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

MiMi: Plataforma computacional para mineração de dados micrometeorológicos

Allan Gonçalves de Oliveira

Orientador: **Prof. Dr. Josiel Maimone de Figueiredo**
Coorientadora: **Profa. Dra. Marta Cristina de Jesus Albuquerque Nogueira**

Cuiabá - MT
Maio/2015

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

MiMi: Plataforma computacional para mineração de dados micrometeorológicos

Allan Gonçalves de Oliveira

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

Maio/2015

Dados Internacionais de Catalogação na Fonte.

D278m de Oliveira, Allan Gonçalves.
MiMi: Plataforma computacional para mineração de dados micrometeorológicos / Allan Gonçalves de Oliveira. -- 2015
102 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. Mineração de Dados. 2. Séries Temporais. 3. Micrometeorologia. I. Título.

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE FÍSICA
Programa de Pós-Graduação em Física Ambiental

FOLHA DE APROVAÇÃO

TÍTULO: **MiMi - PLATAFORMA COMPUTACIONAL PARA MINERAÇÃO DE DADOS MICROMETEOROLÓGICOS**

AUTOR: **ALLAN GONÇALVES DE OLIVEIRA**

Tese de Doutorado defendida e aprovada em 06 de maio 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

Profa. Dra. Marcela Xavier Ribeiro
Examinadora Externa
Centro de Ciências Exatas e de Tecnologia
Universidade Federal de São Carlos - UFSCar

Prof. Dr. Todor Ganchev - Examinador Externo
Department of Electronics
Technical University of Varna – TUV - Bulgária

DEDICATÓRIA

À Deus, aos meus pais, Jessé e Márcia, à minha esposa, Bianca, aos meus irmãos Renan, Isis e Marquinho.

Agradecimentos

- Agradeço primeiramente à Deus, o provedor de tudo que tenho, sou e que posso vir a ser.

- Aos meus pais eu agradeço pelo apoio que me deram em todos os momentos de minha vida. Com certeza eles são a base de tudo que sou. Sem a dedicação que sempre tiveram nada disso seria possível. A eles eu agradeço e dedico este trabalho.

- A minha esposa Bianca, que com tanto carinho e dedicação tem me apoiado não só no doutorado, mas, em todos os projetos que me proponho a realizar. Sem a compreensão dela nada disso seria possível. A ela também dedico este trabalho.

- Ao meu orientador Josiel que tem me guiado na vida acadêmica desde o primeiro semestre da graduação. Além de orientador se tornou amigo da família.

- A professora e hoje colega de profissão Claudia Aparecida que também se tornou amiga e tem uma participação enorme na minha formação.

- A todos os meus professores do Instituto de Computação da Universidade Federal de Mato Grosso.

- Ao professor Paraná e professora Marta que desde o inicio dessa jornada tem apoiado em tudo que é preciso tornando-se além de professores, orientadores e amigos.

- Ao professor Todor Ganchev pela orientação no doutorado sanduíche na Bulgária que com certeza trouxe um crescimento incalculável na minha vida pessoal e acadêmica. Agradeço também aos colegas de laboratório no doutorado sanduíche, Firgan e Nikolay.

- Aos membros do nosso grupo de pesquisa pela ajuda nos momentos necessários. Em especial ao colega Raphael Rosa que está sempre pronto a ajudar.

- Ao meu colega de doutorado, doutorado sanduíche e de trabalho, Thiago Meirelles que muito me ajudou no principalmente no período em que estivemos estudando na Bulgária.

- Aos colegas, amigos desde o mestrado, Thiago Rangel, Leone, Jonathan e Paula que com certeza tem parte neste trabalho.

- Ao CNPq pelo auxílio financeiro em meu doutorado sanduíche na Bulgária.

- E, por fim, a todos os outros professores, técnicos e alunos do PGFA que, direta ou indiretamente, ajudaram em meus estudos.

"Se quer ir rápido, vá sozinho. Se quer ir longe, vá em grupo."

Autor desconhecido

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

RESUMO

OLIVEIRA, A. G. MiMi: Plataforma computacional para mineração de dados micrometeorológicos. Cuiabá, 2015, 84f. Tese (Doutorado em Física Ambiental) - Instituto de Física, Universidade Federal de Mato Grosso.

A evolução da capacidade de aquisição de dados em equipamentos eletrônicos nos ultimos anos impõem desafios ao gerenciamento e análise dos dados coletados em diversas áreas do conhecimento. Pesquisas envolvendo dados micrometeorológicos têm se beneficiado desses avanços e também vivenciado as dificuldades em relação as analise dos dados. Diante disso, o uso de técnicas computacionais para auxiliar o processo de análise dos dados é uma necessidade para usuários desse tipo de dado. Assim, este trabalho apresenta o desenvolvimento da Plataforma MiMi, uma plataforma computacional para mineração de dados micrometeorológicos. A plataforma desenvolvida tem o intuito de apresentar uma arquitetura de software genérica e flexível, na qual é introduzido o conceito de tratar as séries temporais como um novo tipo de dado dentro da plataforma e os algoritmos que o manipulam são considerados operadores do mesmo. Com a definição de operando e operadores foi possível definir a execução de expressões de domínio que representam um fluxo de processamento específico para cada atividade de mineração de dados em um domínio de série temporal. A plataforma MiMi foi validada com a execução de três atividades de mineração de dados, agrupamento, busca por similaridade e detecção de padrões desconhecidos, os testes foram executados utilizando duas bases de dados micrometeorológicos do estado de Mato Grosso, Brasil.

Palavras-chaves: Mineração de Dados, Séries Temporais, Micrometeorologia.

ABSTRACT

OLIVEIRA, A. G. MiMi - Micrometeorological Data Mining Platform, 2015, 84f. Phd Thesis (Doutorado em Física Ambiental) - Instituto de Física, Universidade Federal de Mato Grosso.

The evolution of data acquisition capabilities in electronics devices in recent years created new challenges to the management and analysis of data collected. Researches involving micrometeorological data have been benefited from these developments and also experienced difficulties in relation to data analysis. Thus, the use of computational techniques to aid the process of data analysis is mandatory for their users. This work presents the development of MiMi Platform, a computational platform for mining micrometeorological data. The MiMi platform is intended to provide a general and flexible component based software architecture, which main characteristic involves an algebraic approach for data mining, where the time series are treated as embedded data type in the platform and algorithms that handle them are operators. With the definition of operand and operators, is possible to define domain expressions that represent a specific processing flow for each data mining activity in a time series domain. For this, we defined a domain manager that performs multiple operators in a predefined order. The MiMi platform was validated with the execution of three data mining activities, clustering, similarity search and unknown patterns detection, the tests were performed using two real dataset from micrometeorological stations localized in Mato Grosso, Brazil.

Keywords: Data Mining, Time Series, Micrometeorological.

Capítulo 1

Introdução

O monitoramento ambiental tem ganhado cada vez mais notoriedade nas mais diversas áreas de conhecimento em todo o mundo. Isso se deve principalmente porque acredita-se que a atividade humana tem alterado o equilíbrio ambiental no decorrer da nossa existência e, esse desequilíbrio tem sido notado em todo o mundo. Além disso, o monitoramento ambiental pode auxiliar diversas atividades, como a agricultura, defesa civil, turismo, etc.

Parte desse monitoramento é feito por meio de estudos do comportamento das variáveis meteorológicas de uma determinada região, o que envolve o comportamento do clima dessa região. Geralmente as medições são feitas por períodos longos de tempo para que se tenha séries temporais dessa variáveis e seja possível analisar o seu comportamento passado e tentar prever seu futuro.

Inicialmente a coleta dos dados eram feitas por meio de anotações manuais das medidas realizadas por equipamentos instalados em estações meteorológicas. Isso obviamente coloca o estudo sujeito a erros humanos decorrentes das anotações e impõem limitações à abrangência do mesmo. É comum que nas estações meteorológicas manuais os dados sejam anotados 3 vezes ao dia, ou até uma vez apenas dependendo da dificuldade em acessá-las. Isso claramente limita a análise a ser feita quanto a pequenas variações que poderiam ser analisadas quando se tem uma menor granularidade na coleta de dados.

Entretanto, o avanço nas áreas de eletrônica e informática permitiram melhorias no processo de aquisição e armazenamento de dados. Com isso, criaram-se as estações meteorológicas automatizadas. Nessas, os dados podem ser coletados e armazenados seguindo as especificações do equipamento, permitindo coletas em baixas granularidades como a cada segundo, minuto com a continuidade de ocorrer 24 horas por dia.

Esse avanço permitiu que a coleta de dados sejam ainda mais eficiente,

levando a estudos mais abrangentes, com maiores períodos de estudo e altas resoluções temporais para estudar eventos mais pontuais. No entanto, a grande quantidade de dados gerados pode acarretar em dificuldades de manipulação, controle e interpretação dos mesmos, tanto por limitações das técnicas utilizadas na análise ou limitações humanas em manipular a massa de dados disponíveis.

Não é incomum que as bases de dados geradas nas atividades envolvendo dados meteorológicos ultrapassem gigabytes de dados, o que praticamente impossibilita a análise de todo o conjunto de dados da maneira tradicional, utilizando apenas planilhas eletrônicas. É um processo interativo, em que o usuário executa diversas atividades com o objetivo de encontrar explicações para determinados eventos ou ainda relações entre eventos a partir dos dados disponíveis. Essas atividades envolvem por exemplo seleções de um determinado subconjunto de dados, transformações dos mesmos e ainda geração de novos dados a partir dos que foram coletados, o que pode aumentar ainda mais a dificuldade de análise a medida que aumenta a quantidade de dados.

Tradicionalmente, a solução mais adotada é a fragmentação em várias planilhas de dados compostas por subconjuntos dos dados organizados de acordo com o tempo. Ainda assim, para cada subconjunto de dados gerados, o mesmo processo deve ser executado para realizar as análises dos mesmos, causando retrabalho de todo o processo interativo de análise.

Percebe-se então que a dificuldade na análise dos dados decorre de uma junção de fatores, como a quantidade de dados e organização do processo de análise, visto que o mesmo é composto por etapas interativas. Nesse contexto, com o retrabalho gerado, o conhecimento que é extraído de um processo de análise não é reaproveitado. A cada análise em conjunto de dados semelhantes e mesmos objetivos, todo o processo é executado novamente, refazendo por exemplo a calibração dos modelos utilizados, quando esses poderiam ser reaproveitados.

Dessa forma, é necessário que se tenha uma estrutura computacional para auxiliar o processo de análise dos dados meteorológicos. Uma alternativa é a utilização de técnicas de Mineração de Dados para analisar os dados. Algumas ferramentas computacionais, tanto comerciais quanto gratuitas oferecem um conjunto de soluções para se trabalhar com Mineração de Dados nesse tipo de dado. Entretanto, essas ferramentas fornecem um conjunto de soluções muito genéricas, apenas fornecendo algoritmos para serem utilizados. Além de fornecer algoritmos para utilização, faz-se necessário uma estrutura para tratamento específico desses dados e que atenda ao fluxo de processamento inerente a esse processo, que envolve diversas etapas interativas e que permita a reutilização do conhecimento

extraído de processos anteriores.

Essa estrutura deve permitir que algoritmos sejam implementados e testados de forma simples, automatizando detalhes do processo de análise.

Neste trabalho é apresentado o desenvolvimento da MiMi - Plataforma Computacional para Mineração de Dados Meteorológicos. A arquitetura apresenta um novo conceito na arquitetura de software para mineração de dados que é o tratamento da série temporal como um novo tipo de dado e os algoritmos de mineração como sendo operadores que manipulam o novo tipo de dado.

1.1 Objetivo Geral

O objetivo geral deste trabalho é construir uma plataforma de componentes de software para auxiliar o processo de Mineração de Dados Micrometeorológicos.

1.1.1 Objetivos Específicos

Para alcançar o objetivo geral é necessário atender aos seguintes objetivos específicos:

- Escolha de um modelo de processo de mineração de dados a ser seguido;
- Estudo e implementação de algoritmos de redução de dimensionalidade de séries temporais;
- Definição da organização das Classes que representa a estrutura da plataforma proposta de software proposta;
- Definição de uma álgebra para manipulação de séries temporais;
- Definição e implementação de uma arquitetura de software baseada nos componentes da álgebra;
- Implementação dos principais algoritmos de mineração de dados para Agrupamento, Busca por Similaridade e Detecção de Padrões utilizando as características definidas na plataforma proposta;
- Testes dos algoritmos implementados com dados reais.

1.2 Justificativa

O desenvolvimento dessa plataforma computacional para auxiliar o processo de desenvolvimento e aplicação de algoritmos de mineração de dados em séries de dados meteorológicos se justifica devido a importância desse tipo de dado no contexto atual da humanidade, momento em que se tem grande preocupação com os aspectos ambientais. Muitos dados estão sendo gerados e cada vez mais as técnicas computacionais tendem a ser aplicadas nesse tipo de dado. Com isso novos algoritmos devem ser construídos e as técnicas já existentes devem ser aplicadas e validadas.

Além da aplicação dos algoritmos, para que isso ocorra é estabelecido um conjuntos de passos que devem ser executados, esse passos são interativos, dependem de informações e conhecimento de um especialista de domínio para manipulação dos parâmetros e definição de quais tarefas devem ser executadas. Assim, uma arquitetura computacional que dê suporte a esse processo, facilitando o desenvolvimento, automatizando processos e deixando transparente detalhes específicos do tratamento desses dados pode auxiliar pesquisadores e usuários dessas técnicas e dados.

1.3 Organização do Trabalho

Este trabalho está organizado da seguinte forma: No Capitulo 2 é apresentada a fundamentação teórica necessária para entender o contexto deste trabalho. No Capítulo 3 é mostrada a forma com que o trabalho foi conduzido e executado, são apresentados os dados utilizados e técnicas implementadas e testadas bem como a tecnologia aplicada. O Capítulo 4 apresenta os resultados alcançados neste trabalho a partir do objetivo traçado e a validação da plataforma proposta com a aplicação de algoritmos. Por fim, são apresentadas as conclusões a respeito do trabalho.

Capítulo 2

Fundamentação Teórica

Nas diversas áreas do conhecimento o tipo de dado mais utilizado são as séries temporais (LIN et al., 2012). Esse tipo de dado possui uma dimensão temporal e sua representação se dá por meio de uma sequência ordenada de valores na qual cada um corresponde ao valor observado da variável num determinado instante de tempo. É possível citar a formação de séries temporais em diversos contextos como: valores diários de ações de uma dada companhia na bolsa, quantidade de venda de um determinado produto por mês, produção diária de uma máquina na industria e a temperatura de uma determinada área a cada hora. Todos esses dados, entre outros, são variáveis que tomam valores que podem variar de uma unidade de tempo (minuto, hora, dia, mês, ano, etc) para outra.

Wooldridge (2000) define a séries temporais, como uma sequência de observações de uma variável ao longo do tempo, tomados em intervalos regulares durante um período de tempo. Elas podem ter duas classificações de acordo com a quantidade de variáveis no sistema: univariadas ou multivariadas (PALIT; POPOVIC, 2005). As univariadas são séries obtidas a partir da da medição/observação de uma única variável, por exemplo, os valores de uma única variável física como a Temperatura. As multivariadas são geradas a partir da medição/observação simultânea de duas ou mais variáveis, por exemplo a medição de Temperatura, Umidade Relativa do Ar e Fluxo de Gás Carbônico em uma torre meteorológica.

A análise de séries temporais multivariadas usualmente levam em consideração a relação de influência que uma variável exerce sobre a outra. Isso permite análises mais precisas, no entanto essas análises são mais complexas de serem realizas devido a grande quantidade de dados. Para analisar esse tipo de dados são utilizadas técnicas de Mineração de Dados.

No processo de mineração de dados em Séries Temporais geralmente busca-se retificar a habilidade humana para analisar a forma dos dados, ou seja,

seu comportamento (ESLING; AGON, 2012a). Para Antunes e Oliveira (2001) o objetivo é descobrir relações escondidas entre sequências e subsequências de eventos.

A Mineração de Dados em Séries Temporais tem aplicação em diversas áreas, vários exemplos podem ser citados, Song e Li (2008) apresentam um estudo que envolve previsão econômica no mercado de turismo, em Zhong et al. (2007) é apresentado um trabalho que utilizada mineração de dados na área de redes de computadores com sistema de detecção de intrusos. Em Burkom et al. (2007) são analisados métodos de previsão de séries temporais para aplicação em Vigilância Biológica baseado em dados de indicativos de saúde. Em Ouyang et al. (2010) e Mishra et al. (2013) detecção de padrões e busca por similaridade são aplicados em séries temporais de dados de hidrologia.

Os dados de meteorologia e agrometeorologia também podem ser analisados com a utilização de técnicas de mineração de dados, pois, o foco na análise desse tipo de dado está em seu comportamento e relações entre as diversas variáveis. Por exemplo em Curado et al. (2014) o interesse foi analisar os padrões de comportamento dos fluxos de energia entre as estações seca e chuvosa em uma área de cerrado em Mato Grosso - Brasil. Novais et al. (2013) utilizaram séries de dados micrometeorológicos para modelagem do comportamento geotermal em uma área do Pantanal Norte Matogrossensse. Em Rodrigues et al. (2013) são analisados os padrões do balanço de energia em diferentes condições sazonais. No trabalho de CURADO et al. (2011) foi utilizado algoritmo genético para modelar por meio de dados meteorológicos os parâmetros de uma equação empírica que diz respeito a emissividade atmosférica no cerrado do estado de Mato Grosso.

O processo de análise de séries temporais de dados meteorológicos envolve várias etapas que interagem entre si com uma relação de dependência entre elas, pois, o resultado de uma etapa influencia diretamente o da próxima. Nesse contexto, o papel do especialista de domínio é importante porque guia o processo por meio de calibragens, parametrização e escolha dos métodos mais adequados.

Esse processo de análise dos dados consiste em um conjunto de operações para se extrair informações dos mesmos, o que pode ser visto como uma tarefa de mineração de dados. Assim, neste capítulo o processo de análise de dados meteorológicos é apresentada no contexto de mineração de dados, utilizando um modelo de processo largamente utilizado, o CRISP-DM (SHEARER, 2000; WIRTH, 2000).

São apresentados primeiramente alguns detalhes dos dados meteorológicos, como o que são e suas características gerais. Após é apresentado o processo

de análise e interpretação desses dados por meio de um modelo de processo de mineração de dados.

2.1 Dados Meteorológicos

Dados meteorológicos são valores que representam características do Clima de uma determinada região. O Clima por sua vez, de acordo com Silva (2015) "*..é uma síntese de natureza estatística do estado da atmosfera ou das suas fronteiras, referente a uma determinada área e a um determinado período de tempo*". Quando a análise desses dados são em escala local, eles são chamados de micrometeorológicos.

Esse tipo de dado é utilizado em diversos contextos. Na defesa civil, por exemplo, pode ser utilizado na previsão de inundações (DAMLE; YALCIN, 2007). Na agricultura pode ser utilizado na determinação da melhor temperatura de desenvolvimento de uma determinada espécie Matos et al. (2014), Ventura et al. (2014) e no cálculo da evapotranspiração, soma da água transpirada pelas plantas e evaporada de uma área, que serve para determinar a quantidade de água a ser utilizada na irrigação (SOUZA et al., 2011; SOUZA A. P. ; SILVA, 2014; VOURLITIS et al., 2015). Na biologia os dados meteorológicos são utilizados por exemplo na análise do desenvolvimento de determinadas espécies vegetais em uma área específica (DALMOLIN et al., 2015; DALMAGRO et al., 2014). No turismo podem ser utilizados para a determinação dos períodos ideais para a realização de certas atividades (KOZIEVITH, 2006). Na engenharia civil e urbanismo esses dados são utilizados por exemplo no controle do conforto térmico como visto em (CANEPPELE L. B. ; NOGUEIRA, 2014; SANTOS F. M. M. ; OLIVEIRA, 2014).

A manipulação dos dados meteorológicos em seus diversos contextos é realizada com um conjunto de operaçõesa interativas, nas quais é de vital importância o conhecimento do especialista de domínio para guiar essas operações. Alcântara et al. (2010) apresenta o ciclo de utilização desses dados no contexto de gerenciamento ambiental, esse ciclo é ilutrado pela Figura 1.

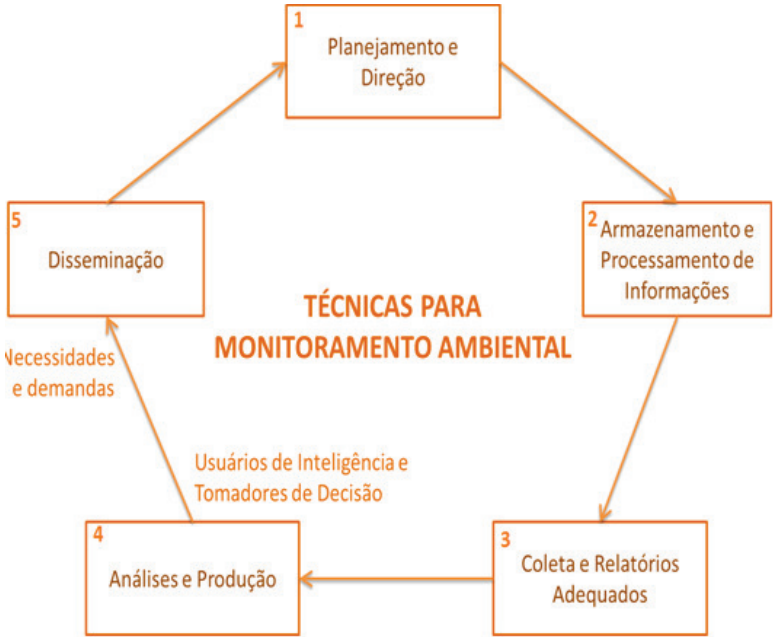


Figura 1: Fluxo de atividades no monitoramento ambiental. FONTE: Alcântara et al. (2010)

O conjunto de atividades apresentado por Alcântara et al. (2010) é composto por 5 etapas, descritas a seguir:

- **Planejamento e Direção:** Nesta etapa é feito o planejamento das próximas atividades a serem desenvolvidas. São elencados os objetivos a serem alcançados e estes guiam o processo. Questões como escopo da pesquisa, dados a serem coletados e período de coleta são definidos neste momento.

- **Armazenamento e Processamento das Informações:** Neste momento os dados da pesquisa são armazenados e processados. São realizadas tarefas que viabilizam a interpretação dos dados, que é feita na próxima etapa.

- **Coleta e Relatórios Adequados:** É o momento em que os dados são analisados, busca-se extrair informações úteis para uma tomada de decisão futura. Esta etapa é fortemente influenciada pelos processamtos realizados na etapa anterior.

- **Análise e Produção:** Nesta fase as informações extraídas na etapa anterior são analisadas. São testadas hipóteses levantadas na fase de planejamento, por exemplo. Os dados podem ainda ser submetidos a simulações para auxiliar as validações e questionamentos.

- **Disseminação:** Na etapa de disseminação os modelos já validados são

aplicados, são tomadas decisões baseadas no conhecimento extraído do processo.

Todo o processo de manipulação e análise dos dados ambientais/meteorológicos como mostrado, consiste num conjunto de atividade interativas em busca da extração de informações úteis do conjunto de dados disponível. Esse processo se assemelha à mineração de dados, termo mais genérico e mais comumente utilizado na área da computação. Dessa forma, a seguir o processo de manipulação e análise dos dados meteorológicos é apresentado sob a perspectiva da mineração de dados. Cada fase do processo de mineração de dados é apresentado no contexto dos dados meteorológicos.

2.2 Mineração de Dados (MD)

Mineração de Dados (MD) é uma área de pesquisa muito dinâmica e que tem se desenvolvido muito nos últimos anos motivado pela grande quantidade de dados que tem sido gerados e disponibilizados para análise (KURGAN; MUSILEK, 2006). A MD surgiu como um nova perspectiva na análise de dados, na qual a análise é vista com um caráter exploratório, devido a grande quantidade de dados que a evolução tecnológica permitiu que fossem coletados e armazenados (GONçALVES et al., 2001). Para Hair et al. (2009) é uma visão da análise dos dados que baseia-se nas tendências de avalanche de formação de dados e o questionamento sobre o significado e conteúdo dos dados.

Para Klosgen e Zytkow (1996) a MD é uma etapa de um processo maior, constituído por várias etapas, cada uma destinada a realização de uma tarefa particular chamado Descoberta de Conhecimento (DC). Outros autores como Fayyad et al. (1996) e Tan et al. (2005) também veem a MD como parte integrante do processo de DC, já Han et al. (2011) por exemplo, tratam a MD como sinônimo de DC.

Mineração de dados é um assunto totalmente interdisciplinar, pois deve considerar fatores desde a aquisição dos dados, passando pelo armazenamento e chegando à interpretação dos dados. Com isso não há uma única definição para esse processo. Han et al. (2011) afirmam que o termo Mineração de Dado não representa o real significado do processo, pois, busca-se conhecimento ou informações úteis em uma grande quantidade de dados, sendo assim, o nome apropriado seria Mineração de Conhecimento.

No processo de MD há diversas tarefas que devem ser executadas, algumas delas mais de uma vez dependendo do contexto, é um processo interativo

entre o usuário e o computador (CôRTES et al., 2002). Há então um conjunto de tarefas definidas que devem ser executadas no processo de mineração de dados, para isso, foram apresentados para a comunidade alguns modelos de processos que definem quais etapas devem ser seguidas. No tópico seguinte são apresentados alguns desses modelos de processo e é detalhado o modelo de processo adotado nesse trabalho. São apresentados os detalhes de cada etapa com considerações no contexto dos dados meteorológicos.

2.3 Modelos de Processos de Mineração de Dados

Em Pressman (2005) um modelo de processo é definido como um conjunto de tarefas a serem executadas para desenvolver um elemento particular, bem como os elementos que são produzidos em cada tarefa e os elementos que são necessários para realizar uma tarefa. Muitos autores definiram modelos de processos para a MD. Os modelos propostos definem os passos necessários para realizar MD mas não dizem como fazer, por isso Marbán et al. (2009) dizem que essas definições não podem ser chamadas de metodologia e sim modelos de processo.

Todos os modelos de processos consistem em várias etapas executadas sequencialmente incluindo *loops* e interações entre etapas (KURGAN; MUSILEK, 2006). O primeiro modelo de processo relatado é o modelo de 9 etapas proposto por Fayyad et al. (1996) ilustrado pela Figura 2.

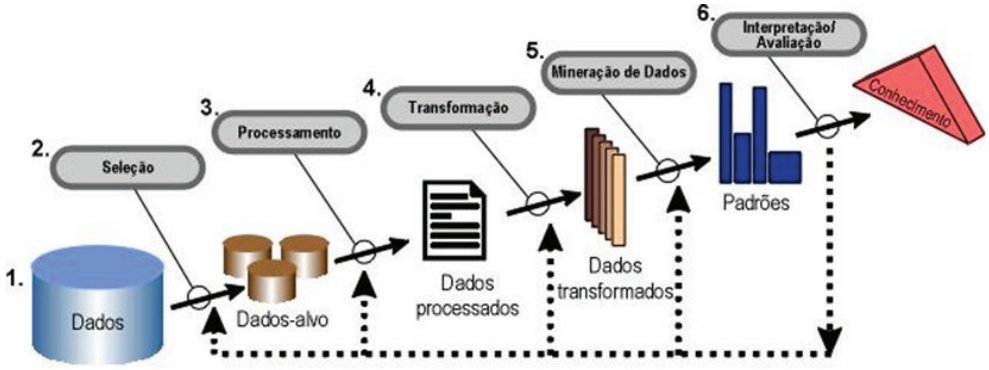


Figura 2: Mineração de Dados como parte do Processo de Descoberta de Conhecimento. Adaptado de (FAYYAD, 1998).

Cabena et al. (1998) propôs um modelo composto por 5 etapas. No mesmo ano Anand e Büchner apresentaram um modelo de 8 etapas (ANAND; BüCHNER, 1998). No ano de 2000 algumas empresas que atuam com análise de dados (Teradata,SPSS – ISL, Daimler-Chrysler and OHRA) analisando o que havia até então a respeito do processo de MD, propuseram um modelo de processo

de 6 etapas o CRISP-DM (*CRoss Industry StandardProcess for Data Mining*) (SHEARER, 2000; WIRTH, 2000). Ainda em 2000, Cios et al. (2000) apresentaram outro modelo com 6 etapas adaptado do CRISP-DM. As adaptações foram feitas para atender a demanda acadêmica, basicamente adicionando *feedbacks* explícitos e alterando as definições das últimas etapas para que fique claro que o conhecimento gerado em um domínio específico pode ser aplicado em outros domínios. A Tabela 1 mostra uma comparação entre as etapas de cada um dos modelos citados anteriormente.

Tabela 1: Comparação entre modelos de processo de MD. Adaptado de Kurgan e Musilek (2006)

Model	Fayyad et al.	Cabena et al.	Anand & Buchner	CRISP-DM	Cios et al.
N. de Passos	9	5	8	6	6
Passos	Desenvolvimento e entendimento do domínio de aplicação	Determinação dos objetivos de negócio	Identificação de Recursos Humanos	Entendimento do negócio	Entendimento dos dados
			Especificação do Problema		
	Criação de um conjunto de dados alvo	Preparação dos dados	Prospecção dos dados	Entendimento dos dados	Entendimento dos dados
	Limpeza e pré-processamento dos dados		Elicitação do domínio de conhecimento		
	Projeção e redução dos dados		Identificação da metodologia	Preparação dos dados	Preparacao dos dados
	Escolha das tarefas de mineração		Pré-processamento dos dados		
	Escolha dos algoritmos de mineração				
	Mineracao de dados	Mineração de dados	Descoberta de padrões	Modelagem	Mineração dos dados
	Interpretação dos padrões minerados	Elicitação do dominio de conhecimento	Pós-processamento do conhecimento	Avaliação	Avalicacao da Descoberta de Conhecimento
	Consolidação do donhecimento descoberto	Assimilação do conhecimento		Desenvolvimento	Uso dos conhecimentos descobertos

Outros modelos podem ser encontrados na literatura, em Marbán et al. (2009) e MARISCAL et al. (2010) há uma revisão em que podem ser vistos outros modelos, neste trabalho foram citados apenas 5 por se tratarem dos modelos que mais se destacaram na literatura. No entanto, como destaca Kurgan e Musilek (2006) os outros modelos não são focados em atender a especificidades acadêmicas ou industriais mas sim fornecer um modelo que é independente de uma determinada ferramenta, vendedor, ou aplicativo.

O CRISP-DM apesar de ser inicialmente voltado para industria é de fato o mais utilizado para desenvolver projetos de mineração de dados, de acordo com pesquisas feitas por um site especialista em mineração de dados feitas em 2002, 2004 e 2007 KdNuggets.Com (2002), KdNuggets.Com (2004), KdNuggets.Com (2007) e por isso o CRISP-DM é o modelo adotado neste trabalho e descrito com mais detalhes a na seção seguinte.

As técnicas que são aplicadas em cada etapa do processo de MD dependem diretamente do contexto em que o processo é aplicado, o que influencia a origem e o formato de dados que são utilizados. Assim, no contexto meteorológico, os dados são resultantes de observações feitas ao longo do tempo, o que de acordo com Keogh (2002) permitem que os mesmos sejam denominados séries

temporais. Dessa forma, a seção seguinte apresenta cada etapa do CRISP-DM mostrando as técnicas utilizadas em séries temporais.

2.4 CRISP-DM

O CRISP-DM é um modelo de processo independente de fornecedor podendo ser utilizado em qualquer aplicação ou qualquer projeto de MD. O modelo foi inicialmente proposto em 1996 por um consórcio de 4 companhias: SPSS, NCR , Daimler Chrysler, e OHRA (KURGAN; MUSILEK, 2006). No entanto, o CRISP-DM foi oficialmente apresentado à comunidade científica em 2000 nos trabalhos de Shearer (2000) e Wirth (2000). É composto por 6 etapas que definem os passos para conduzir um projeto de MD, essas etapas são mostradas na Figura 3 e descritas a seguir.

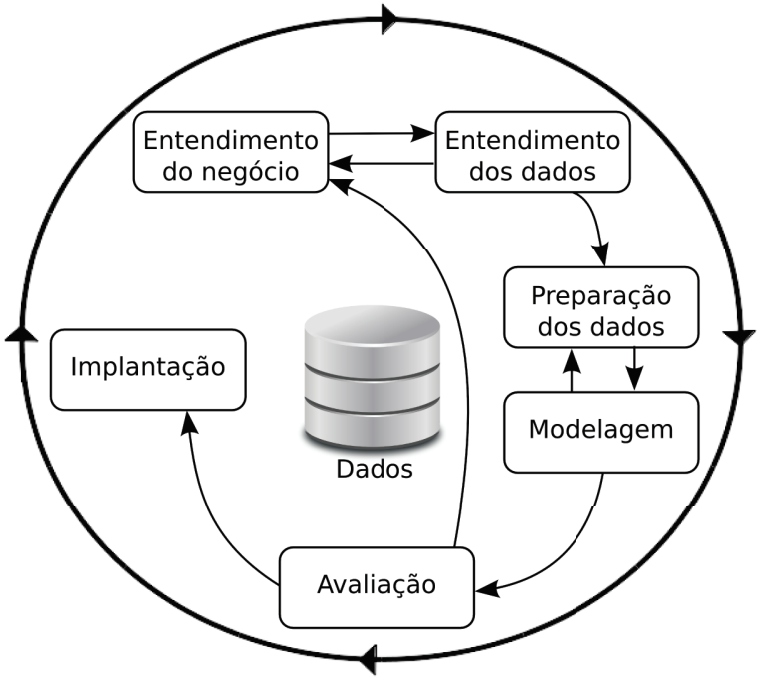


Figura 3: Modelo de Processos para Mineração de Dados, CRISP-DM. Adaptado de Chapman et al. (2000)

2.4.1 Entendimento do Negócio

Essa fase concentra-se em entender os objetivos do projeto e converter esse conhecimento em uma definição de um problema de mineração de dados bem como um plano preliminar de objetivos a serem alcançados. Por exemplo, em um projeto de mineração de dados que utilize dados de meteorologia é importante

elencar certos aspectos sobre cada variável a ser medida. Detalhes como faixa de valores permitidos, por exemplo, pode influenciar em etapas posteriores, como detecção de anomalias. Ainda nessa fase deve ser planejada a política de coleta dos dados, por exemplo quais variáveis devem ser medidas e qual a periodicidade das medições. Nessa etapa também são coletadas informações a respeito do local a ser estudado, detalhes como o clima predominante, sazonalidade das variáveis climáticas, relevo e altitude são listadas para que possam servir de base para a interpretação dos resultados da mineração.

2.4.2 Entendimento dos Dados

Essa fase inicia-se com a coleta dos dados e depois de coletados procede-se com atividades para se familiarizar com os dados, identificando por exemplo problemas de qualidade nos dados. Nesse momento devem ser executadas tarefas que trazem um prévio entendimento a respeito do comportamento dos dados e o significado de cada variável.

Todo o conhecimento que deve ser levantado nessa etapa depende do tipo de dado utilizado. Cada área tem suas especificidades, principalmente relacionado ao tipo de dado utilizado. Nos dados de meteorologia, por exemplo, os dados tem faixas de valores conhecidos para uma determinada região.

2.4.3 Preparação dos Dados

Nesta etapa os dados passam por operações que alteram de alguma forma os dados para construir o conjunto de dados final, partindo dos dados brutos na fase anterior, para realizar a mineração. Nessa etapa, o conhecimento das fases anteriores podem auxiliar muito o processo.

Pode-se ter a necessidade de por exemplo, de aplicar filtros para retirada de ruídos e considerar apenas o comportamento da série, como filtros digitais ou wavelet Jansen (2001). É importante destacar que processamentos realizados, como os filtros, dependem das características de cada variável. Essas informações são levantadas na etapa de entendimento do negócio e entendimento dos dados.

Em dados provenientes de sensores comumente ocorre ausência de dados e/ou a presença de *outliers*. Nesses casos deve ser feita a detecção do momento em que houve falha ou há um *outlier* e posteriormente deve ser feita a correção. Em Ventura et al. (2013) é apresentado um método para preenchimento de falhas em séries temporais de micrometeorologia utilizando Algoritmos Genéticos e Redes Neurais. O trabalho de Ooba et al. (2006) utiliza ideia semelhante utilizando a

combinação de algoritmos genéticos e redes neurais para preenchimento de falhas em séries de fluxo de carbono.

No que se refere a preenchimento de falhas, não são utilizadas apenas técnicas de inteligência artificial como os citados anteriormente, 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. Outras técnicas estatísticas como a *Multiple Imputation* pode ser utilizada (HUI et al., 2004; TATSCH et al., 2007). Em alguns casos como em dados de finanças de acordo com Moerchen (2006) não é feita a correção mas o último valor disponível é utilizado.

Além de procedimentos para garantir a qualidade dos dados nessa etapa podem ser executadas tarefas que alteram a forma como os dados são representados. É uma característica comum nas séries temporais a alta dimensionalidade e alta granularidade, ou seja, muitos dados coletados em pouco tempo, geralmente são bases de dados grandes o que pode dificultar a análise dos mesmos, desde o ponto de vista do utilizador dos dados até o custo computacional para realizar análises. Dessa forma, existem técnicas de representação de séries temporais que objetivam simplificar a representação, diminuir custo computacional, destacar ou esconder um determinado comportamento da série.

A forma de representação escolhida pode ser muito útil para determinadas aplicações. Por exemplo, dados coletados a cada minuto durante o dia todo e durante 1 ano podem ser agrupados em dados mensais quando o evento a ser analisado não precise de uma granularidade tão alta. Esses mesmos dados por exemplo podem ser divididos entre os dados que representam o dia e dados que representam a noite, quando essa divisão é necessária, é o caso por exemplo de análises envolvendo a Radiação Solar em torres meteorológicas. Nesse caso os dados são agrupados em apenas dados coletados durante o dia, pois durante a noite a radiação solar incidente é mínima.

Fu (2011) diz que a principal razão de se ter diferentes técnicas de representação de séries temporais é a redução do número de pontos da séries ou a redução de dimensionalidade. Além da diminuição de pontos, ou alteração na granularidade da série, a escolha da forma de representação deve levar em consideração o objetivo da análise a ser feita. Esling e Agon (2012a) destaca que deve-se buscar destacar características fundamentais da série a ser analisada, deve-se ainda derivar a noção da forma da série. Esses detalhes envolvidos na representação da série dependem do entendimento do negócio e conhecimento dos dados, o que destaca a importância dessas etapas no processo de MD. Devido a

importância dos métodos de representação de dados no contexto de dados temporais, especificamente nos dados meteorológicos, a seguir são apresentados os principais métodos de representação de séries temporais com mais detalhes.

2.4.4 Modelagem

Nessa fase técnicas de modelagem são selecionadas e aplicadas. Nesse momento os modelos são também calibrados para atender melhor ao problema, é comum que se dê um passo atrás para a fase anterior para corrigir problemas identificados apenas nessa fase. É a fase em que é extraído o conhecimento, como detecção de padrões ou agrupamento dos dados.

Para Fu (2011), descoberta de padrões é a tarefa mais comumente executada quando se trata de mineração em séries temporais. Outras técnicas são também muito aplicadas, como busca por conteúdo ou busca por similaridade Faloutsos et al. (1994), Keogh (2002), Wu et al. (2005), detecção de anomalias na série Weiss (2004), Park et al. (2013), Sipes et al. (2014), *clustering* ou agrupamento Keogh e Lin (2005a), Xing et al. (2011), Zhang et al. (2011) e classificação Jeong et al. (2011), Douzal-Chouakria e Amblard (2012), Harvey et al. (2014).

A técnica ser utilizada depende do contexto de utilização dos dados, na Seção 2.5.2 são apresentadas principais técnicas utilizadas em séries temporais de meteorologia.

2.4.5 Avaliação

Nessa fase são avaliados os modelos utilizados ou criados na fase anterior. São avaliados por exemplo se os padrões encontrados não são padrões triviais, como o comportamento horário da temperatura, que tende a ter seu menor valor no início da manhã e o valor mais alto ao meio dia. No caso de tarefas de agrupamento pode ser avaliado por exemplo se o número de clusters escolhido para a análise é adequado e caso não seja, executa-se a fase anterior novamente com outros parâmetros. Ao final dessa fase é tomada uma decisão de como utilizar o resultado da mineração.

É importante destacar que nessa etapa é imprescindível a atuação do especialista de domínio, que é quem pode avaliar o resultados dos modelos gerados. Além disso, a avaliação pode envolver a opinião de domínios diferentes, o que se caracteriza como um processo multidisciplinar. Na utilização de dados meteorológicos para auxiliar atividades agronômicas por exemplo, é comum a presença de um físico para interpretar os resultados do ponto de vista físico da interação entre

as variáveis e um agrônomo para extrair informações a respeito da influência das variáveis meteorológicas nos processos ecofisiológicos da planta.

2.4.6 Implantação

Nessa fase o conhecimento adquirido a respeito dos dados deve ser organizado de forma que o usuário final, ou cliente possa utilizá-lo, é a consolidação do processo todo.

Essa consolidação pode ser por exemplo a publicação dos resultados por meio de relatório técnias ou artigos científicos, por exemplo. Podem ainda ser tomadas decisões ou feitas recomendações a serem consideradas em contextos que envolvem os dados utilizados.

Nesse momento são também utilizadas um conjunto importante de técnicas que fazem parte da mineração de dados, que são as técnicas de visualização. A visualização adequada dos resultados auxilia em todo o processo de tomada de decisões decorrente dos resultados alcançados.

2.5 Processamento de Dados Meteorológicos

Tanto o modelo apresentado por Alcântara et al. (2010) que apresenta o ciclo de atividades para realização do monitoramento ambiental como o modelo de processos de mineração de dados CRISP-DM definem apenas de forma geral as etapas a serem relizadas no processo. Os detalhes a respeito das atividades e técnicas a serem utilizadas depende do contexto de aplicação e do tipo de dado utilizado.

Os trabalhos envolvendo meio ambiente utilizam comumente séries temporais, pois busca-se entender o comportamento de uma determinada variável ao longo do tempo. Nesse tipo de dados duas atividades são mais comuns: A preparação ou pré-processamento e a análise propriamente dita. Essas atividades são executadas nas fases de **Preparação dos dados** e **Modelagem** do CRISP-DM, as demais etapas envolvem mais planejamento e aplicação dos resultados objetivos do que processamento dos dados em si. As atividades desenvolvidas em cada uma dessas etapas são guiadas pelo conhecimento do especialista de domínio, que responsável por escolher quais técnicas utilizar e seus parâmetros.

A preparação dos dados diz respeito à qualidade dos mesmos e a forma de representação. A modelagem é a aplicação de modelos para a extração de conhecimento.

2.5.1 Preparação dos Dados

A correção dos dados pode ser necessária porque podem ocorrer falhas no processo de aquisição dos dados. Essas falhas podem ser humanas ou falhas dos equipamentos de medidas.

A transformação consiste em um processamento nos dados de forma que sua representação inicial é alterada. Essa nova representação pode por exemplo ser aplicada para diminuir a quantidade de pontos na séries temporal, ou seja, alterar a granularidade dos dados (FU, 2011). É importante que forma de representação destaque as características fundamentais da série a ser analisada (ESLING; AGON, 2012a). As transformações podem ainda ser para destacar um determinado comportamento ou excluir uma característica indesejavel.

A seguir são apresentados mais detalhes sobre a correção dos dados e métodos de representação.

2.5.1.1 Correção

Os dados meteorológicos são geralmente coletados por estações meteorológicas compostas por equipamentos eletrônicos que ficam instalados em campo, sujeitos a vários intemperes do meio ambiente como: corrosão, acumulo de insetos e falta de energia (VENTURA et al., 2013; DIAS, 2007).

Estes problemas podem fazer com que os equipamentos deixem de coletar ou de armazenar os dados em um determinado instante de tempo. Além da falha nos dados podem ocorrer de valores absurdos serem armazenados, a esse acontecimento dá-se o nome de *outliers*.

As falhas ou *outliers* prejudicam a análise dos dados e, por isso esses casos devem ser tratados antes da análise dos dados. Existe duas possibilidades nesses casos: corrigir os dados ou ignorar períodos de medições.

Os dados meteorológicos são de natureza complexa, há a interação entre diversas variáveis. Isso dificulta o processo de correção desses dados, por isso mesmo, muitos trabalho apresentam como alternativa excluir um período de dados ou cálculo da média dos valores próximos à falha (VENTURA, 2015).

Para o preenchimento de falhas é comum a utilização de técnicas específicas para cada tipo de dado, buscando-se aproveitar da do comportamento de cada variável. Em Oliveira et al. (2010) é apresentada uma comparação de técnicas estatísticas de ponderação regional, regressão linear e regressão potencial para preenchimento de falhas em dados de precipitação.

Outra alternativa adotada é a utilização de dados de estações próximas à que ocorreu a falha, como apresentado em (CHIBANA et al., 2005; PINHEIRO et

al., 2013; FERRARI; OZAKI, 2014). Outros trabalhos em vez de apenas copiarem os valores medidos em estações próximas, utilizam técnicas de geoestatística para estimar os dados como em (WANDERLEY et al., 2012).

Além das técnicas puramente estatísticas, foram apresentados também trabalho utilizando técnicas de inteligência artificial para o preenchimento de falhas em dados meteorológicos, como em Ventura et al. (2013) e Ooba et al. (2006) que utilizaram a combinação de Redes Neurais e Algoritmos Genéticos.

2.5.1.2 Métodos de Representação de Dados

As análises envolvendo os dados meteorológicos baseam-se principalmente no comportamento dos dados. Desse modo, a forma de representação dos dados pode influenciar na análise a ser feita. Por isso, são aprentados a seguir alguns dos principais métodos de representação de dados utilizados nesse tipo de dado.

2.5.1.2.1 Método da Amostragem O método mais simples de representaçãoe largamente utilizado é a amostragem (ASTROM, 1969). O Resultado de uma representação por amostragem é representado na Figura 4. A série temporal é representada como segue: uma taxa de n/m é utilizada, onde n é o comprimento de uma série temporal x e m é a dimensão após a redução dimensionalidade. Este método de amostragem tem o inconveniente de distorcer a forma da série amostrada em tempo, se a taxa de amostragem for muito baixa.

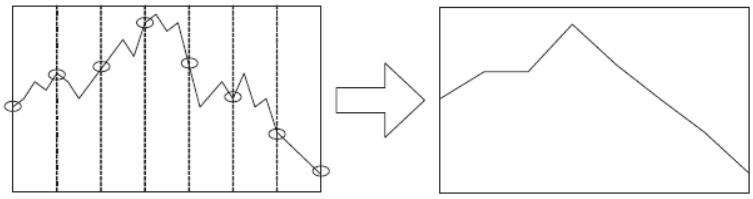


Figura 4: Exemplo de amostragem de uma série temporal.

2.5.1.2.2 Piecewise Aggregate Approximation (PAA) Uma melhoria do método de amostragem foi proposto em Keogh et al. (2001) e Keogh e Pazzani (2000), chamado de *Piecewise Aggregate Approximation* (PAA). O método utiliza a média de cada segmento para representar o conjunto correspondente de pontos de dados.

O algoritmo divide a a série de tamanho n em N segmentos de tamanhos fixos, adjacentes um ao outro e depois calcula a média dos valores dos pontos dentro de cada segmento. A série com dimensionalidade reduzida utilizando-se PAA é data pela equação 1:

$$\widehat{x} = \frac{1}{e_k - s_k + 1} \sum_{i=s_k}^{e_k} x_i \quad (1)$$

Onde, $\widehat{x}$ é a série após a redução de dimensionalidade e s_k e e_k denotam os pontos inicial e final do segmento kth dados da série x. A Figura 5 mostra um exemplo de série depois de aplicado o método PAA.

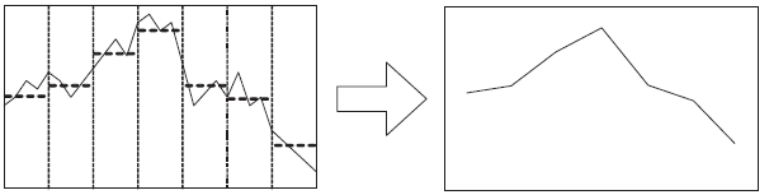


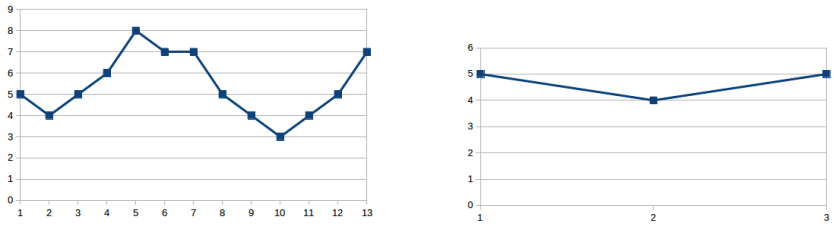
Figura 5: Exemplo de redução de dimensionalidade utilizando PAA.

Este método é amplamente utilizado, por ser simples e tratar diretamente a granularidade dos dados. Por exemplo, dada uma série temporal de temperatura com medidas a cada 15 minutos durante 1 ano, deseja-se obter os dados tem médias horárias, nesse caso a cada 4 medidas é calculada a média dos valores.

2.5.1.2.3 Segmented Sum of Variation (SSV) Este método utiliza uma ideia parecida com o PAA, primeiro a série temporal é dividida em N segmentos, os segmentos são conectados, ou seja, o fim de um segmento k é o início do segmento $k+1$. Isso é necessário porque o método calcula a variação de cada segmento, se os segmentos forem desconectados a variação seria perdida. A SSV de uma série temporal x de tamanho n utilizando-se segmentos de tamanho l é dada como segue(LEE et al., 2003):

$$SSV_x = < \sum_{i=1}^{l-1} |a_i + 1 - a_i|, \sum_{i=l}^{2(l-1)} |a_i + 1 - a_i|, .., \sum_{i=(s-1)l+(2-s)}^{s(l-1)} |a_i + 1 - a_i| > \quad (2)$$

A Figura 6 mostra o resultado do método SSV aplicado em uma série de tamanho 13 e com $l = 3$. A série original (Figura 6(a)) $x = 5, 4, 5, 6, 8, 7, 7, 5, 4, 3, 4, 5, 7$ foram calculadas as variações de 3 segmentos, $l_1 = 5, 4, 5, 6, 8$, $l_1 = 8, 7, 7, 5, 4$, $l_3 = 4, 3, 4, 5, 7$ que resultaram na série com dimensionalidade reduzida (Figura 6(b)) $X = 5, 4, 5$.



(a)Série Original (b)Resultado do Método SSV

Figura 6: Redução de dimensionalidade utilizando SSV

Esse método foi proposto por Lee et al. com o objetivo de criar um vetor de características para busca por similaridade em séries temporais deslocadas verticalmente utilizando a medida de distância mínima.

2.5.1.2.4 Bit Level Representation O método da *Bit Level Representation* foi proposto pelos autores Ratanamahatana et al. (2005), Bagnall e Janacek (2005) e Bagnall et al. (2006), e trabalha substituindo cada valor real da série por um único bit, baseando-se num valor μ, como mostrada a Figura 7 para μ igual a 0.

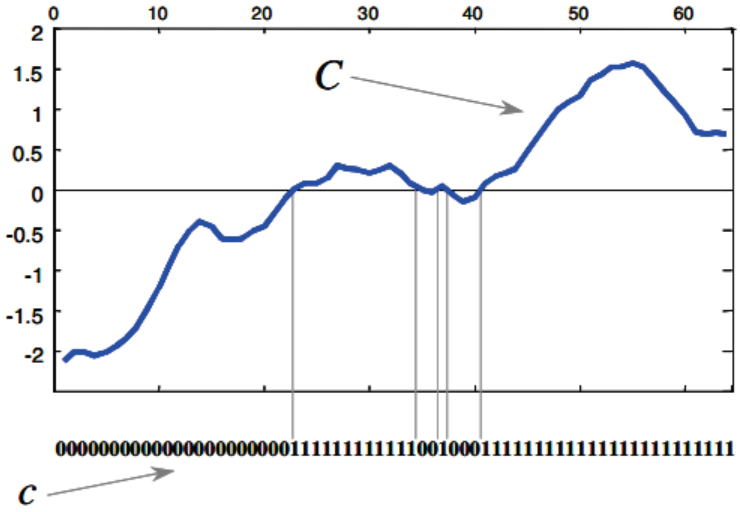


Figura 7: Uma série temporal, C, comprimento de 64, convertida para a representação por bit, c, por meio da observação cada elemento de C; se o seu valor é estritamente superior a zero, o correspondente bit é ajustado para 1, e a 0 caso contrário (RATANAMAHATANA et al., 2005)

A escolha dos bits messe método é feita como segue, na equação 3:

$$c(i) = \begin{cases} 1 \text{ se } C(i) > \mu \\ 0 \text{ outros casos} \end{cases} \tag{3}$$

Onde: μ é a média de C.

A redução da dimensionalidade se dá em um próximo passo, no qual os bits são contados, ou uma representação utilizando-se, por exemplo, bit de paridade é utilizada. No caso do exemplo da Figura 7 a série 00000000000000000000-1111111111110010001111111111111111111111111 pode ser representada por 22♯0-, 11♯1, 2♯0, 1♯1, 3♯0, 24♯1.

2.5.1.2.5 Perceptually Important Points (PIP) Apresentado pela primeira vez por (CHUNG F.L., 2001), esse método faz a redução de dimensionalidade selecionando apenas pontos mais importantes da série original, por exemplo aqueles que se diferenciam mais abruptamente dos seus vizinhos. Esses pontos são chamados de *perceptually importante points* ou pontos perceptualmente importantes. Esse método foi e é bastante utilizado para comparações de séries temporais do mercado financeiro, exatamente porque nesse cenário grandes oscilações nos movimentos das ações de empresas são os fatos mais importantes a serem analisados. Uma grande utilidade desse método é permitir que séries de tamanhos diferentes possam ser comparadas (FU et al., 2004).

A Figura 8 ilustra o exemplo de identificação de 7 PIPs em uma série temporal

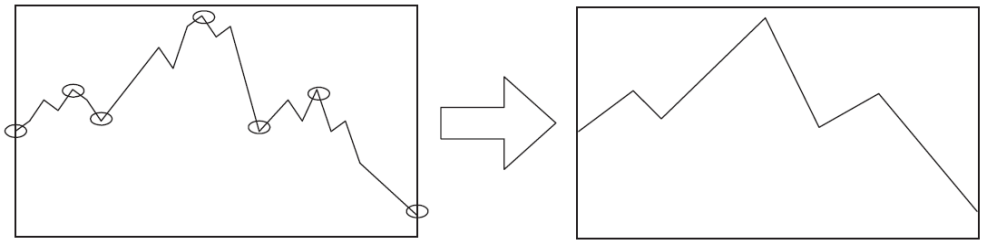


Figura 8: Redução de dimensionalidade por PIP. A série histórica da esquerda é representada por sete PIP à direita (FU et al., 2008)

O algoritmos para encontrar os PIPs é um algoritmo iterativo. Inicia-se considerando o primeiro e o último ponto da série como sendo dois PIPs, os PIPs iniciais. Posteriormente é criado um segmento de reta que liga os dois PIPs inciais, o terceiro PIP é encontrado procurando-se o ponto na série original que possui a maior distância vertical ao segmento de reta. Para o próximo PIP busca-se o ponto que tem a maior distância vertical aos dois segmentos de retas adjacentes, ou seja, o que liga o primeiro ao terceiro PIT e o segundo ao terceiro PIP. Esse processo iterativo continua até que o número de PIP seja igual ao tamanho da série original, nesse caso não há redução de dimensionalidade, ou até que um número pré-determinado de PIP seja encontrado. Depois de encontrados

todos os PIPs desejados o algoritmo armazena os PIPs pela ordem em que foram encontrados e não considera a sua dimensão temporal. Dessa forma, caso seja necessário, apenas poucos PIPs podem ser escolhidos para representar a série original.

A Figura 9 ilustra o funcioamento do algortimo em sua primeira parte e a Figura 10 ilustra a ordenação dos PIPs.

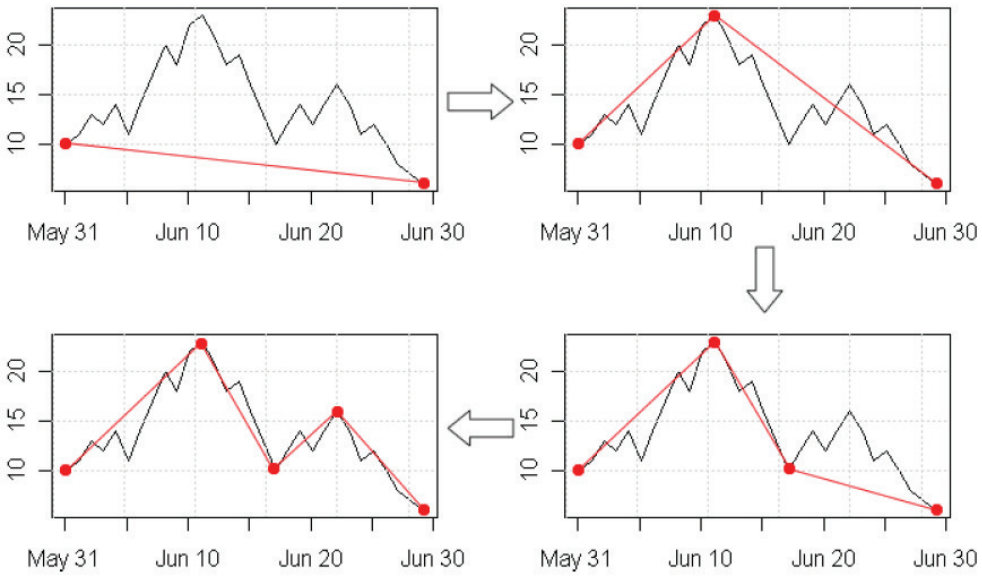


Figura 9: Identificação de 5 PIPs (SANCHES, 2006)

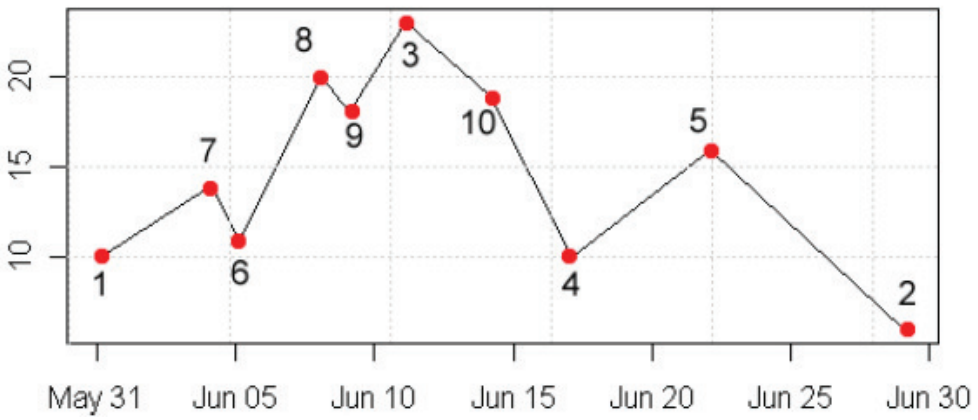


Figura 10: Ordenação de 10 PIPs identificados (SANCHES, 2006)

A ideia dessa técnica é similar a o que foi apresentado a mais de 40 anos por Douglas e Peucker (1973) e melhorado por Hershberger e Snoeyink (1992), em que os autores apresentaram dois algoritmos para reduzir a quantidade de pontos necessários para representar linhas ou suas caricaturas digitalizadas. Outras ideias parecidas podem ainda ser encontradas, Perng et al. (2000) apresentou um

modelo de referência para representar os pontos mais importantes na série e realizar busca por similaridade. Em Man e Wong (2001) é introduzido um algoritmo que utiliza uma estrutura de rede para representar os picos e depressões identificados na série os quais dá o nome de chamados de Pontos de Controle. Pratt e Fink (2002) e Fink et al. (2003) propuseram um algoritmo baseado em extremos mínimos e máximos na série para representá-las, eles utilizaram o algoritmo em dados de valores de ações, temperatura do ar e do oceano e velocidade do vento.

2.5.1.2.6 Representação Linear Uma outra abordagem utilizada na redução de dimensionalidade é a representação linear das séries, nesse contexto, existem duas categorias principais: Interpolação Linear e Regressão Linear.

A Interpolação Linear é implementada já a muito tempo utilizando a *Piecewise Linear Representation* (PLR) (KEOGH, 1997a; KEOGH, 1997b; SMYTH et al., 1997). Basicamente, o método faz uma linha de aproximação entre os pontos $X(x_i, ..., x_j)$ conectando por meio de uma reta os pontos x_i e x_j.

2.5.2 Modelagem

No processo de análise dos dados são aplicadas técnicas com o objetivo de extrair informações úteis da base de dados disponíveis. Em séries temporais de meteorologia a análise envolve principalmente extrair informações a respeito do comportamento da série, ou parte dela ao longo do tempo.

A seguir são apresentadas as principais técnicas de mineração de dados aplicadas a séries temporais de meteorologia.

2.5.2.1 Detecção de Padrões e Agrupamento

Detecção de padrões e Agrupamento em séries temporais significa identificar em uma dada série padrões que se repetem ao longo do tempo (*recurring patterns*) (FU et al., 2001). Esses padrões podem ser inicialmente desconhecidos ao especialista humano ou essa tarefa pode ser executada para provar alguma suposição inicial a respeito dos padrões. O mesmo tipo de técnica ainda pode ser utilizada para encontrar padrões surpresa, ou seja, que não são comuns (*surprising patterns*) na série temporal (KEOGH et al., 2002b). Em ambos os casos é importante ressaltar que não é uma tarefa trivial como destaca Fu (2011).

Para essas tarefas alguns autores utilizam outros nomes. Para *surprising patterns* alguns autores dão o nome de Detecção de Anomalias (*anomaly detection*) Chan e Mahoney (2005), Wei et al. (2005), Discordâncias (*finding discords*) Keogh et al. (2005b), Keogh et al. (2006) ou ainda Detecção de Novidades

(*novelty detection*) Ma e Perkins (2003). No caso de detecção de anomalias ou discordâncias é importante ressaltar que é uma tarefa diferente da detecção de *outlier*, pois não busca-se um determinado ponto que difere dos demais, mas sim um padrão que difere do comportamento da série (KEOGH et al., 2002b). A Figura 11 ilustra esse processo, é mostrada uma série temporal e percebe-se claramente que há um espaço de tempo em que os dados fogem ao padrão da série.

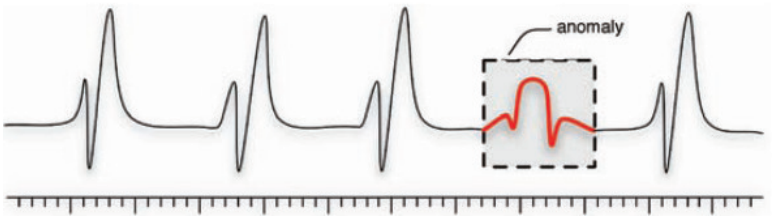


Figura 11: Exemplo de detecção de anomalia em uma série temporal.

Diversas técnicas são utilizadas no processo de Detecção de Padrões e Agrupamento, podendo ser citadas, Mapas Auto-Organizáveis de Kohonen (*Self-Organized Maps* - SOM) (KOHONEN, 1997). Em Li e Kuo (2008) SOM foram utilizados para descoberta de conhecimento em dados de finanças buscando-se tendências e previsões. Em Mörchen et al. (2005) a técnica foi utilizada para identificar sinais de ativações que se repedem em dados de cinesiologia, um exame feito para medir ativações do músculo esquelético. Em (GUO et al., 2007) redes SOM são utilizadas para descoberta de padrões em dados do mercado de ações. Em hidrologia, Chang et al. (2014) utiliza SOM para modelos de previsão de inundações.

Suport Vector Machine (SVM) também são utilizados para realizar agrupamento, em Boecking et al. (2014) é apresentado um método baseado em SVM para agrupamento em séries temporais. SVM pode ainda ser utilizado para previsão de valores em séries temporais Sapankevych e Sankar (2009), em Kim (2003) é utilizado para previsão de valores em séries de preços de índices na bolsa de valores. Em Samsudin et al. (2011) é apresentado um modelo baseado em SVM para previsão de vazão de um rio.

Hidden Markov Model (HMM) tem sido utilizado também para realização de agrupamentos em séries temporais. Em Yin e Yang (2005) HMM é utilizado em conjunto com análise espectral para agrupamento de dados de sensores. Em Duan et al. (2005) é proposto um algoritmo de HMM recursivo para agrupamento de séries temporais. Zhao e Deng (2010) é apresentado um algoritmo de agrupamento hieráquico utilizando HMM para agrupamento de expressões de gene em dados de DNA. Ghassempour et al. (2014) apresentam um algoritmo utilizando HMM

para agrupamento de séries multivariadas com dados contínuos e discretos de indicadores de saúde.

Dentre os métodos existentes para agrupamento destaca-se o *k-means* apresentado à comunidade científica pela primeira vez em 1955 (JAIN, 2010). O algoritmo para agrupar dados utilizando o *k-means* divide os dados em determinado número de grupos (cluster), seleciona um centróide para cada grupo e seleciona iterativamente a qual grupo cada item do conjunto de dados pertente, baseado em uma função de distância. O *k-means* necessita de um parâmetro que é importante e que pode alterar significativamente o resultado do agrupamento, é o número de clusters a ser considerado, a Figura 12 ilustra esse problema.

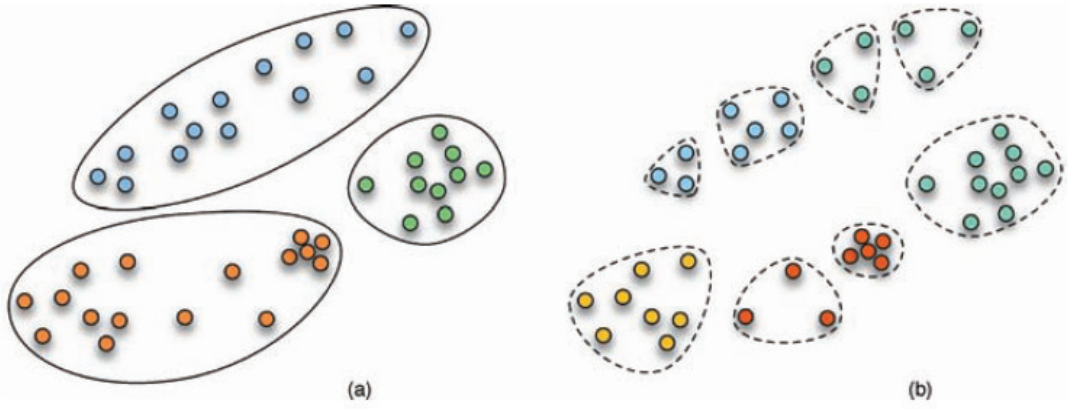


Figura 12: Exemplo de agrupamento de dados. A figura ilustra duas possibilidades de agrupamento, na primeira (a) foram definidos 3 grupos e na segunda (b) 8 grupo.

A definição da função de distância é importante para todos os métodos de agrupamento e deteção de padrões, pois é a medida que define o quanto um dado ou conjunto de dados difere ou é parecido com os demais, funções de distância são discutidas na Seção **??**.

2.5.2.2 Classificação

Classificação é talvez a tarefa mais clássica executada quando se trata de mineração de dados e pode ser definida de acordo com Alencar (2007) como segue: Dado um conjunto de dados $X = \{x_1, ..., x_n\}$ cujas classes são conhecidas y, $y = \{1, ...J\}$, o objetivo da classificação é o aprendizado de uma função $f : X \rightarrow Y$ que mapeie um objeto $x \in X$ para a sua classe $y \in Y$.

Para o aprendizado da função f são utilizados dados de treinamento para generalizar o modelo, nesse contexto, Tan et al. (2005) alerta para que, o objetivo

é criar um modelo que possa categorizar dados do conjunto de treinamento e de um conjunto de dados nunca visto, o que significa que o modelo deve ser o mais genérico possível, sem perda acurácia.

Muitas técnicas são aplicadas na tarefa de classificação de dados em séries temporais, no entanto como destaca Xi et al. (2006) a combinação de *Nearest-Neighbor* com *Dynamic Time Warping*(DTW) é difícel de ser superada quanto a precisão da classificação. Entretanto, Xi et al. (2006) afirma que essa combinação tem um problema, que é cara computacionalmente para aplicações em tempo real.

2.5.2.3 Descoberta de Motifs

Patel et al. (2002) afirmam que encontrar padrões previamente conhecidos em uma série temporal (busca por similaridade) é interessante, no entanto do ponto de vista da descoberta de conhecimento é mais interessante a descoberta de padrões recorrentes desconhecidos. Para isso a abordagem mais comum é utilizar algoritmos de agrupamento e utilizar como entrada subsequências extraídas por meio de Janela Deslizante em uma dada série. O processo de Janela Deslizante consiste em: definido um tamanho de uma janela, subsequências são extraídas deslocando-se essa janela na série original. No entanto, essa abordagem sofreu duras críticas, Keogh et al. (2003), Keogh e Lin (2005b), Idé (2006) afirmam que a utilização dessa abordagem gera resultados sem sentido (*meanless*) no agrupamento, entretanto, é feita uma observação por Keogh et al. (2003) que o problema não está no algoritmo de agrupamento utilizado, mas em como as entradas são geradas, no caso a janela deslizante.

Para contornar então os problemas da abordagem comum de descoberta de padrões recorrentes desconhecidos em uma série temporal, Patel et al. (2002) apresentaram o algoritmo para descoberta de *Motifs*, que são os padrões recorrentes na série temporal.

Formalmente, segundo Esling e Agon (2012b), a descoberta de *Motifs* é: *Dada uma série temporal* $T = \{t_1, ..., t_n\}$ *encontre todas as subsequências* $T' \in S_T^n$ *que ocorrem repetidamente na série original.*

A Figura 13 ilustra o processo de descoberta de *Motifs*, é possível ver que não é um processo trivial, pois a mesma série temporal pode conter diferentes *Motifs* e eles podem inclusive se sobrepor.

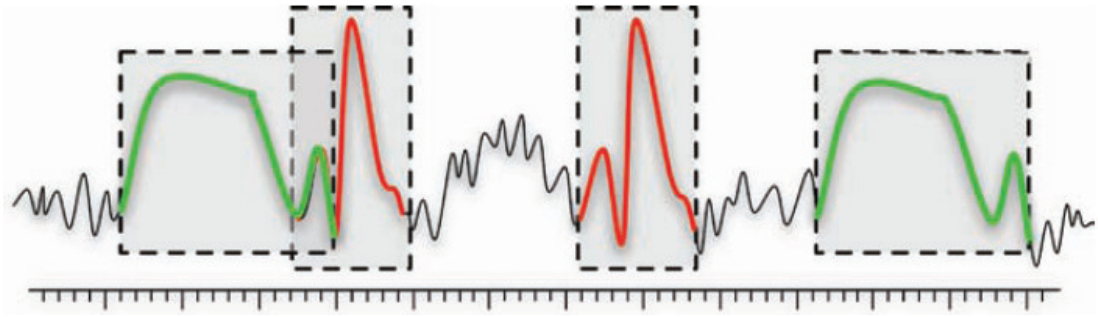


Figura 13: Exemplo de descoberta de *Motifs*. Fonte (ESLING; AGON, 2012a)

Para Yankov et al. (2007) *Motifs* são ocorrências aproximadas de subsequências em uma dada série temporal em posições significativamente distintas. Para encontrar subsequências aproximadas, primeiro é necessário definir uma função de distância (D) e um limiar (*range*) de aceitação (r). Dada uma série temporal Z de tamanho m e sejam p e q posições iniciais de duas subsequências de tamanho n. Para que p e q sejam denominadas significativamente distintas é necessário existir uma posição p' tal que $p < p' < q$ e $D(Z_p, ...,_{p+n-1}, Z_{p'}, ...{}_{p'+n-1}) > r$ (MALETZKE, 2009). Essa definição faz com que subsequências muito próximas não sejam consideras *Motifs*, evitando o problema do *trivial matching*. Esse e outros conceitos são discutidos a seguir.

2.5.2.3.1 Casamento (*Match*) Seja R um número real positivo o limiar de aceitação, e dada uma série temporal T contendo uma subsequência C com inicio na posição p e outra subsequência M na posição q, e considerando a distância D entre dois objetos, tem-se que se $D(C, M) \leq R$, então assume-se que M é similar a subsequência C (LEE et al., 2003).

A Figura 14 mostra um exemplo de casamento, na figura C é uma subsequência e M uma outra subsequência que possui casamento com C. Percebe-se que as subsequências não são exatamente iguais, mas são próxima de acordcom com a função de distância D e o limiar R.

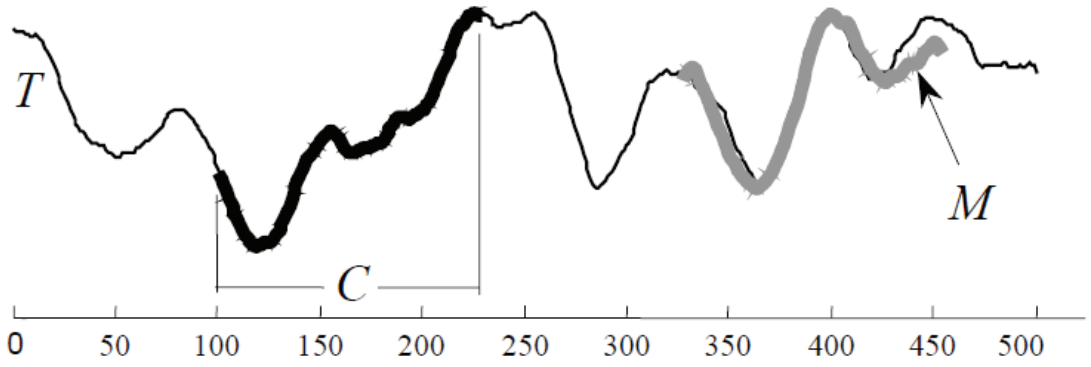


Figura 14: Exemplo de Casamento. Fonte (LEE et al., 2003)

Por meio da definição de Casamento, e da definição da Reflexividade (Seção 2.5.2.4.1), percebe-se que o melhor casamento de uma subsequência é quando comparada a ela mesma. As subsequências mais próximas, com 1 ou 2 unidades de deslocamento para um lado ou outro da série (eixo do tempo) tendem a ser também muito similares, nesse caso tem-se o que se chama de Casamento Trivial.

2.5.2.3.2 Casamento Trivial (*Trivial Macth*) Data uma série temporal T contendo uma subsequência C com início na posição p e a subsequência M que possui casamento com C com início na posição q, M é chamada de casamento trivial de C se, $p = q$ ou se não existe uma subsequência M' com início em q' tal que $D(C, M') > R$, e q $< q' < p$ ou $p < q' < q$ (LEE et al., 2003).

A Figura 15 mostra um exemplo de casamento trivial.

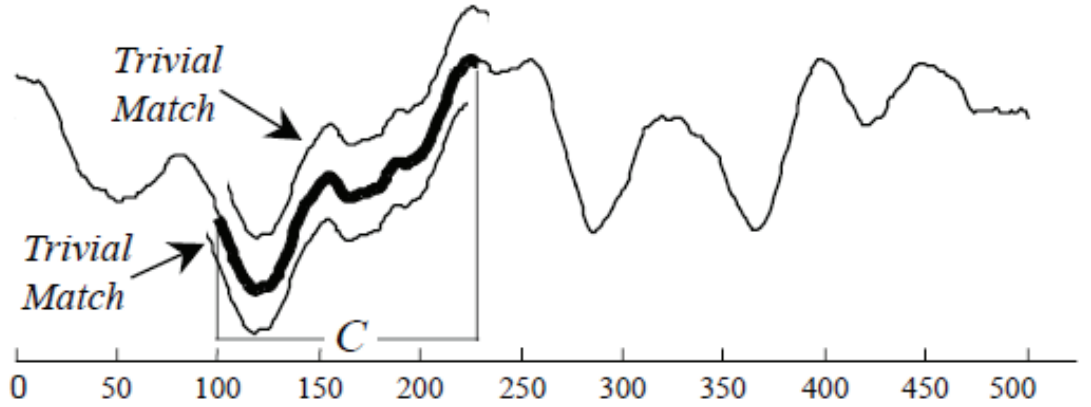


Figura 15: Exemplo de Casamento Trivial. Fonte (LEE et al., 2003)

Com essas definições é possível então definir o problema da descoberta de *k-motifs* mais significantes em uma série temporal.

2.5.2.3.3 *K-Motifs* Dada uma série temporal T, uma subsequência de tamanho n e um limiar R, o *Motif* mais significante em T, chamado de *1-Motif*, é a subsequência C_1 que tem a maior quantidade de casamento não trivial. Os K^th *Motifs* mais significantes em T, chamados *K-Motifs* subsequentes é a subsequência C_k que tem a maior quantidade de casamentos não triviais e satisfaz $D(C_k, C_i) > 2R$, para todo $1 \leq i < K$ (LEE et al., 2003).

Por meio dessa definição percebe-se a importância de que duas subsequências estejam a pelo menos $2R$ de distância uma da outra, o que força com que sejam mutualmente exclusivas, fazendo com que duas subsequências não compartilhem muitos elementos, o que cairia num caso de casamento trivial, a Figura 16 ilustra um caso em que primeiro é considerado distâncias maiores que $1R$ apenas e para o caso de distâncias maiores que $2R$. No primeiro caso, percebe-se que os dois *Motifs* compartilham muitos elementos e no segundo caso estão em posições totalmente diferentes.

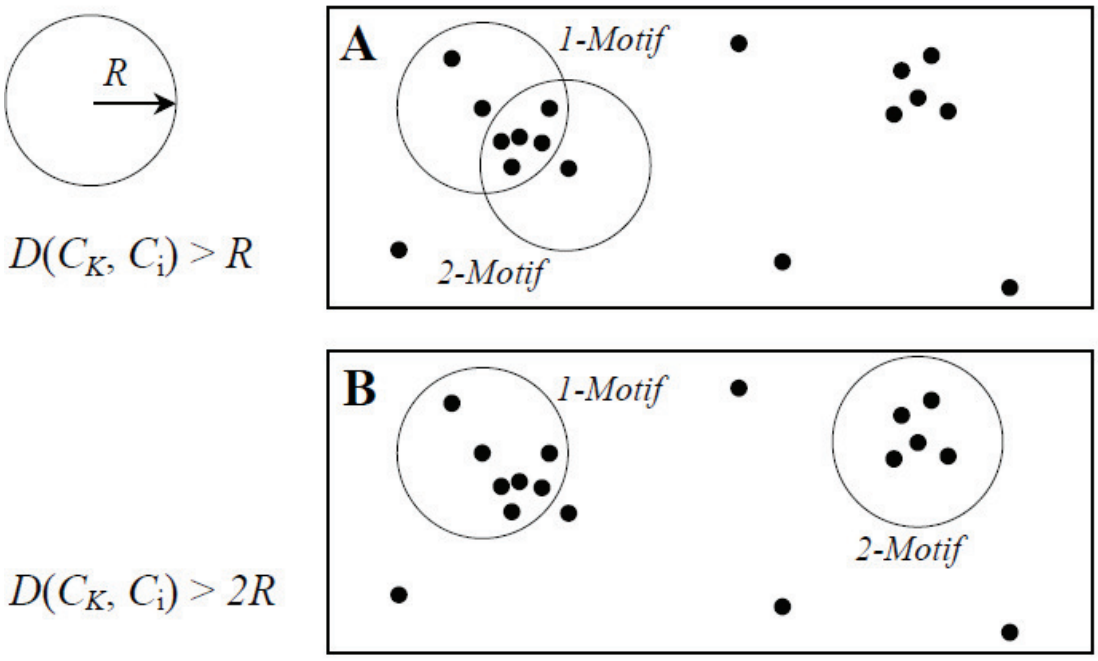


Figura 16: Exemplificação visual do motivo pelo qual a definição de *K-Motifs* requer que a distância entre duas subsequências seja maior que $2R$. Fonte (LEE et al., 2003)

Além do tratamento do problema de casamento trivial, Chiu et al. (2003) apontam outro cuidado que deve ser tomado, o grau de significância dos *motifs* encontrados. Dependendo do tamanho do *motif* que é buscado, o tamanho é parâmetro do algoritmo, pode-se encontrar padrões que não agregam conhecimento ao processo de mineração de dados. A Figura 17 ilustra esse caso, na figura

pequenas retas foram consideradas *motifs*.

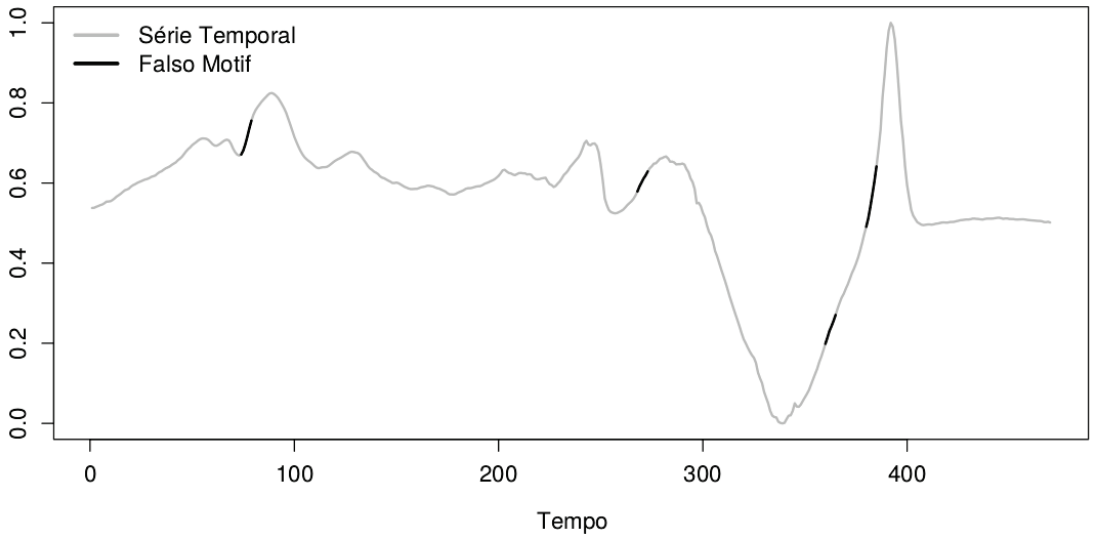


Figura 17: Exemplo de falsos motifs. Fonte (MALETZKE, 2009)

O algoritmos mais simples para detecção de *Motifs* em uma série temporal é o algoritmo de força bruta mostrado no Algoritmo a seguir.

```
Algorithm Find-1-Motif-Brute-Force(T,n,R)
1.  best_motif_count_so_far = 0;
2.  best_motif_location_so_far = null;
3.  for i = 1 to length(T)- n + 1
4.   count    = 0;
5.   pointers = null;
6.   for j = 1 to length(T)- n + 1
7.     if non_trival_match(C[i:i+n-1],C[j:j+n-1],R)
8.       count = count + 1;
9.       pointers = append(pointers,j);
10.    end;
11.  end;
12.  if count > best_motif_count_so_far
13.    best_motif_count_so_far = count;
14.    best_motif_location_so_far = i;
15.    motif_matches = pointers;
16.  end;
17. end;
```

Figura 18: Algoritmos Motifs Força Bruta. Fonte (MALETZKE, 2009)

O algoritmo deve receber parâmetros, a série temporal T de tamanho m, o tamanho dos *Motifs* que deseja-se buscar n e o limiar R. O algoritmo utiliza ideia de janela deslizante, entretanto é feito o controle do casamento trivial citado anteriormente. O algoritmos percorre toda a série temporal com deslocamento de uma unidade de tempo e extrai subsequências de tamanho n, para cada subsequência é percorrido o restante da série em busca de outras subsequências que sejam similares porém não sejam um caso de casamento trivial. Para cada subsequência que satisfaça a condição é guardada posição inicial e a final da subsequência.

O algoritmo por força bruta tem ordem de execução $O(m^2)$, o que pode tornar o algoritmo inviável para séries de tamanhos muito grande.

2.5.2.4 Busca por Similaridade

Em bases de dados tradicionais uma busca é feita por correspondência exata, por exemplo: Cliente cujo nome é igual a João. Em séries temporais no entanto, a busca não se baseia em uma correspondência exata, mas em uma aproximação, isso devido a natureza dos dados que são geralmente numéricos e contínuos (FU, 2011). Um exemplo de uma busca em uma série temporal pode ser: "Busque os períodos de venda que se comportem de forma parecida ao último mês".

A busca por similaridade em séries temporais pode ser conduzida de duas formas, comparando a série a ser buscada com toda a outra série alvo (*Whole Sequence Matching*) ou em outro caso, quando é dada uma série Q e uma série mais longa P e busca-se subséries que sejam similares à Q em P (*Subserie Matching*) ou (*Subsequence Matching*) (AGRAWAL et al., 1993).

Para melhor acompanhamento dessa Seção a Tabela 2 é apresentada para padronização dos símbolos utilizados.

Tabela 2: Notações matemáticas utilizadas na seção.

Símbolo	Descrição
$\mathbb{U}$	Universo de Objetos Válidos (Descritores)
$\mathbb{S} \subset \mathbb{U}$	Banco de Dados de Objetos Válidos
$x, y, o_i \in \mathbb{U}$	Objetos de $\mathbb{U}$
$q \in \mathbb{U}$	Objeto de pesquisa de $\mathbb{U}$
$s : \mathbb{U} \times \mathbb{U} \to \mathbb{R}$	Função de similaridade entre pares de objetos em $\mathbb{U}$
$\delta : \mathbb{U} \times \mathbb{U} \to \mathbb{R}$	Função de dissimilaridade entre pares de objetos em $s : \mathbb{U}$

Quando se trata de mineração de dados em séries temporais, comumente as análises envolvem medidas de similaridade entre séries de dados, o que implica no uso de uma função de distância, ou seja, uma função matemática que diga o quão parecida ou não é uma série da outra (ALENCAR, 2007). Portanto, nessa seção são descritas algumas medidas de similaridade entre séries temporais.

Um função de similaridade (s) é definida como uma função de dois pares (*pairwise similarity function*) de objetos x, y do universo $\mathbb{U}$ e a similaridade assume um valor no universo dos números reais $\mathbb{R}$, essa função é definida como segue:

$$s : \mathbb{U} \times \mathbb{U} \mapsto \mathbb{R} \tag{4}$$

A função produz um número real que representa o quão similar são os dois

objetos do universo $\mathbb{U}$ que é composto por um conjunto de todos os descritores do objeto, a estrutura do objeto $x \in \mathbb{U}$ pode ser por exemplo, um vetor, uma série tepmoral, uma string, etc (SKOPAL; BUSTOS, 2011).

Uma outra função também é utilizada, esta define não a similaridade, mas a dissimilaridade ou distância entre os objetos (δ). Nesse caso, valores altos de dissimilaridade representam baixa similaridade entre os pares e vice-versa (SKOPAL; BUSTOS, 2011). Dessa forma s e δ devem cumprir a seguinte regra:

$$s(x, y) \geq s(x, z) \Leftrightarrow \delta(x, y) \leq \delta(x, z), \forall 1 x, y, z \in \mathbb{U} \tag{5}$$

A dupla (U, s) é chamada espaço de similaridade e (U, δ) espaço de dissimilaridade. A escolha do uso da função similaridade ou dissimilaridade depende do algoritmo implementado.

A busca por similaridade pode-se ter dois tipos de busca: Por faixa (Query Range) que busca objetos que estejam dentro de uma faixa de similaridade e os k-vizinhos mais próximos (*k-nearest-neighbors*) que busca os k objetos mais próximos, ambos os casos de acordo com δ. Os dois casos são definidos formalmente a seguir.

- **Query Range:** ((SKOPAL; BUSTOS, 2011)) Dada a consulta do objeto q com um determinado limite de distância r, $q \in \mathbb{U}$, $r \in \mathbb{R}^+$, retorna-se todos os objetos contidos em $\mathbb{S}$ que estão a uma distância de no máximo r de q, ou seja:
$$(q, r) = \{x \in \mathbb{S} | \delta(x, q) \geq r\} \tag{6}$$

- **K-Nearest-NeighBors:** ((SKOPAL; BUSTOS, 2011)) Dada a consulta do objeto q com um determinado limite de distância r, $q \in \mathbb{U}$, $r \in \mathbb{R}^+$, retorna-se um conjunto $\mathbb{C}$ contendo os k objetos mais similares a q, ou seja $\mathbb{C} \subseteq \mathbb{S}$ tal que $|\mathbb{C}| = k$ e $\forall x \in \mathbb{C}, y \in \mathbb{S} - \mathbb{C}, \delta(x, q) \geq \delta(y, q)$.

A Figura 19 exemplifica os dois tipos de busca, no primeiro caso ($q1$) é executada a busca por *Range Query*. No segundo caso a consulta por Knn. Observa-se que no primeiro caso é definido um alcance (r) e são retornados todos os objetos dentro desse alcance. Na consulta por Knn são selecionados apenas o k mais próximos.

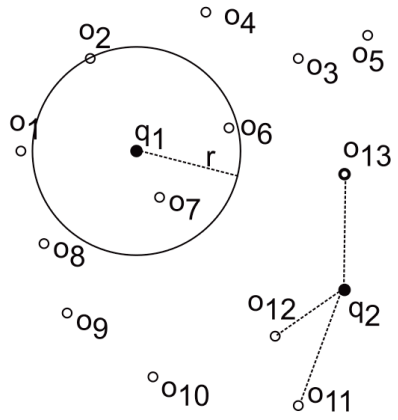


Figura 19: Exemplo de funcionamento do *Range Query* e KNN.

2.5.2.4.1 Funções de Distância Como dito anteriormente, tanto a tarefa de agrupamento, detecção de padrões e classificação em séries temporais envolvem um item importante que é a função de distância a ser utilizada. A função de distância diz basicamente o quão diferente (ou similar) uma determinada série temporal é de outra.

As funções de distância mais utilizadas são as Funções Métricas que seguem os 4 Postulados Métricos descritos a seguir (SKOPAL; BUSTOS, 2011):

$\delta x, y = 0$	$\Leftrightarrow x = y$	Relexividade
$\delta x, y > 0$	$\Leftrightarrow x \neq y$	Não negatividade
$\delta x, y = \delta y, z$		Simetria
$\delta x, y + \delta y, z \geq \delta x, z$		Desigualdade Triangular

A Reflexividade diz que a distância entre dois objetos a serem comparados é Zero somente se os dois objetos forem idênticos, ou forem o mesmo objeto. A Não Negatividade diz que a distância entre dois objetos será sempre maior que zero se os objetos forem diferentes. De acordo com a o postulado da Simetria, a distância entre dois objetos y e z é igual a distância entre z e y. A Desigualdade Triangular diz que a soma da distância entre x, e y com a distância entre y e z será sempre maior ou igual que a distância entre os objetos x e z. É importante ressaltar que, para alguns tipos de dados complexos nem sempre as funções métricas são suficiente para a comparação, como destaca Skopal e Bustos (2011) ao fazer uma revisão sobre funções não métricas para domínios complexos.

A função de distância mais utilizada é a Euclidiana Krislock e Wolkowicz (2012), que satisfaz aos postulados métricos descritos. A distância euclidiana dada duas séries temporais $Q = q_1, ...q_n$ e $C = c_1, ..., c_n$ é definida como segue:

$$\delta(Q, C) = \sqrt{\sum_{i=1}^{n}(q_i - c_i)^2} \tag{7}$$

Apesar de muito utilizada a distância euclidiana, no caso de séries temporais a distância euclidiana necessita que as duas séries a serem comparadas tenham o mesmo tamanho, sendo necessário um pré-processamento caso duas séries de tamanhos diferentes precisem ser comparadas.

Além do tamanho das séries, o valor da distância euclidiana pode mascarar o valor da distância entre duas séries que estejam deslocadas horizontalmente uma em relação a outra. Para esse caso a alternativa utilizada é a *Dynamic Time Warping*, que faz um alinhamento entre as duas séries a serem comparadas. A Figura 20 ilustra o comportamento das duas funções de distância citadas.

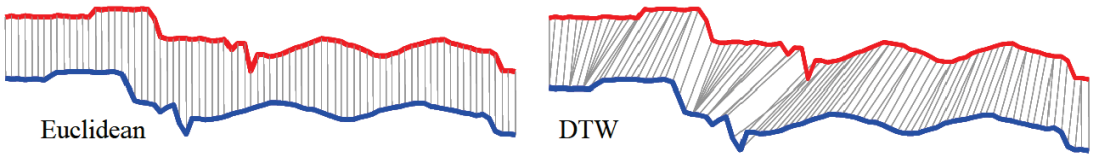


Figura 20: Comparação entre Distância Euclidiana e Dynamic Time Warping. Fonte: (KEOGH, 2002)

Devido a sua importância, principalmente para busca por similaridade em séries temporais, a Seção a seguir trás mais detalhes a respeito da *Dynamic Time Warping*.

2.5.2.4.2 Dynamic Time Warping (DTW) O *Dynamic Time Warping*(DTW) Fu et al. (2008), é largamente utilizada em busca por similaridade entre séries temporais por fazer o alinhamento entre a séries deslocadas horizontalmente entre si como o exemplo mostrado na Figura 20.

Para alinhar duas séries temporais Q e C de tamanhos n e m respectivamente é construída uma matriz de dimensão $n-por-m$ onde cada elemento (i, j) contém a distância (que pode ser inclusive a euclidiana) $\delta(q_i, c_j)$ entre dois pontos q_i e c_j das duas séries. Após calculadas as distâncias entre todos os elementos das séries temporais (nxm) e construída a matriz de distâncias, é traçado um caminho W, contíguo que representa o mapeamento entre as séries deslocadas horizontalmente Q e C, cujo cada elemento k de W é definido como $w_k = (i, j)_k$. Para encontrar o caminho W algumas restrições devem ser satisfeitas, são elas:

- Limites: $w_1 = (1, 1)$ e $w_k = (m, n)$. Essa condição faz com que o caminho W comece e termine em cantos diagonalmente opostos da matriz.

- Continuidade: Sendo $w_k = (a, b)$ então $w_{k-1} = (a, b)$ onde $a - a <= 1$ e $b - b <= 1$. Essa condição restringe os possíveis passos de caminhamento para células que sejam adjacentes, incluindo diagonalmente adjacentes.

- Monotonicidade: Sendo $w_k = (a, b)$ então $w_{k-1} = (a, b)$ onde $a - a >= 0$ e $b - b >= 0$. Essa condição força os pontos de W a serem monotonicamente espaçados no tempo.

A Figura 21 ilustra um exemplo de duas séries temporais semelhantes porém deslocadas um em relação a outra no tempo e seu posterior alinhamento.

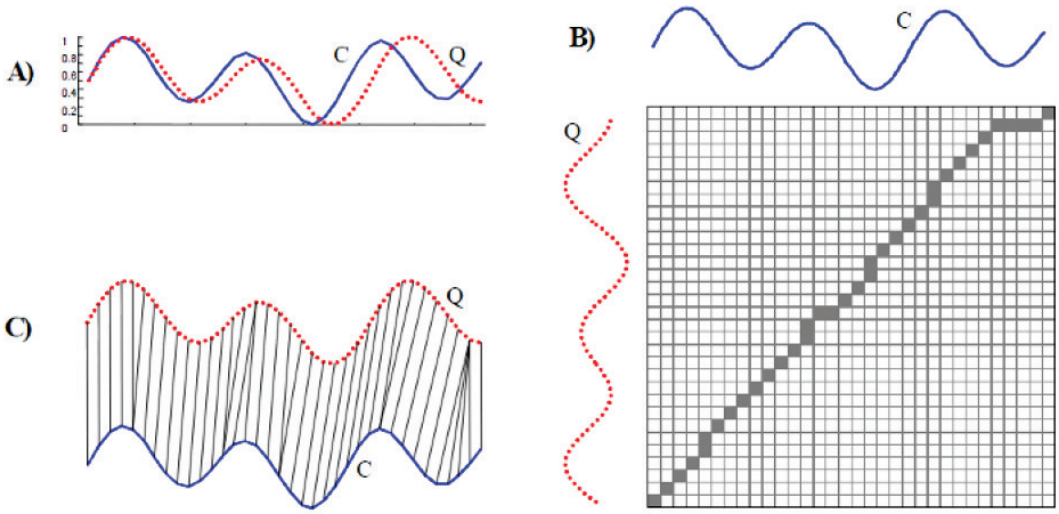


Figura 21: Exemplo de duas séries similares porém deslocadas horizontalmente e o seu alinhamento após a construção da matriz de distância e o caminho W. Fonte: (KEOGH, 2002)

Na Figura 21 em **B** tem-se a matriz de distâncias encontradas após o calculo de distâncias ponto a ponto entre as duas séries e o caminho W encontrado seguindo as condições citadas acima. Em **C** tem-se as duas séries alinhadas o que possibilita ter o calculo da distância entre elas, ou seja, o valor DTW entre as duas séries.

Como existem vários caminhos que satisfazem as condições acima é necessário então encontrar o menor caminho, ou seja, o caminho que minimize a distância, como descrito na equação abaixo.

$$DTW(Q, C) = min\{\sqrt{\sum_{k=1}^{K} w_k}\} \tag{8}$$

2.6 Conclusão do Capítulo

A utilização de dados meteorológicos para análises em diversos contextos envolve um processo interativo, no qual o conhecimento de um especialista do domínio é necessário para guiar o processo. Detalhes desde a aquisição dos dados, preparação e o uso em sí, são dependentes do objetivo da utilização e consequentemente do especialista do domínio. Este por sua vez, deve possuir conhecimento genérico o suficiente, ou amplo como a interação das variáveis em diversos contextos bem como conhecimento específico do comportamento de cada variável.

Percebe-se a clara aplicação da mineração de dados nos dados meteorológicos. Entretanto, Côrtes et al. (2002) fazem uma observação importante, de que difunde-se erroneamente o conceito de que a mineração de dados pode automaticamente minerar conceitos ou conhecimentos valiosos escondidos em uma grande quantidade de dados sem intervenção ou direcionamento humano. Quando, na verdade *".. é um processo altamente cooperativo entre homens e máquinas, que visa a exploração de grandes banco de dados, com o objetivo de extrair conhecimento através do reconhecimento de padrões e relacionamento entre variáveis.."*.

Para que seja possível extrair informações úteis dos dados, eles devem ser preparados ou transformados para se adequarem a um modelo a ser aplicado para se extrair informações. Existe um conjunto de ferramentas disponíveis na literatura que pode auxiliar o processo de mineração de dados neste tipo de dado. O SPSS SPSS (2014) por exemplo é uma ferramenta com enfoque mais estatístico que fornece um conjunto de técnicas para análise de dados. O Weka Weiss (2004), R Team (2014) e o Matlab Mathworks (2014) por exemplo oferecem diversos recursos tanto estatísticos quanto de inteligência artificial. Existem ainda outras ferramentas disponíveis, no entanto, além de fornecer técnicas para manipulação de dados, é necessário uma ferramenta que atenda as características específicas dos dados meteorológicos e permita a que o conhecimento do especialista de domínio faça parte do processo de análise dos dados.

Diante disso, este trabalho apresenta uma nova ferramenta, uma plataforma computacional para mineração de dados meteorológicos que considera as séries temporais como um novo tipo de dado e os algoritmos que as manipulam como os seus operadores. Dessa forma, é possível criar expressões de domínio que, representam o conhecimento do especialista do domínio em um determinado processo. A construção desta plataforma e a sua validação são descritas nas próximas seções.

Capítulo 3

Material e Métodos

Os modelos de processo de mineração de dados definem as etapas a serem executadas em uma determinada ordem, no entanto, não especifica quais atividades, tarefas, manipulações e processamentos devem ser realizadas em cada etapa, e por isso mesmo são chamados de modelos de processo e não de metodologia (MARBáN et al., 2009). As atividades a serem desenvolvidas em cada etapa do processo depende do contexto de aplicação, do tipo de dado utilizado e do tipo de análise a ser feita, entre outros fatores. Em meteorologia são realizados processamentos nos dados em si, principalmente em duas etapas: na **Preparação dos dados** e na **Modelagem**.

Os processamentos realizados internamente nessas duas etapas são também interativos e guiados pelo especialista de domínio. O fluxo de execução das atividades é guiado por uma série de parâmetros e escolhas que dependem da *expertise* desse especialista.

Nesse contexto, este trabalho concentra-se em fornecer uma plataforma computacional que dê suporte à execução das atividades nessas duas etapas fornecendo um mecanismo que permita que o conhecimento do especialista de domínio possa ser inserido em cada processamento realizado em cada parte do processo. Isso pode ser feito por meio da definição do fluxo e parametrização do processamento, podendo esse conhecimento ser reutilizado ou combinado com outros pré-existentes. Esta seção descreve os materiais e métodos utilizados no desenvolvimento do trabalho.

3.1 Desenvolvimento da Plataforma MiMi

O CRISP-DM mostra que há uma conexão entre o conhecimento levantado na fase de **Entendimento do negócio** e o **Entendimento dos dados**. No

entanto, esse conhecimento serve de suporte para a execução também das demais etapas do processo. Nesse sentido a Plataforma MiMi modifica o fluxo original e torna mais natural a interação entre fases como as incluídas pelas setas azuis na Figura 22.

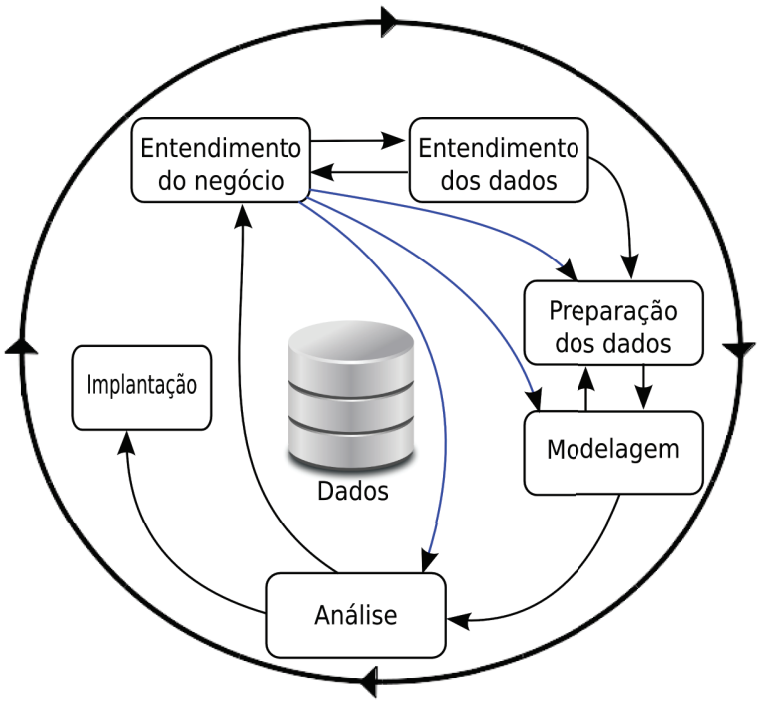


Figura 22: Entendimento do Negócio influenciado demais etapas do processo de Mineração de Dados. Adaptado de Chapman et al. (2000).

O conhecimento da fase de **Entendimento do negócio** influencia todo o processo, o que gera a necessidade de vinculá-lo com a parametrização de cada etapa de processamento. Entretanto, no modelo tradicional CRISP-DM não fica explicito essa influência, bem como as ferramentas de DM também não refletem a necessidade de interação do Especialista em todo o processo. A plataforma MiMi busca resolver esse problema com a explicitação da influência da expertise do Especialista nas outras etapas, conforme ilustrado nas setas azuis da Figura 22.

Os modelos de processo definem as etapas do processo de mineração, entretanto, cada etapa é composta por atividades internas específicas. As etapas de **Preparação dos dados** e **Modelagem** envolvem manipulação direta nos dados. Há um conjunto interno de atividades que engloba essas duas etapas e que é também interativo e guiado pelo conhecimento do especialista como mostrado na Figura 23.

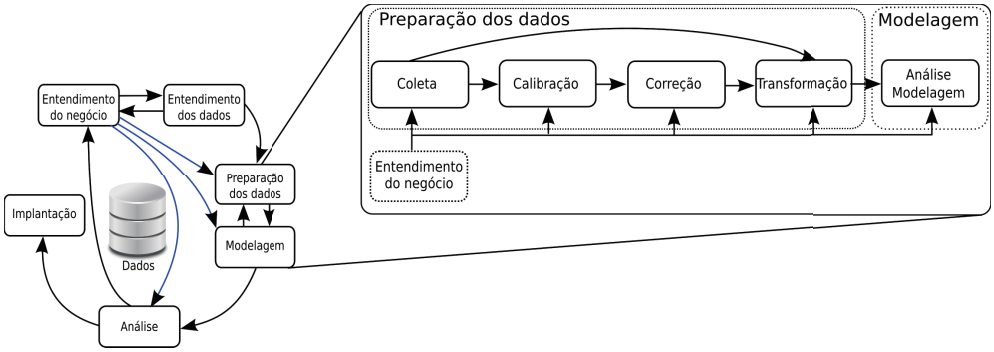


Figura 23: Detalhamento das fases de Preparação dos Dados e Modelagem. Adaptado de Chapman et al. (2000).

Após a **Coleta** de dados é comum os dados serem calibrados, isso internamente é feito utilizando por exemplo, um aparelho de referência que mede a mesma variável por um determinado período de tempo. Após, são extraídos os coeficientes de correlação entre os dados do equipamento de referência e o equipamento de trabalho para a correção (BIUDES et al., 2015).

Depois de feita a **Calibração** dos dados, mas não obrigatoriamente, os dados passam por um processo de **Correção**, onde são identificadas possíveis falhas nos dados e são então corrigidos (VENTURA et al., 2014).

Com os dados já corrigidos eles podem passar por um processo de **Transformação**. Essa transformação pode ocorrer na representação dos dados, na maioria das vezes essas transformações são a respeito da granularidade dos mesmos. Os dados podem por exemplo ser coletados a cada 5 minutos e depois ser calculada a média diária para a posterior análise. Além desse tipo de transformação, nesse momento podem ser gerados novos dados a partir dos dados coletados.

Após a transformação dos dados de acordo com o interesse da análise a ser feita, são aplicados modelos de **Análise** ou de **Visualização** dos dados.

É importante notar que o modelo de processo proposto pelo CRISP-DM (Figura 3) não apresenta uma ligação direta entre o **entendimento do negócio** e as etapas de **Preparação dos dados** e **Modelagem**. Nos trabalhos envolvendo dados meteorológicos essa ligação existe e por isso ela é apresentada na Figura 22. Por exemplo, se na etapa de **entendimento do negócio** foi definido que seriam necessários dados de radiação solar, os dados coletados entre o pôr e o nascer do sol devem ser descartados, pois nesse período não há radiação solar direta incidente sobre a área estudada.

Nesse contexto, este trabalho tem como enfoque as etapas do processo de mineração de dados que envolvem processamento de dados, como mostrado na Figura 23. Dessa forma, tendo o dado como objeto central do processo, foi

definida uma arquitetura para tratamento dos dados meteorológicos no processo de mineração de dados.

A plataforma MiMi foi desenvolvida para dar suporte ao desenvolvimento de aplicações de mineração de dados. Para isso foi definida uma arquitetura que contempla o processo de acesso aos dados, preparação dos dados e mineração. Nessa arquitetura foi definida uma álgebra na qual as séries temporais são tratadas como um tipo de dado. Esse tipo de dado é definido como o operando série temporal e os operadores representam os algoritmos que o manipulam. As definições do operando e operadores foram baseadas no trabalho de Traina et al. (2005). Através da agregação de operações em expressões de domínio é possível embutir o conhecimento do especialista. Maiores detalhes são apresentados no Capítulo 4.

3.1.1 Tecnologia Utilizada

A plataforma proposta foi implementada utilizando a linguagem de programação Java Oracle (2014). Foi definida uma API (Application Program Interface) seguindo os padrões de desenvolvimento utilizando-se de Classes *Abstratas* e *Interfaces* com o intuito de se ter um código genérico para que a plataforma seja facilmente estendida para outros contextos e para facilitar o desenvolvimento de novos operadores.

Outro recurso da linguagem Java que foi utilzado foi o padrão de desenvolvimento *Singleton* (Freeman et al. (2004)) que permite que seja mantido apenas um objeto para todas as instâncias de uma determinada classe. essa funcionnalidade foi utilizada para implementar o acesso ao operando série temporal.

Foi utilizado também o padrão XML (Bibeck et al. (2001)) para a configuração da execução dos operadores. Para criar os objetvos java que representam os operadores em tempo real, a partir da leitura do arquivo XML foi utilizada a API Java Reflection, que permite a manipulação de códigos java em tempo de execução.

A IDE (*Integrated Development Environment*) utilizada para o desenvolvimento foi o NetBeans IDE 8.0 NetBeans (2014).

3.2 Testes

Para validar a plataforma proposta foram realizados testes para 3 tipos diferentes de tarefas de mineração de dados para cada conjunto de dados.

3.2.1 Descrição dos dados

Para a realização dos restes foram utilizados dados de duas estações meteorológicas. A primeira, denominada Torre UFMT está localizada no Campus da Universidade Federal de Mato Grosso (UFMT) , Cuiabá - Mato Grosso. A segunda, denominada Torre Santo Antônio está localizada na Fazenda Experimental da UFMT no município de Santo Antônio de Leverger - Mato Grosso denominada Torre. As duas torres meteorológicas estão em locais com características distintas, a Torre UFMT está localizada na zona urbana e a Torre Santo Antônio está na zona rural.

Para ambas as torres foram utilizados dados de:

- Temperatura do Ar (Ta) ;
- Umidade Relativa do Ar (Ur)
- Radiação Solar (RN)
- Precipitação (P)

A periodicidade de coleta dos dados para cara torre é mostrado na Tabela 3

Tabela 3: Organização dos dados para teste.

	Período de Coleta	Periodicidade	Obs.
Torre UFMT	Jan-2011 até Set-2014	15 minutos	-
Torre Santo Antônio	Jan-2005 até Dez-2009	Diário	Média Diária

Os dados da Torre Santo Antônio foram fornecidos pelo Instituto Nacional de Meteorologia (INMET) na estação meteorológica os dados são coletados 3 vezes ao dia, no entanto foi fornecido dados resultantes da médias desses valores, dados de média diária.

3.2.2 Agrupamento

Para realizar o agrupamento de séries temporais foi utilizado o algoritmo *k-means* utilizando médias mensais de Temperatura Ar, Umidade Relativa do Ar e Radiação Solar, ou seja, foi utilizado para cada torre meteorológica um vetor de características com 3 atributos como mostrado na Figura 24. Essa configuração gerou para a Torre Santo Antônio um vetor de características com 60 elementos (1 para cada mês de 2005 a 2009) e 3 atributos. Para a Torre UFMT foi gerado

um vetor com 36 elementos (Maio a Dezembro de 2011, Janeiro a Dezembro de 2012, Janeiro a Outubro de 2013 e Fevereiro a Setembro de 2014) e 3 atributos.

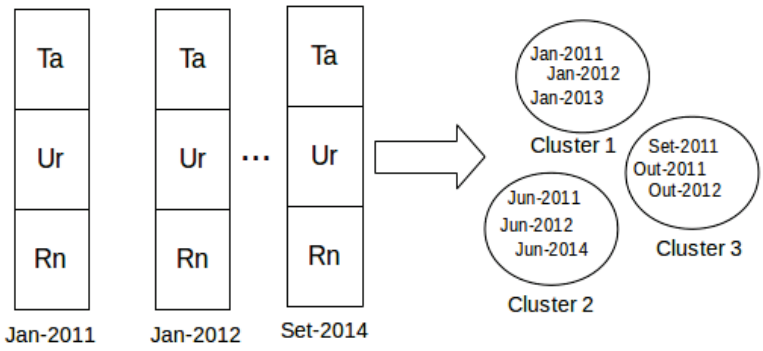


Figura 24: Ilustração do funcionamento do algoritmo k-means para 3 clusters e com vetores de 3 características.

O k-means tem como parâmetros principais a quantidade de *clusters* e a função de distância. Foi utilizado a medida de 3 *clusters* com o intuito de validar a definição de estações Chuvosa, Seca e Intermediária que é utilizada na literatura para o estado de Mato Grosso. A função de distância utilizada foi a Euclidiana.

3.2.3 Preenchimento de Falhas

O preenchimento de falhas nos dados foi realizado utilizando o trabalho desenvolvido por Ventura et al. (2013) que utiliza uma combinação de algoritmos genéticos e redes neurais para o preenchimento de falhas.

A técnicas proposta destina-se ao preenchimento de falhas em dados multivariados de meteorologia. O preenchimento das falhas é feito pela rede neural, que estima os valores faltantes baseando-se nos valores das outras variáveis disponíveis. O Algoritmo genético é reponsável por determinar quais variáveis devem ser utilizadas pela rede neural bem como os seus principais parâmetros.

3.2.4 Busca Por Similaridade

A busca por similaridade foi implementada utilizando o algoritmo DTW para a comparação entre duas séries temporais. Para a busca foi passada uma série temporal equivalente a 1 mês de dados de média diária de temperatura da Torre PGFA e a busca foi realizada na série da Torre Santo Antônio. Para realizar a busca a série passada como parâmetro é comparada com subséries da série total da Torre Santo Antônio geradas por meio da técnica de Janela Deslizante, ilustrada na Figura 25.

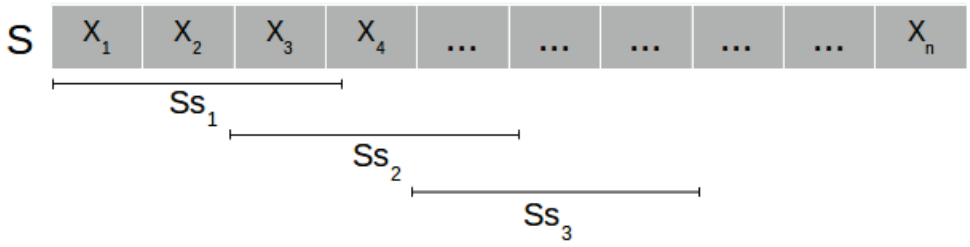


Figura 25: Ilustração do funcionamento da técnica de Janela Deslizante.

A Janela Deslizante funciona recebendo uma série temporal S com os elementos $x_1, x_2, ..x_n \in S$ e dois outros parâmetros, o tamanho da janela w e o tamanho do salto d, extrai-se então a primeira janela contendo os elementos x_1 até x_w depois extrai-se as demais janelas iniciando-se em x_{i+d} até x_{i+w+d} até que $i <= n - w$.

Para cada subsequência extraída pela janela deslizante a mesma é submetida a uma comparação com a série passada como parâmetro para o algoritmo DTW, o algoritmo retorna então uma distância que diz o quão parecidas são as duas séries comparadas, cada valor de distância é armazenado em um vetor de distância, após o término do processo o vetor de distâncias é ordenado para que se tenha acesso as subsequências mais similares. Esse processo é ilustado na Figura 26.

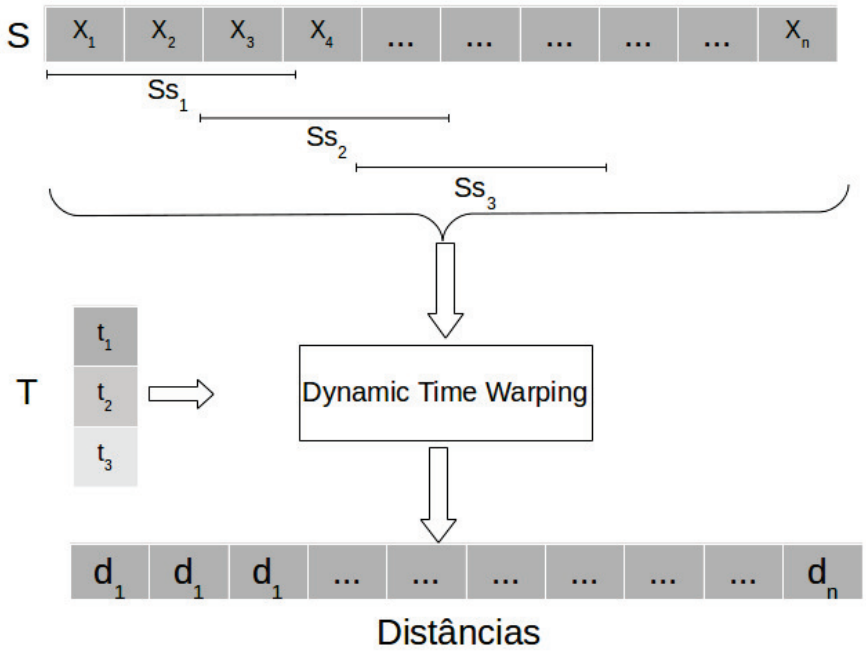


Figura 26: Ilustração do funcionamento da técnica de Janela Deslizante e do Algoritmo DTW para a busca por similaridade.

3.2.5 Descoberta de Padrões Desconhecidos

Para a tarefa de descoberta de padrões recorrentes desconhecidos na série temporal foi utilizado o algoritmo de Descoberta de MOTIFS (PATEL et al., 2002). Existem diversas versões do algoritmo de descoberta de MOTIFS, nesse trabalho foi utilizado o algoritmo proposto por Castro e Azevedo (2010).

Foram utilizados os dados da Torre Santo Antônio com médias diárias entre os anos de 2005 e 2009. Buscou-se padrões de tamanhos 7,10,15,20 e 30.

Capítulo 4

Resultados e Discussões

A plataforma MiMi baseia-se em uma formalização algébrica proposta por Traina et al. (2005). A diferença é que a álgebra original trata imagens como sendo um novo tipo de dados cuja principal característica é suportar comparações entre instâncias de imagens. Já a formulação proposta na plataforma MiMi tem como enfoque a manipulação de séries temporais como sendo o tipo de dado básico.

Foi definido o Operando Série Temporal e os Operadores que o manipulam. Assim, é possível, por meio dessas definições com um conjunto de operadores, definir o fluxo de processamento para cada atividade de mineração de dados com a parametrização e encapsulamento seguindo a expertise do especialista.

4.1 Plataforma Computacional para Mineração de Dados Micrometeorológicos - MiMi (Micrometeorological Data Mining Plataform)

A plataforma MiMi acrescenta dois novos componentes à arquitetura de software para mineração de dados se comparada às demais descritas na literatura: (1) o **Operando** Série Temporal, que mantém um conjunto de séries temporais que pode ser acessado por qualquer componente dessa arquitetura; (2) e o **processador** que é responsável por implementar a sequência de processamento que deve ser realizada nas tarefas de mineração de dados de acordos com os modelos de processos.

A Figura 27 mostra a representação gráfica dos módulos da plataforma MiMi.

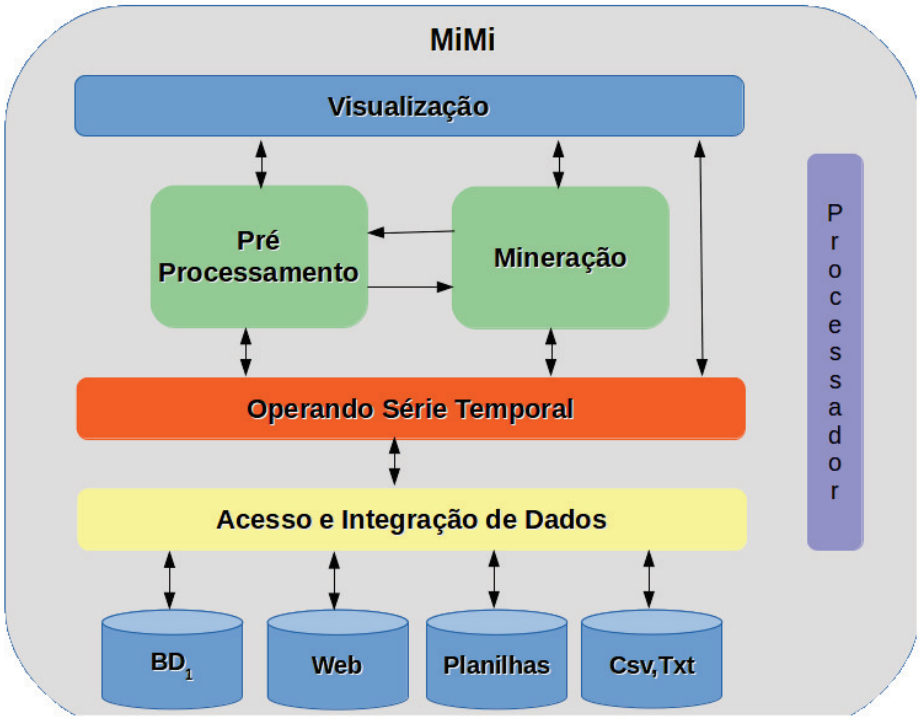


Figura 27: Modularização da plataforma MiMi.

O módulo de **Acesso e Integração de Dados** é responsável por acessar uma base de dados para disponibilizar a série temporal que será utilizada no processo por meio do operando série temporal, essa base pode ser um Sistema Gerenciador de Banco de Dados, um arquivo texto ou ainda a ligação direta com uma rede de sensores, por exemplo. A Integração de dados pode ser feita, quando mais de uma base de dados está disponível.

O módulo de **Pré-Processamento** é onde são executados os algoritmos que preparam a série temporal para a fase posterior. Nessa camada geralmente são executados algoritmos que trabalham a qualidade dos dados ou sua representação computacional.

O módulo de **Mineração** é onde são executados os algoritmos responsáveis pela extração do conhecimento em si. Essa camada acessa a série temporal já trabalhada na fase anterior.

O módulo de **Visualização** é responsável por mostrar para o usuário especialista de domínio o resultado da fase anterior. É possível perceber que essa camada tem acesso à série temporal desde a camada de acesso aos dados, pois ela pode servir também de auxílio para as camadas inferiores por meio da análise visual dos dados.

A plataforma MiMi baseia-se no conceito de que o processo de mineração

de dados em séries temporais ocorre como um fluxo de processamento consecutivo, podendo haver voltas a passos anteriores como ilustra a Figura 3. Em todas as etapas do processo há um elemento em comum, a Série Temporal, que é o dado tratado em todo o processo. Por isso, foi definida uma arquitetura que trata a série temporal como sendo um Operando e os métodos que atuam sobre esse Operando são chamados de Operadores. Essa definição é baseada no trabalho de Traina et al. (2005) em que foi definido o Operando Imagem e os Operadores que atuam sobre esse operando.

O Operando Série Temporal é definido como segue:

$$\lambda(<\phi>) \tag{9}$$

Onde :

$$\phi = \{S_1, S_2, ..., S_n\} \tag{10}$$

e:

$$S = \{(time_1, < var_1, ..., var_d >), ..., (time_t, < var_1, ..., var_d >)\} \tag{11}$$

Sendo: $time$ o instante de tempo em que os valores das variáveis (var) foram medidas. d é a dimensão da série temporal, ou seja, a quantidade de variáveis medidas.

Cada etapa do processo de mineração de dados é composta por algoritmos que manipulam a série temporal. Para a implementação de uma plataforma mais flexível e de fácil manipulação e entendimento, os algoritmos que manipulam as séries temporais são tratados como sendo operadores do operando imagem.

Os operadores são definidos como:

$$\Theta(args) : \lambda_i \rightarrow \lambda_{i+1} \tag{12}$$

O operador $\Theta(args)$ recebe como parâmetro $args$ que pode ser um ou mais parâmetros e manipula uma ou mais série temporal, podendo ou não adicionar uma nova série à λ.

Os operadores são divididos em três tipos: Operadores de Controle, Operadores de Pré-Processamento e Operadores de Mineração.

Os Operadores de Controle são descritos a seguir.

- Θ_{load}: carrega uma série temporal e aloca como primeiro elemento de λ;

- Θ_{add}: adiciona uma série temporal após a última posição de λ;
- Θ_{get} : recupera a série corrente de λ;
- Θ_{move} : transforma série presente na posição n recebida como parâmetro como sendo a série corrente;
- Θ_{swap}: faz uma cópia da série atual em λ;

Esses operadores têm a função de manipular diretamente o operando série temporal. Por meio desses, os outros operadores que vierem a ser implementados manipular o conjunto de série temporal. Esses operadores são necessários para encapsular o acesso aos dados, uma vez o conjunto λ deve ser o mesmo para todos os operadores.

Os operadores de pré-processamento não são pré-definidos, eles devem ser implementados para cada aplicação. São os operadores executados geralmente na fase de entendimento e preparação dos dados. Esses operadores geralmente adicionam uma nova série temporal à λ.

Os operadores de mineração de dados são os responsáveis por executar as tarefas de mineração de dados, eles não adicionam necessariamente uma nova série temporal em λ, o resultado de cada operador depende da aplicação em questão.

Com a definição de operando e operadores é possível definir o que Traina et al. (2005) chamou de Expressão de Domínio (Θ_∂) que é exatamente uma sequência de operações que devem ser executadas na tarefa de mineração de dados, ou seja, a execução sequencial de diversos operadores, como em: $\Theta_1 : \Theta_2 : \Theta_3$. O símbolo ':' é usado para expressar a sequência de execução dos operadores, de forma que $\Theta_1 : \Theta_2$ indica que Θ_2 é executado depois de Θ_1 ser processado tendo acesso às alterações feitas no operando λ por Θ_1. Pelo fato de que cada Θ tem como resultado um operando λ, o resultado final da Expressão de Domínio (Θ_∂) também será um operando λ. Com isso, uma $\Theta_{\partial 1}$ pode ser utilizada dentro de outra $\Theta_{\partial 2}$ como um operador comum. Como uma expressão de domínio Θ_∂ pode ser considerada um ciclo de processamento conforme descrito pelo CRISP-DM, o reuso de uma expressão de domínio por outra, permite que o ciclo seja percorrido novamente.

A expressão seguinte ilustra um exemplo definição de uma expressão de domínio Θ_∂.

$$\Theta_{\partial FindPatterns} = \Theta_{load}(\text{“}dados.csv\text{”}) : \Theta_{gapFilling}() : \Theta_{motifDiscovery}(10) \qquad (13)$$

A expressão, denominada FindPatterns, possui o primeiro operador $\Theta_{load}(\text{“}dados.csv\text{”})$ que lê os dados do arquivo “dados.csv” e carrega a primeira série temporal em λ, após o operador $\Theta_{gapFilling}()$ realiza a operação de preenchimento de falhas na série corrente, para isso, internamente o operador de Pré-Processamento deve fazer o uso de um dos operadores de controle como o *get* que da acesso a última série temporal adicionada. Posteriormente, é executado o operador $\Theta_{motifDiscovery}(10)$ que recebe como parâmetro um número inteiro.

4.2 MiMi API

A implementação da plataforma MiMi seguiu a definição de uma API (*Aplication Program Interface*) com o objetivo de ser genérica, para facilitar a inclusão de novos componentes e possibilitar a reprodução do fluxo de processamento apresentado na Figura 23 no processo de mineração de dados. Para isso definiu-se uma organização de Classes base para implementar a plataforma, essas classes são: *MiMiData*, *GenericTimeSeries*, *TimeSerieOperand* e *Operator*. A Figura 28 mostra o diagrama de Classes da plataforma MiMi.

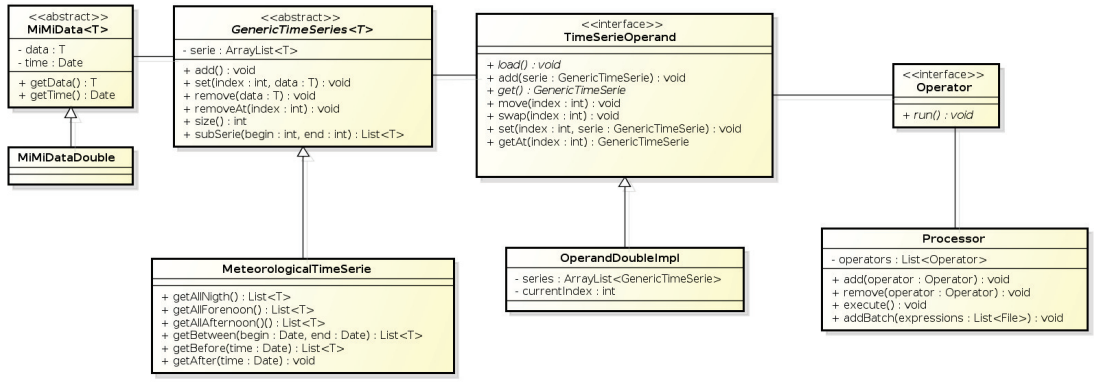


Figura 28: Diagrama de Classes da plataforma MiMi.

4.2.1 Classe MiMiData

A Classe *MiMiData* é utilizada para representar o tipo de dado o qual a série temporal será composta. A Figura 29 ilustra a modelagem da Classe *MiMiData* e uma subclasse *MiMiDataDouble*.

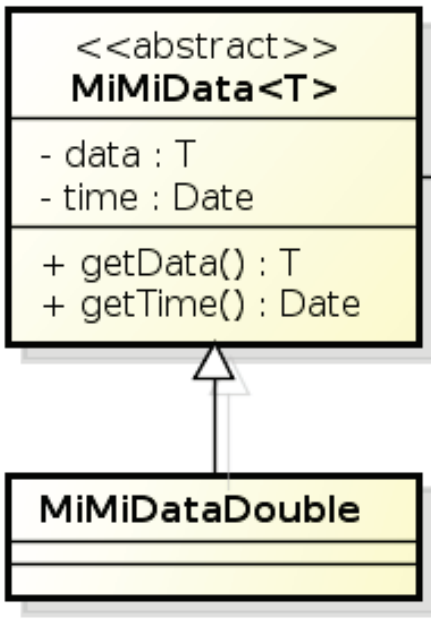


Figura 29: Representação da Classe Abstrata *MiMiData* e a Classe *MiMiDataDouble*.

A Classe *MiMiData* é uma Classe abstrata, para não permitir uma instância de objeto *MiMiData*, utilizou-se então do recurso de tipo de dado genérico ($< T >$) disponível na linguagem Java, dessa forma é obrigatório que se tenha uma subclasse que estenda a Classe MiMiData e defina qual o tipo de dado T será utilizado. Na Figura 29 é mostrada a subclasse *MiMiDataDouble* que foi criada para utilizar o tipo de dado *Double*.

A Classe MiMiData contém dois atributos o primeiro *time* é do tipo *Date* que representa a unidade de tempo para cada dado. O atributo *data* que é do tipo T, definido a subclasse, representa o dado em si.

É importante destacar que a subclasse pode definir qualquer tipo de dado para o atributo *data* inclusive tipos definidos pelo programador, permitindo por exemplo, representações multidimensionais.

4.2.2 Classe GenericTimeSeries

A Classe *MiMiData* faz com que o tipo de dado utilizado seja genérico, definiu-se também a Classe *GenericTimeSeries* com o objetivo de se ter uma série temporal genérica. A Figura 30 mostra a definição da classe *GenericTimeSerie*.

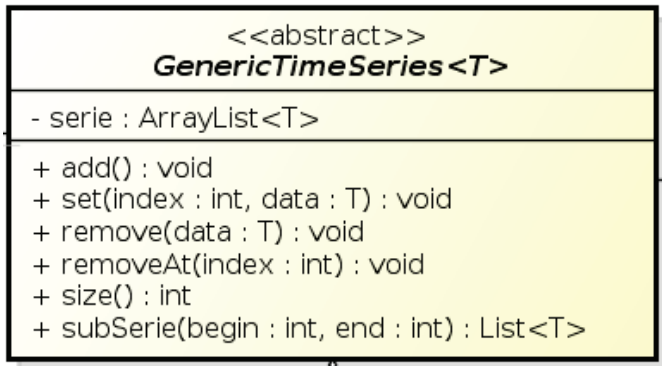


Figura 30: Representação da Classe Abstrata *GenericTimeSerie*.

O estereótipo << *abstract* >> faz com que não seja possível existir uma instância da classe *GenericTimeSerie*. Para que ela seja utilizada é necessário que uma outra classe a estenda e defina por qual tipo de dados será composta a série passando o parâmetro $< T >$ que define o tipo de dado, neste trabalho essa definição foi feita para utilizar os tipos de dados definidos por meio da Classe *MiMiData*. A Classe contém os métodos comuns na manipulação de listas em Java, o método *add*() adiciona um elemento do tipo T à série. O método *set*() adiciona um objeto mento do tipo T em uma posição específica da série. O método *remove*() recebe como parâmetro um objeto e o remove da série, o método *removeAt*() diferentemente, recebe um índice e remove o objeto da série no índice correspondente. O método *size*() retorna o tamanho atual da série. O método *subSerie*() recebe como parâmetro dois números inteiros, o primeiro representando o índice na serie original correspondente ao início da subsérie e o segundo um índice da série original correspondente ao final da subsérie.

4.2.3 Classe TimeSerieOperand

Neste trabalho definiu-se o Operando $TimeSeries\lambda$ representado pela Classe Interface *TimeSerieOperand* mostrado na Figura 31. O Operando λ deve ser acessível à todos os operadores dentro da plataforma proposta para possibilitar a execução do fluxo de processamento dos algoritmos de mineração de dados.

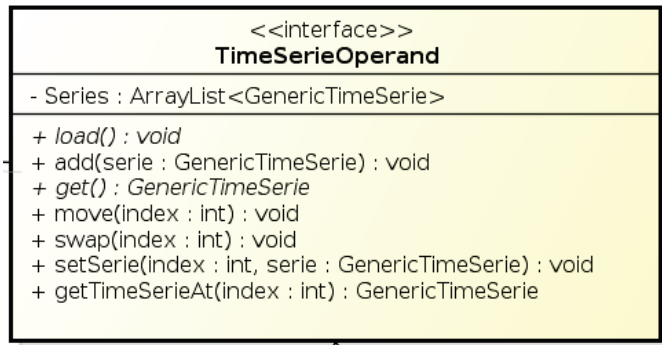


Figura 31: Representação da Classe Interface do Operando Time Series (λ)

A Figura 31 mostra a definição da Classe Interface $TimeSerieOperand$, o estereótipo $<< Interface >>$ define que essa classe precisa ser implementada e que todos os seus métodos devem ser sobrescritos. Essa definição foi feita para que o operando se tornasse genérico, percebe-se na Figura 31 que a Classe tem um atributo do tipo $GenericTimeSerie$, que é um tipo de dado genérico que também deve ser definido de acordo com a aplicação. Foram definidos na interface os métodos que funcionam como os operadores base Θ_{load}, Θ_{add}, Θ_{get}, Θ_{move}, Θ_{swap} (Seção 4.1), que devem ser sobrescritos na classe que implementar a interface.

O operador Θ_{load} é responsável por carregar a série inicial na lista de series temporais ($ArrayListSeries$), e é implementado no método $load()$. Após a execução do método $load()$ os outros métodos podem ser acionados. O método $add()$ recebe como parâmetro uma série e adiciona-a ao final lista de séries. O método $set()$ é similar ao $add()$ a diferença é que ele recebe como parâmetro além de uma série um índice e adiciona a série passada como parâmetro à posição correspondente na lista de séries. O método $get()$ retorna a última série da lista de séries. Similar ao $get()$ o $getAt()$ recebe como parâmetro um número inteiro ($index$) representando um índice na lista de séries e retorna a série localizada no índice passado como parâmetro. Os métodos $move()$ e $swap()$ atuam sobre as séries já adicionadas na lista. O $move()$ recebe como parâmetro um índice e transforma série presente na lista de série nesse índice e a transforma na série corrente. O método $swap()$ faz uma cópia da série atual.

É interessante observar que nos casos em que o tamanho da série temporal seja muito grande a utilização do operador $add()$ pode prejudicar o desempenho do processo ou causar um estouro de memória. Sendo assim, a utilização do mesmo deve ser feita quando há a necessidade de manter as séries anteriores, caso não seja necessário, deve ser utilizado o método $set()$ que não adiciona um novo elemento, mas sobrepõem um existente.

4.2.4 Classes Operator e Processor

Como citado anteriormente, os algoritmos que manipulam as séries temporais são vistos agora como Operador que manipulam o Operando *TimeSerieOperand*. Isso foi feito com o objetivo de construir uma arquitetura para que seja possível automatizar o fluxo de processamento comum nas tarefas de mineração de dados. Para isso, foi definida uma Classe Interface chamada *Operator* que define um método que deve ser implementado, o método *run*(). A Figura 32 mostra a definição da classe *Operator*.

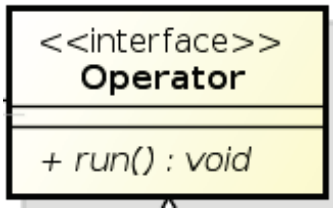


Figura 32: Representação da Classe interface Operator.

Os algoritmos que atuam sobre a série temporal nas tarefas de mineração de dados devem implementar a classe *Operator* e implementar o método *run*(). Para reproduzir o fluxo de processamento foi definida uma classe responsável por isso, denominada *Processor*, Figura 33.

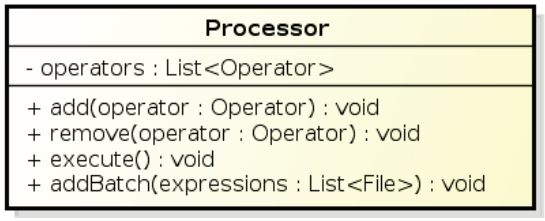


Figura 33: Representação da Classe Processor.

A classe *Processor* contém uma lista de *Operator* que deve armazenar todos os operadores a serem executados, para isso foi definido o método *add*() que recebe como parâmetro um objeto *Operator* e o adiciona à lista de *Operator* (atributo $operators : List < Operator >$). O método *remove*() remove um objeto da lista de operadores. O método *addBatch*() recebe como parâmetro uma lista de arquivos, esses arquivos seguem o formato *xml* contendo cada um, uma ou mais expressões de domínio Θ_∂. No arquivo *xml* é possível descrever cada operador da expressão de domínio com os seus parâmetros. Após receber a lista de arquivos o método adiciona os operadores na sequência definida nos arquivos à lista de operadores, essa inserção é feita utilizando *Reflections* da linguagem Java, que

permite objetos serem criados e instanciados em tempo de execução. O método *execute*() é responsável por reproduzir o fluxo de processamento, ele executa os operadores adicionados à lista invocando o método *run* de cada operador na ordem em que foram adicionados. O código seguinte mostra uma implementação do método *processor*(), o método percorre toda a lista de operadores e para cada operador executa o método *run*.

```
1
2  public void execute(){
3          for (Operator operator : operators) {
4                  operator.run();
5          }
6  }
```

Na seção seguinte são apresentados testes reais feitos com a plataforma apresentada.

4.3 Testes

Para validar a modelagem da plataforma proposta foram realizados testes com 3 tarefas comuns na mineração de dados, Agrupamento, Descoberta de Padrões e Busca por Similaridade. Foram implementados então 3 operadores, *kmeans*, $MotifsDiscovery$ e $SimilaritySearch$ como mostrado na Figura 35.

Entretanto, antes da definição dos operadores é necessário definir o tipo de dado a ser utilizado na série temporal, para isso deve-se estender a classe $GenericTimeSerie$ e definir o tipo de dado que compõem a lista e depois é necessário implementar a interface $TimeSerieOperand$ e adicionar um objeto do tipo da classe que estende a $GenericTimeSerie$. Isso é mostrado na Figura 34.

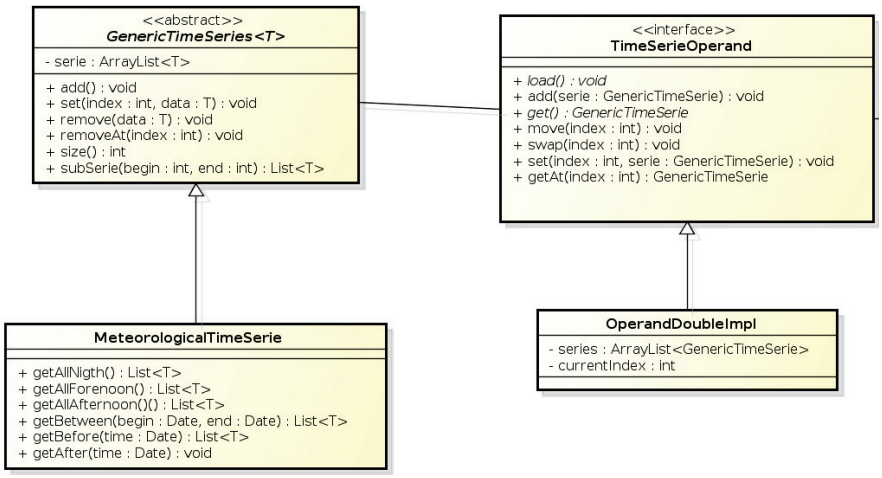


Figura 34: Definição do tipo de série e implementação do operando série temporal λ.

A Classe *GenericTimeSeries* fornece alguns métodos básicos para manipulação de listas de dados. Além desses métodos comuns na manipulação de listas, a Classe $MeteorologicalTimeSerie$ fornece métodos específicos para manipulação da dimensão temporal de dados meteorológicos. Foram definidos 6 métodos, descritos a seguir:

1. $getAllNigth()$: Este método retorna uma série temporal contendo apenas os dados noturnos da série original.

2. $getAllForenoon()$: Este método retorna uma série temporal contendo apenas os dados do período da tarde da série original.

3. $getAllAfternoon()$: Este método retorna uma série temporal contendo apenas os dados do período da manhã da série original.

4. $getBetween()$: Este método recebe como parâmetro duas datas, ou duas representações de tempo e retorna uma série temporal contendo os dados compreendidos entre as duas datas.

5. $getAfter()$ e $getBefore()$: Ambos os métodos recebem como parâmetro uma data e retornam uma série temporal contento todos os dados antes e depois da data especificada respectivamente.

A Figura 35 mostra a modelagem dos operadores que foram implementados.

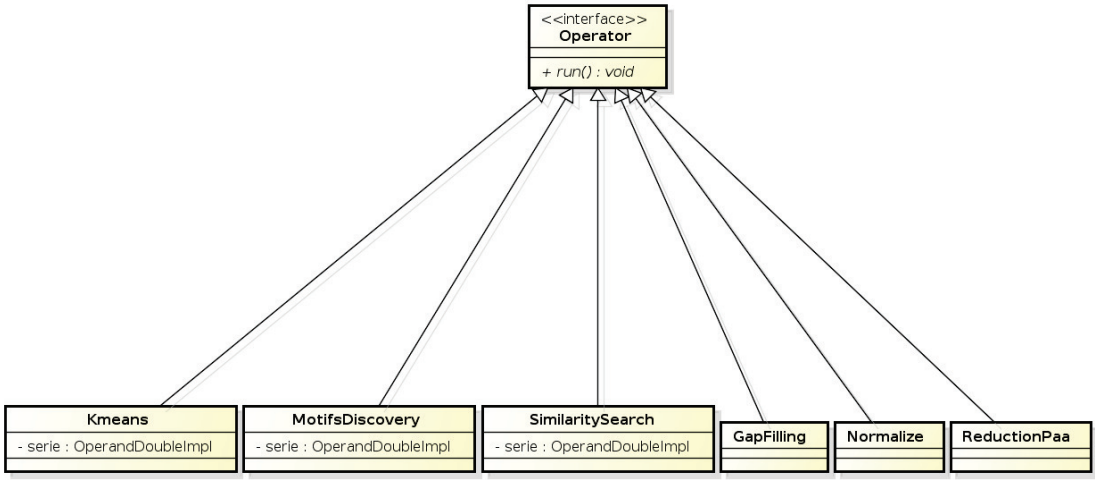


Figura 35: Operadores implementados para teste da plataforma proposta.

Além dos operadores Θ_{kmeans}, $\Theta_{MotifsDiscovery}$ e $\Theta_{SimilaritySearch}$ foram implementados operadores que são geralmente utilizados no pré-processamento dos dados, um operador de preenchimento de falhas $\Theta_{GapFilling}$, outro de normalização de dados $\Theta_{Normalize}$ e um de redução de dimensionalidade $\Theta_{ReductionPaa}$.

4.3.1 Execução dos Testes e Discussão

Depois de definidos e implementados os operadores para testes eles foram validados. Para cada teste de operador foi definido um fluxo de processamento dividido em duas expressões de domínio. A primeira $\Theta_{\partial PreProcessamento}$ contemplando as fases de seleção de dados e pré-processamento. A segunda expressão $\Theta_{\partial 2}$ contém apenas o operador de mineração. Os testes foram conduzidos dessa maneira para validar a capacidade da plataforma de gerar fluxo de processamento de forma simples baseado em expressões de domínio.

A primeira expressão de domínio $\Theta_{\partial 1}$ foi utilizada em todos os testes, ela é definida como segue:

$$\Theta_{\partial PreProcessamento} = \Theta_{load}(\text{“}dados.csv\text{”}) : \Theta_{gapFilling}() : \Theta_{PAA}(30) \qquad (14)$$

O operador $\Theta_{load}(\text{“}dados.csv\text{”})$ recebe como parâmetro o nome de um arquivo contendo os dados para serem processados. O operador $\Theta_{gapFilling}$ executa o preenchimento de falhas no conjunto de dados, caso elas existam. O operador $\Theta_{PAA}(30)$ executa o algoritmo do método de representação de dados *Piecewise Aggregate Approximation* recebendo como parâmetro o número 30 que representa o tamanho da janela utilizada para realizar a média dos dados.

A Figura mostra o código XML que representa a expressão de domínio descrita anteriormente.

```
<expression name="PreProcessing">
        <operator name="load" class="org.br.ufmt.ic.gaiia.mimi.operator.manipulation.Load">
                <param name="file" values="dados.csv"/>
        </operator>
        <operator name="gapFilling" class="org.br.ufmt.ic.gaiia.mimi.operator.preprocessing.GapFilling"/>
        <operator name="paa" class="org.br.ufmt.ic.gaiia.mimi.operator.preprocessing.Paa">
                <param name="window" values="30"/>
        </operator>
</expression>
```

Figura 36: Expressão de Domínio Θ_{∂} de pré-processamento.

4.3.1.1 Agrupamento

Para a execução da tarefa de agrupamento dos dados foi necessário primeiro realizar a normalização dos dados. Assim, foi definido operador $\Theta_{normalizeData}()$ e o operador $\Theta_{kmeans}(numCluster, DistanceFunction)$. O operador Θ_{kmeans} recebe dois parâmetros: o número de clusters a ser utilizado e a função de distância. Para realizar todo o processamento foi combinado a execução da expressão de domínio $\Theta_{\partial PreProcessamento}$ e os dois operadores descritos anteriormente. Nota-se que a atividade de normalização dos dados é também uma atividade de preparação dos dados, o que evidencia a interação nos dois sentidos entre a fase de **Preparação dos Dados** e **Modelagem**. A Equação 15 mostra a representação da expressão de domínio utilizada para o teste de agrupamento dos dados.

$$\Theta_{\partial Kmeans} = \Theta_{normalizeData}() : \Theta_{kmeans}(3, \text{“}euclidean\text{”}) \tag{15}$$

Os dados utilizados como teste para o algoritmo *kmeans* estavam organizados no arquivo com médias diárias, o preenchimento de falhas corrigiu apenas falhas menores que 1 mês de dados, quando um mês inteiro estava ausente na série, este foi ignorado. Após o preenchimento de falhas foi executado o algoritmo do método de representação PAA (Seção 2.5.1.2.2) utilizando como tamanho do intervalo para cálculo da média 30 dias, o que equivale a utilizar dados de média mensal. Como os dados tem faixas de valores muito diferentes foi executado posteriormente um operador de normalização dos dados e depois executado o algoritmo *kmeans*. A Figura 37 mostra o código XML que representa a junção das duas expressões de domínio para execução do pré-processamento e do algoritmo *kmeans*.

```
<expression name="PreProcessing">
        <operator name="load" class="org.br.ufmt.ic.gaiia.mimi.operator.manipulation.Load">
                <param name="file" value="dados.csv"/>
        </operator>
        <operator name="gapFilling" class="org.br.ufmt.ic.gaiia.mimi.operator.preprocessing.GapFilling"/>
        <operator name="paa" class="org.br.ufmt.ic.gaiia.mimi.operator.preprocessing.Paa">
                <param name="window" value="30"/>
        </operator>
</expression>
<expression name="kmeans">
        <operator name="normalizeData" class="org.br.ufmt.ic.gaiia.mimi.operator.preprocessing.Normalize"/>
        <operator name="kmeans" class="org.br.ufmt.ic.gaiia.mimi.operator.datamining.clustering.Kmeans">
                <param name="clusterNumber" value="dados.csv"/>
        </operator>

</expression>
```

Figura 37: Expressão de Domínio Kmeans.

Para a execução do processamento o arquivo XML deve ser passado como parâmetro para o método *addBatch* da classe *Processor*, essa classe utilizando *Reflection* constrói os objetos de acordo com as especificações do arquivo XML.

Caso não seja utilizado o arquivo XML para a configuração da expressão de domínio, pode-se construir a expressão via código java, como mostrado no código a seguir.

```
1  public class TestKmeans {
2          public static void main(String[] args){
3                  Processor processor = new Processor();
4                  processor.addOperator(new Load("dados.csv"))
                       ;
5                  processor.addOperator(new GapFilling());
6                  processor.addOperator(new ReductionPaa(30));
7                  processor.addOperator(new NormalizeData());
8                  processor.addOperator(new Kmeans(3,"
                       euclidean");
9                  processor.execute();
10         }
11 }
```

Percebe-se pelo código mostrado que, uma vez implementados os operadores, a plataforma consegue reproduzir um fluxo de processamento de forma simples, bastando que se tenha um objeto *Processor* (Linha 3) depois adicionar os operadores à lista de operadores (linhas 4 a 8) e invocar o método *execute* que é responsável por executar os operadores na ordem em que foram adicionados. Percebe-se que não é necessário passar a série temporal como parâmetro para cada operador, isso porque em sua implementação cada operador já tem acesso ao operando *DoubleOperandImpl* que é responsável por manter um único objeto que representa a lista de séries temporais para todos os operadores.

A seguir são apresentados os resultados da execução do *kmeans* para as

Torres Santo Antônio e UFMT respectivamente. Os testes foram feitos utilizando 3 clusters e a função de distância euclidiana para os dois conjuntos de dados. Os dados utilizados foram, Temperatura do Ar, Umidade Relativa do Ar e Radiação Solar.

4.3.1.2 Torre Santo Antônio

Para a execução do *kmeans* com os dados da Torre Santo Antônio foram utilizados dados de Janeiro de 2005 a Dezembro de 2009. A Tabela 4 mostra o resultado do agrupamento dos dados encontrado na execução do *kmeans*

Tabela 4: Resultado do agrupamento encontrado pelo *kmeans* para os dados da Torre Santo Antônio.

Cluster 1		Cluster 2	Cluster 3
Janeiro de 2005	Abril de 2009	Abril de 2008	Abril de 2007
Janeiro de 2006	Setembro de 2006	Maio de 2006	Maio de 2005
Janeiro de 2007	Outubro de 2005	Maio de 2007	Junho de 2005
Janeiro de 2008	Outubro de 2006	Maio de 2008	Junho de 2006
Janeiro de 2009	Outubro de 2007	Maio de 2009	Julho de 2005
Fevereiro de 2005	Outubro de 2008	Junho de 2007	Julho de 2006
Fevereiro de 2006	Outubro de 2009	Junho de 2008	Julho de 2008
Fevereiro de 2007	Novembro de 2005	Junho de 2009	Agosto de 2005
Fevereiro de 2008	Novembro de 2006	Julho de 2007	Agosto de 2006
Fevereiro de 2009	Novembro de 2007	Julho de 2009	Agosto de 2007
Março de 2005	Novembro de 2009		Agosto de 2008
Março de 2006	Dezembro de 2005		Agosto de 2009
Março de 2007	Dezembro de 2006		Setembro de 2005
Março de 2008	Dezembro de 2007		Setembro de 2007
Março de 2009	Dezembro de 2008		Setembro de 2008
Abril de 2005	Dezembro de 2009		Setembro de 2009
Abril de 2006			Novembro de 2008

Percebe-se na divisão dos dados encontrada pelo *kmeans* que os dados foram agrupados seguindo a divisão que a literatura já reporta sobre o estado de Mato Grosso, o qual diz que tem-se no estado duas estações, uma chuvosa e outra seca (RODRIGUES et al., 2013). E existe um período entre as duas estações que é chamada de período de transição. Para verificar, foi plotado o gráfico da Precipitação Acumulada média para os meses de cada cluster, Figura 38.

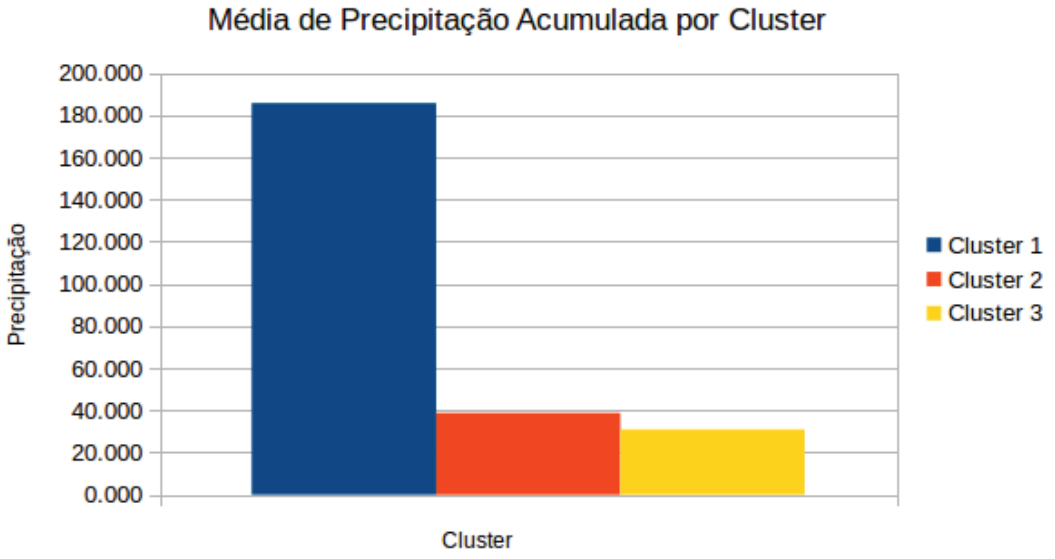


Figura 38: Média da Precipitação acumulada para cada cluster nos dados da Torre Santo Antônio.

O Cluster 1 apresenta a média de precipitação acumulada de 185,71 mm/m^2, os clusters 2 e 3 apresentam valores de 38,74 mm/m^2 e 30,92 mm/m^2 respectivamente. As números mostram claramente a divisão entre dois períodos muito diferentes, o cluster 1 com grande quantidade de chuva e o 3 caracteriza-se por pouca chuva. O cluster 2 apresenta muitos dos meses caracterizados como sendo do período de transição entre os dois períodos (RODRIGUES et al., 2011).

4.3.1.3 Torre UFMT

Para o teste do *kmeans* com os dados da Torre UFMT foram utilizados dados de Janeiro de 2011 até Setembro de 2014. A Tabela 5 mostra o resultado do agrupamento dos dados encontrado na execução do *kmeans*

Nesse caso percebe-se também a clara divisão entre os meses considerados como sendo do período chuvoso, seco e intermediário. A Figura 41 apresenta os dados da precipitação acumulada média para os meses de cada cluster para a Torre UFMT.

Tabela 5: Resultado do agrupamento encontrado pelo *kemans* para os dados da Torre UFMT.

Cluster 1	Custer 2	Cluster 3
Agosto de 2011	Janeiro de 2012	Maio de 2011
Agosto de 2012	Janeiro de 2013	Maio de 2012
Agosto de 2014	Fevereiro de 2012	Maio de 2013
Setembro de 2011	Fevereiro de 2013	Maio de 2014
Setembro de 2012	Fevereiro de 2014	Junho de 2011
Setembro de 2014	Março de 2012	Junho de 2012
Outubro de 2011	Março de 2013	Junho de 2014
Outubro de 2012	Março de 2014	Julho de 2011
Outubro de 2013	Abril de 2012	Julho de 2012
Novembro de 2011	Abril de 2013	Julho de 2013
	Abril de 2014	Julho de 2014
	Junho de 2013	
	Novembro de 2012	
	Dezembro de 2011	
	Dezembro de 2012	

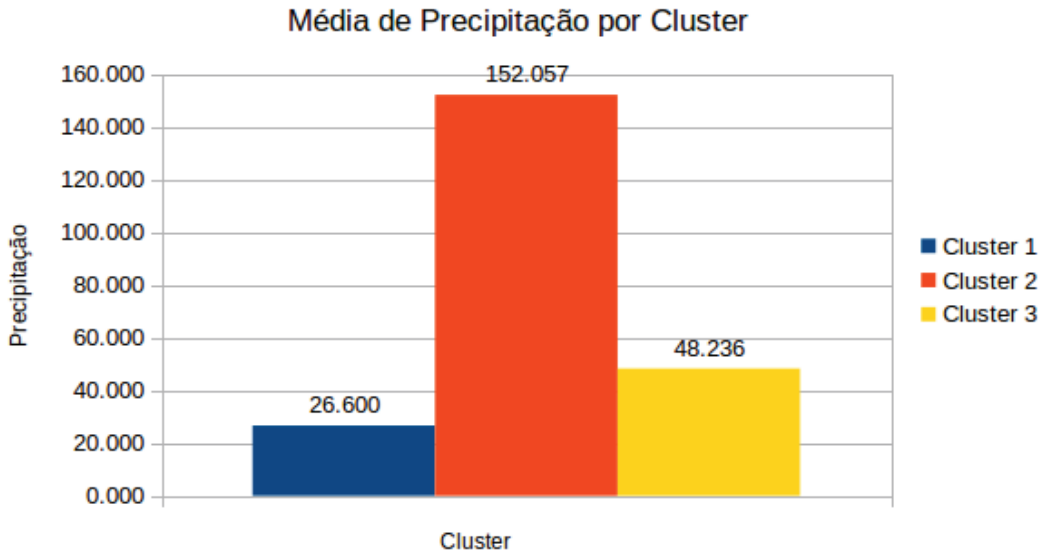


Figura 39: Média da Precipitação acumulada para cada cluster nos dados da Torre PgFA.

Nos dados da Torre UFMT fica mais evidente a divisão entre os períodos,

com diferenças maiores entre cada cluster. Nesse caso o primeiro cluster representa os meses do período de seca com valor médio de precipitação acumulada de 26,5 mm/m^2 o cluster 3 representa os meses do período de transição com valor igual a 48,23 mm/m^2 e o cluster 2 representa o período chuvoso com valor de 152,05 mm/m^2 de média de precipitação acumulada.

Observando o resultado do cluster percebe-se por exemplo que o mês de Novembro de 2011 foi alocado no cluster de meses secos e Novembro de 2012 no cluster de meses chuvosos. Observando os dados de precipitação para cada um desses meses tem-se que em Novembro de 2011 a precipitação acumulada foi de 106 mm/m^2 e Novembro de 2012 216,60 mm/m^2 evidenciando que os meses foram muito diferentes entre si, e apesar de a média de precipitação acumulada do cluster 1 ser muito menor que o mês de Novembro de 2011, este se difere muito dos meses do período chuvoso.

4.3.2 Busca por Similaridade

Para o teste da busca por similaridade foram utilizados apenas dados de temperatura. Foi selecionada a manhã mais fria do conjunto de dados disponível da Torre UFMT. Para realizar esse teste foi implementado o método $getAllForenoon()$ da Classe $MetheorologicalTimeSeries$ considerando o período da manhã como sendo de 5 a 12 horas. Os dados da Torre UFMT foram coletados a cada 15 minutos, foi feito então a média horária para esse teste, ou seja, representação dos dados utilizando o método PAA com segmento de tamanho 4 medidas. A manhã selecionada foi a do dia 24 de Julho de 2013 cuja temperatura média horária mínima foi de $9,33$ °C. Foi utilizado o método *k nearest neighbors* para a busca com $k = 4$, ou seja, as 4 manhãs mais similares com a selecionada. Foi utilizada a função DTW para realizar a comparação entre as séries. A Figura 40 apresenta os gráficos comparando cada manhã selecionada (linha preta) pelo algoritmo com a manha do dia 24 de Julho de 2013 (linha azul)

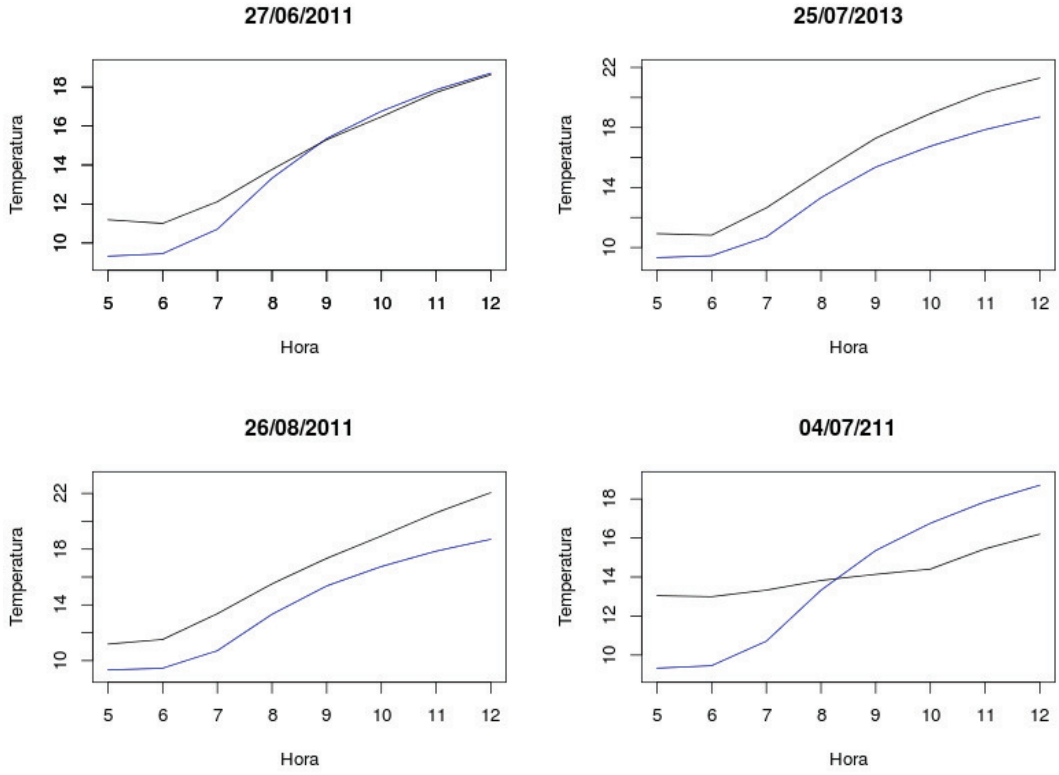


Figura 40: 4 manhãs mais similares à manha do dia 24 de julho de 2013.

Todas as 4 manhãs selecionadas foram do próprio ano de 2013 ou de 2011. A primeira manhã foi a do dia 27 de Junho de 2011 com temperatura horária mínima de 11°C e máxima de 18,75°C. A segunda manhã selecionada foi a do dia 25 de Julho de 2013, um dia após o dia selecionado para busca, com temperatura horária mínima de 10,82°C e máxima de 18,62°C. A terceira manhã foi a do dia 26 de Agosto de 2011 com temperatura horária mínima de 11,18 °C e máxima de 22,05°C. A quarta manhã selecionada foi a do dia 4 de Julho de 2011 com temperatura horária mínima de 12,99°C e máxima de 16,20°C.

4.3.3 Detecção de Padrões Desconhecidos

O teste de descoberta de padrões desconhecidos tem como objetivo encontrar padrões de comportamento na série sem que que se tenha conhecimento prévio dos padrões, a única informação que é passada ao algoritmo é o tamanho do padrão que se busca. Foram utilizados os dados da Torre UFMT para o teste. Foram utilizados dados de média diária dos dados de temperatura.

O algoritmo utilizado foi o proposto por Castro e Azevedo (2010). Foi feita a busca por padrões de tamanho 10 dias, 20 dias e 30 dias, em todos os

casos utilizou-se como passo no algoritmo de janela deslizante 50% do tamanho da janela. Os teste com 20 e 30 dias apresentaram 1 padrão que se repetiram apenas 3 vezes em ambos os casos, na busca por padrões de 10 dias foram retornados 4 padrões com 3 ou 4 repetições. A Figura

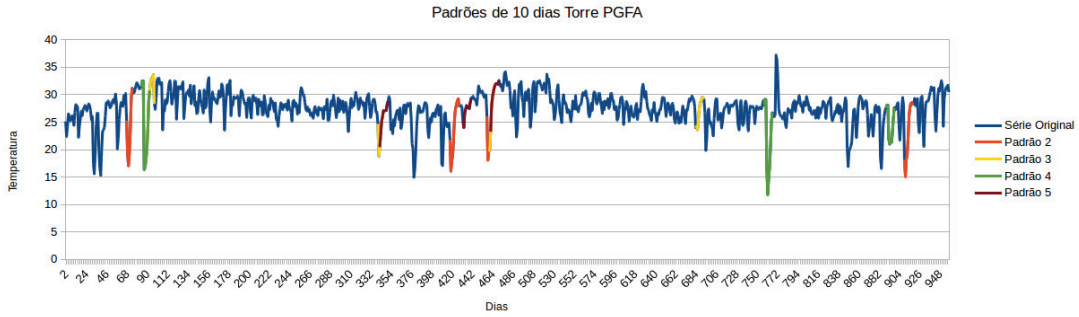


Figura 41: Padrões encontrados nos dados de média diária de temperatura da Torre UFMT.

Os padrões são destacados na figura por cores diferentes, nota-se que alguns padrões sobrepõe outros, isso acontece porque o algoritmo percorre toda a série para cada padrão, isso pode ser evitado utilizando, por exemplo, algum tipo de flag na série temporal. Percebe-se que todos os padrões encontrados foram detectados em dias que contém temperaturas baixas para o padrão da região, o que evidencia que para este local os eventos entre queda de temperatura e retorno à temperatura normal para a região podem durar até 10 dias.

É importante destacar que esses testes foram feitos com médias diárias, a busca por padrões é fortemente influenciada pela granularidade dos dados, por exemplo, se fossem utilizados dados de médias horárias os padrões encontrados seriam diferentes.

A busca por padrões pode ser útil para planejamento na área ambiental e agronômica por exemplo, áreas cujo conhecimento do comportamento dos dados auxilia na tomada de decisões.

Capítulo 5

Conclusões

Este trabalho apresenta o desenvolvimento da MiMi - Plataforma Computacional para Mineração de Dados Micrometeorológicos. A plataforma tem como objetivo facilitar o desenvolvimento e validação de algoritmos de mineração de dados aplicados a séries de dados micrometeorológicos bem como automatizar a execução do fluxo de processamento que deve ser executado nas tarefas de mineração de dados.

A plataforma MiMi inclui um novo conceito em relação às arquiteturas de software para mineração de dados, que é o tratamento das séries temporais como um novo tipo de dado e um novo componente que atua entre as camadas da arquitetura, que é o processador, responsável por gerar a sequência de processamento dos algoritmos.

A manipulação de um conjunto de séries temporais como sendo um operando e os algoritmos que manipulam as séries temporais como operadores possibilita a padronização no desenvolvimento dos algoritmos e facilita o acesso às séries por esses algoritmos por meio dos operadores base definidos.

A definição da Classe Interface *Operator* permitiu que o fluxo de processamento pudesse ser executado pela classe *Processor* que é responsável por armazenar uma sequência de operadores e executá-los na ordem em que foram adicionados à lista de operadores. Essa estratégia permite que aplicações possam ser construídas utilizando a plataforma e que esse fluxo seja facilmente alterado por meio de uma interface com o usuário.

A definição dos operadores permite ainda que os operadores sejam tratados como componentes de software, permitindo que novos algoritmos sejam incluídos de forma simples com poucas alterações para acessar o operando *TimeSerieOperand*.

Os testes realizados puderam validar a plataforma mostrando que a cons-

trução do fluxo de processamento acontece de forma simples utilizando a classe *Processor*. Por meio dos testes informações importantes ainda puderam ser extraídas. O agrupamento realizado nos dois conjuntos de dados utilizando dados de Temperatura, Umidade Relativa do Ar e Radiação Solar que agrupou os dados mensais de acordo com o que já revela a literatura a respeito dos locais aonde os dados foram coletados, dividindo em 3 períodos distintos, um de muita chuva, outro de seca e outro chamado de período intermediário ou de transição. O teste de busca por similaridade foi executado para encontrar as 4 manhãs mais similares à manha do dia 24 de Julho de 2013. O teste de detecção de padrões desconhecidos revelou que eventos de quedas bruscas e retorno à normalidade nos dados de temperatura na região denominada Torre PGFA duram 10 dias.

As atividades envolvendo os dados meteorológicos podem se beneficiar da plataforma proposta uma vez que as definições estabelecidas por um especialista de domínio pode ser reaproveitada, poupando tempo de calibração dos modelos. Por meio das expressões de domínio as soluções encontradas podem ainda ser integradas, facilitando o uso dos algoritmos por novos usuários.

5.1 Contribuições

Estas foram as principais contribuições geradas no desenvolvimento deste trabalho:

- O uso das expressões de domínio permitem que o conhecimento do especialista seja embutido no processo de mineração de dados em suas diversas etapas.
- A possibilidade de encapsular o conhecimento do especialista de domínio por meio das expressões de domínio permite que o especialista concentre esforços em partes específicas do processo com a reutilização de expressões de domínio já parametrizadas.
- A expressões de domínio permitem a reutilização de soluções já encontradas por outros especialistas.
- A definição dos operadores permite a integração das etapas do processo de mineração de dados.
- A construção da plataforma MiMi em forma de uma API permite que novas aplicações que necessitem ser desenvolvidas se beneficiem de uma pla-

taforma extensível e flexível que fornece mecanismos para manipulação dos dados e integração de operadores.

5.2 Publicações

1. OLIVEIRA, A. G. de, VENTURA, T., GANCHEV, T. D., FIGUEIREDO, J. M. de, JAHN, O., MARQUES, M. I., & K.-L. SCHUCHMANN. 2014. Bird Acoustic Activity Detection Based on Morphological Filtering of the Spectrogram. Applied Acoustics, Elsevier.

2. OLIVEIRA, A. G., FIGUEIREDO, J. M., NOGUEIRA, M. C. J. A. SISTEMA PARA MINERAÇÃO DE DADOS EM SÉRIES DE DADOS MICROMETEOROLÓGICOS In: VI Mostra de Pós-Graduação, 2014, Cuiabá. 2014. v.1.

3. DE SOUZA, R. R. G.S, FIGUEIREDO, J. M., MARTINS, C. A., OLIVEIRA, A. G., DE SOUZA, J. S. A framework for automating the configuration of OpenCL. Environmental Modelling & Software. , v.53, p.81 - 86, 2014.

4. VENTURA, T., OLIVEIRA, A. G. de, GANCHEV, T. D., FIGUEIREDO, J. M. de, JAHN, O., MARQUES, M. I., & K.-L. SCHUCHMANN. 2014. Audio Parameterization for Improved Bird Identification. Expert Systems with Applications, Elsevier (Em avaliação)

5. CURADO, L. F. A., RODRIGUES, T. R., OLIVEIRA, A. G., NOVAIS, J. W. Z., PAULO, I. J. C., BIUDES, M. S., NOGUEIRA, J. S. ANALYSIS OF THERMAL CONDUCTIVITY IN A SEASONAL FLOODED FOREST IN THE NORTHERN PANTANAL. Revista Brasileira de Meteorologia (Impresso). , v.28, p.125 - 128, 2013.

6. NOVAIS, J. W. Z., RODRIGUES, T. R., de OLIVEIRA, A. G., CURADO, L. F. A, PAULO,S.R, NOGUEIRA, J.S, OLIVEIRA, R. G. Geothermal Dynamics in Vochysia Divergens Forest in a Brazilian Wetland. Air, Soil and Water Research. , v.6, p.47 - 52, 2013.

7. VENTURA, T. M., de 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.

8. RODRIGUES, T. R., CURADO, L. F. A., NOVAIS, J. W. Z., de OLIVEIRA, A. G., NOVAIS, J. W. Z., PAULO, S. R., BIUDES, M. S., NOGUEIRA, J. S. Distribuição Sazonal da Energia Disponível no Norte do Pantanal. Revista de Ciências Agro-Ambientais (Online). , v.9, p.165 - 175, 2012.

9. NOVAIS, J. W. Z., RODRIGUES, T. R., CURADO, L. F. A., de OLIVEIRA, A. G., PAULO, S. R., NOGUEIRA, J. S. Variabilidade sazonal horária das propriedades térmicas em gleissolo háplico no norte do pantanal. Semina. Ciências Agrárias (Impresso). , v.33, p.2563 - 2570, 2012.

10. CURADO, L. F. A., RODRIGUES, T. R., NOVAIS, J. W. Z., de OLIVEIRA, A. G., VENTURA, T. M., DE MUSIS, C. R, NOGUEIRA, J. S. Adjustment of the Brunt‘s equation parameters for the Northern Brazilian Pantanal. Journal of Ecology and The Natural Environment (JENE). , v.3, p.157 - 162, 2011.

11. RODRGUES, T.R, CURADO,L.F.A, de OLIVEIRA, A. G., NOVAIS, J. W. Z., PAULO,S.R, BIUDES, Marcelo Sacardi, NOGUEIRA, J.S Distribuição dos componentes do balanço de energia do Pantanal Mato-grossense. Revista de Ciências Agro-Ambientais (Online). , v.9, n.2, p.165 - 175, 2011.

5.3 Trabalhos Futuros

A plataforma MiMi atingiu os objetivos propostos neste trabalho, e seu funcionamento abre novas perspectivas na área de mineração de dados meteorológicos. Nessas novas perspectivas, as principais melhorias e pesquisas a serem desenvolvidas envolvem:

- Implementação de novos operadores de pré-processamento de dados e mineração para que seja possível realizar mais testes e expandir a possibilidade de análises.

- Implementação de técnicas de visualização de dados para que o usuário possa visualizar na plataforma MiMi os resultados da mineração.

- Desenvolvimento de uma interface gráfica web para que os usuários possam utilizar os algoritmos implementados e visualizar os resultados, dessa forma outros especialistas podem utilizar a plataforma e contribuir para o desenvolvimento.

- Desenvolvimento de um editor de expressões de domínio na interface gráfica para que seja possível o usuário especialista de domínio manipular e utilizar as expressões existentes, bem como armazenar novas expressões.
- Disponibilização dos operadores por meio de serviços web para que seja possível integração por outras ferramentas, bem como a possibilidade de consumo de serviços em forma de operadores.
- Integração com ferramentas já existentes como R, MatLab e Weka. Isso é necessário porque essas ferramentas já fornecem um conjunto de algoritmos de mineração de dados com eficiência já comprovada.
- Validar a plataforma em outros contextos que envolvam dados temporais.

REFERÊNCIAS

AGRAWAL, R.; FALOUTSOS, C.; SWAMI, A. N. Efficient similarity search in sequence databases. In: *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*. London, UK, UK: Springer-Verlag, 1993. (FODO '93), p. 69–84. ISBN 3-540-57301-1. Disponível em: <http://dl.acm.org/citation.cfm?id=645415.652239>. Citado na página 32.

ALCâNTARA, A.; E., M. A. D.; H., P. A. Cenários prospectivos, monitoração ambiental e metadados. *Revista de Ciência da Informação*, 2010. Citado 4 vezes nas páginas I, 7, 8 e 16.

ALENCAR, A. B. *Mineração e Visualização de Coleções de Séries Temporais*. Dissertação de Mestrado, 2007. Citado 2 vezes nas páginas 25 e 32.

ANAND, S. S. S.; BüCHNER, A. G. *Decision Support Using Data Mining*. [S.l.: s.n.], 1998. 168 p. Citado na página 10.

ANTUNES, C. M.; OLIVEIRA, A. L. Temporal data mining: An overview. In: *KDD Workshop on Temporal Data Mining*. [S.l.: s.n.], 2001. p. 1–13. Citado na página 6.

ASTROM, K. On the choice of sampling rates in parametric identification of time series. *Information Sciences*, v. 1, n. 3, p. 273 – 278, 1969. ISSN 0020-0255. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025569800137>. Citado na página 18.

BAGNALL, A.; JANACEK, G. Clustering time series with clipped data. *Machine Learning*, Kluwer Academic Publishers, v. 58, n. 2-3, p. 151–178, 2005. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1007/s10994-005-5825-6>. Citado na página 20.

BAGNALL, A.; RATANAMAHATANA, C.; KEOGH, E.; LONARDI, S.; JANACEK, G. A bit level representation for time series data mining with shape based similarity. *Data Mining and Knowledge Discovery*, Springer US, v. 13, n. 1, p. 11–40, 2006. ISSN 1384-5810. Disponível em: <http://dx.doi.org/10.1007/s10618-005-0028-0>. Citado na página 20.

BIBECK, M.; DIAMOND, J.; DUCKETT, J.; GUDMUNDSSON, O. G.; KOBAL, P.; LENZ, E.; LIVINGSTON, S.; MARCUS, D.; MOHR, S.; OZU, N.; PINNOCK, J.; VISCO, K.; WATT, A.; WILLIAMS, K.; ZAEV, Z. *Professional XML*. [S.l.]: Wrox Press Ltd, 2001. Citado na página 41.

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*, p. 65–77, 2009. Citado na página 14.

BIUDES, M. S.; VOURLITIS, G. L.; MACHADO, N. G.; ARRUDA, P. H. Z. de; NEVES, G. A. R.; LOBO, F. de A.; NEALE, C. M. U.; NOGUEIRA, J. de S. Patterns of energy exchange for tropical ecosystems across a climate gradient in mato grosso, brazil. *Agricultural and Forest Meteorology*, v. 202, n. 0, p. 112 – 124, 2015. ISSN 0168-1923. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0168192314003189>. Citado na página 40.

BOECKING, B.; CHALUP, S. K.; SEESE, D.; WONG, A. S. Support vector clustering of time series data with alignment kernels. *Pattern Recognition Letters*, v. 45, n. 0, p. 129 – 135, 2014. ISSN 0167-8655. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167865514000956>. Citado na página 24.

BURKOM, H. S.; MURPHY, S. P.; SHMUELI, G. Automated time series forecasting for biosurveillance. *Statistics in Medicine*, v. 26, n. 22, p. 4202–4218., 2007. Citado na página 6.

CABENA, P.; HADJINIAN, P.; STADLER, R.; VERHEES, J.; ZANASI, A. *Discovering Data Mining: From Concept to Implementation*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1998. ISBN 0-13-743980-6. Citado na página 10.

CANEPPELE L. B. ; NOGUEIRA, M. C. J. A. AvaliaÇÃo de desenpenho tÉrmico e eficiÊncia energÉtica de habitaÇÃo em condomÍnio residencial de cuiabÁ/ mt. *Revista Monografias Ambientais,*, v. 14, 2014. Citado na página 7.

CASTRO, N.; AZEVEDO, P. Multiresolution Motif Discovery in Time Series. In: SIAM. *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, 2010, Columbus, Ohio, USA*. [S.l.], 2010. p. 665–676. Citado 2 vezes nas páginas 45 e 64.

CHAN, P. K.; MAHONEY, M. V. Modeling multiple time series for anomaly detection. In: *Proceedings of the Fifth IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2005. (ICDM '05), p. 90–97. ISBN 0-7695-2278-5. Disponível em: <http://dx.doi.org/10.1109/ICDM.2005.101>. Citado na página 23.

CHANG, L.-C.; SHEN, H.-Y.; CHANG, F.-J. Regional flood inundation nowcast using hybrid {SOM} and dynamic neural networks. *Journal of Hydrology*, v. 519, Part A, n. 0, p. 476 – 489, 2014. ISSN 0022-1694. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022169414005538>. Citado na página 24.

CHAPMAN, P.; CLINTON J.; KERBER, R.; KHABAZA, T.; REINARTZ, T.; SHEARER, C.; WIRTH, R. Relatório Técnico, *CRISPDM 1.0 step-by-step data mining guide*. 2000. Citado 5 vezes nas páginas I, II, 12, 39 e 40.

CHIBANA, E. Y.; FLUMIGNAN, D.; MOTA R. G.; VIEIRA, A. d. S. F. R. T. Estimativa de falhas em dados meteoroloógicos. In: . [S.l.: s.n.], 2005. Citado 2 vezes nas páginas 17 e 18.

CHIU, B.; KEOGH, E.; LONARDI, S. Probabilistic discovery of time series motifs. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2003. (KDD '03), p. 493–498. ISBN 1-58113-737-0. Disponível em: <http://doi.acm.org/10.1145/956750.956808>. Citado na página 29.

CHUNG F.L., F. T. L. R. N. V. Flexible time series pattern matching based on perceptually important points. In: *International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Learning from Temporal and Spatial Data*. [S.l.: s.n.], 2001. Citado na página 21.

CIOS, K.; TERESINSKA, A.; KONIECZNA, S.; POTOCKA, J.; SHARMA, S. Diagnosing myocardial perfusion from pect bull's-eye maps—a knowledge discovery approach. *IEEE Engineering in Medicine and Biology Magazine, Special issue on Medical Data Mining and Knowledge Discovery*, v. 4, n. 19, p. 17–25, 2000. Citado na página 11.

CURADO, L. F. A.; NOGUEIRA, J. S.; SANCHES, L.; RODRIGUES, T. R.; LOBO, F. A.; BíUDES, M. S. Inter seasonality of the energy fluxes in brazilian savana—mato grosso—brazil. *Atmospheric and Climate Sciences*, v. 4, p. 219–230, 2014. Citado na página 6.

CURADO, L. F. A.; RODRIGUES, T. R.; NOVAIS, J.; OLIVEIRA, A. de; VENTURA, T.; MUSIS, C. D.; NOGUEIRA, J. . Adjustment of the brunt's equation parameters for the northern brazilian pantanal. *Journal of Ecology and The Natural Environment (JENE)*, v. 3, p. 157–162, 2011. Citado na página 6.

CôRTES, S. C.; PORCARO, R. M.; LIFSCHITS, S. *Mineração de Dados – Funcionalidades, Técnicas e Abordagens*. [S.l.], 2002. Citado 2 vezes nas páginas 10 e 37.

DALMAGRO, H.; LOBO, F.; VOURLITIS, G.; DALMOLIN, A.; ANTUNES M.Z., J.; ORTíZ, C.; NOGUEIRA, J. The physiological light response of two tree species across a hydrologic gradient in brazilian savanna (cerrado). *Photosynthetica*, The Institute of Experimental Biology of the Czech Academy of Sciences, v. 52, n. 1, p. 22–35, 2014. ISSN 0300-3604. Citado na página 7.

DALMOLIN, A. C.; LOBO, F. de A.; VOURLITIS, G.; SILVA, P. R.; DALMAGRO, H. J.; ANTUNES MARIO ZORTéA, J.; ORTíZ, C. E. R. Is the dry season an important driver of phenology and growth for two brazilian savanna tree species with contrasting leaf habits? *Plant Ecology*, Springer Netherlands, v. 216, n. 3, p. 407–417, 2015. ISSN 1385-0237. Citado na página 7.

DAMLE, C.; YALCIN, A. Flood prediction using time series data mining. *Journal of Hydrology*, v. 333, n. 2 - 4, p. 305 – 316, 2007. ISSN 0022-1694. Citado na página 7.

DIAS, C. A. A. *Procedimentos de Medição e Aquisição de Dados de uma Torre Micrometeorológica em Sinop-MT*. Dissertação (Mestrado em Fisica Ambiental), 2007. Citado na página 17.

DOUGLAS, D. H.; PEUCKER, T. K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, UT Press, v. 10, n. 2, p. 112–122, 1973. Citado na página 22.

DOUZAL-CHOUAKRIA, A.; AMBLARD, C. Classification trees for time series. *Pattern Recognition*, v. 45, n. 3, p. 1076 – 1091, 2012. ISSN 0031-3203. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320311003578>. Citado na página 15.

DUAN, J.; WANG, W.; LIU, B.; XUE, Y.; ZHOU, H.; SHI, B. Incorporating with recursive model training in time series clustering. In: *Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on*. [S.l.: s.n.], 2005. p. 105–109. Citado na página 24.

ESLING, P.; AGON, C. Time-series data mining. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 45, n. 1, p. 12:1–12:34, dez. 2012. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/2379776.2379788>. Citado 5 vezes nas páginas I, 6, 14, 17 e 27.

ESLING, P.; AGON, C. Time-series data mining. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 45, n. 1, p. 12:1–12:34, dez. 2012. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/2379776.2379788>. Citado na página 26.

FALOUTSOS, C.; RANGANATHAN, M.; MANOLOPOULOS, Y. Fast subsequence matching in time-series databases. *SIGMOD Rec.*, ACM, New York, NY, USA, v. 23, n. 2, p. 419–429, maio 1994. ISSN 0163-5808. Disponível em: <http://doi.acm.org/10.1145/191843.191925>. Citado na página 15.

FAYYAD, U. M. Mining Databases: Towards Algorithms for Knowledge Discovery. *IEEE Data Engineering Bulletin*, v. 21, p. 39–48, 1998. Citado 2 vezes nas páginas I e 10.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P. Advances in knowledge discovery and data mining. In: FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P.; UTHURUSAMY, R. (Ed.). Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996. cap. From Data Mining to Knowledge Discovery: An Overview, p. 1–34. ISBN 0-262-56097-6. Disponível em: <http://dl.acm.org/citation.cfm?id=257938.257942>. Citado 2 vezes nas páginas 9 e 10.

FERRARI, G. T.; OZAKI, V. Missing data imputation of climate datasets: implications to modeling extreme drought events. *Revista Brasileira de Meteorologia*, 2014. Citado 2 vezes nas páginas 17 e 18.

FINK, E.; PRATT, K.; GANDHI, H. Indexing of time series by major minima and maxima. In: *Systems, Man and Cybernetics, 2003. IEEE International Conference on*. [S.l.: s.n.], 2003. v. 3, p. 2332–2335 vol.3. ISSN 1062-922X. Citado na página 23.

FREEMAN, E.; ROBSON, E.; BATES, B.; SIERRA, K. *Head first design patterns*. [S.l.]: "O'Reilly Media, Inc.", 2004. Citado na página 41.

FU, A. W.-C.; KEOGH, E.; LAU, L. Y.; RATANAMAHATANA, C. A.; WONG, R. C.-W. Scaling and time warping in time series querying. *The VLDB Journal—The International Journal on Very Large Data Bases*, Springer-Verlag New York, Inc., v. 17, n. 4, p. 899–921, 2008. Citado na página 35.

FU, T.; CHUNG, F.; LUK, R.; NG, C. Financial time series indexing based on low resolution clustering. In: CITESEER. *FU, T. C. et al. Financial time series indexing based on low resolution clustering. In: 4th IEEE Intl Conference on Data Mining (ICDM 2004) Workshop on Temporal Data Mining: Algorithms, Theory and Applications*. [S.l.], 2004. p. 5–14. Citado na página 21.

FU, T.; CHUNG, F.; LUK R. ANS NG, V. Pattern discovery from stock time series using self-organizing maps. In: *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Workshop on Temporal Data Mining*. [S.l.: s.n.], 2001. p. 27–37. Citado na página 23.

FU, T.-c.; CHUNG, F.-l.; LUK, R.; NG, C.-m. Representing financial time series based on data point importance. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 21, n. 2, p. 277–300, 2008. Citado 2 vezes nas páginas I e 21.

FU, T. chung. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, v. 24, n. 1, p. 164 – 181, 2011. ISSN 0952-1976. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0952197610001727>. Citado 5 vezes nas páginas 14, 15, 17, 23 e 32.

GHASSEMPOUR, S.; GIROSI, F.; MAEDER, A. Clustering multivariate time series using hidden markov models. *International Journal of Environmental Research and Public Health*, v. 3, n. 11, p. 2741–2763, 2014. ISSN 1660-4601. Disponível em: <http://www.mdpi.com/1660-4601/11/3/2741/htmsthash.HfoB3MEC.dpuf>. Citado na página 24.

GONçALVES, A. L.; PACHECO, R. C. S.; MORALES, A. B. T. Utilização de técnicas de mineração de dados em bases de c t: uma análise dos grupos de pesquisa no brasil. In: *XXI ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO – ENEGEP, Salvador–Bahia*. Rio de Janeiro–RJ: Anais do XXI Encontro Nacional de Engenharia de Produção – XXXI ENEGEP, 2001. Citado na página 9.

GUO, X.; LIANG, X.; LI, N. Automatically recognizing stock patterns using rpcl neural networks. In: *Proceedings of the 2007 International Conference on*

Intelligent Systems and Knowledge Engineering,. [S.l.: s.n.], 2007. p. 997–1004. Citado na página 24.

HAIR, J. F.; BLACK, W. C.; BABIN, B. J.; ANDERSON, R. E. *Multivariate Data Analysis*. 7. ed. [S.l.]: Prentice Hall, 2009. Citado na página 9.

HAN, J.; KAMBER, M.; PEI, J. *Data mining : concepts and techniques*. 3. ed. [S.l.]: ELSEVIER, 2011. ISBN 9780123814791. Citado na página 9.

HARVEY, D. Y.; WORDEN, K.; TODD, M. D. Robust evaluation of time series classification algorithms for structural health monitoring. In: *Proc. SPIE 9064, Health Monitoring of Structural and Biological Systems*. [s.n.], 2014. p. 90640K–90640K–8. Disponível em: <http://dx.doi.org/10.1117/12.2044790>. Citado na página 15.

HERSHBERGER, J.; SNOEYINK, J. Speeding up the douglas-peucker line-simplification algorithm. In: *Proc. 5th Intl. Symp. on Spatial Data Handling*. [S.l.: s.n.], 1992. p. 134–143. Citado na página 22.

HUI, D.; WAN, S.; SU, B.; KATUL, G.; MONSON, R.; LUO, Y. Gap-filling missing data in eddy covariance measurements using multiple imputation (mi) for annual estimations. *Agricultural and Forest Meteorology*, v. 121, n. 1–2, p. 93 – 111, 2004. ISSN 0168-1923. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0168192303001588>. Citado na página 14.

IDé, T. Why does subsequence time-series clustering produce sine waves? In: FüRNKRANZ, J.; SCHEFFER, T.; SPILIOPOULOU, M. (Ed.). *Knowledge Discovery in Databases: PKDD 2006*. Springer Berlin Heidelberg, 2006, (Lecture Notes in Computer Science, v. 4213). p. 211–222. ISBN 978-3-540-45374-1. Disponível em: <http://dx.doi.org/10.1007/11871637_23>. Citado na página 26.

JAIN, A. K. Data clustering: 50 years beyond k-means. *Pattern Recogn. Lett.*, Elsevier Science Inc., New York, NY, USA, v. 31, n. 8, p. 651–666, jun. 2010. ISSN 0167-8655. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2009.09.011>. Citado na página 25.

JANSEN, M. *Noise Reduction by Wavelet Thresholdings*. [S.l.]: Springer, 2001. Citado na página 13.

JEONG, Y.-S.; JEONG, M. K.; OMITAOMU, O. A. Weighted dynamic time warping for time series classification. *Pattern Recognition*, v. 44, n. 9, p. 2231 – 2240, 2011. ISSN 0031-3203. Computer Analysis of Images and Patterns. Disponível em: <http://www.sciencedirect.com/science/article/pii/S003132031000484X>. Citado na página 15.

KDNUGGETS.COM. *Data Mining Methodology*. 2002. Disponível em: <http://www.kdnuggets.com/polls/2002/methodology.htm>. Citado na página 11.

KDNUGGETS.COM. *Data Mining Methodology*. 2004. Disponível em: <hhttp://www.kdnuggets.com/polls/2004/data_mining_methodology.htm>. Citado na página 11.

KDNUGGETS.COM. *Data Mining Methodology*. 2007. Disponível em: <http://www.kdnuggets.com/polls/2007/data_mining_methodology.htm>. Citado na página 11.

KEOGH, E. A fast and robust method for pattern matching in time series databases. In: *Proceedings of the Ninth IEEE International Conferenceon Tools with Artificial Intelligence*. [S.l.: s.n.], 1997. p. 145–150. Citado na página 23.

KEOGH, E. Fast similarity search in the presence of longitudinal scaling in time series databases. In: *Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on*. [S.l.: s.n.], 1997. p. 578–584. ISSN 1082-3409. Citado na página 23.

KEOGH, E. Exact indexing of dynamic time warping. In: *Proceedings of the 28th International Conference on Very Large Data Bases*. VLDB Endowment, 2002. (VLDB '02), p. 406–417. Disponível em: <http://dl.acm.org/citation.cfm?id=1287369.1287405>. Citado 5 vezes nas páginas II, 11, 15, 35 e 36.

KEOGH, E.; CHAKRABARTI, K.; PAZZANI, M.; MEHROTRA, S. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, Springer-Verlag London Limited, v. 3, n. 3, p. 263–286, 2001. ISSN 0219-1377. Disponível em: <$http://dx.doi.org/10.1007/PL00011669$>. Citado na página 18.

KEOGH, E.; LIN, J. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems*, Springer, v. 8, n. 2, p. 154–177, 2005. Citado na página 15.

KEOGH, E.; LIN, J. Clustering of time-series subsequences is meaningless: Implications for previous and future research. *Knowl. Inf. Syst.*, Springer-Verlag New York, Inc., New York, NY, USA, v. 8, n. 2, p. 154–177, ago. 2005. ISSN 0219-1377. Disponível em: <http://dx.doi.org/10.1007/s10115-004-0172-7>. Citado na página 26.

KEOGH, E.; LIN, J.; FU, A. Hot sax: Efficiently finding the most unusual time series subsequence. In: *Proceedings of the Fifth IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2005b. (ICDM '05), p. 226–233. ISBN 0-7695-2278-5. Disponível em: <http://dx.doi.org/10.1109/ICDM.2005.79>. Citado na página 23.

KEOGH, E.; LIN, J.; LEE, S.-H.; HERLE, H. V. Finding the most unusual time series subsequence: Algorithms and applications. *Knowl. Inf. Syst.*, Springer-Verlag New York, Inc., New York, NY, USA, v. 11, n. 1, p. 1–27, dez. 2006. ISSN 0219-1377. Disponível em: <http://dx.doi.org/10.1007/s10115-006-0034-6>. Citado na página 23.

KEOGH, E.; LIN, J.; TRUPPEL, W. Clustering of time series subsequences is meaningless: Implications for previous and future research. In: *Proceedings of the Third IEEE International Conference on Data Mining*. Washington, DC,

USA: IEEE Computer Society, 2003. (ICDM '03), p. 115–. ISBN 0-7695-1978-4. Disponível em: <http://dl.acm.org/citation.cfm?id=951949.952156>. Citado na página 26.

KEOGH, E.; LONARDI, S.; CHIU, B. Y.-c. Finding surprising patterns in a time series database in linear time and space. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2002b. (KDD '02), p. 550–556. ISBN 1-58113-567-X. Disponível em: <http://doi.acm.org/10.1145/775047.775128>. Citado 2 vezes nas páginas 23 e 24.

KEOGH, E.; PAZZANI, M. A simple dimensionality reduction technique for fast similarity search in large time series databases. In: TERANO, T.; LIU, H.; CHEN, A. (Ed.). *Knowledge Discovery and Data Mining. Current Issues and New Applications*. Springer Berlin Heidelberg, 2000, (Lecture Notes in Computer Science, v. 1805). p. 122–133. ISBN 978-3-540-67382-8. Disponível em: <$http://dx.doi.org/10.1007/3-540-45571-X_14$>. Citado na página 18.

KIM, K. jae. Financial time series forecasting using support vector machines. *Neurocomputing*, v. 55, n. 1–2, p. 307 – 319, 2003. ISSN 0925-2312. Support Vector Machines. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925231203003722>. Citado na página 24.

KLOSGEN, W.; ZYTKOW, J. Advances in knowledge discovery and data mining. In: FAYYAD U, P.-S. G. S. P.; UTHURUSAMY, R. (Ed.). [S.l.]: AAAI Press, 1996. cap. Knowledge discovery in databases terminology, p. 573–592. Citado na página 9.

KOHONEN, T. (Ed.). *Self-organizing Maps*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997. ISBN 3-540-62017-6. Citado na página 24.

KOZIEVITH, N. P. *Dados Meteorológicos: um estudo de viabilidade utilizando um SGBD em plataforma de baixo custo*. Dissertação de Mestrado, 2006. Citado na página 7.

KRISLOCK, N.; WOLKOWICZ, H. Euclidean distance matrices and applications. In: ANJOS, M. F.; LASSERRE, J. B. (Ed.). *Handbook on Semidefinite, Conic and Polynomial Optimization*. Springer US, 2012, (International Series in Operations Research and Management Science, v. 166). p. 879–914. ISBN 978-1-4614-0768-3. Disponível em: <http://dx.doi.org/10.1007/978-1-4614-0769-0_30>. Citado na página 34.

KURGAN, L. A.; MUSILEK, P. A survey of knowledge discovery and data mining process models. *The Knowledge Engineering Review*, v. 21, p. 1–24, 3 2006. ISSN 1469-8005. Disponível em: <http://journals.cambridge.org/article_S0269888906000737>. Citado 5 vezes nas páginas IV, 9, 10, 11 e 12.

LEE, S.; KWON, D.; LEE, S. Dimensionality reduction for indexing time series based on the minimum distance. *J. Inf. Sci. Eng.*, p. 697–711, 2003. Citado 6 vezes nas páginas II, 19, 20, 27, 28 e 29.

LI, S.-T.; KUO, S.-C. Knowledge discovery in financial investment for forecasting and trading strategy through wavelet-based {SOM} networks. *Expert Systems with Applications*, v. 34, n. 2, p. 935 – 951, 2008. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417406003551>. Citado na página 24.

LIN, J.; WILLIAMSON, S.; BORNE, K.; DEBARR, D. Pattern recognition in time series. In: _____. [S.l.]: Advances in Machine Learning and Data Mining for Astronomy, 2012. cap. 1. Citado na página 5.

MA, J.; PERKINS, S. Online novelty detection on temporal sequences. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2003. (KDD '03), p. 613–618. ISBN 1-58113-737-0. Disponível em: <http://doi.acm.org/10.1145/956750.956828>. Citado na página 24.

MALETZKE, A. G. *Uma Metodologia para Extração de Conhecimento em Séries Temporais por meio da Identificação de Motifs e Extração de Características*. Dissertação de Mestrado, 2009. Citado 4 vezes nas páginas II, 27, 30 e 31.

MAN, P. W. P.; WONG, M. H. Efficient and robust feature extraction and pattern matching of time series by a lattice structure. In: *Proceedings of the Tenth International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2001. (CIKM '01), p. 271–278. ISBN 1-58113-436-3. Disponível em: <http://doi.acm.org/10.1145/502585.502631>. Citado na página 23.

MARBáN, O.; MARISCAL, G.; SEGOVIA, J. Data mining and knowledge discovery in real life applications. In: PONCE, J.; KARAHOCA, A. (Ed.). [S.l.]: InTech, 2009. cap. A Data Mining and Knowledge Discovery Process Model. Citado 3 vezes nas páginas 10, 11 e 38.

MARISCAL, G.; MARBÁN, O.; FERNANDÉZ, C. A survey of data mining and knowledge discovery process models and methodologies. *The Knowledge Engineering Review*, v. 25, n. 2, p. 137–166, 2010. Citado na página 11.

MATHWORKS. *MatLab*. 2014. Disponível em: <http://www.mathworks.com/products/matlab/>. Citado na página 37.

MATOS, V. A. T.; PIVETA, F.; SOBRINHO, S. P.; TISSIANI, A. S. O.; PEREIRA, A. P. M. S.; RAMOS F. T. ; CAMPELO JúNIOR, J. H. Temperaturas basais e exigência térmica para a maturação do caju. *Bioscience Journal*, v. 30, 2014. Citado na página 7.

MISHRA, S.; DWIVEDI, V. K.; SARAVANAN, C.; PATHAK, K. K. Pattern discovery in hydrological time series data mining during the monsoon period of the high flood years in brahmaputra river basin. *International Journal of Computer Applications*, v. 67, n. 6, p. 0975 – 8887, 2013. Citado na página 6.

MOERCHEN, F. Algorithms for time series knowledge mining. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2006. (KDD '06), p. 668–673. ISBN 1-59593-339-5. Disponível em: <http://doi.acm.org/10.1145/1150402.1150485>. Citado na página 14.

MöRCHEN, F.; ULTSCH, A.; HOOS, O. Extracting interpretable muscle activation patterns with time series knowledge mining. *Int. J. Know.-Based Intell. Eng. Syst.*, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 9, n. 3, p. 197–208, ago. 2005. ISSN 1327-2314. Disponível em: <http://dl.acm.org/citation.cfm?id=1233864.1233868>. Citado na página 24.

NETBEANS. *NetBeans IDE 8.0*. 2014. Disponível em: <https://netbeans.org/>. Citado na página 41.

NOVAIS, J. W. Z.; RODRIGUES, T. R.; CURADO, L. F. A.; OLIVEIRA, A. G.; PAULO, S. R. de; NOGUEIRA, J. de S.; OLIVEIRA, R. G. de. Geothermal dynamics in vochysia divergens forest in a brazilian wetland. *Air, Soil and Water Research*, Libertas Academica, v. 6, p. 47–52, 03 2013. Disponível em: <www.la-press.com/geothermal-dynamics-in-vochysia-divergens-forest-in-a-brazilian-wetlan-article-a3603>. Citado na página 6.

OLIVEIRA, L. F. C. d.; FIOREZI, 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. Citado na página 17.

OOBA, M.; HIRANO, T.; MOGAMI, J.-I.; HIRATA, R.; FUJINUMA, Y. Comparisons of gap-filling methods for carbon flux dataset: A combination of a genetic algorithm and an artificial neural network. *Ecological Modelling*, v. 198, n. 3–4, p. 473 – 486, 2006. ISSN 0304-3800. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0304380006002766>. Citado 2 vezes nas páginas 13 e 18.

ORACLE. *Java*. 2014. Disponível em: <https://www.java.com/>. Citado na página 41.

OUYANG, R.; REN, L.; CHENG, W.; ZHOU, C. Similarity search and pattern discovery in hydrological time series data mining. *Hydrological Processes*, John Wiley e Sons, Ltd., v. 24, n. 9, p. 1198–1210, 2010. ISSN 1099-1085. Disponível em: <http://dx.doi.org/10.1002/hyp.7583>. Citado na página 6.

PALIT, A.; POPOVIC, D. *Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications*. Xxii. [S.l.]: Springer, 2005. 372 p. (Advances in Industrial Control). 978-1-84628-184-6. Citado na página 5.

PARK, Y.; PRIEBE, C.; YOUSSEF, A. Anomaly detection in time series of graphs using fusion of graph invariants. *Selected Topics in Signal Processing,*

IEEE Journal of, v. 7, n. 1, p. 67–75, Feb 2013. ISSN 1932-4553. Citado na página 15.

PATEL, P.; KEOGH, E.; LIN, J.; LONARDI, S. Mining motifs in massive time series databases. In: *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. [S.l.: s.n.], 2002. p. 370–377. Citado 2 vezes nas páginas 26 e 45.

PERNG, C.-S.; WANG, H.; ZHANG, S.; PARKER, D. Landmarks: a new model for similarity-based pattern querying in time series databases. In: *Data Engineering, 2000. Proceedings. 16th International Conference on*. [S.l.: s.n.], 2000. p. 33–42. ISSN 1063-6382. Citado na página 22.

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. Citado 2 vezes nas páginas 17 e 18.

PRATT, B.; FINK, E. Search for patterns in compressed time series. *International Journal of Image and Graphics*, v. 1, n. 2, p. 89–106, 2002. Citado na página 23.

PRESSMAN, R. *Software Engineering: A Practitioner's Approach*. [S.l.: s.n.], 2005. Citado na página 10.

RATANAMAHATANA, C.; KEOGH, E.; BAGNALL, A.; LONARDI, S. A novel bit level time series representation with implication of similarity search and clustering. In: HO, T.; CHEUNG, D.; LIU, H. (Ed.). *Advances in Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3518). p. 771–777. ISBN 978-3-540-26076-9. Disponível em: <http://dx.doi.org/10.1007/11430919_90>. Citado 2 vezes nas páginas I e 20.

RODRIGUES, T. R.; CURADO, L. F. A.; ; NOVAIS, J.; OLIVEIRA, A. de; PAULO, S. R. D.; BIDES, M. S.; NOGUEIRA, J. . DistribuiÇÃo dos componentes do balanÇo de energia do pantanal mato-grossense. *Revista de Ciências Agro-Ambientais*, v. 9, n. 2, p. 165 – 175, 2011. Citado na página 61.

RODRIGUES, T. R.; PAULO, S. R. de; NOVAIS, J. W. Z.; CURADO, L. F. A.; NOGUEIRA, J. S.; OLIVEIRA, R. G. de; LOBO, F. A.; VOURLITIS, G. L. Temporal patterns of energy balance for a brazilian tropical savanna under contrasting seasonal conditions. *International Journal of Atmospheric Sciences*, v. 2013, p. 9, 2013. Citado 2 vezes nas páginas 6 e 60.

SAMSUDIN, R.; SAAD, P.; SHABRI, A. River flow time series using least squares support vector machines. *Hydroly and Earth System Sciences*, v. 15, p. 1835 – 1852, 2011. Citado na página 24.

SANCHES, A. R. *Redução de Dimensionalidade em Séries Temporais*. Dissertação de Mestrado, 2006. Citado 2 vezes nas páginas I e 22.

SANTOS F. M. M. ; OLIVEIRA, A. . N. M. C. J. A. . D. M. C. R. . N. J. S. AnÁlise do clima urbano de cuiabÁ-mt-brasil por meio de transectos mÓveis. *Revista Monografias Ambientais,*, v. 114, 2014. Citado na página 7.

SAPANKEVYCH, N.; SANKAR, R. Time series prediction using support vector machines: A survey. *Computational Intelligence Magazine, IEEE*, v. 4, n. 2, p. 24–38, May 2009. ISSN 1556-603X. Citado na página 24.

SHEARER, C. The CRISP-DM Model: The new blueprint for data mining. *Journal of Data Warehousing*, v. 5, n. 4, p. 13–22, 2000. Citado 3 vezes nas páginas 6, 11 e 12.

SILVA, S. T. *Reconstrução da dinâmica não linear da temperatura do ar em Cuiabá-MT*. Tese de Doutorado (Doutorado em Fisica Ambiental), 2015. Citado na página 7.

SIPES, T.; KARIMABADI, H.; JIANG, S.; MOORE, K.; LI, N.; BARR, J. Anomaly detection in time series radiotherapy treatment data. In: *Semantic Computing (ICSC), 2014 IEEE International Conference on*. [S.l.: s.n.], 2014. p. 324–329. Citado na página 15.

SKOPAL, T.; BUSTOS, B. On nonmetric similarity search problems in complex domains. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 43, n. 4, p. 34:1–34:50, out. 2011. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/1978802.1978813>. Citado 2 vezes nas páginas 33 e 34.

SMYTH, P.; ; SMYTH, P.; KEOGH, E. *Clustering and Mode Classification of Engineering Time Series Data*. 1997. 24-30 p. Citado na página 23.

SONG, H.; LI, G. Tourism demand modelling and forecasting—a review of recent research. *Tourism Management*, v. 29, n. 2, p. 203 – 220, 2008. ISSN 0261-5177. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0261517707001707>. Citado na página 6.

SOUZA A. P. ; SILVA, A. C. . L. S. . T. A. A. . S. M. E. Evapotranspiração e eficiência do uso da água no primeiro ciclo produtivo da figueira roxo de valinhos submetida a cobertura morta. *Bioscience Journal*, v. 30, 2014. Citado na página 7.

SOUZA, P. R. F.; JUNIOR, O. B. P.; JúNIOR, J. H. C. Evapotranspiração do algodoeiro estimada pelo método do balanço de energia e pelo método de penman-monteith. *Bioscience Journal*, v. 33, 2011. Citado na página 7.

SPSS. *IBM SPSS*. 2014. Disponível em: <http://www-01.ibm.com/software/analytics/spss/>. Citado na página 37.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321321367. Citado 2 vezes nas páginas 9 e 25.

TATSCH, J.; ROCHA, H. D.; CABRAL, O.; FREITAS H.AND 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*, p. 09–112, 2007. Citado na página 14.

TEAM, R. D. C. *R: A language and environment for statistical computing. R Foundation for Statistical Computing*. 2014. Disponível em: <http://www.r-project.org/>. Citado na página 37.

TRAINA, C.; FIGUEIREDO, J. M.; TRAINA, A. J. M. Image domain formalization for content-based image retrieval. In: *Proceedings of the 2005 ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2005. (SAC '05), p. 604–609. ISBN 1-58113-964-0. Disponível em: <http://doi.acm.org/10.1145/1066677.1066818>. Citado 4 vezes nas páginas 41, 46, 48 e 49.

VENTURA, T. M. *Criação de um Ambiente Computacional para Detecção de Outliers e Preenchimento de Falhas em Dados Meteorológicos*. Tese (Doutorado em Fisica Ambiental), 2015. Citado na página 17.

VENTURA, T. M.; MACHADO N. G. AND; MARTINS, A. L.; JúNIOR, J. H. C.; LOBO, F. A.; ORTIZ, C. E. R.; MARTINS, C. A. . Temperaturas basais para o crescimento de frutos de mangueira alfa. *Bioscience Journal*, v. 30, 2014. Citado 2 vezes nas páginas 7 e 40.

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*, n. 27, 2013. Citado 4 vezes nas páginas 13, 17, 18 e 43.

VOURLITIS, G.; NOGUEIRA, J. de S.; LOBO, F. de A.; PINTO OSVALDO-BORGES, J. Variations in evapotranspiration and climate for an amazonian semi-deciduous forest over seasonal, annual, and el niño cycles. *International Journal of Biometeorology*, Springer Berlin Heidelberg, v. 59, n. 2, p. 217–230, 2015. ISSN 0020-7128. Citado na página 7.

WANDERLEY, H. S.; AMORIM, R. F. C. d.; CARVALHO, F. O. d. Variabilidade espacial e preenchimento de falhas de dados pluviométricos para o estado de alagoas. *Revista Brasileira de Meteorologia*, 2012. Citado na página 18.

WEI, L.; KUMAR, N.; LOLLA, V.; KEOGH, E. J.; LONARDI, S.; RATANAMAHATANA, C. Assumption-free anomaly detection in time series. In: *Proceedings of the 17th International Conference on Scientific and Statistical Database Management*. Berkeley, CA, US: Lawrence Berkeley Laboratory, 2005. (SSDBM'2005), p. 237–240. ISBN 1-88888-111-X. Disponível em: <http://dl.acm.org/citation.cfm?id=1116877.1116907>. Citado na página 23.

WEISS, G. M. Mining with rarity: A unifying framework. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 6, n. 1, p. 7–19, jun. 2004. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/1007730.1007734>. Citado 2 vezes nas páginas 15 e 37.

WIRTH, R. Crisp-dm: Towards a standard process model for data mining. In: *Proceedings of the Fourth International Conference on the Practical Application*

of Knowledge Discovery and Data Mining. [S.l.: s.n.], 2000. p. 29–39. Citado 3 vezes nas páginas 6, 11 e 12.

WOOLDRIDGE, J. M. *Introductory Econometrics: a Modern Approach*. [S.l.]: South–Western College Publishing, a division of Thomson Learning, 2000. Citado na página 5.

WU, H.; SALZBERG, B.; SHARP, G. C.; JIANG, S. B.; SHIRATO, H.; KAELI, D. Subsequence matching on structured time series data. In: *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. [S.l.: s.n.], 2005. Citado na página 15.

XI, X.; KEOGH, E.; SHELTON, C.; WEI, L.; RATANAMAHATANA, C. A. Fast time series classification using numerosity reduction. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York, NY, USA: ACM, 2006. (ICML '06), p. 1033–1040. ISBN 1-59593-383-2. Disponível em: <http://doi.acm.org/10.1145/1143844.1143974>. Citado na página 26.

XING, Z.; PEI, J.; YU, P. S.; WANG, K. Extracting interpretable features for early classification on time series. In: _____. *Proceedings of the 2011 SIAM International Conference on Data Mining*. [S.l.: s.n.], 2011. cap. 22. Citado na página 15.

YANKOV, D.; KEOGH, E.; MEDINA, J.; CHIU, B.; ZORDAN, V. Detecting time series motifs under uniform scaling. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2007. (KDD '07), p. 844–853. ISBN 978-1-59593-609-7. Disponível em: <http://doi.acm.org/10.1145/1281192.1281282>. Citado na página 27.

YIN, J.; YANG, Q. Integrating hidden markov models and spectral analysis for sensory time series clustering. In: *Data Mining, Fifth IEEE International Conference on*. [S.l.: s.n.], 2005. p. 8 pp.–. ISSN 1550-4786. Citado na página 24.

ZHANG, X.; LIU, J.; DU, Y.; LV, T. A novel clustering method on time series data. *Expert Systems with Applications*, v. 38, n. 9, p. 11891 – 11900, 2011. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417411004908>. Citado na página 15.

ZHAO, G.; DENG, W. An hmm-based hierarchical clustering method for gene expression time series data. In: *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on*. [S.l.: s.n.], 2010. p. 219–222. Citado na página 24.

ZHONG, S.; KHOSHGOFTAAR, T. M.; SELIYA, N. Clustering-based network intrusion detection. *International Journal of Reliability, Quality and Safety Engineering*, v. 14, n. 2, p. 169 – 187, 2007. Citado na página 6.