos e verificados se os mesmos são detectados como *outliers* pelo método, obtendo assim um índice de erro.

O método ainda disponibiliza duas funções para retornar os *outliers*. Para tanto, deve ser informado a quantidade de *outliers* que se deseja retornar ou um *score* específico. No primeiro caso, o método retorna os registros que obtiveram as maiores pontuações (módulo dos *scores*). No segundo caso, todos os registros que tiveram uma pontuação maior que o *score* informado são classificados como *outliers*.

4.4 Execução dos Testes

Para avaliar os métodos criados e implementados neste trabalho, diversas séries de dados foram selecionadas. Essas séries se diferenciam com relação às suas variáveis, periodicidades e frequências de leitura. Desta forma, é possível verificar o desempenho dos métodos em diferentes cenários.

Para os dados do INMET, que possuem cinco séries de dados com as mesmas variáveis climáticas, foi calculado uma média dos valores encontrados, resultando em um único EMA e coeficiente de correlação por variável climática para cada método. Para os dados do TRMM, que contém apenas dados de precipitação, também foi calculado uma média entre os resultados.

Nas seções a seguir são apresentados os parâmetros utilizados, o tempo

de processamento e os resultados obtidos nos testes para cada variável climática, dos três métodos criados e dos três obtidos da literatura.

4.4.1 MANNGA para Preenchimento de Falhas

No método MANNGA, os parâmetros configurados para executar os testes foram:

- Quantidade máxima de iterações no treinamento da RNA: 300.
- Erro aceitável para parar o treinamento da RNA: 0,002.
- Número máximo de gerações do AG: 60.
- Erro aceitável para parar de criar novas gerações no AG: 0,001.
- Tamanho da população de indivíduos do AG: 50.
- Porcetagem de mutação que os indivíduos sofrem no AG: 10%.
- Porcentagem de dados utilizados no treinamento: 70%.

O valor para cada parâmetro pode ser definido efetuando testes iniciais para verificar as melhores opções. Com os parâmetros configurados foi possível realizar o treinamento e o preenchimento das falhas. Na Tabela 6 estão os valores do EMA resultante dos testes realizados para as variáveis de temperatura do INMET (T), umidade relativa do ar do INEMT (UR), ponto de orvalho (*dew*), pressão atmosférica (P), radiação solar (Rg), velocidade do vento (u), precipitação (*prec*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

Tabela 6: Erro médio absoluto (EMA) para cada teste realizado com o MANNGA de preenchimento de falhas para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

% de falhas	EMA									
	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
	°C	%	°C	*hPa*	kJ/m^2	*m/s*	*mm*	*umol/mol*	°C	%
5%	1,14	3,36	1,19	1,48	204,69	0,84	1,90	132,21	0,16	4,00
15%	0,91	3,21	**1,10**	1,52	210,49	0,79	**1,16**	**81,49**	**0,13**	5,37
30%	**0,76**	3,30	1,11	**1,45**	202,25	0,80	1,42	148,84	0,56	**4,36**
50%	0,79	**2,62**	1,16	1,51	**201,57**	**0,79**	1,75	103,87	0,17	6,39

É importante lembrar que quanto menor o valor do EMA, melhor é o desempenho do método. Além disso, os valores do EMA são baseados na unidade de cada variável. Logo, pode ser observado que, em média, para o sensor de temperatura do INMET, esse método teve um erro de 0,9 °C no preenchimento das falhas, tendo como melhor desempenho a correção dos dados na simulação de 30%, com apenas 0,76 °C de erro.

A mesma lógica pode ser aplicada para todos os outros sensores, no qual os erros mínimos foram de 2,62% para umidade do INMET, 1,1 °C para ponto de orvalho, 1,45hPa para pressão, 201,57kJ/m^2 para radiação solar, 0,79m/s para velocidade do vento, 1,16mm para precipitação, 81,49$umol/mol$ para concentração de CO_2, 0,13 °C para temperatura do Ameriflux e 4% para umidade do Ameriflux.

O coeficiente de correlação também foi calculado para cada teste. Na Tabela 7 consta os resultados obtidos para esse coeficiente.

Tabela 7: Coeficiente de correlação (r) para cada teste realizado com o MANNGA de preenchimento de falhas para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	r									
% de falhas	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
5%	0,82	0,93	**0,40**	0,16	0,92	0,09	**0,77**	0,25	0,91	1,00
15%	0,88	0,94	0,32	0,18	0,91	0,14	0,73	**0,56**	**0,99**	1,00
30%	0,93	0,93	0,33	**0,25**	0,91	**0,14**	0,62	0,22	0,86	1,00
50%	**0,93**	**0,95**	0,34	0,16	**0,92**	0,13	0,54	0,39	0,98	**1,00**

É possível observar que as variáveis de temperatura, umidade e radiação solar tiveram ótimos resultados, com coeficientes de correlação maiores ou igual a 0,82. A precipitação teve um desempenho regular, com uma média de 0,67. Outra importante observação é que a quantidade de falhas nas série de dados não afetou a precisão do método.

Os resultados dos testes com dados do Ameriflux, que possui 1 ano de dados, mostram que o método não é afetado pelas características de sazonalidade das variáveis climáticas. O método é capaz de absorver as características de cada período e estimar corretamente os valores das falhas.

O tempo de processamento do método para cada variável também é importante para avaliar o desempenho do método. Essa informação pode ser útil na escolha de um método. Na Tabela 8 encontra-se essa informação para cada teste realizado com esse método.

Tabela 8: Tempo de processamento para cada teste realizado com o MANNGA de preenchimento de falhas para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	Tempo (mm:ss.ms)									
%	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
5%	13:14.372	15:25.481	11:10.337	10:41.571	11:38.155	13:19.171	01:06.097	274:15.672	139:54.789	318:55.053
15%	10:49.097	12:34.893	08:40.427	07:55.878	10:18.736	14:14.310	01:04.991	185:47.931	102:50.556	127:39.761
30%	08:35.983	11:45.263	07:47.633	06:19.577	08:21.294	09:36.091	01:03.071	120:45.588	90:57.012	231:18.149
50%	07:32.761	09:36.496	05:51.382	05:56.655	06:03.099	08:05.362	01:05.527	74:40.795	76:30.261	133:54.238

O tempo gasto para realizar o processamento em cada teste depende da quantidade de dados e do que foi configurado nos parâmetros do método, principalmente nos valores referentes às iterações de treinamento. De maneira geral, se for aumentado o número de iterações, um resultado melhor será alcançado, mas um tempo maior será exigido. O processo de correção mais rápido levou 01:03 *min* e o mais longo levou mais de 5 horas.

4.4.2 Regressão Linear Múltipla para Preenchimento de Falhas

No método de Regressão Linear Múltipla não há a necessidade de definir parâmetros. O próprio método trata de encontrar os padrões necessários na série de dados. Logo, o método pode ser executado diretamente. Na Tabela 9 estão os valores do EMA resultante dos testes realizados com este método. Baseado nesses valores, o método de Regressão Linear Múltipla teve bons resultados para algumas variáveis, principalmente na temperatura (0,44 °C), umidade (1,84%) e ponto de orvalho (0,47 °C). Por outro lado, não teve um desempenho favorável com a variável de pressão (15,55hPa).

Tabela 9: Erro médio absoluto (EMA) para cada teste realizado com o método RLM de preenchimento de falhas para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	EMA									
% de falhas	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
	°C	%	°C	hPa	kJ/m^2	m/s	mm	$umol/mol$	°C	%
5%	0,45	2,01	0,53	**15,55**	**517,57**	0,80	1,15	136,73	**0,56**	3,61
15%	0,44	**1,84**	**0,47**	15,71	548,80	**0,78**	**0,97**	**83,65**	0,61	**3,57**
30%	**0,44**	1,91	0,48	15,63	543,38	0,79	1,32	128,19	0,61	3,67
50%	0,45	1,88	0,48	16,16	538,84	0,79	1,38	122,07	0,60	3,76

Assim como no EMA, a análise com o coeficiente de correlação apresentada na Tabela 10 mostra que esse método teve bom desempenho com as variáveis de temperatura, umidade e ponto de orvalho. É mostrado também o baixo desempenho com as variáveis de pressão e velocidade do vento.

Tabela 10: Coeficiente de correlação (r) para cada teste realizado com o método RLM de preenchimento de falhas para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	r									
% de falhas	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
5%	0,97	0,98	0,85	0,03	**0,68**	0,10	**0,85**	0,25	**0,87**	1,00
15%	0,97	0,98	0,85	**0,04**	0,61	**0,16**	0,76	**0,59**	0,84	1,00
30%	0,97	0,98	**0,87**	0,02	0,63	0,14	0,69	0,23	0,84	1,00
50%	**0,97**	**0,98**	0,86	0,03	0,63	0,12	0,68	0,38	0,84	**1,00**

Na Tabela 11 estão os tempos gastos na execução de cada teste. Por se tratar de um método que não exige um fase intensiva de treinamento, o tempo de processamento para cada teste com esse método é muito rápido, sendo praticamente instantâneo com a máquina utilizada nos testes. Todos os testes demoraram menos que 1 segundo.

Tabela 11: Tempo de processamento para cada teste realizado com o método RLM de preenchimento de falhas para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	Tempo (mm:ss.ms)									
%	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
5%	00:00.001	00:00.002	00:00.002	00:00.002	00:00.002	00:00.043	00:00.001	00:00.057	00:00.234	00:00.169
15%	00:00.005	00:00.001	00:00.004	00:00.003	00:00.002	00:00.001	00:00.001	00:00.731	00:00.060	00:00.164
30%	00:00.013	00:00.001	00:00.002	00:00.013	00:00.001	00:00.001	00:00.001	00:00.064	00:00.019	00:00.151
50%	00:00.001	00:00.001	00:00.001	00:00.003	00:00.001	00:00.001	00:00.001	00:00.342	00:00.087	00:00.090

4.4.3 Média Móvel para Preenchimento de Falhas

O principal parâmetro no método de Média Móvel é a quantidade de itens a serem utilizados para calcular a média. Foram feitos alguns testes iniciais para determinar esse valor nas avaliações deste trabalho, ficando decidido que 5 valores antes da falha e 5 valores depois da falha seriam utilizados. Na Tabela 12 estão os valores do erro médio absoluto resultante dos testes com o método de Média Móvel.

Em se tratando de EMA, o método teve os melhores desempenhos com a variável de ponto de orvalho (0,58 °C no melhor resultado) e com a temperatura da fonte de dados Ameriflux (0,74 °C). Para as outras variáveis climáticas não houve bom desempenho, principalmente com a umidade do ar do Ameriflux. Analisando essa série de dados, foram encontrados valores inválidos nessa variável climática, o que justifica um erro tão ruim quando realizado uma estimativa por média.

Tabela 12: Erro médio absoluto (EMA) para cada teste realizado com o método Média Móvel de preenchimento de falhas para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	EMA									
% de falhas	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
	°C	%	°C	*hPa*	kJ/m^2	*m/s*	*mm*	*umol/mol*	°C	%
5%	**1,40**	**5,63**	0,63	**0,99**	**541,64**	0,69	2,34	69,51	**0,74**	298,15
15%	1,60	6,65	**0,58**	1,24	604,48	**0,63**	**2,31**	**65,57**	0,83	**278,21**
30%	1,96	8,41	0,67	2,38	735,25	0,66	2,51	106,53	1,02	336,44
50%	2,71	11,27	0,73	2,43	955,18	0,72	2,75	83,28	1,31	339,66

Um desempenho semelhante pode ser visto na Tabela 13 que mostra os coeficientes de correlação. O maior índice de coeficiente de correlação foi com a umidade relativa do ar do INMET (0,85 na simulação com 5% de falhas). O alto índice de erro se deve a simplicidade do método, além da sua característica de ser univariado.

Tabela 13: Coeficiente de correlação (r) para cada teste realizado com o método Média Móvel de preenchimento de falhas para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	r									
% de falhas	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
5%	**0,79**	**0,85**	0,68	**0,64**	**0,71**	0,25	**0,62**	**0,67**	**0,81**	0,00
15%	0,75	0,74	**0,73**	0,20	0,58	**0,36**	0,15	0,41	0,72	0,02
30%	0,59	0,63	0,65	0,18	0,43	0,33	0,18	0,41	0,57	**0,11**
50%	0,22	0,32	0,58	0,01	0,05	0,22	0,06	0,46	0,30	0,05

A vantagem desse método é a sua facilidade de aplicação e sua agilidade, como pode ser visto na Tabela 14 que apresenta o tempo de processamento para cada teste. Todos os testes com esse método levaram 28 milésimo de segundo ou menos.

Tabela 14: Tempo de processamento para cada teste realizado com o método Média Móvel de preenchimento de falhas para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	Tempo (mm:ss.ms)									
%	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
5%	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.028	00:00.001	00:00.002	00:00.006	00:00.002
15%	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.003	00:00.003	00:00.024
30%	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.003	00:00.015	00:00.002	00:00.002
50%	00:00.001	00:00.001	00:00.005	00:00.001	00:00.001	00:00.001	00:00.001	00:00.002	00:00.002	00:00.002

4.4.4 MANNGA para Detecção de *Outliers*

Para testar o método de detecção de *outliers* com MANNGA, foram utilizados os mesmos parâmetros configurados na Seção 4.4.1.

Uma diferença com relação aos testes já mostrados, é que para os métodos de detecção de *outliers*, ao invés de utilizar o EMA e o coeficiente de correlação, foram calculados a precisão e a *area under the curve* (AUC) em cada teste. Além disso, as simulações variam em porcentagem de *outliers* nas séries de dados e porcentagem de modificação dos valores para torná-los *outliers*, conforme detalhado na Seção 3.2.2.

Na Tabela 15 estão os valores de precisão, na Tabela 16 estão os valores de AUC e na Tabela 17 estão os tempos demandados para cada teste com esse método.

Tabela 15: Precisão do MANNGA na detecção de *outliers* para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	Precisão (%)									
Simulação	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
2% - 30%	98,02	88,10	67,46	99,60	8,33	0,79	-	56,49	100,00	20,13
2% - 50%	**100,00**	97,22	89,29	99,60	17,06	5,56	-	1,30	100,00	5,19
5% - 30%	97,62	90,79	76,51	**100,00**	16,19	6,67	0,00	23,32	99,74	44,82
5% - 50%	99,68	**98,89**	**93,65**	99,84	**28,10**	**7,62**	**10,00**	**58,03**	**100,00**	**73,58**

O método MANNGA para detectar *outliers* teve ótimos desempenhos nas variáveis de temperatura, umidade relativa e pressão atmosférica, atingido 100% de precisão em algumas simulações. Para radiação solar, velocidade do vento, precipitação e concentração de CO_2 não houve bons desempenhos.

Os valores encontrados com a AUC confirmam o baixo desempenho com as variáveis de radiação solar, velocidade do vento e precipitação. Entretanto, houve bons resultados para a temperatura, umidade, ponto de orvalho, pressão e

concentração de CO_2.

Tabela 16: AUC do MANNGA na detecção de *outliers* para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	AUC									
Simulação	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
2% - 30%	1,00	0,99	0,99	1,00	0,54	0,45	-	**0,99**	1,00	0,91
2% - 50%	1,00	0,99	0,99	1,00	0,55	**0,55**	-	0,98	1,00	0,96
5% - 30%	**1,00**	0,99	0,97	**1,00**	0,53	0,47	0,48	0,84	**1,00**	0,90
5% - 50%	1,00	**1,00**	**1,00**	1,00	**0,60**	0,48	**0,58**	0,88	1,00	**0,97**

O tempo de processamento foi similar à operação de preenchimento de falhas com as mesmas técnicas de IA, variando de 2 minutos a 5 horas. O tempo médio para cada teste gasto desse método pode ser considerado aceitável para um tipo de operação complexa como a de detecção de *outliers* e levando em consideração a quantidade de dados.

Tabela 17: Tempo de processamento para cada teste realizado com o MANNGA na detecção de *outliers* para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	Tempo (mm:ss.ms)									
Sim.	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
2% - 30%	14:44.836	16:23.884	13:31.688	12:00.937	15:48.852	12:13.660	-	201:07.904	222:14.828	82:43.802
2% - 50%	14:29.909	15:55.166	12:42.780	12:28.723	16:30.202	15:03.732	-	120:42.577	279:08.239	235:38.631
5% - 30%	14:30.099	15:14.294	15:26.834	14:04.059	13:52.270	12:16.955	02:03.532	126:39.105	346:24.185	89:51.960
5% - 50%	15:08.428	15:09.097	14:07.607	14:25.029	15:48.976	11:36.315	02:07.982	137:39.992	103:31.330	102:29.640

4.4.5 ODHiMM para Detecção de *Outliers*

Para o método ODHiMM para detectar *outliers* foram configuradas 100 iterações e 100% dos dados para o treinamento. Dessa forma haveria uma quantidade mínima de iterações para os modelos aprenderem o comportamento dos dados, ao mesmo tempo que ainda haveria um processamento rápido, mesmo com a quantidade de dados selecionada para o treinamento. Além disso, foi configurado para os dados serem alterados em 90%, ou seja, tanto a variação mínima quanto a máxima foram configuradas em 90%. Por fim, 3 modelos de HMM foram selecionados para modelar os dados válidos. Esses valores se mostraram os mais eficazes durante os testes iniciais.

Os altos valores de simulação de *outliers* pelo método na fase de treinamento mostrou-se mais eficaz para construir os modelos que representam os comportamentos de registros com *outliers*.

Os resultados da precisão do método podem ser visualizados na Tabela 18, os valores da AUC estão na Tabela 19 e o tempo de processamento de cada teste está na Tabela 20.

O melhor desempenho do método foi com a variável de pressão atmosférica (84,47% em uma simulação) e a pior precisão foi com a variável de precipitação (errando toda a estimativa). Na maioria das variáveis não houve um desempenho constante, sugerindo que são necessários mais testes para ajustar os melhores parâmetros do *Hidden Markov Model*.

Tabela 18: Precisão do método ODHiMM na detecção de *outliers* para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	Precisão (%)									
Simulação	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
2% - 30%	65,39	33,69	50,59	81,61	0,66	0,83	-	4,29	2,93	1,39
2% - 50%	71,31	63,86	55,24	81,26	2,05	0,69	-	2,86	4,38	1,40
5% - 30%	71,29	36,59	44,44	**84,47**	1,60	1,39	0,00	**7,14**	9,49	**4,90**
5% - 50%	**74,23**	**70,22**	**67,09**	83,63	**3,15**	**1,56**	**2,50**	2,90	**14,56**	3,52

Os valores obtidos com a AUC mostram bons resultados apenas para a variável de pressão atmosférica. O desempenho regular para as variáveis de temperatura e ponto de orvalho. Entretanto, para as outras variáveis, o resultado foi praticamente aleatório em sua detecção de *outliers*. Os resultados sugerem que melhorias no funcionamento do método devem ser realizadas, além de mais testes para verificar o motivo do método não estar conseguindo modelar corretamente boa parte das variáveis meteorológicas.

Tabela 19: AUC do método ODHiMM na detecção de *outliers* para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	AUC									
Simulação	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
2% - 30%	0,70	0,60	0,70	0,96	0,53	0,31	-	0,47	0,64	0,51
2% - 50%	**0,75**	0,64	0,79	0,94	**0,60**	**0,36**	-	0,48	**0,73**	0,50
5% - 30%	0,64	0,59	0,71	**0,96**	0,50	0,31	0,49	**0,50**	0,68	**0,51**
5% - 50%	0,71	**0,66**	**0,81**	0,95	0,55	0,30	**0,51**	0,50	0,72	0,50

O tempo de processamento desse método foi bem consistente, demandando no máximo 14 minutos para cada teste. Pela quantidade de dados e pela complexidade do método, pode ser considerado um bom tempo de processamento.

Tabela 20: Tempo de processamento para cada teste realizado com o método ODHiMM na detecção de *outliers* para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	Tempo (mm:ss.ms)									
Sim.	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
2% - 30%	04:09.658	03:43.348	03:20.664	03:08.702	03:40.124	04:29.399	-	13:14.883	08:01.264	05:16.126
2% - 50%	03:21.838	04:04.346	03:43.493	03:05.697	03:49.860	04:37.919	-	13:13.898	08:02.508	05:15.546
5% - 30%	03:51.527	03:27.202	04:10.050	03:10.291	03:12.050	02:49.768	00:27.027	13:22.652	08:05.855	05:15.009
5% - 50%	03:55.240	04:02.799	03:51.023	02:54.509	03:35.915	03:58.176	00:23.750	05:15.584	05:11.572	05:14.903

4.4.6 Z-Score para Detecção de *Outliers*

Assim como no método de Regressão Linear Múltipla, o método que utiliza o Z-Score não necessita de definições de parâmetros que influenciam em seu desempenho. O método calcula a média e o desvio padrão da variável que está sendo tratada e, assim, pode calcular a pontuação para cada dado.

A Tabela 21 mostra a precisão do método em cada teste. Esse método conseguiu uma precisão de 100% nas quatro simulações com a variável de pressão atmosférica. Alguns bons resultados também podem ser encontrados na variável de temperatura. Em contrapartida, houve um desempenho regular para o dado de ponto de orvalho (atingindo 86,35%) e ruim para todas as outras variáveis (menor que 58%).

Tabela 21: Precisão do método Z-Score na detecção de *outliers* para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	Precisão (%)									
Simulação	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
2% - 30%	35,71	21,43	54,76	100,00	6,35	0,79	-	48,05	90,91	9,09
2% - 50%	**81,75**	46,03	79,37	100,00	**11,90**	3,97	-	49,35	100,00	7,14
5% - 30%	48,25	34,29	61,59	**100,00**	5,71	5,71	2,50	**52,85**	93,78	38,60
5% - 50%	79,68	**57,14**	**86,35**	100,00	8,89	**7,94**	**5,00**	50,00	**100,00**	**46,63**

Como o Z-Score é um método univariado, a quantidade de atributos na série de dados não interfere no desempenho do método, por isso a possibilidade de

conseguir bons resultados mesmo em séries com um número grande de atributos, como a do Ameriflux.

A Tabela 22 mostra os resultados obtidos quando calculada a AUC com esse método, no qual praticamente em todas as variáveis houve um resultado ruim, independente da simulação. A detecção de *outliers* em dados meteorológicos é uma tarefa de muita dificuldade, e métodos mais simples como o Z-Score tendem a não ter um bom desempenho.

Tabela 22: AUC do método Z-Score na detecção de *outliers* para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	AUC									
Simulação	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
2% - 30%	0,43	0,45	0,55	0,52	0,50	**0,72**	-	0,46	0,49	0,50
2% - 50%	0,47	0,42	**0,58**	0,50	0,50	0,71	-	0,48	0,43	**0,54**
5% - 30%	0,49	**0,50**	0,48	0,49	0,50	0,71	0,56	**0,52**	0,50	0,46
5% - 50%	**0,50**	0,49	0,56	**0,57**	**0,51**	0,71	**0,58**	0,48	**0,53**	0,50

O tempo de processamento de cada teste está na Tabela 23. Assim como no método de Média Móvel, o método Z-Score detecta os *outliers* de forma praticamente instantânea, levando 6 milessegundo ou menos para executar o procedimento.

Tabela 23: Tempo de processamento para cada teste realizado com o método Z-Score na detecção de *outliers* para as variáveis de temperatura do INMET (T), umidade do INMET (UR), ponto de orvalho (*dew*), pressão (P), radiação solar (Rg), vento (u), precipitação (*ppt*), concentração de CO_2 (CO_2), temperatura do Ameriflux (TA) e umidade do Ameriflux (URA).

	Tempo (mm:ss.ms)									
Sim.	T	UR	*dew*	P	Rg	u	*ppt*	CO_2	TA	URA
2% - 30%	00:00.002	00:00.001	00:00.001	00:00.001	00:00.001	00:00.002	-	00:00.003	00:00.006	00:00.001
2% - 50%	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	-	00:00.002	00:00.001	00:00.001
5% - 30%	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001
5% - 50%	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.001	00:00.003	00:00.001

4.5 Comparação dos Métodos

Com a obtenção dos resultados nos testes realizados, é possível fazer uma análise da eficácia de cada método de preenchimento de falhas e detecção de *outliers*. Dessa forma, pôde ser avaliado qual método é mais indicado para cada variável, considerando tanto o nível de acerto do método quanto o tempo de processamento utilizado. Também foi possível comparar o desempenho dos

três métodos criados neste trabalho (MANNGA para preenchimento de falhas, MANNGA para detecção de *outliers* e ODHiMM) com os três métodos obtidos da literatura (RLM, MM e Z-Score).

4.5.1 Métodos de Preenchimento de Falhas

Na Figura 12 é mostrado um gráfico para cada variável climática, sendo possível comparar o desempenho de cada método de preenchimento de falhas com base no EMA obtido.

O método RLM teve o melhor resultado para os dados de temperatura, umidade, ponto de orvalho e precipitação. Esse método também teve um bom desempenho com os dados de umidade do Ameriflux, praticamente igual ao método MANNGA. Por sua vez, o método MANNGA teve os melhores resultados com os dados de radiação, pressão e temperatura do Ameriflux. Por fim, o método de Média Móvel teve o melhor resultado com os dados de vento e CO_2.

Duas variáveis climáticas, temperatura e umidade, estavam presentes em duas fontes de dados diferentes: INMET e Ameriflux. A principal diferença entre elas é a quantidade de atributos presentes em cada série de dados, 6 e 23, respectivamente. A quantidade de atributos pode afetar a precisão dos métodos. Nos testes realizados, o MANNGA obteve melhores resultados com a série de dados com o maior número de atributos. Isso indica que o MANNGA é mais indicado em cenários com um número grande de variáveis climáticas.

Quanto mais falhas, maior as ocorrências de falhas em sequência. Dependendo da característica do método, isso pode afetar diretamente a sua precisão. Os gráficos da Figura 12 mostram que o método de Média Móvel teve uma menor precisão quando a quantidade de falhas na série de dados aumentaram. O mesmo não aconteceu com o método MANNGA, o qual obteve um desempenho similar em todos os cenários.

O coeficiente de correlação, por apresentar o mesmo índice independente da unidade da variável climática, permite que o desempenho dos métodos possam ser comparados de maneira geral. A Figura 13 mostra gráficos dos resultados dos testes realizados com esse índice.

Na simulação de 5% de falhas, 5 das 8 variáveis climáticas únicas tiveram ótimos resultados, com um coeficiente maior que 0,8 (máximo 1). Os métodos de preenchimento de falhas do *framework* conseguiram uma boa precisão para as variáveis climáticas de temperatura, umidade, ponto de orvalho, radiação e precipitação. O *framework* teve um desempenho regular para as variáveis de pressão e CO_2, mas não houve um bom desempenho para velocidade do vento.

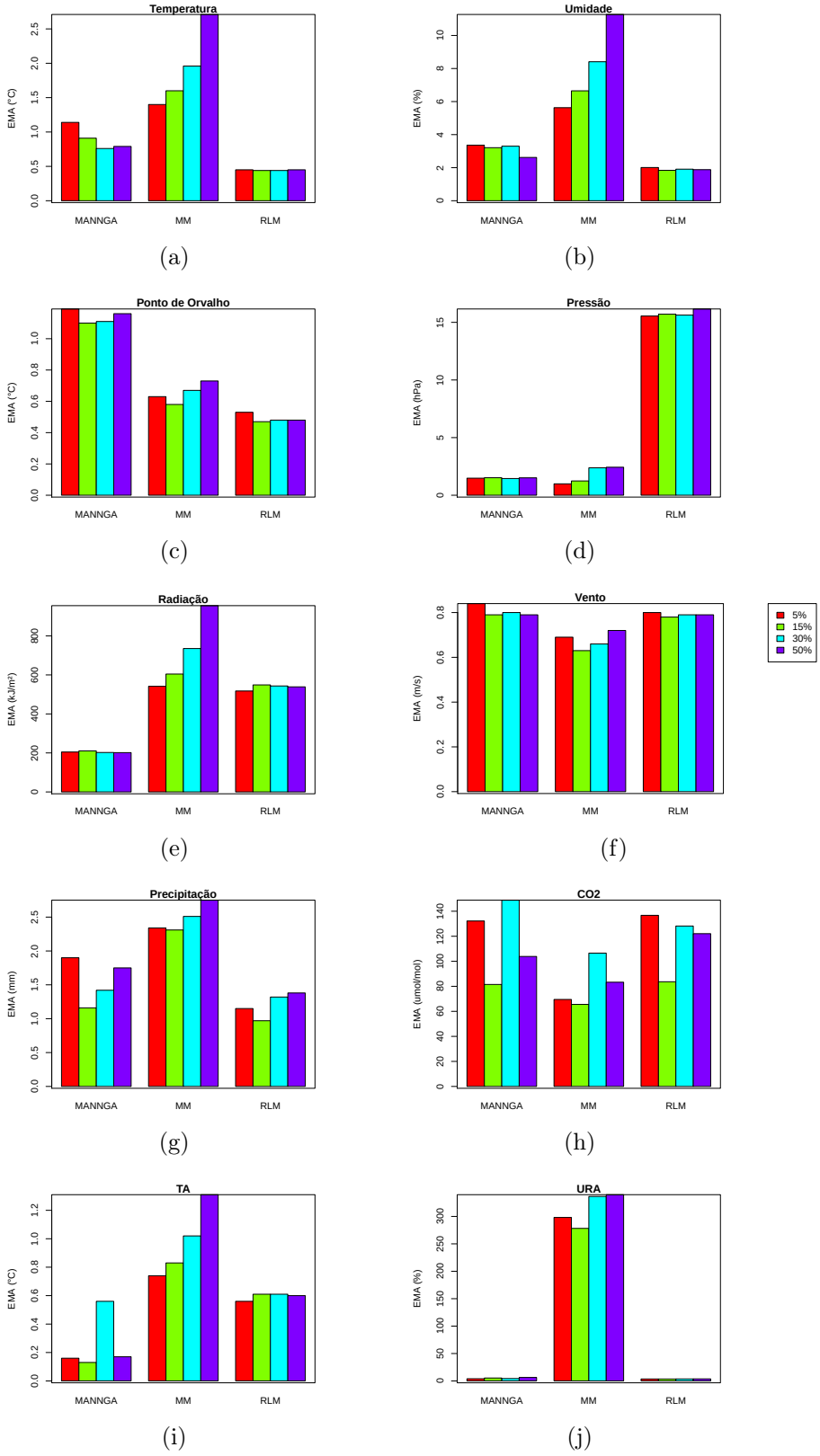

Figura 12: Valores do EMA para os testes de preenchimento de falha em (a) temperatura, (b) umidade, (c) ponto de orvalho, (d) pressão, (e) radiação solar, (f) vento, (g) precipitação, (h) CO_2, (i) temperatura do Ameriflux e (j) umidade do Ameriflux.

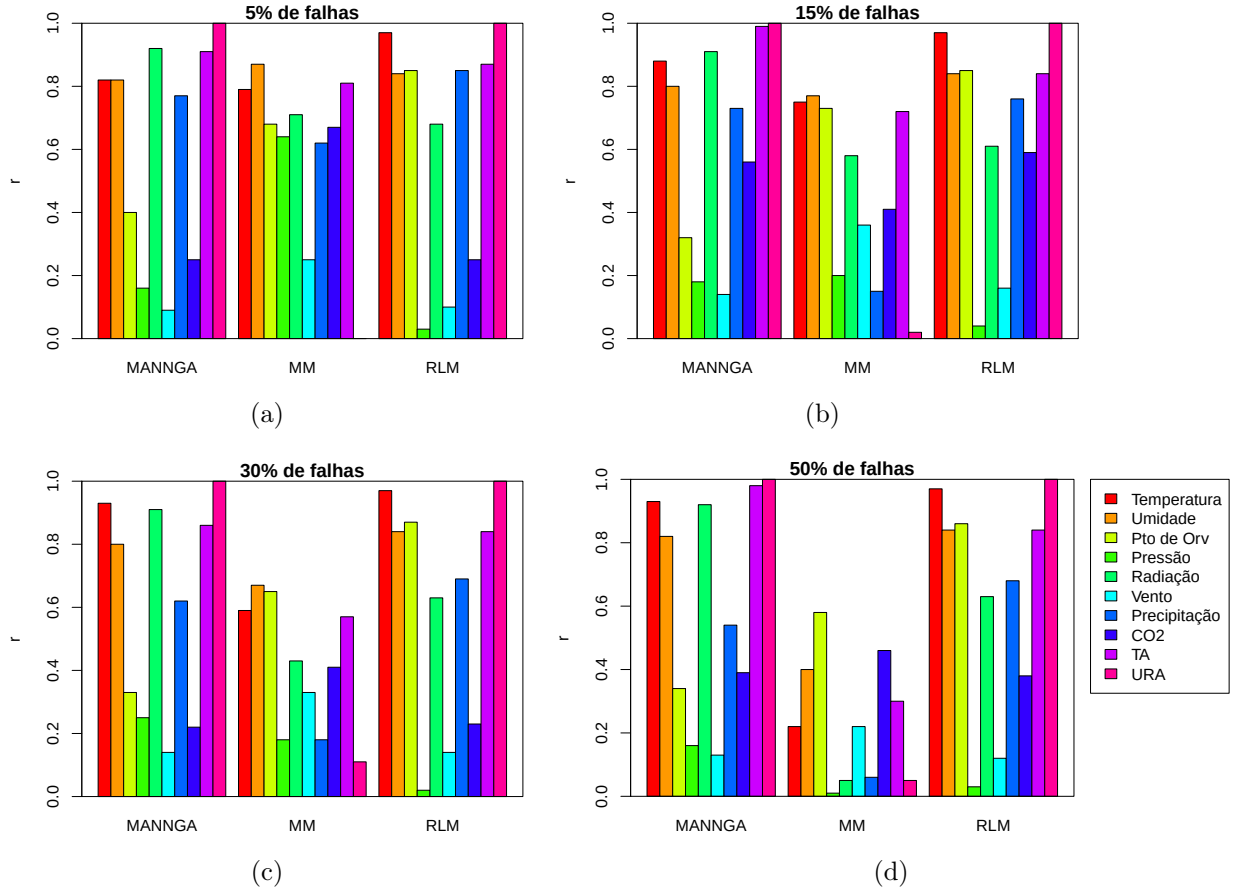

Figura 13: Resultados dos testes com os métodos de preenchimento de falhas avaliando o coeficiente de correlação com (a) 5%, (b) 15%, (c) 30% e (d) 50% de falhas.

Além disso, mais uma vez pode ser observado a diminuição na precisão do método univariado (Média Móvel) com o aumento da quantidade de falhas na série de dados. Pode ser visto também a regularidade dos métodos multivariados nos mesmos cenários, sendo que o MANNGA obteve melhores resultados quando havia uma grande quantidade de atributos.

Uma média do tempo de processamento gasto nos testes também foi calculado para realizar um comparativo entre os métodos. Na Tabela 24 é apresentado o tempo médio de cada método.

Tabela 24: Tempo médio de processamento para cada método de preenchimento de falhas.

Método	Tempo (mm:ss.ms)
MANNGA	21:28.600
Média Móvel (MM)	00:00.001
Regressão Linear Múltipla (RLM)	00:00.020

Os métodos de Média Móvel e Regressão Linear Múltipla têm um processamento muito rápido (menos de 1 segundo por execução). Já o método que envolve técnicas de Inteligência Artificial demandam mais tempo, mas não suficiente para inviabilizar o uso do método.

É importante lembrar que, de acordo com o ciclo de vida do *framework* apresentado na Figura 11, é possível realizar o treinamento apenas uma vez e realizar várias correções em seguida, considerando que os dados treinados tenham as mesmas características dos dados a serem corrigidos. Se for comparado apenas o tempo de correção, ou seja, sem contabilizar o tempo da fase de treinamento, a diferença entre os métodos de IA e os de estatística serão menores.

Em um cenário real, seria viável criar modelos para cada variável climática e treiná-los intensivamente. Posteriormente, apenas o processamento computacional para correção dos dados seria notado pelos usuários, o que resultaria em algo tão ágil quanto os métodos estatísticos.

4.5.2 Métodos de Detecção de *Outliers*

Para os métodos de detecção de *outliers* foram calculados a precisão, a AUC e o tempo gasto em cada execução. A Figura 14 mostra a precisão obtida por cada método em cada variável testada, nos quatro cenários criados.

Pode ser observado que o MANNGA teve os melhores desempenhos, se saindo melhor para os dados de temperatura, umidade, ponto de orvalho e pressão. O método Z-Score também atingiu ótimos resultados no dado de pressão, sendo regular com o dado de temperatura, ponto de orvalho e CO_2. O método ODHiMM teve bom desempenho com o dado de pressão, sendo regular com temperatura, umidade e ponto de orvalho.

O Z-Score, um método univariado, teve um bom resultado apenas nos dados de pressão porque esta variável tem um comportamento mais constante, sendo teoricamente mais fácil de detectar comportamentos anormais. Mas o desempenho regular em outras três variáveis climáticas mostra que é um método que pode ser utilizado em alguns casos, principalmente se o tempo de processamento for essencial.

O MANNGA, de comportamento multivariado, conseguiu bons resultados mesmo em variáveis mais dinâmicas, sendo auxiliado pelos comportamentos também dinâmicos das outras variáveis. No método ODHiMM não houve o mesmo desempenho, mesmo sendo também multivariado. A diferença entre um método e o outro é a ação do AG ao selecionar as variáveis que realmente afetarão a modelagem, sendo que no ODHiMM todas as variáveis sempre participam dos

modelos. Então, é possível que alguma variável auxiliar esteja prejudicando os modelos HMM.

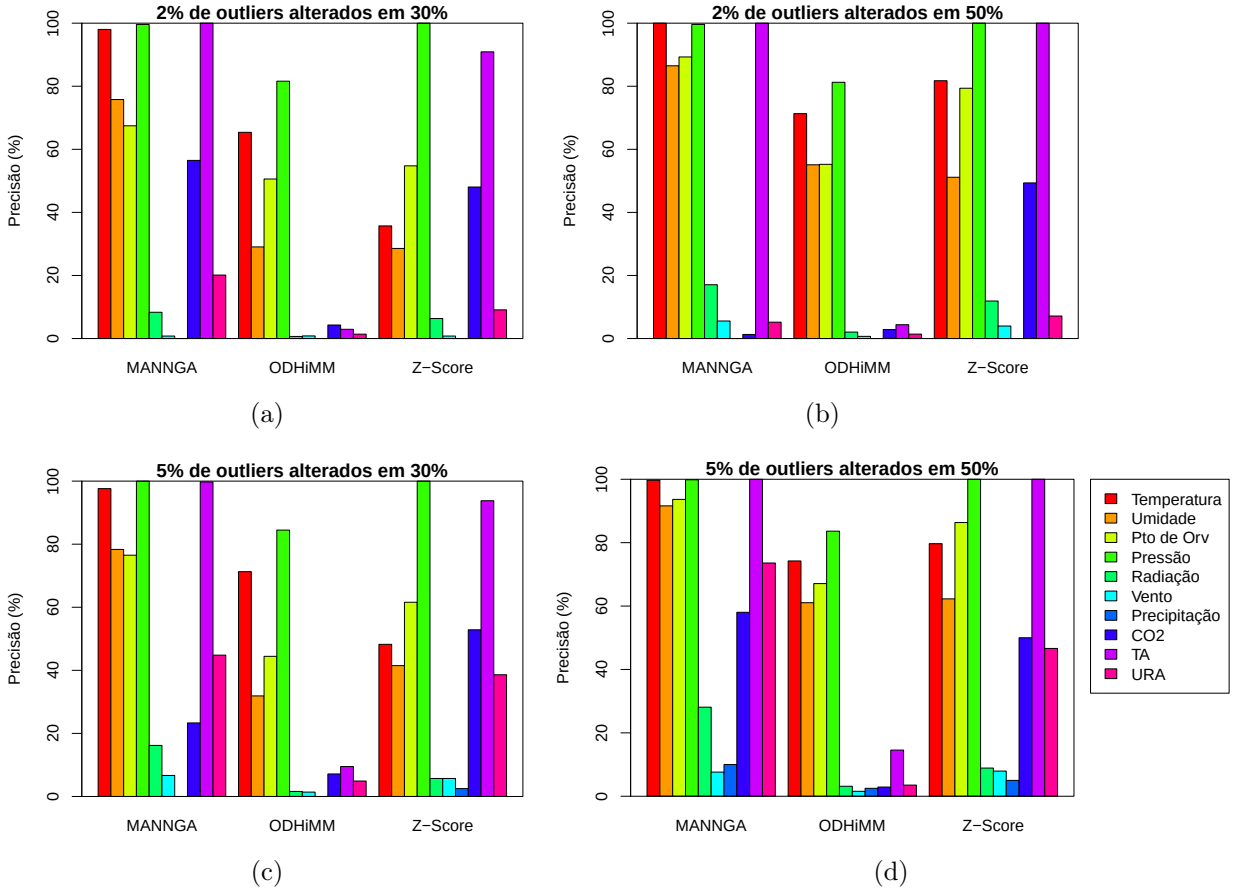

Figura 14: Resultados dos testes com os métodos de detecção de *outliers* avaliando a precisão de (a) 2% de *outliers* alterando-os em 30%, (b) 2% alterando-os em 50%, (c) 5% alterando-os em 30% e (d) 5% alterando-os em 50%.

Esse comportamento pode ser notado comparando os resultados dos testes com os dados do INMET e do Ameriflux. Com o método ODHiMM, a precisão dos testes com os dados do Ameriflux (TA e URA) são consideravelmente inferiores aos testes com as mesmas variáveis climáticas dos dados do INMET (Temperatura e Umidade). Isso indica que o grande número de atributos das séries de dados do Ameriflux prejudica a precisão do método ODHiMM.

É importante notar que não houve bons resultados na detecção de *outliers* para as variáveis climáticas de radiação solar, velocidade do vento e precipitação. O baixo desempenho da variável de radiação solar é justificada porque as medidas oscilam muito durante o dia com a presença de nuvens. Melhores resultados podem ser obtidos caso a série de dados possua variáveis que representem tais influências. Com relação a velocidade do vento e a precipitação, há a dificuldade

da modelagem por causa das influências que essas variáveis sofrem por fatores externos ao local da medição, sendo difícil estimar o seu comportamento apenas observando medições da estação meteorológica.

A AUC também pode ser comparada entre os métodos de detecção de *outliers*. A Figura 15 apresenta os dados desta comparação. É possível observar que os métodos tiveram desempenho semelhantes nos quatro cenários criados. Como esperado, houve melhores desempenhos nas simulações em que os dados foram alterados em 50%, já que desta forma a anomalia se torna mais aparente.

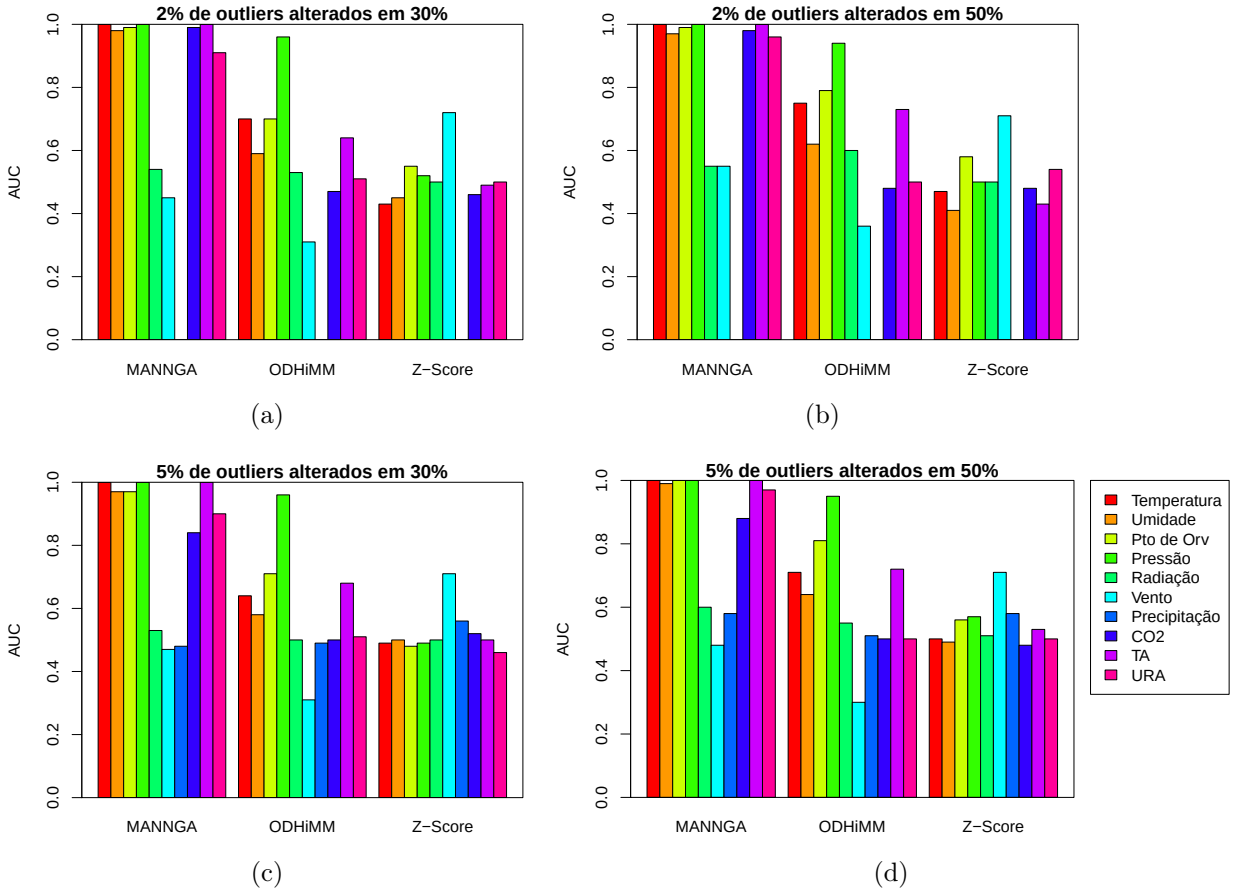

Figura 15: Resultados dos testes com os métodos de detecção de *outliers* avaliando a AUC de (a) 2% de *outliers* alterando-os em 30%, (b) 2% alterando-os em 50%, (c) 5% alterando-os em 30% e (d) 5% alterando-os em 50%.

A AUC confirma que a detecção de *outliers* do *framework* tem capacidade para obter bons resultados para as variáveis climáticas de temperatura, umidade, ponto de orvalho, pressão e CO_2. É possível obter melhores resultados para a variável de precipitação caso os locais de coleta de dados estejam mais próximos uns dos outros. Nos testes realizados, por causa da limitação do TRMM, um ponto de dados estava no mínimo $25km$ distante do ponto mais próximo, prejudicando

a modelagem de precipitação.

A Tabela 25 mostra a média do tempo de processamento para esses métodos. O método Z-Score é o mais rápido dos métodos de detecção de *outliers* testados, exigindo um tempo menor que 1 segundo para detectar os *outliers*. O método MANNGA requer mais tempo de processamento, principalmente se houver muitos atributos na série de dados. O método ODHiMM, apesar de ser configurado com várias iterações, não obteve um tempo de processamento longo, necessitando em média menos de quatro minutos para cada teste.

Tabela 25: Tempo médio de processamento para cada método de detecção de *outliers*.

Método	Tempo (mm:ss.ms)
MANNGA	27:59.174
ODHiMM	03:36.641
Z-Score	00:00.001

4.6 Sistema *Web-based*

O ambiente computacional desenvolvido tem como foco o *framework* que possibilita que aplicações possuam funcionalidades complexas de tratamento de dados meteorológicos. Para testar a integração entre o *framework* e um novo sistema, foi desenvolvido uma aplicação *web* que possibilita o envio de uma planilha de dados meteorológicos e, com poucas ações, a aplicação das operações de detecção de *outliers* e de preenchimento de falhas.

Desta forma, é feita uma abstração dos métodos implementados. Ou seja, o usuário não terá conhecimento dos métodos complexos da área de Inteligência Artificial e estatística, tendo que se preocupar apenas com a correção dos seus dados. O sistema pode ser acessado e testado pelo endereço *http://ceda.ic.ufmt.br/web*.

O modo como esta integração foi feita e os detalhes do sistema desenvolvido encontra-se no Apêndice A deste trabalho, atestando a possibilidade da criação de aplicações com base no *framework* desenvolvido.

4.7 Conclusão

Soluções como as propostas em Dias et al. (2014), de criar redundâncias em todas as etapas de obtenção e armazenamento dos dados, podem ter ótimos

resultados, mas em contrapartida há um alto custo de instalação, além da difícil implantação e manutenção. Para muitos casos, corrigir os dados de forma inteligente é a melhor opção.

Com o *framework* criado, a aplicação dos métodos existentes por outros sistemas é facilitada, bastando poucas linhas de código de programação para executar funcionalidades complexas da área computacional e da estatística.

Inicialmente, foram criados ou incorporados três métodos de preenchimento de falhas e três métodos de detecção de *outliers*. Entretanto, a arquitetura do *framework* possibilita que novos métodos sejam adicionados no futuro.

Para avaliar os métodos criados neste trabalho diversos testes foram realizados. Foram obtidas séries de dados de três fontes diferentes (INMET, TRMM e Ameriflux). Para cada variável climática foram simulados preenchimento de falhas e detecção de *outliers*. Ao todo, 267.180 dados foram processados. Foram efetuados 1.032 testes, no qual em cada resultado uma análise estatística foi feita a fim de possibilitar a avaliação do desempenho de cada método criado em comparação com os métodos obtidos da literatura.

Nas séries do INMET e Ameriflux, os dados foram coletados de hora em hora. Séries de dados deste tipo tem uma maior complexidade de ser corrigida porque há muitas variações entre as leituras. É comum em algumas pesquisas realizar análises em médias diárias dos dados, no qual há uma maior regularidade entre os dados. Testes de preenchimento de falhas com o MANNGA teve resultados ainda melhores quando testado com dados de média diária, como pode ser visto em Ventura et al. (2013b).

Outro ponto importante que influencia no desempenho dos métodos multivariados são as variáveis disponíveis nas séries de dados. Quanto mais variáveis relacionadas, maior a chance de preencher as falhas e detectar os *outliers* corretamente. Em Ventura et al. (2013a) os testes de detecção de *outliers* em dados de temperatura do ar tiveram ótimos resultados (mais que 90% de precisão) quando haviam dados de temperatura do solo e saldo de radiação na mesma série de dados. Em contrapartida, foi visto que a presença de variáveis que não estão relacionadas à variável que está sendo tratada, pode prejudicar a precisão dos métodos.

Outros métodos se mostraram eficientes com dados meteorológicos, como em Tsukahara et al. (2010) que preencheu falhas utilizando RNA, mas dependendo de várias estações meteorológica, alcançando coeficientes de correlação maiores que 0,9 para variáveis de pressão, temperatura, radiação solar e umidade. Ou em Rihbane (2014), que usou a técnica de Monte Carlo e obteve coeficientes de cor-

relação da ordem de 0,95 para preenchimento de falhas em dados de temperatura do ar, umidade relativa do ar e fluxo de calor no solo. Em Sadik e Gruenwald (2010) um novo método foi apresentado para detectar *outliers* em dados de temperatura do solo. Os resultados avaliados alcançaram um índice de 0,8 com o coeficiente de Jaccard (de 0 a 1). Entretanto, como as variáveis envolvidas influenciam muito o resultado final, há uma dificuldade em realizar comparações entre métodos existentes e os criados neste trabalho caso eles não sejam testados com a mesma base de dados. Neste trabalho, três métodos da literatura foram implementados e aplicados com as mesmas séries de dados dos métodos criados, podendo realizar uma comparação justa entre esses métodos.

De forma geral, o *framework* teve ótimos resultados, sendo que cada método se sobressaiu em uma ou mais variáveis climáticas. Também como resultado deste trabalho, foi implementado um sistema *web-based* para testar a integração entre o *framework* e novos sistemas, facilitando as operações de tratamento de dados meteorológicos.

Capítulo 5

Considerações Finais

As falhas nos dados são um problema recorrente em pesquisas que envolvem dados meteorológicos. Decisões devem ser tomadas para tentar evitar essas ocorrências ou corrigí-las. O grande obstáculo na correção dos dados meteorológicos é a dificuldade em aplicar os métodos existentes na literatura. Análises de pesquisas são prejudicadas por decisões de remover períodos inteiros de dados por causa de falhas não corrigidas, assim como quando métodos simples de correção de dados são aplicados, obtendo uma baixa precisão na estimativa do valor a ser preenchido.

Além dos dados ausentes, *outliers* presentes nas séries de dados, mas não detectados, mostram um comportamento dos dados que não é normal, podendo dificultar as análises. Caso não seja aplicado um método para detectá-los e corrigí-los, falsos resultados podem ser gerados.

Neste trabalho foi desenvolvida uma forma para auxiliar pesquisadores de diversas áreas na correção dos seus dados, principalmente se forem de natureza meteorológica, tanto para detectar os possíveis *outliers* nas séries de dados quanto para corrigir as falhas encontradas.

Um novo método para preenchimento de falhas foi criado, além de outros dois novos métodos para detecção de *outliers*. Esses métodos foram integrados a um *framework* para facilitar o seu uso, abstraindo as etapas complexas de suas teorias, configurações e processamentos. Além disso, foram adicionados outros três métodos já existentes ao *framework*, oferecendo mais opções para corrigir as séries de dados meteorológicos.

O *framework* permite que sistemas possam ser desenvolvidos utilizando de suas funcionalidades. Logo, vários sistemas com ações de correção de dados podem facilmente serem criados, eliminando o grande obstáculo da dificuldade em aplicar métodos de preenchimento de falhas ou detecção de *outliers*.

Um sistema *web-based* foi implementado para mostrar a integração entre uma aplicação e o *framework*. No sistema desenvolvido o usuário, mesmo sem conhecimentos em inteligência artificial ou estatística, pode executar sem nenhuma dificuldade as operações de correções de dados, bastando basicamente enviar o arquivo de dados, selecionar a operação desejada e exportar os dados corrigidos.

5.1 Contribuições

Estas foram as contribuições geradas com o desenvolvimento deste trabalho:

- Novo método de preenchimento de falhas foi criado - O MANNGA utiliza de RNA e AG para o preenchimento de falhas e tem como uma das principais vantagens conseguir preencher as falhas mesmo que elas estejam em sequência, já que não são utilizados dados do mesmo sensor que houve a falha, mas sim os dados de outros sensores.

- Novo método de detectar *outliers* foi criado - Aproveitando da mesma metodologia do método anterior, o MANNGA para detectar *outliers* utiliza do conceito de distância para ordenar os dados que são os mais prováveis de serem *outliers*. Entre os métodos testados, este obteve os melhores resultados.

- Outro novo método de detectar *outliers* foi criado - O ODHiMM cria modelos de *Hidden Markov Model* para classificar os dados como válidos ou *outliers*. Um modelo serve para modelar o comportamento de dados que estão desviando para mais, outro para modelar os dados que estão desviando para menos e um ou mais modelos para modelar o comportamento dos dados comuns. Uma vantagem desse método é que não é necessário configurar um ponto de corte ou dizer quantos *outliers* há na série de dados, porque o próprio método fica responsável por isso.

- Encapsulamento dos métodos criados e obtidos da literatura em um único *framework* - O *framework* criado possui uma série de métodos a ser escolhido de acordo com as características dos dados a serem corrigidos, permitindo que seja realizado a correção de dados configurando poucos parâmetros. Logo, qualquer pesquisador com experiência em programação poderia estar utilizando o *framework* e, assim, adicionando funcionalidades de correção de dados aos seus próprios sistemas.

- Um sistema *web-based* de tratamento de dados ambientais foi desenvolvido - Para beneficiar os usuários sem habilidades em programação de computadores, um sistema foi desenvolvido (integrado ao *framework*) para facilmente enviar um arquivo com os dados meteorológicos e obter, com poucos ajustes, a série de dados corrigida. Desta forma, está disponível uma forma eficaz e simplificada para detectar os *outliers* e realizar preenchimento de falhas em dados meteorológicos.

Com essas contribuições espera-se que os pesquisadores tenham uma ferramenta a mais para aumentar a qualidade dos seus dados e, consequentemente, melhorar as análises nas pesquisas.

5.2 Publicações

1. **VENTURA, T. M.**; FIGUEIREDO, J. M.; MARTINS, C. A.; GOMES, R. S.; OLIVEIRA, A. G.; NOGUEIRA, M. C. J. A Framework for Gap Filling in Meteorological Data. Expert Systems with Applications (em avaliação).

2. **VENTURA, T. M.**; OLIVEIRA, A. G.; MARTINS, C. A.; FIGUEIREDO, J. M.; NOGUEIRA, M. C. J. A. GapFiM: uma plataforma de preenchimento de falhas de dados meteorológicos. In: XVIII CBMET - Congresso Brasileiro de Meteorologia, 2014, Recife – PE.

3. MACHADO, N. G.; **VENTURA, T. M.**; DANELICHEN, V. H. M.; BIUDES, M. S. Performance of Neural Network for Estimating Rainfall over Mato Grosso State, Brazil. In: DailyMeteo.org/2014, 2014, Belgrade, p. 95-99.

4. **VENTURA, T. M.**; FIGUEIREDO, J. M.; NOGUEIRA, M. C. J. A. Desenvolvimento de uma Framework para Tratamento de Dados Multivariados. In: V Semana Acadêmica, 2014, Cuiabá.

5. **VENTURA, T. M.**; OLIVEIRA, A. G.; MARQUES, H. O.; OLIVEIRA, R. S.; MARTINS, C. A.; FIGUEIREDO, J. M.; BONFANTE, A. G. Uma abordagem computacional para preenchimento de falhas em dados micrometeorológicos. Revista Brasileira de Ciências Ambientais (Online), v. 1, p. 61-70, 2013.

6. **VENTURA, T. M.**; MARQUES, H. O.; OLIVEIRA, A. G.; MARTINS, C. A.; NOGUEIRA, M. C. J. A.; TEIXEIRA, W. R. S.; FIGUEIREDO, J. M.; BONFANTE, A. G. Detecção de Outliers em Dados Micrometeorológicos. In: IX Congresso Brasileiro de Agroinformática, 2013, Cuiabá.

7. NOITE, A. A.; **VENTURA, T. M.**; MARTINS, C. A. Modelagem e Prototipagem de uma Plataforma Web para Tratamento de Dados Ambientais. In: IV Escola Regional de Informática da SBC (Regional Mato Grosso), 2013, Alto Araguaia – MT.

8. TEIXEIRA, W. R. S.; **VENTURA, T. M.**; MARTINS, C. A. Comparativo entre Métodos de Detecção de Outliers para Dados Ambientais. In: IV Escola Regional de Informática da SBC (Regional Mato Grosso), 2013, Alto Araguaia – MT.

9. **VENTURA, T. M.**; NOGUEIRA, M. C. J. A. Uma Framework para Tratamento de Dados Ambientais. In: IV Semana Acadêmica, 2013, Cuiabá. V Mostra de Pós-Graduação, 2013. p. 625-625.

10. **VENTURA, T. M.**; MARQUES, H. O.; NOGUEIRA, M. C. J. A.; MARTINS, C. A. Uma Alternativa para Detecção de Outliers em Dados Micrometeorológicos Multivariados. Coletânea Física Ambiental II. 1ed.Baraúna: Editora São Paulo, 2012, v. 2, p. 50-54.

11. **VENTURA, T. M.**; NOGUEIRA, M. C. J. A.; MARTINS, C. A. Proposta de um Método de Detecção de Outliers para Dados Micrometeorológicos. In: III Semana Acadêmica, 2012, Cuiabá – MT.

12. **VENTURA, T. M.**; MARQUES, H. O.; OLIVEIRA, A. G.; MARTINS, C. A.; NOGUEIRA, M. C. J. A.; FIGUEIREDO, J. M.; BONFANTE, A. G. Detectando Outliers em Dados Micrometeorológicos. In: III Escola Regional de Informática da SBC, 2012, Rondonópolis - MT.

5.3 Trabalhos Futuros

Considerando os resultados obtidos neste trabalho, recomenda-se a continuação no aperfeiçoamento do *framework*. Os métodos criados podem ser melhorados, modificando as técnicas de RNA, AG e HMM para aproveitar o máximo possível das suas capacidades. Além do mais, outros métodos podem ser adicionados, uma vez que toda a estrutura do *framework* já está pronta.

Segundo Gupta et al. (2014), existem inúmeras formulações de detecção de *outliers* para dados temporais que ainda não foram suficientemente exploradas, principalmente por causa das inúmeras combinações de definição de problemas envolvendo este tipo de dado. Logo, novos métodos podem ser criados, tanto para preenchimento de falhas quanto para detecção de *outliers*. Então, não só a adição de métodos já existentes na literatura pode ser realizada, mas sim, de métodos inovadores, que consigam alcançar melhores resultados para as variáveis climáticas que, atualmente, o *framework* não alcança boa precisão. Uma possibilidade é adicinoar métodos específicos para uma determinada variável climática, tornando o *framework* mais completo.

Além de novos métodos, outras funcionalidades podem ser implementadas no *framework*. Uma forma automatizada do próprio *framework* decidir qual método deverá ser utilizado pode ser muito útil, caso o usuário não tenha conhecimento sobre os métodos ou não queira testar todos os disponíveis. Além disso, ações encadeadas podem ser planejadas, como o resultado de um método ser a entrada de outro método, possibilitando tratamentos ainda mais complexos. Outras tarefas da metodologia CRISP-DM também podem ser implementadas no *framework*.

Sistemas em geral podem ser implementados aproveitando de todos os recursos que o *framework* tem a oferecer, apresentando mais opções ao usuário e, assim, possibilitando uma melhor correção dos dados enviados.

Por fim, um estudo mais aprofundado pode ser realizado sobre a precisão do *framework* com dados não meteorológicos. Teoricamente, os métodos presentes atualmente no *framework* tem a capacidade de preencher falhas e detectar *outliers* em dados de outra natureza, como dados de bolsa de valores, dados multimídia, dados geoespaciais, dados de questionários, dentre outros. Se isso for testado e comprovado, este trabalho não beneficiará apenas as pesquisas da área ambiental, o que já é de muita importância, mas também várias áreas da comunidade científica.

REFERÊNCIAS

AGGARWAL, C.; YU, S. An effective and efficient algorithm for high-dimensional outlier detection. *The VLDB Journal*, 2005. Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 14, n. 2, p. 211–221, 2005. Citado na página 12.

AHMAD, F.; MAT-ISA, N.; HUSSAIN, Z.; BOUDVILLE, R.; OSMAN, M. Genetic algorithm-artificial neural network (ga-ann) hybrid intelligence for cancer diagnosis. In: *Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on*. [S.l.: s.n.], 2010. p. 78–83. Citado na página 18.

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*, 2006. v. 141, n. 1, p. 57–66, dez. 2006. Citado na página 15.

AMERIFLUX. *AmeriFlux Site and Data Exploration System*. 2015. Http://ameriflux.ornl.gov/. Acessado em Janeiro/2015. Citado na página 26.

ARNING, A.; AGRAWAL, R.; RAGHAVAN, P. A linear method for deviation detection in large databases. In: *KDD*. [S.l.]: AAAI Press, 1996. p. 164–169. Citado na página 13.

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*, 2010. Universidade Estadual Paulista Júlio de Mesquita Filho, Brasil, v. 35, n. 4, p. 133–138, 2010. Citado na página 17.

BARDELI, R.; WOLFF, D.; KURTH, F.; KOCH, M.; TAUCHERT, K. H.; FROMMOLT, K. H. Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring. *Pattern Recogn. Lett.*, 2010. Elsevier Science Inc., New York, NY, USA, v. 31, n. 12, p. 1524–1534, set. 2010. Citado na página 19.

BAUM, L. E.; PETRIE, T.; SOULES, G.; WEISS, N. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Statist.*, 1970. The Institute of Mathematical Statistics, v. 41, n. 1, p. 164–171, 02 1970. Citado na página 19.

BELLMAN, R. E. *Dynamic programming*. Princeton, NY: Princeton University Press, 1957. Citado na página 12.

BIUDES, M. S.; JÚNIOR, J. H. C.; ESPINOSA, M. M.; NOGUEIRA, J. S. Uso de séries temporais em análise de fluxo de seiva de mangabeira. *Ciência e Natura*, 2009. v. 31, n. 1, p. 65–77, 2009. Citado na página 20.

BODEN, T. A.; KRASSOVSKI, M.; YANG, B. The ameriflux data activity and data system: an evolving collection of data management techniques, tools, products and services. *Geoscientific Instrumentation, Methods and Data Systems*, 2013. v. 2, n. 1, p. 165–176, 2013. Citado na página 28.

BOSCHI, R. S.; OLIVEIRA, S. R. de M.; ÁVILA, A. M. H. de. Análise da variabilidade espaço-temporal da precipitação pluviométrica no estado do rio grande do sul. In: *VII Congresso Brasileiro de Agroinformática, SBIAgro*. Viçosa, MG: Associação Brasileira de Agroinformática, 2009. Citado na página 8.

BRAUNER, N.; MORDECHAI. Considering precision of data in reduction of dimensionality and {PCA}. *Computers and Chemical Engineering*, 2000. v. 24, n. 12, p. 2603 – 2611, 2000. Citado na página 12.

BREUNIG, M. M.; KRIEGEL, H.-P.; NG, R. T.; SANDER, J. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 2000. ACM, New York, NY, USA, v. 29, n. 2, p. 93–104, maio 2000. Citado na página 12.

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 14.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. *Outlier detection: a survey*. [S.l.], 2007. Citado na página 13.

CHAPMAN, P.; CLINTON, J.; KERBER, R.; KHABAZA, T.; REINARTZ, T.; SHEARER, C.; WIRTH, R. *CRISP-DM 1.0: step-by-step data mining guide*. Illinois: SPSS, 2000. Citado 2 vezes nas páginas I e 6.

CHÁVEZ, E.; NAVARRO, G.; BAEZA-YATES, R.; MARROQUÍN, J. L. Searching in metric spaces. *ACM Comput. Surv.*, 2001. ACM, New York, NY, USA, v. 33, n. 3, p. 273–321, set. 2001. Citado na página 12.

CHEN, S.; WANG, W.; ZUYLEN, H. van. A comparison of outlier detection algorithms for its data. *Expert Systems with Applications*, 2010. v. 37, n. 2, p. 1169 – 1178, 2010. Citado 2 vezes nas páginas 11 e 12.

CHIBANA, E. Y.; FLUMIGNAN, D.; MOTA, R. G.; VIEIRA, A. d. S.; FARIA, R. T. Estimativa de falhas em dados meteorológicos. In: *V Congresso Brasileiro de Agroinformática, SBI-AGRO*. Londrina, PR: [s.n.], 2005. Citado na página 14.

CHU, W.; BLUMSTEIN, D. T. Noise robust bird song detection using syllable pattern-based hidden markov models. In: *ICASSP*. [S.l.]: IEEE, 2011. p. 345–348. Citado na página 19.

DENGEL, S.; ZONA, D.; SACHS, T.; AURELA, M.; JAMMET, M.; PARMENTIER, F. J. W.; OECHEL, W.; VESALA, T. Testing the applicability of neural networks as a gap-filling method using CH4 flux data from high latitude wetlands. *Biogeosciences*, 2013. v. 10, p. 8185 – 8200, 2013. Citado na página 15.

DESWAL, S.; PAL, M. Artificial neural network based modeling of evaporation losses in reservoirs. *International Journal of Mathematical Physical and Engineering Sciences*, 2008. v. 39, n. 2, p. 177–181, 2008. Citado na página 15.

DIAS, A.; FRARE, B.; JOURDAN, P.; D'ORSI, R. Desenvolvimento e implantação de um conjunto resiliente de ferramentas computacionais para a operação de sistemas de alerta. In: *XVIII Congresso Brasileiro de Meteorologia, CBMET*. Recife, PE: Sociedade Brasileira de Meteorologia, 2014. Citado na página 68.

DIAS, C. A. A. *Procedimentos de Medição e Aquisição de Dados de uma Torre Micrometeorológica em Sinop-MT*. 89 p. Dissertação (Dissertação) — UFMT, 2007. Citado na página 9.

DOURADO, C. da S.; OLIVEIRA, S. R. de M.; AVILA, A. M. H. de. Análise de zonas homogêneas em séries temporais de precipitação no estado da bahia. *Bragantia*, 2013. SciELO Brasil, v. 72, n. 2, p. 192–198, 2013. Citado na página 8.

EDDY, S. What is a hidden markov model? *Nature Biotechnology*, 2004. Nature Publishing Group, v. 22, n. 10, p. 1315–1316, 2004. Citado na página 20.

ENCOG. *Encog Machine Learning Framework*. 2014. Http://www.heatonresearch.com/encog. Acessado em Outubro/2014. Citado na página 41.

FALGE, E.; BALDOCCHI, D.; OLSON, R.; ANTHONI, P.; AUBINET, M.; BERNHOFER, 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*, 2001. v. 107, n. 1, p. 43–69, 2001. Citado na página 15.

FAWCETT, T. An introduction to roc analysis. *Pattern Recogn. Lett.*, 2006. Elsevier Science Inc., New York, NY, USA, v. 27, n. 8, p. 861–874, jun. 2006. Citado na página 29.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P.; UTHURUSAMY, R. (Ed.). *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996. Citado na página 6.

FERRARI, G. T.; OZAKI, V. Missing data imputation of climate datasets: implications to modeling extreme drought events. *Revista Brasileira de Meteorologia*, 2014. scielo, v. 29, p. 21 – 28, 03 2014. Citado na página 15.

FIGUEIREDO J, M. *Formalização do domínio imagem para buscas por conteúdo em SGBDs relacionais*. 125 p. Dissertação (Tese) — USP - São Carlos, 2005. Citado na página 12.

FRANCOIS, J.-M. *Jahmm - An implementation of HMM in Java*. 2014. Https://code.google.com/p/jahmm/. Acessado em Outubro/2014. Citado na página 41.

GALES, M.; YOUNG, S. The application of hidden markov models in speech recognition. *Foundations and Trends in Signal Processing*, 2008. v. 1, n. 3, p. 195–304, 2008. Citado na página 19.

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 14.

GALVÃO, C. O.; VALENÇA, M. J. S.; VIEIRA, V. P. P. B.; DINIZ, L. S.; LACERDA, E. G. M.; CARVALHO, A. C. P. L. F. *Sistemas inteligentes: Aplicações a recursos hídricos e ciências ambientais*. [S.l.]: UFRGS/ABRH, 1999. 246 p. Citado na página 16.

GANCHEV, T.; POTAMITIS, I.; FAKOTAKIS, N. Acoustic Monitoring of Singing Insects. In: IEEE. *International Conference on Acoustics, Speech, and Signal Processing*. [S.l.], 2007. v. 4, p. 721–724. Citado na página 19.

GENÇAY, R. Non-linear prediction of security returns with moving average rules. *Journal of Forecasting*, 1996. John Wiley and Sons, Ltd., v. 15, n. 3, p. 165–174, 1996. Citado na página 24.

GHAHRAMANI, Z. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 2001. v. 15, n. 01, p. 9–42, 2001. Citado na página 19.

GRUBBS, F. E. Procedures for detecting outlying observations in samples. *Technometrics*, 1969. v. 11, p. 1–21, 1969. Citado na página 11.

GUHA, S.; RASTOGI, R.; SHIM, K. Rock: A robust clustering algorithm for categorical attributes. In: *ICDE*. [S.l.]: IEEE Computer Society, 1999. p. 512–521. Citado na página 13.

GUPTA, M.; GAO, J.; AGGARWAL, C. C.; HAN, J. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2014. IEEE Computer Society, Los Alamitos, CA, USA, v. 26, n. 9, p. 1–1, 2014. Citado 2 vezes nas páginas 13 e 75.

HAASE, R. F. *Multivariate General Linear Models*. [S.l.]: SAGE Publications, 2011. Citado na página 23.

HASAN, M. M.; CROKE, B. F. W. Filling gaps in daily rainfall data: a statistical approach. In: *20th International Congress on Modelling and Simulation*. Adelaide, Australia: [s.n.], 2013. p. 380–386. Citado na página 15.

HAYKIN, S. S. *Redes Neurais - Principios E Pratica*. Porto Alegre, RS: Bookman, 2001. Citado 4 vezes nas páginas I, 16, 17 e 36.

HEWITT, C. N. *Handbook of atmospheric science - principles and applications*. [S.l.]: Wiley-Blackwell, 2003. 600 p. Citado na página 1.

HODGE, V.; AUSTIN, J. A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 2004. Kluwer Academic Publishers, Norwell, MA, USA, v. 22, n. 2, p. 85–126, out. 2004. Citado na página 12.

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*, 2009. v. 149, n. 9, p. 1410–1420, set. 2009. Citado na página 15.

HUI, D. Gap-filling missing data in eddy covariance measurements using multiple imputation (mi) for annual estimations. *Agricultural and Forest Meteorology*, 2004. v. 121, n. 1–2, p. 93–111, 2004. Citado 2 vezes nas páginas 2 e 15.

INMET. *Instituto Nacional de Meteorologia*. 2014. Http//www.inmet.gov.br. Acessado em Outubro/2014. Citado na página 26.

JAIN, S. K.; NAYAK, P. C.; SUDHEER, K. P. Models for estimating evapotranspiration using artificial neural networks, and their physical interpretation. *Hydrological Processes*, 2008. v. 2234, n. February, p. 2225– 2234, 2008. Citado na página 15.

JIA, L.; XI, G.; LIU, S.; HUANG, C.; YAN, Y.; LIU, G. Regional estimation of daily to annual regional evapotranspiration with modis data in the yellow river delta wetland. *Hydrology and Earth System Sciences*, 2009. v. 13, n. 10, p. 1775–1787, 2009. Citado na página 15.

JÚNIOR, S. S. Histórico de instalação das estações meteorológicas do inmet no estado de minas gerais. In: *Simpósio de Pós-Graduação em Geografia do Estado de São Paulo*. Rio Claro, SP: UNESP, 2008. Citado na página 9.

KNORR, E. M.; NG, R. T.; TUCAKOV, V. Distance-based outliers: Algorithms and applications. *The VLDB Journal*, 2000. Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 8, n. 3-4, p. 237–253, fev. 2000. Citado na página 12.

KOVÁCS, Z. L. *Redes Neurais Artificiais: Fundamentos e Aplicações*. [S.l.]: Collegium Cognitio, 1996. Citado na página 35.

KOZIEVITCH, N. P. *Dados Meteorológicos: um estudo de viabilidade utilizando um SGBD em plataforma de baixo custo*. 60 p. Dissertação (Dissertação) — UFPR, 2005. Citado na página 5.

KUMAR, M.; RAGHUWANSHI, N.; SINGH, R.; WALLENDER, W.; PRUITT, W. Estimating evapotranspiration using artificial neural network. *Journal of Irrigation and Drainage Engineering*, 2002. American Society of Civil Engineers, v. 128, n. 4, p. 224–233, out. 2002. Citado na página 10.

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.

LIMA, C. H. R. Preenchimento de falhas em dados espaciais binários de precipitação utilizando máquinas de vetor de suporte (support vector machines). In: *Simpósio Nacional de Probabilidade e Estatística*. [S.l.: s.n.], 2010. Citado na página 15.

LUXBURG, U. von. *Statistical Learning with Similarity and Dissimilarity Functions*. 166 p. Dissertação (Tese) — Max Planck Institute for biological cybernetics, Germany, 2004. Citado na página 13.

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 14.

MEIRA, C. A. A.; RODRIGUES, L. H. A. Mineração de dados no desenvolvimento de sistemas de alerta contra doenças de culturas agrícolas. In: *V Congresso Brasileiro de Agroinformática, SBIAgro*. Londrina, PR: Associação Brasileira de Agroinformática, 2005. Citado na página 8.

MENZER, O. *Eddy covariance gap-filling and flux-partitioning tool*. 2014. Http://www.bgc-jena.mpg.de/ MDIwork/eddyproc/. Acessado em Outubro/2014. Citado na página 92.

MICHALEK, S.; TIMMER, J. Estimating rate constants in hidden markov models by the em algorithm. *IEEE Transactions on Signal Processing*, 1999. v. 47, n. 1, p. 226–228, 1999. Citado na página 19.

MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. New York: Springer-Verlag, 1994. Citado na página 18.

MISHUROV, M.; KIELY, G. Gap-filling techniques for the annual sums of nitrous oxide fluxes. *Agricultural and Forest Meteorology*, 2011. v. 151, p. 1763–1767, 2011. Citado na página 15.

MOFFAT, A. M.; PAPALE, D.; REICHSTEIN, M.; HOLLINGER, D. Y.; RICHARDSON, A. D.; BARR, A. G.; BECKSTEIN, C.; BRASWELL, B. H.; CHURKINA, G.; DESAI, A. R. et al. Comprehensive comparison of gap-filling techniques for eddy covariance net carbon fluxes. *Agricultural and Forest Meteorology*, 2007. Elsevier, v. 147, n. 3, p. 209–232, 2007. Citado na página 15.

MURUGAVEL, P.; PUNITHAVALLI, D. M. Performance evaluation of density-based outlier detection on high dimensional data. *International Journal on Computer Science and Engineering*, 2013. v. 5, n. 2, p. 62–67, 2013. Citado na página 13.

NASCIMENTO, T. S. do; SARAIVA, J. M. B.; SENNA, R.; AGUIAR, F. E. O. Preenchimento de falhas em banco de dados pluviométricos com base em dados do CPC (Climate Prediction Center): estudo de caso do rio Solimões-Amazonas. *Revista Brasileira de Climatologia*, 2009. v. 7, p. 143 – 158, 2009. Citado na página 15.

NETO, P. S. G. M.; PETRY, G. G.; ATAIDE, J. P. M.; FERREIRA, T. A. E. Combinação de redes neurais artificiais com algoritmo genético modificado para a previsão de séries temporais. In: *XXV Congresso da Sociedade Brasileira de Computação*. [S.l.: s.n.], 2005. Citado 2 vezes nas páginas 17 e 18.

NEVES, G. A. R. *Desenvolvimento de Estação Micrometeorológica com Armazenamento de Dados*. 61 p. Dissertação (Dissertação) — UFMT, 2011. Citado na página 9.

NOORMETS, A.; CHEN, J.; CROW, T. R. Age-dependent changes in ecosystem carbon fluxes. *Ecosystems*, 2007. v. 10, p. 187–203, 2007. Citado na página 15.

OLIVEIRA, A. G. de; VENTURA, T. M.; GANCHEV, T. D.; FIGUEIREDO, J. M.; JAHN, O.; MARQUES, M. I.; SCHUCHMANN, K.-L. Acoustic activity detection based on morphological filtering of the spectrogram. *Applied Acoustics*, 2014. Elsevier, 2014. No prelo. Citado na página 19.

OLIVEIRA, L. F. C. d.; FIOREZE, A. P.; MEDEIROS, A. M. M.; SILVA, M. A. S. Comparação de metodologias de preenchimento de falhas de séries históricas de precipitação pluvial anual. *Revista Brasileira de Engenharia Agrícola e Ambiental*, 2010. scielo, v. 14, p. 1186 – 1192, 11 2010. Citado 2 vezes nas páginas 14 e 20.

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*, 2006. v. 198, n. 3-4, p. 473 – 486, 2006. Citado 2 vezes nas páginas 15 e 18.

ORTON, N. J. H.; IPSITZ, S. R. L. Multiple imputation in practice : Comparison of software packages for regression models with missing variables. *American Statistician*, 2001. v. 55, n. 3, p. 244–254, 2001. Citado na página 15.

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 14.

PEARSON, K. Mathematical contributions to the theory of evolution. III. Regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A*, 1896. The Royal Society, v. 187, p. 253–318, jan. 1896. Citado na página 29.

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 1.

PINHEIRO, A.; GRACIANO, R. L. G.; SEVERO, D. L. Tendência das séries temporais de precipitação da região sul do brasil. *Revista Brasileira de Meteorologia*, 2013. scielo, v. 28, p. 281 – 290, 09 2013. Citado na página 15.

RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. In: *PROCEEDINGS OF THE IEEE*. [S.l.: s.n.], 1989. p. 257–286. Citado na página 18.

RFC4180. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. 2005. Http://tools.ietf.org/html/rfc4180. Acessado em Janeiro/2015. Citado na página 43.

RIHBANE, F. E. C. *Preenchimento de Falhas Aleatórias de Séries Temporais Micrometeorológicas pela Técnica de Monte Carlo*. 48 p. Dissertação (Dissertação) — UFMT, 2014. Citado na página 69.

RIHBANE, F. E. C.; GAIO, D. C.; SANCHES, L. Estudo da variabilidade em séries históricas para preenchimento de falhas. In: *XVII Congresso Brasileiro de Meteorologia*. [S.l.]: CBMET, 2012. Citado na página 20.

RUSSELL, S.; NORVIG, P. *Inteligência Artificial*. [S.l.]: Campus, 2004. Citado 2 vezes nas páginas 16 e 18.

SADIK, M.; GRUENWALD, L. Dbod-ds: Distance based outlier detection for data streams. 2010. Springer Berlin Heidelberg, v. 6261, p. 122–136, 2010. Citado 2 vezes nas páginas 14 e 70.

SANTOS, S. R. *A framework for the visualization of multidimensional and multivariate data*. 225 p. Dissertação (PhD thesis) — University of Leeds, 2004. Citado na página 10.

SCHIFFLER, R. E. Maximum z scores and outiers. *American Statistician*, 1988. p. 79–80, 1988. Citado na página 20.

SEDGEWICK, R.; WAYNE, K. *Introduction to Computer Science*. 2014. Http://introcs.cs.princeton.edu/java/cs/. Acessado em Outubro/2014. Citado na página 24.

SELLITTO, M. A. Inteligência artificial: uma aplicação em uma indústria de processo contínuo. *Gestão e Produção*, 2002. scielo, v. 9, p. 363 – 376, 12 2002. Citado na página 16.

SERRANO-ORTIZ, P.; DOMINGO, F.; CAZORLA, A.; WERE, A.; CUEZVA, S.; VILLAGARCÍA, L.; ALADOS-ARBOLEDAS, L.; KOWALSKI, A. S. Interannual co2 exchange of a sparse mediterranean shrubland on a carbonaceous substrate. *Journal of Geophysical Research: Biogeosciences*, 2009. v. 114, n. G4, p. 1–11, 2009. Citado na página 15.

SFERRA, H. H.; CORRÊA, A. M. C. J. Conceitos e aplicações de data mining. *Revista de Ciência e Tecnologia*, 2003. v. 11, n. 22, p. 19–34, 2003. Citado na página 7.

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*, 2011. v. 8, n. 2, p. 2001–2033, 2011. Citado na página 15.

SHEIKH, R. H.; RAGHUWANSHI, M.; JAISWAL, A. N. Genetic algorithm based clustering: A survey. *Emerging Trends in Engineering and Technology, International Conference on*, 2008. IEEE Computer Society, Los Alamitos, CA, USA, p. 314–319, 2008. Citado na página 17.

SHEN, J.; YANG, M.; ZHONG, R.; ZHANG, C. A hidden markov model based method for anomaly detection of precipitation series. *Journal of Information and Computational Science*, 2011. v. 8, n. 9, p. 1551–1560, 2011. Citado na página 19.

SILVA, D. F. da; ROCHA, J. V. Interpolação dos dados observados de precipitação pluvial e comparados com dados estimados pelo satélite trmm. In: *XVI Simpósio Brasileiro de Sensoriamento Remoto, SBSR*. Foz do Iguaçu, PR: INPE, 2013. p. 4086–4092. Citado na página 27.

SOARES, F. S.; FRANCISCO, C. N.; SENNA, M. C. A. Distribuição espaço-temporal da precipitação na região hidrográfica da baía da ilha grande-rj. *Revista Brasileira de Meteorologia*, 2014. v. 29, n. 1, p. 125 – 138, 2014. Citado na página 14.

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*, 2003. v. 129, p. 214–218, 2003. Citado na página 36.

SUN, Y.; GENTON, M. G. Adjusted functional boxplots for spatio-temporal data visualization and outlier detection. *Environmetrics*, 2012. John Wiley and Sons, Ltd, v. 23, n. 1, p. 54–64, 2012. Citado na página 14.

TARDIVO, G.; BERTI, A. The selection of predictors in a regression-based method for gap filling in daily temperature datasets. *International Journal of Climatology*, 2014. John Wiley and Sons, Ltd, v. 34, n. 4, p. 1311–1317, 2014. Citado na página 15.

TATSCH, J.; ROCHA, H.; CABRAL, O.; FREITAS, H.; LLOPART, M.; ACOSTA, R.; LIGO, M. Avaliação do método de multiple imputation no preenchimento de falhas de fluxos de energia sobre uma área de cana-de-açúcar. *Ciência e Natura*, 2007. p. 109–112, 2007. Citado na página 20.

THAMADA, T. T.; NETO, C. D. G.; MEIRA, C. A. A. Sistema de alerta da ferrugem do cafeeiro: resultado de um processo de mineração de dados. In: *IX Congresso Brasileiro de Agroinformática, SBIAgro*. Cuiabá, MT: Associação Brasileira de Agroinformática, 2013. Citado na página 8.

TRIFA, V.; KIRSCHEL, A.; TAYLOR, C. E.; VALLEJO, E. E. Automated species recognition of antbirds in a mexican rainforest using hidden markov models. *Journal of the Acoustical Society of America*, 2008. v. 123, n. 4, p. 2424–2431, April 2008. Citado na página 19.

TRMM. *Tropical Rainfall Measuring Mission*. 2014. Http://trmm.gsfc.nasa.gov/. Acessado em Janeiro/2015. Citado na página 26.

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*, 2010. p. 1–5, 2010. Citado 2 vezes nas páginas 15 e 69.

UYANıK, G. K.; GÜLER, N. A study on multiple linear regression analysis. *Procedia - Social and Behavioral Sciences*, 2013. v. 106, n. 0, p. 234 – 240, 2013. Citado na página 23.

VENTURA, T. M.; MARQUES, H. O.; OLIVEIRA, A. G. de; MARTINS, C. A.; NOGUEIRA, M. C. de J. A.; TEIXEIRA, W. R. S.; FIGUEIREDO, J. M. de; BONFANTE, A. G. Detecção de outliers em dados micrometeorológicos. In: *IX Congresso Brasileiro de Agroinformática, SBIAgro*. Cuiabá, MT: Associação Brasileira de Agroinformática, 2013a. Citado na página 69.

VENTURA, T. M.; OLIVEIRA, A. G. de; GANCHEV, T. D.; FIGUEIREDO, J. M.; JAHN, O.; MARQUES, M. I.; SCHUCHMANN, K.-L. Audio parameterization with robust frame selection for improved bird identification. *Expert Systems with Applications*, 2014. Elsevier, 2014. No prelo. Citado na página 19.

VENTURA, T. M.; OLIVEIRA, A. G. de; MARQUES, H. O.; OLIVEIRA, R. S.; MARTINS, C. A.; FIGUEIREDO, J. M. de; BONFANTE, A. G. Uma abordagem computacional para preenchimento de falhas em dados micro meteorológicos. *Revista Brasileira de Ciências Ambientais*, 2013b. n. 27, p. 61–70, 2013b. Citado na página 69.

VENTURA, T. M. V. *Preenchimento de Falhas de Dados Micrometeorológicos Utilizando Técnicas de Inteligência Artificial*. 73 p. Dissertação (Dissertação) — UFMT, 2012. Citado 3 vezes nas páginas I, 31 e 33.

WANDERLEY, H. S.; AMORIM, R. F. C. de; CARVALHO, F. O. de. Variabilidade espacial e preenchimento de falhas de dados pluviométricos para o estado de Alagoas. *Revista Brasileira de Meteorologia*, 2012. v. 27, p. 347 – 354, 2012. Citado na página 15.

WEEKLEY, R. A.; GOODRICH, R. K.; CORNMAN, L. B. An algorithm for classification and outlier detection of time-series data. *Journal of Atmospheric and Oceanic Technology*, 2010. v. 27, n. 1, p. 94–107, 2010. Citado na página 14.

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*, 2005. v. 30, p. 79–82, 2005. Citado na página 29.

WILLMOTT, C. J.; MATSUURA, K.; ROBESON, S. M. Ambiguities inherent in sums-of-squares-based error statistics. *Atmospheric Environment*, 2009. Elsevier Ltd, v. 43, n. 3, p. 749–752, 2009. Citado na página 29.

WU, E.; LIU, W.; CHAWLA, S. Spatio-temporal outlier detection in precipitation data. In: *Proceedings of the Second International Conference on Knowledge Discovery from Sensor Data*. Berlin, Heidelberg: Springer-Verlag, 2010. (SensorKDD'08), p. 115–133. Citado na página 14.

XIONG, J.; WU, B.; YAN, N.; ZENG, Y.; LIU, S. Estimation and validation of land surface evaporation using remote sensing and meteorological data in north china. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 2010. v. 3, n. 3, p. 337–344, Sept 2010. Citado na página 15.

YOUNG, S. J.; EVERMANN, G.; GALES, M. J. F.; HAIN, T.; KERSHAW, D.; MOORE, G.; ODELL, J.; OLLASON, D.; POVEY, D.; VALTCHEV, V.; WOODLAND, P. C. *The HTK Book, version 3.4*. Cambridge, UK: Cambridge University Engineering Department, 2006. Citado 2 vezes nas páginas I e 19.

ZANETTI, S.; SOUSA, E.; OLIVEIRA, V.; ALMEIDA, F.; BERNARDO, S. Estimating evapotranspiration using artificial neural network and minimum climatological data. *Journal of Irrigation and Drainage Engineering*, 2007. v. 133, n. 2, p. 83–89, 2007. Citado na página 16.

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. *Revista Brasileira de Engenharia Agrícola e Ambiental*, 2008. p. 174–180, 2008. Citado na página 17.

ZHANG, J. Advancements of outlier detection: A survey. *EAI Endorsed Trans. Scalable Information Systems*, 2013. v. 1, p. e2, 2013. Citado na página 13.

ZHANG, Z.; FENG, X. New methods for deviation-based outlier detection in large database. In: *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery - Volume 1*. Piscataway, NJ, USA: IEEE Press, 2009. (FSKD'09), p. 495–499. Citado na página 13.

Anexo I: Função para a Regressão Linear Múltipla

Código 10: Função para tratar a Regressão Linear Múltipla

```
1  private final int N;
2  private final int p;
3  private final Matrix beta;
4  private double SSE;
5  private double SST;
6
7  public MultipleLinearRegression(double[][] x, double[] y) {
8      if (x.length != y.length)
9          throw new RuntimeException("dimensions don't agree");
10     N = y.length;
11     p = x[0].length;
12     Matrix X = new Matrix(x);
13     // create matrix from vector
14     Matrix Y = new Matrix(y, N);
15     // find least squares solution
16     QRDecomposition qr = new QRDecomposition(X);
17     beta = qr.solve(Y);
18     // mean of y[] values
19     double sum = 0.0;
20     for (int i = 0; i < N; i++)
21         sum += y[i];
22     double mean = sum / N;
23     // total variation to be accounted for
24     for (int i = 0; i < N; i++) {
25         double dev = y[i] - mean;
26         SST += dev*dev;
27     }
28     // variation not accounted for
29     Matrix residuals = X.times(beta).minus(Y);
30     SSE = residuals.norm2() * residuals.norm2();
31 }
```

Anexo II: Exemplo de Dados do INMET

Tabela 26: Exemplo de dados obtidos de estações meteorológicas do INMET.

hora	temp_inst	umid_inst	pto_orvalho_inst	pressao	radiacao	vento_vel
0	23,8	89	21,8	968,4	-2,49	0
1	24	86	21,5	969	-0,47	1,3
2	23,8	86	21,4	969,4	-0,95	1,1
3	23,6	88	21,5	969,3	-3,1	0,8
4	23,4	89	21,5	968,6	-3,16	0,1
5	22,9	90	21,3	967,9	-3,5	0,5
6	22,7	91	21,2	967,6	-3,25	0
7	22,8	92	21,3	968,1	-2,59	0
8	22,8	91	21,3	968,8	-2,44	0,2
9	22,6	91	21,1	968,9	-2,7	0
10	22,9	91	21,3	969,3	72,06	0
11	24	87	21,8	970,4	461,9	1,7
12	24,7	82	21,5	971,2	978,7	3,5
13	26,1	77	21,7	971,6	1002	3,2
14	26,9	71	21,2	971,2	1298	0,6
15	28,7	64	21,2	970,7	1649	0
16	29,5	60	20,8	970,1	2045	0,2
17	29,7	57	20,3	968,6	1794	0
18	30,6	54	20,3	967,1	1152	0
19	30,1	55	20,1	966,4	763,3	0
20	29,7	60	21,1	966	839	0
21	27,1	70	21,2	966,6	181,9	0
22	26,2	74	21,2	967,5	42,34	0
23	24,7	81	21,3	968,4	-3,46	0
0	24,4	80	20,7	969,1	-3,3	0
1	24,2	82	20,9	970,2	-3,47	0
2	23,9	81	20,4	971	-3,52	0
3	24,1	79	20,3	970,8	-3,5	3,8
4	23,5	83	20,4	970,1	-3,08	1,6
5	23,3	83	20,3	969,4	-3,48	0,1
6	23,3	84	20,5	969,2	-3,25	1,4

Anexo III: Exemplo de Dados do TRMM

Tabela 27: Exemplo de dados obtidos de pontos do TRMM.

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
8,54624	1,22664	2,02526	3,65691	6,36387	3,54631	7,8796	2,53001	5,50507	7,95705
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
4,79625	7,40608	5,37477	6,75288	5,87116	4,75333	4,81294	7,143	4,60541	3,63525
0,72513	0,35047	0,37	0,36677	2,30947	0,39522	0,48981	0,20321	0,51409	0,54022
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0,01068	0	0	0	0
0,15538	0,09223	0,01947	0,14022	0,15396	0,29908	0,10648	0	0	0,22509
0,08287	0	0	0,21574	0,02052	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0,2142	0
0	0,15679	0,31158	0,05393	0	0,27772	0	0,0508	0,18207	0
8,7845	10,56958	11,66484	7,16281	11,61917	8,31033	14,42816	6,47238	9,71421	11,92994
0,60082	1,17132	2,00581	0,16181	0,34898	1,90134	0,34074	0,87382	1,18884	1,15923
0,06215	0	0	0,29125	0,45162	0	0,41527	0	0	0
0	0	0	0	0	0	0,17037	0	0,38556	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0,79765	3,39407	1,98632	2,95574	0,45162	2,41405	0,39396	3,91188	2,22773	1,44059
1,16021	0,45192	0,60367	1,06794	0,72875	0,64089	1,05415	1,08719	1,30664	1,80072
0,76656	7,50753	1,99607	0	3,46932	8,08602	1,01155	0,07112	1,27451	1,33929
0,89087	1,31888	0,82763	0,57171	1,33435	0,54476	1,48008	0,35562	0,35343	0,42767
3,51171	9,72105	8,35428	2,37322	2,95611	2,39268	5,70738	3,30224	4,74463	3,15129
6,71267	0,8854	0,38946	7,95028	11,50625	0,34181	4,65322	1,9102	1,33877	1,51936
0	0	0	0	0	0	0	0	0	0,14631
0,41436	0	0	1,00322	2,2684	0	0,13842	0,01016	0	0
0	0	0	0	0,13343	0	0	0	0	0
0,08287	0,79317	4,9853	0	0,23607	0,97202	0	0,07112	0,29988	0
8,2251	8,31915	4,18687	9,19084	3,9928	2,63835	0,84119	16,58232	5,23731	0,81032
1,83164	0,1163	0,50062	8,1518	3,16602	6,56611	3,11425	7,86701	0,24507	4,23674
4,96471	1,18635	1,56863	3,82152	6,70453	1,48303	3,65837	1,53389	3,48004	1,49317
0	5,61775	0,4895	0,21481	0	0	0	1,51065	0,57592	0
0,43381	0,36055	0,45612	0	0,38489	1,04152	0,97248	0,19754	0,80874	1,18969
5,03702	8,79299	10,6022	13,0022	3,60059	4,4944	7,71038	11,0742	10,84451	6,16695
8,2424	3,14034	8,18806	8,4797	6,97768	4,34722	11,47295	6,42608	7,18067	6,26407
0	0	0	0	0	0,18113	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0,22895	0,18609	0,089	0	0	1,40379	0,17365	0	0	0
0,0964	0	0	0,1922	0	0	0	0	0,3431	0,15781
0	0,87232	0,30037	0	0	1,50568	1,11141	0,0581	0	3,87256
0	0	0	0	0	0	0	0	0	0,02427
0	1,62833	0,92338	0	0	0,35094	0	0,1162	0,44113	0,29135
0	0	0,34487	0	0	0,31698	0,59043	0	0,29408	0,49772

Anexo IV: Exemplo de Dados do Ameriflux

Tabela 28: Exemplo de dados obtidos do Ameriflux (parte 1).

DOY	HRMIN	UST	TA	WD	WS	NEE	FC	SFC	H	LE	PREC
1	1500	0,366	29,301	6,863	1,688	-6,232	-3,309	-2,923	13,617	73,073	0
1	1600	0,542	29,469	40,686	2,228	0,759	-3,52	4,279	8,734	181,661	0
1	1700	0,211	29,002	24,68	1,439	6,264	-0,018	6,091	-5,829	30,827	0
1	1800	0,078	28,452	51,99	1,246	8,299	0,094	5,916	-3,142	2,659	0
1	1900	0,092	27,962	52,211	2,064	8,299	1,014	9,393	-6,872	5,68	0
1	2000	0,055	27,576	67,98	1,893	8,299	0,698	4,226	-3,126	2,904	0
1	2100	0,031	27,049	94,855	1,394	8,299	0,047	0,246	-0,463	0,038	0
1	2200	0,031	26,896	73,192	1,955	8,299	0,14	0,958	-0,367	0,27	0
1	2300	0,067	26,239	106,215	2,181	8,299	0,094	0,749	-1,864	-0,961	0
2	0	0,1	25,328	99,064	2,418	8,299	2,221	3,445	-8,873	1,521	0
2	100	0,204	25,107	89,206	2,48	8,299	4,83	-2,32	-17,409	2,695	0
2	200	0,197	25,112	95,034	2,49	8,299	3,502	0,129	-19,504	-0,758	0
2	300	0,162	25,243	90,608	2,234	8,299	3,662	0,664	-13,492	-0,931	0
2	400	0,256	25,007	83,246	2,641	6,276	11,857	-5,581	-28,806	7,112	0
2	500	0,384	24,266	66,872	3,086	13,189	13,672	-0,482	-18,193	-0,784	0
2	600	0,662	23,945	50,846	3,215	12,187	23,755	-11,568	-9,265	19,832	0
2	700	0,656	24,237	51,09	3,427	-13,167	3,268	-16,434	43,175	97,632	0
2	800	0,744	25,462	45,915	3,492	-27,636	-25,42	-2,216	125,255	348,698	0

Tabela 29: Exemplo de dados obtidos do Ameriflux (parte 2).

RH	PRESS	CO2	VPD	Rn	PAR	PARout	H2O	RE	GPP	CO2top
64,029	97,272	373,737	1,479	280,7	725,328	31,347	27,064	8,299	14,53	377,61
63,215	97,272	374,534	1,529	163,75	478,088	22,99	26,996	8,299	7,539	375,895
64,037	97,286	373,76	1,457	-8,8	72,25	6,485	26,66	8,299	2,035	378,125
66,672	97,366	375,667	1,313	-40,85	-1,061	3,578	26,975	8,299	0	379,09
66,762	97,446	376,067	1,28	-39,3	-1,144	3,032	26,368	8,299	0	381,065
69,091	97,521	378,94	1,164	-35,55	-1,434	3,1	26,67	8,299	0	380,8
69,22	97,548	376,95	1,129	-32,35	-1,328	3,024	26,011	8,299	0	382,2
71,004	97,539	377,344	1,052	-32,85	-1,27	3,012	26,391	8,299	0	382,725
76,425	97,521	376,677	0,82	-33,15	-1,269	2,83	27,241	8,299	0	382,225
86,712	97,455	379,8	0,439	-32,15	-1,246	2,532	29,392	8,299	0	385,97
87,512	97,437	385,354	0,405	-24,6	-1,308	2,723	29,082	8,299	0	391,55
89,189	97,432	382,444	0,351	-28,55	-1,326	2,591	29,707	8,299	0	385,745
89,07	97,432	380,85	0,357	-24,4	-1,356	2,622	29,832	8,299	0	386,045
89,587	97,437	377,6	0,336	-20,65	-1,336	2,674	29,607	6,276	0	386,39
93,524	97,517	398,54	0,199	-6,75	0	2,886	29,378	8,299	-4,891	405,73
97,313	97,579	407,987	0,081	36,85	91,392	4,998	29,943	8,299	-3,888	409,62
92,717	97,65	379,894	0,223	213,35	512,911	22,26	29,003	8,299	21,465	382,78
86,988	97,699	373,01	0,427	406,35	947,458	38,122	29,214	8,299	35,935	372,3

Apêndice A: Desenvolvimento de Sistemas Integrados ao *Framework*

O *framework* criado disponibiliza métodos para que facilmente possam realizar operações de tratamento de dados meteorológicos, mesmo que o método que esteja sendo executado envolva técnicas complexas de estatística ou da área de Inteligência Artificial.

Entretanto, para que o *framework* seja utilizado, é necessário um sistema que faça a ligação entre o usuário e as funcionalidades do *framework*. Dessa forma, o sistema desenvolvido irá abstrair o *framework* mostrando componentes gráficos que possam ser utilizados sem a necessidade de conhecimentos em linguagem de programação ou das técnicas utilizadas.

A.1 Integração com o Framework

Para que a integração seja realizada, deve haver uma importação do *framework*, por meio do arquivo no formato JAR, para que o projeto do sistema a ser desenvolvido tenha a capacidade de acionar os métodos disponíveis pelo *framework* e, assim, conseguir disponibilizar tais operações para o usuário do sistema.

Na Figura 16 está ilustrado um diagrama que demonstra a integração entre o sistema desenvolvido localmente e o *framework*. Pode ser observado três componentes principais: o computador local que tem acesso ao sistema desenvolvido, um banco de dados (BD) com os valores a serem analisados e tradados e o núcleo de processamento de dados (NPD) no qual consiste do *framework*.

Com esta integração, o sistema desenvolvido consegue uso direto do *framework*, oferendo total liberdade para carregamento de dados, uso dos métodos e manipulação dos resultados obtidos, como a exportação dos dados ou emissão de relatórios.

Lembrando que, para que o sistema desenvolvido possa utilizar os méto-

dos do *framework*, os códigos exemplificados do Capítulo 4 devem ser utilizados. Assim, tendo conhecimentos em linguagem de programação, em específico na linguagem Java, é possível o desenvolvimento dos próprios sistemas utilizando os recursos do *framework*, atendendo a necessidade de cada demanda.

Figura 16: Diagrama mostrando a integração entre um sistema local e o *framework*.

Por outro lado, por mais que o uso direto do *framework* ofereça total liberdade para carregamento de dados e uso dos métodos, há também uma necessidade por parte de pessoas que não possuem os conhecimentos em linguagem de programação para tratar os seus dados. Pensando nisso, foi desenvolvido um sistema *web-based*, integrado ao *framework* criado neste trabalho, para disponibilizar em um ambiente *web* os métodos de preenchimento de falhas e de detecção de *outliers*.

A.2 Sistema Web-Based

Existem produtos, como Menzer (2014), que disponibilizam funcionalidades de tratamento de dados. Como características comum entre esses produtos estão a especificidade de uma única variável para tratamento e a determinação de parâmetros às vezes complexos para o usuário. No sistema desenvolvido neste trabalho, buscou-se a possibilidade de tratar diversas variáveis climáticas e a simplicidade no uso das operações de tratamento de dados.

Na Figura 17 é mostrado um diagrama da solução *web-based*. Nessa solução, um computador acessando a Interface Web (IW) utiliza, por meio da internet, as funcionalidades disponíveis no NPD. A parte de IW foi desenvolvida

também com a linguagem de programação Java e a tecnologia JavaServer Faces (JSF).

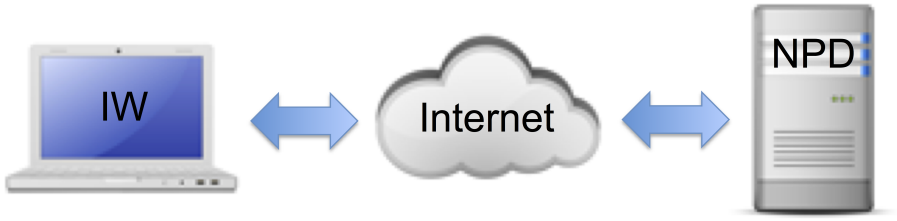

Figura 17: Diagrama mostrando a integração entre um sistema *web-based* e o *framework*.

Este sistema permite que o usuário acesse uma página na internet utilizando uma interface com boa usabilidade e realize as etapas necessárias para carregar os dados no sistema, validá-los e processá-los. Em cada etapa, uma nova versão dos dados é gerada e pode ser visualizada e exportada. As Figuras 18, 19, 20, 21 e 22 mostram telas do sistema desenvolvido.

Como um dos objetivos deste sistema era a simplicidade, várias decisões foram tomadas para limitar o número de opções a serem configuradas e, assim, deixar o uso do sistema muito mais fácil e direto. O formato de dados, por exemplo, foi limitado para o tipo CSV, com cabeçalho para identificar o nome das variáveis. Com este sistema, o usuário pode selecionar um arquivo CSV de sua máquina e carregar os dados no sistema (Figura 18).

Figura 18: Tela de carregamento de dados do sistema *web-based*.

Depois de carregado os dados, pode ser visualizado todos os valores contidos no arquivo selecionado (Figura 19). Há também as opções de iniciar os dois

tipos de operações para tratamento dos dados. É possível verificar também que existe a opção de exportação dos dados, podendo nesse caso selecionar alguma versão gerada pelo sistema após uma operação.

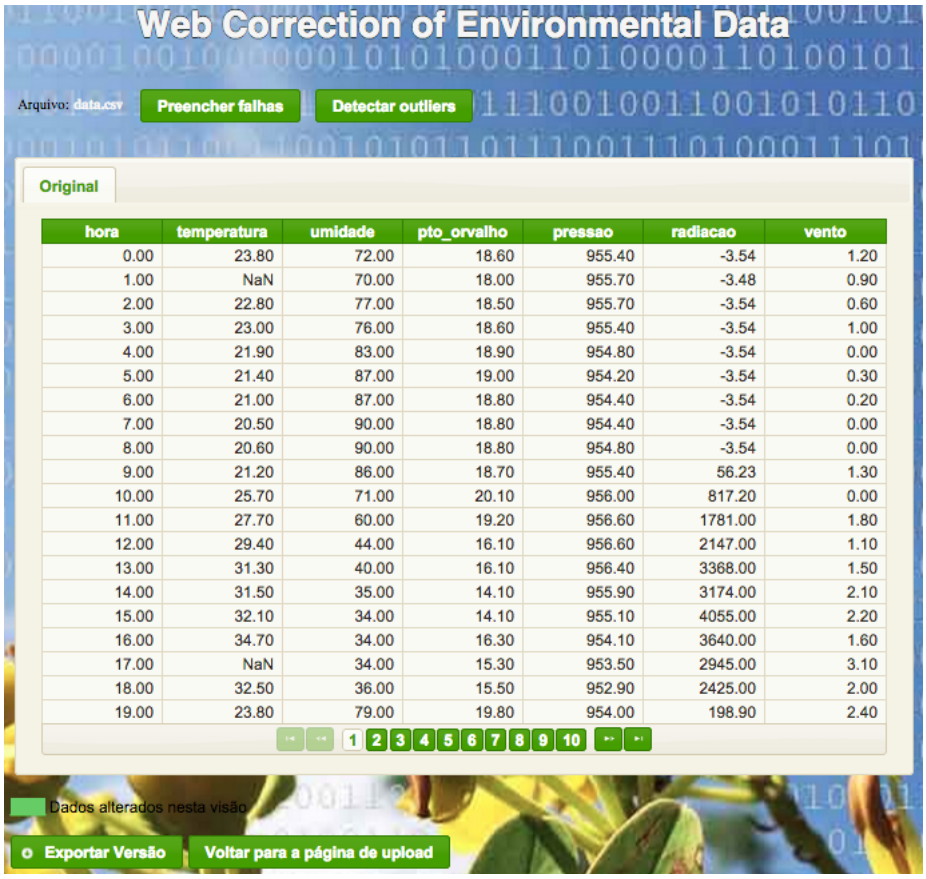

Figura 19: Tela de visualização dos dados no sistema *web-based*.

Uma das operações existentes, o preenchimento de falhas, pode ser iniciado clicando no respectivo botão. Pode ser visto que o único parâmetro a ser selecionado é a variável a ser tratada (Figura 20). Tanto o método quanto os parâmetros do método são atribuídos pelo próprio sistema, visando a facilidade e simplicidade da utilização do sistema. Para esse sistema, todas as falhas encontradas na coluna da variável especificada seriam tratadas. As falhas são os dados em branco ou sinalizado como *Not a Number* (NaN).

Depois de enviar o comando para iniciar o tratamento, o sistema processa os dados e gera uma nova versão da série de dados, desta vez com os dados da coluna selecionada corrigidos.

Um processo semelhante acontece com a operação de detecção de *outliers*. Selecionando esta opção, deve ser escolhido apenas a variável climática que deve ser tratada e determinado um valor de corte (Figura 21). O valor de corte é uma porcentagem que determina quando um dado é classificado como *outlier* ou como

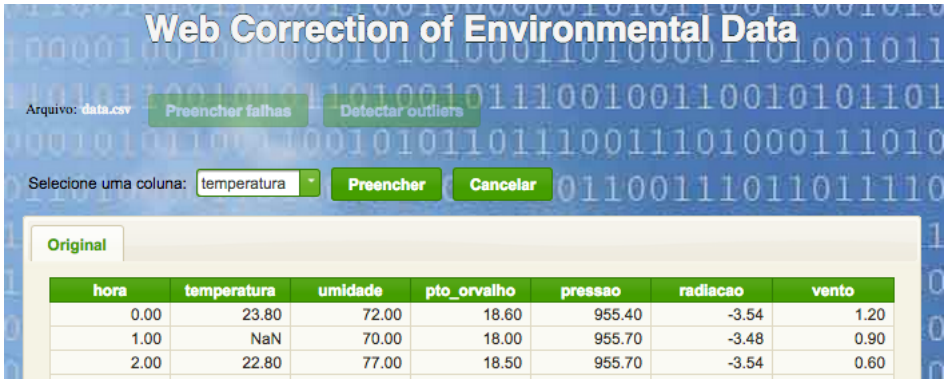

Figura 20: Opção para preenchimento de falhas no sistema *web-based*.

um dado normal. E, mais uma vez, tanto o método quanto outros parâmetros não precisam ser determinados, simplificando o processo. Quando iniciado a operação, os dados são analisados e sinalizados quando os seus valores estiverem distante do comportamento normal da série de dados.

Figura 21: Opção de detecção de *outliers* no sistema *web-based*.

A cada operação realizada, uma nova versão da série de dados é gerada (Figura 22). O sistema permite que mais de uma operação seja realizada. Então, é possível realizar a detecção de *outliers* em uma coluna e, na nova versão, realizar o preenchimento de falhas na mesma coluna. Dessa forma, várias combinações de resultados podem ser realizadas.

Isso permite também que uma versão passada possa ser tratada novamente. Isso quer dizer que se um tratamento não teve desempenho satisfatório, essa versão pode ser ignorada e, a partir da versão anterior, iniciar novos tratamentos. Com isso, é possível analisar vários resultados para a mesma variável climática, auxiliando na tomada de decisões com relação ao tratamento desses dados.

A exportação dos dados pode ser realizada para qualquer versão gerada. Essa funcionalidade permite que sejam obtidos novamente os dados enviados an-

teriormente, mas desta vez com os dados tratados.

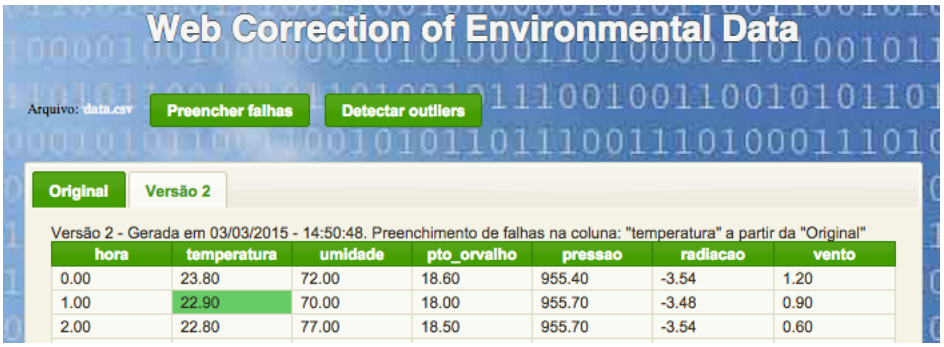

Figura 22: Opção para visualizar versões da base de dados após cada operação de tratamento de dados no sistema *web-based*.

Com o *framework* criado, é possível desenvolver sistemas com funcionalidades complexas de tratamento de dados meteorológicos, já que não há dificuldades de realizar a integração entre o *framework* e um novo sistema. O exemplo do sistema *web-based* mostrado demonstra como torna-se trivial a detecção de *outliers* e o preenchimento de falhas por qualquer pessoa que tenha conhecimentos básicos em informática.