Erick Giovani Sperandio Nascimento

Um Algoritmo Baseado em Técnicas de Agrupamento para Detecção de Anomalias em Séries Temporais Utilizando a Distância de Mahalanobis

Vitória – ES, Brasil

27 de julho de 2012

Dissertação de Mestrado sob o título "Um Algoritmo Baseado em Técnicas de Agrupamento para Detecção de Anomalias em Séries Temporais Utilizando a Distância de Mahalanobis", defendida por Erick Giovani Sperandio Nascimento e aprovada em _____, em Vitória, estado do Espírito Santo, Brasil, pela banca examinadora constituída pelos professores:

Prof. D.Sc. Orivaldo de Lira Tavares - UFES
Orientador

Prof. Ph.D. Alberto Ferreira De Souza - UFES
Co-orientador

Prof. D.Sc. Crediné Silva de Menezes - UFRGS
Examinador Interno

Prof. D.Sc. Ricardo Shitsuka - UNIFEI
Examinador Externo

AGRADECIMENTOS

Agradeço primeiramente a Deus, minha esposa Quezia, meu pai Adiel e meus irmãos Thiago e Felipe, que incodicionalmente me apoiaram neste trajeto até aqui. Aos meus orientadores Prof. D.Sc. Orivaldo de Lira Tavares e Prof. Ph.D. Alberto Ferreira De Souza, que foram fundamentais no desenvolvimento deste trabalho. Sou grato à FAPES (Fundação de Amparo à Pesquisa do Espírito Santo), que financiou parcialmente este trabalho. Meu obrigado também aos nossos colaboradores, que nos deram acesso aos dados reais que foram utilizados na avaliação do algoritmo e na produção dos resultados, que são: o Instituto Ambiental do Estado do Espírito Santo, que forneceram os dados que estão relacionados aos casos I e II, e a ArcelorMittal Tubarão, localizada no município de Serra - ES, que gentilmente apoiou a nossa pesquisa fornecendo as séries temporais relacionadas com os casos III e IV. Além disso, agradecemos ao Sr. Eammon Keogh que nos deu acesso ao arquivo de dados UCR Time Series Data Mining Archive [Keogh, 2011].

RESUMO

Este trabalho propõe um algoritmo para a detecção de anomalias em séries temporais, baseado em técnicas de agrupamento, utilizando a função de distância de Mahalanobis. Após uma revisão das principais e mais recentes contribuições feitas neste campo de pesquisa, uma descrição formal e detalhada do algoritmo é apresentada, seguida por uma discussão sobre como configurar seus parâmetros. A fim de avaliar sua efetividade, ele foi aplicado a casos reais, e seus resultados foram comparados com outra técnica aplicável ao mesmo problema. Os resultados obtidos sugerem que esta proposta pode ser aplicada com sucesso na detecção de anomalias em séries temporais.

ABSTRACT

This work proposes an algorithm for anomaly detection in time series data, based on clustering techniques, using the Mahalanobis distance function. After a brief review of the main and recent contributions made in this research field, a formal and detailed description of the algorithm is presented, followed by a discussion on how to set its parameters. In order to evaluate its effectiveness, it was applied to real cases, and its results were compared with another techniques proposed to the same problem. The obtained results suggest that this proposal can be successfully applied to detect anomaly in time series.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

UCR – University of California at Riverside;

C-AMDATS – Cluster-based Algorithm using Mahalanobis distance for Detection of Anomalies in Time Series;

ECG – Eletrocardiograma;

SAX – Symbolic Aggregate approXimation;

DTW – Dynamic Time Warping;

DFT – Discrete Fourier Transform;

DWT – Discrete Wavelet Transform;

SVD – Single Value Decomposition;

PLA – Piecewise Linear Approximation;

LSI – Latent Semantic Indexing;

PAA – Piecewise Aggregate Approximation;

APCA – Adaptative Piecewise Constant Approximation;

TSA-Tree – Trend and Surprise Abstractions – Tree;

PCAD – Periodic Curve Anomaly Detection;

RS – Random Swap;

ODAC – Online Divisive Agglomerative Clustering;

1 INTRODUÇÃO

Hoje em dia, em muitos domínios, tais como de processos industriais, monitoramento meteorológico, cardiologia ou mercados de ações, há a geração de séries temporais de dados sequenciais relevantes continuamente. Em geral, esses dados são coletados e armazenados por equipamentos específicos e, posteriormente, são analisados e mantidos por pessoas especializadas a fim de verificar a capacidade dos dados em representar com exatidão o estado real dos processos.

Em muitas situações, é crítico o processo de identificar padrões incomuns que poderiam ser gerados por um comportamento inesperado. Tal comportamento indesejável pode ser devido a qualquer problema que o processo relacionado possa estar experimentando. Por exemplo, uma indústria pode monitorar algumas variáveis de seu processo produtivo atual para diagnosticar os gargalos, as violações dos requisitos de qualidade, violação de requisitos ambientais, tais como a emissão de um poluente específico para o ambiente acima do permitido por lei, ou qualquer outra situação que poderia ser inadequada à indústria. Outro exemplo é um determinado instituto ou agência ambiental que monitora alguns parâmetros atmosféricos, a fim de avaliar a qualidade do ar de uma área urbana, sujeita a falhas nos equipamentos de monitoramento que, se não forem detectadas a tempo, podem gerar dados falsos e assim levar os especialistas a uma má interpretação desses parâmetros atmosféricos. Ainda outro exemplo: uma empresa de cartão de crédito pode monitorar cada transação do usuário a fim de procurar comportamentos incomuns que poderiam apontar para operações fraudulentas.

Esses comportamentos anormais, indesejados, são muitas vezes chamados como comportamentos anormais ou anômalos, e podem ser identificados por indução a partir

dos dados, devido a uma variedade de razões que podem ser descobertas por especialistas ao analisar os dados do respectivo processo. É importante que essa análise leve em conta quaisquer mudanças no comportamento do parâmetro monitorado, a fim de identificar oportunidades de melhorar, prevenir ou corrigir a causa da anomalia.

1.1 Objetivos

O objetivo central desta dissertação é apresentar a proposta e implementação de um algoritmo baseado em técnicas de agrupamento utilizando a distância Mahalanobis, como sua função de distância, visando o problema de detecção de anomalias em séries temporais, aqui chamado de *C-AMDATS*, acrônimo de ***C****luster-based* ***A****lgorithm using* ***M****ahalanobis distance for* ***D****etection of* ***A****nomalies in* ***T****ime* ***S****eries*.

Outros objetivos são:

i. Avaliar os principais e mais recentes trabalhos correlatos que visam à detecção de anomalias em séries temporais;
ii. Apresentar uma descrição detalhada de um novo algoritmo para a detecção de anomalias em séries temporais, evidenciando seu grau de inovação em comparação com os trabalhos correlatos;
iii. Avaliar a eficácia do algoritmo, aplicando-o a casos reais e comparando-o com classificações feitas por especialistas humanos com outra técnica já aplicada com sucesso na detecção de anomalias em séries temporais.

1.2 Metodologia

Para a avaliação das técnicas correlatas, foi feita uma extensa pesquisa sobre os principais trabalhos publicados mais recentes que usam métodos de agrupamento de dados com o objetivo de detectar anomalias em séries temporais. Para avaliação da

eficácia do algoritmo, foram comparados os dados reais com as classificações feitas previamente por especialistas e foi analisada a matriz de confusão de cada caso, conforme está descrito mais detalhadamente na **Seção 4.3**.

Para avaliar se o algoritmo proposto aqui é uma inovação na área de detecção de anomalias em séries temporais foi feita uma revisão dos trabalhos correlatos mais recentes e destacados na área, notadamente os trabalhos que utilizam uma abordagem de agrupamento de dados. Para avaliar sua eficácia, foi necessário encontrar e selecionar dados temporais que tivessem anotações sobre anomalias em suas séries, feitas por especialistas humanos, de modo a comparar com o algoritmo *C-AMDATS* e assim avaliar a capacidade desse algoritmo em detectar as mesmas anomalias. Também foi selecionada uma técnica já publicada e aplicada anteriormente nessa área de pesquisa, dentre as que foram avaliadas na revisão dos trabalhos correlatos, com o fim de se comparar seus resultados com os encontrados pelo algoritmo *C-AMDATS*.

1.2.1 Motivação e Processo de Criação do Algoritmo

Inicialmente, após contato com especialistas na área de avaliação e tratamento de dados ambientais, verificou-se que havia um problema comum que necessitava de uma solução. Em vários momentos era necessário analisar as séries temporais dos dados coletados por equipamentos de monitoramento ambiental e de processos industriais a fim de identificar situações em que houve algum tipo de comportamento inesperado ou anômalo. Tais comportamentos, em geral, podem ser causados por diversos motivos, dentre eles: mau funcionamento de equipamento, manutenção mal executada, falta de energia etc. Independente da causa, a sua identificação é fundamental a fim de garantir que o processo de monitoramento se mantenha adequado na representação do estado real dos dados amostrados. Motivado por este contexto. Iniciou-se um processo de

pesquisa e avaliação das técnicas atualmente disponíveis para detecção de anomalias em séries temporais. Após extensa avaliação, feita primeiramente em entrevistas com especialistas tanto do setor público quanto do setor privado, e posteriormente através da análise e testes de métodos estatísticos e algoritmos outrora já desenvolvidos para este fim, verificou-se haver ainda uma lacuna nessa área que necessitava ser preenchida.

Foi durante o curso de graduação e posteriormente neste curso de mestrado que foi elaborada a base do algoritmo. Após vários testes, percebeu-se que seria possível aplicar o algoritmo *k-means* de uma forma diferenciada em uma série temporal individual. Essa forma envolvia, dentre outros aspectos, a utilização da distância de Mahalanobis, dadas as suas interessantes propriedades, como será mais especificamente descrito no **Capítulo 3**, associada a uma maneira de extrair dos grupos calculados os padrões anômalos, o algoritmo foi por fim concebido,

A fim de avaliar sua eficácia, o algoritmo precisava ser submetido a testes utilizando dados reais. Para tanto, os colaboradores forneceram tais dados, já avaliados, analisados e classificados anteriormente por especialistas nas áreas afins, que foram utilizados na avaliação do algoritmo. Os resultados da execução, dos testes e a avaliação dos resultados estão presentes no **Capítulo 4**.

1.3 Estrutura da dissertação

Este trabalho está organizado da seguinte forma: o **Capítulo 2** apresenta uma revisão dos conceitos e da pesquisa recente sobre os trabalhos que visam à detecção de anomalias em séries temporais. O **Capítulo 3** apresenta os fundamentos do algoritmo, com uma descrição detalhada e formal. O **Capítulo 4** apresenta cinco casos reais com padrões anômalos que foram avaliados, a fim de avaliar a capacidade da abordagem *C-AMDATS* em detectar estas anomalias, em conjunto com uma comparação com outra

técnica que objetiva o mesmo problema, isto é, a detecção de anomalias em séries temporais. O **Capítulo 5** apresenta uma conclusão e recomendações para trabalhos futuros. Esta dissertação é concluída com as referências bibliográficas.

2 TRABALHOS RELACIONADOS

Vários trabalhos têm sido desenvolvidos para identificar padrões em séries temporais, e alguns deles foram especializados para detectar padrões anormais nessas séries. Algumas técnicas foram desenhadas para trabalhar com dados de domínios como: na identificação de fraudes em operações bancárias e de crédito, ou na identificação de problemas cardiológicos através da análise de eletrocardiograma (ECG). Outras técnicas foram projetadas para detectar anomalias em domínios diversos. Far-se-á uma breve revisão das pesquisas mais recentes que visam detectar anomalias em séries temporais, a fim de identificar se a técnica proposta nesta dissertação introduz uma nova contribuição para a comunidade científica.

Algumas dessas obras foram extraídas a partir de [Kavitha, 2010], que traz um levantamento bibliográfico sobre agrupamento de séries temporais. Foram escolhidas algumas das pesquisas mais recentes sobre a detecção de anomalias em séries temporais.

2.1 Técnicas Baseadas em Distância e *Outliers*

Como afirmado por [Rebbapragada, 2009], alguns algoritmos foram projetados para usar uma técnica baseada em distância para a detecção de *outliers*, ou valores atípicos. Um *outlier* é uma amostra que é numericamente distante do restante das demais amostras [Barnet, 1994]. Uma abordagem estatística é a ideia de encontrar pontos que estão mais distantes do que, pelo menos, p por cento do conjunto de dados, utilizando uma distância especificada D [Knorr, 1998]. Outra variante desta abordagem é a análise da distância de um ponto ao seu k-ésimo vizinho mais próximo [Ramaswamy, 2000], ou a soma das distâncias aos seus k vizinhos mais próximos [Angiulli, 2002]. Outro

trabalho visa encontrar cada exemplo de *k* vizinhos mais próximos no que diz respeito a uma amostra aleatória, em vez de analisar o conjunto de dados inteiro [Wu, 2006].

A definição de *outliers* baseada em distância não permite a correta identificação dos valores atípicos em dados de variância mista, que é um problema comum quando se lida com dados de séries temporais. Buscando resolver esse aspecto, o trabalho em [Breunig, 2000] introduziu o fator de *outlier* local (LOF) como uma solução. Em vez de definir um limite rígido *D*, LOF considera a densidade de um exemplo em comparação com a densidade da vizinhança para calcular o quanto aquele objeto é atípico em relação ao restante da série, conforme ilustra a **Figura 2.1**. Outros trabalhos baseados em densidade, nesta mesma área, visam melhorar a eficiência computacional da LOF, através da sumarização dos dados [Jin, 2001] e da sua indexação [Ren, 2000].

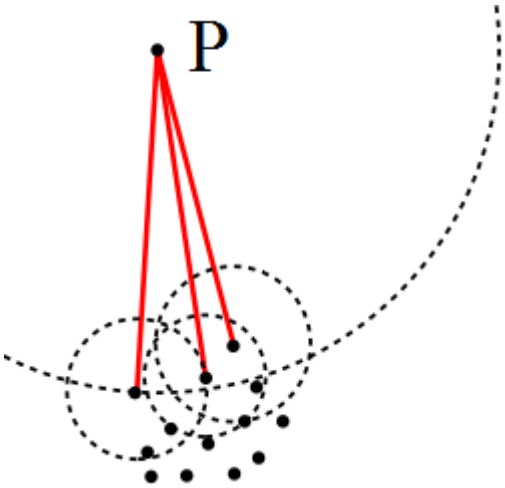

Figura 2.1. Ilustração da ideia básica de LOF, onde a densidade de uma amostra P é comparada com a densidade de seus vizinhos mais próximos. P possui uma densidade bem menor de que seus vizinhos

2.2 Técnicas Baseadas em Janelas Deslizantes e de Discretização

A meta da detecção de anomalias em uma única série temporal é encontrar uma sub-região anômala. Em alguns casos, uma única série temporal é convertida em um conjunto de dados compostos por subséries temporais através da utilização de janelas deslizantes, de maneira incremental, como nos trabalhos de [Dasgupta, 1996], [Keogh, 2002], [Ma, 2003], [Wei, 2005] e [Keogh, 2005], ou em passos discretos de acordo com um período conhecido, como nos trabalhos de [Yang, 2001] e [Yang, 2004].

Os métodos apresentados em [Dasgupta, 1996], [Keogh, 2002] e [Keogh, 2005] utilizam janelas deslizantes e discretização, em uma única série temporal contínua, e definem o que é uma anomalia no que diz respeito a uma série temporal de referência. Eles convertem tanto a série temporal de entrada quanto a de referência em duas bases de dados de séries temporais, deslizando uma janela de comprimento especificado pelo usuário e convertendo os valores na janela em uma cadeia de valores discretos – por exemplo, uma cadeia de caracteres, ou uma *string*. Especificamente em [Keogh, 2005], os autores apresentam uma técnica, denominada *HOT SAX – Symbolic Aggregate approXimation* – que aborda a detecção de anomalias usando o conceito de divergências em séries temporais – ou *time series discords* – aplicando-a a casos reais. A abordagem geral para encontrar uma subsérie temporal divergente – ou divergência – em uma dada série temporal de referência T começa por empregar um algoritmo em toda a série, de modo a transformá-la de valores contínuos para valores discretos. Essa discretização será melhor entendida mais à frente no texto. Em seguida, T é dividida em j janelas, ou subsequências de T, w_1, w_2,..., w_j, de modo que $j = |T| + m-1$, por uma janela deslizante de tamanho m, iterativamente. Em seguida, para cada janela w_i é calculada a chance de ela ser uma subsequência anômala na série T, o que é feito calculando-se sua distância das demais janelas. Quanto mais distante for uma janela das demais, menos similar ela

será e maiores serão suas chances de ser uma anomalia. As janelas cuja dissimilaridade for maior do que todas as outras serão tratadas como subsequências divergentes, e, portanto, anomalias. Neste processo de avaliar a similaridade, é utilizada uma determinada função de distância, tais como a distância Euclidiana, de Manhatan ou a *Dynamic Time Warping* (DTW), que geralmente são utilizadas nesses tipos de técnicas. Em se tratando especificamente de *HOT SAX*, a função de distância utilizada para o cálculo da similaridade entre as janelas deslizantes é a distância Euclidiana. A **Figura 2.2** mostra um esquema gráfico ilustrando o funcionamento da técnica *HOT SAX*.

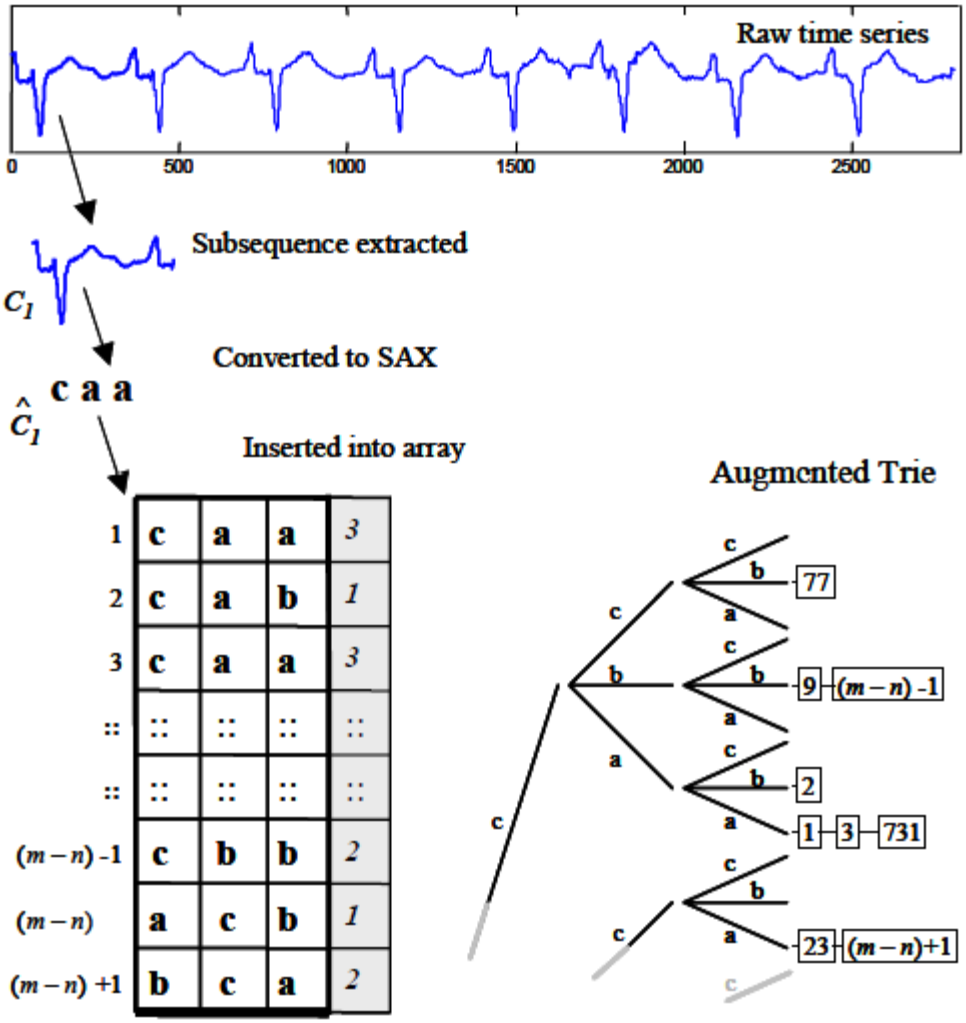

Figura 2.2. As duas estruturas usadas no *HOT SAX* a fim de dar suporte à sua heurística [Keogh, 2005]

Na **Figura 2.2**, uma determinada série temporal é transformada em uma sequência de caracteres, conforme é possível ver na **Figura 2.3**. Nessa fase, é executada uma aproximação PAA, técnica que será descrita na **Seção 2.2.1**. Na **Figura 2.3** foram utilizados três símbolos, os caracteres "a", "b" e "c", com uma janela deslizante de tamanho 8 e tamanho de palavra igual a 128 [Keogh, 2005]. A série temporal na **Figura 2.3** é então transformada na cadeia de caracteres "cbccbaab".

Continuando na **Figura 2.2**, uma janela da série é convertida em uma cadeia de caracteres que é então inserida em uma lista (mostrada à esquerda), contendo a própria cadeia de caracteres e a quantidade de vezes em que ela aparece na série. À direita é apresentada uma amostra da árvore de sufixos gerada a partir da série, cujo objetivo é determinar a frequência, a posição e o grau de cada subsequência da série em termos de ser uma anomalia. Uma subsequência cuja frequência é menor do que as outras tende a ser uma anomalia na série temporal.

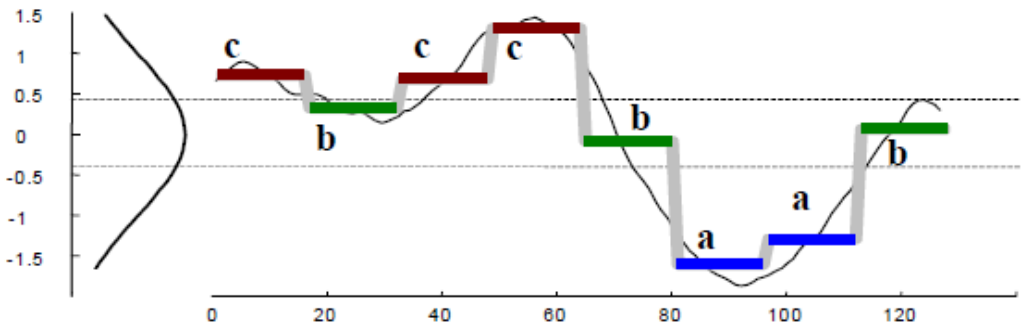

Figura 2.3. Uma série temporal é discretizada durante a execução da *HOT SAX* [Keogh, 2005]

O método proposto por [Ma, 2003], também para uma série temporal contínua, usa uma janela deslizante de comprimento especificado pelo usuário, mas não emprega discretização, ao invés disso, usa a regressão de vetor de suporte para modelar todas as subsequências vistas anteriormente durante a execução de seu algoritmo. A discretização que é empregada nesses tipos de trabalhos visa transformar a série

temporal de valores contínuos para uma representação mais simples, com o uso de valores discretos. Por exemplo, ao se transformar uma série temporal contínua para uma cadeia de caracteres, permite-se empregar técnicas e funções bem conhecidas na área de processamento de *strings* para os mais diversos fins – como para o reconhecimento de padrões. Contudo, existem outras formas de transformações de séries temporais, e por isso serão analisadas as principais formas, a fim de determinar seus prós e contras.

2.2.1 Análise de Transformações em Séries Temporais

Segue um resumo do trabalho realizado em [Keogh, 2011]:

a. **Transformada de Fourier Discreta (DFT – Discrete Fourier Transform)**: representa a série temporal como uma combinação linear de senos e cossenos, mas mantém somente os *n/2* coeficientes, devido ao fato de cada onda requerer dois parâmetros, para a fase *w* e amplitude *A,B*.

    i. Prós: boa habilidade para representar sinais naturais; rápido, já existe um algoritmo $O(n \log n)$ disponível;

    ii. Contras: dificuldade para lidar com sequências de diferentes tamanhos; não oferece suporte a medidas de distâncias ponderadas;

b. **Transformada Discreta de Ondaletas (DWT – Discrete Wavelet Transform)**: representa a série temporal como uma combinação de funções básicas de *ondaletas* (*wavelets*), mantendo somente os *N* coeficientes. A abordagem mais utilizada é a de Haar.

    i. Prós: boa habilidade para representar sinais estacionários; algoritmos de complexidade linear tornam o algoritmo rápido; habilidade para trabalhar com medidas não-Euclidianas;

ii. Contras: sinais devem ter um tamanho de $2^{\text{algum-inteiro}}$ e também não suporta medidas de distância ponderadas;

c. **Decomposição de Valor Singular (SVD – Single Value Decomposition)**: representa a série temporal como uma combinação de *auto-ondas* (*eigenwaves*), mas mantém somente os *N* coeficientes. Assemelha-se a DFT e DWT pelo fato de representar as séries em termos da combinação linear de formas. SVD tem sido muito utilizado na área de processamento de texto, onde é conhecido como *Latent Semantic Indexing – LSI*.

   i. Prós: técnica de redução de dimensionalidade linear ótima; os autovalores e autovetores extraídos desta técnica dizem algo sobre a estrutura dos dados;

   ii. Contras: é computacionalmente pouco viável ($O(mn^2)$, em tempo e espaço; também não pode suportar medidas de distância ponderadas ou não-Euclidianas;

d. **Aproximação Linear Seccional (PLA – Piecewise Linear Approximation)**: representa a série temporal como uma sequência de linhas retas. Estas podem ser conectadas, e neste caso são permitidas *N/2* linhas; se são desconexas, tem-se até *N/3* linhas.

   i. Prós: boa habilidade para lidar com sinais naturais; há uma versão linear para esta técnica; habilidade para suportar distâncias não-Euclidianas; é bastante aceita na comunidade científica;

   ii. Contras: não é atualmente indexável por nenhuma estrutura de dados;

e. **Aproximação Agregada Seccional (PAA – Piecewise Aggregate Approximation)**: representa a série temporal como uma sequência de funções básicas formato caixa, onde cada caixa tem o mesmo tamanho.

   i. Prós: muito rápida de se calcular; tão eficiente quanto as demais técnicas (empiricamente); suporta consultas de tamanho variável; suporta medidas de distância não-Euclidianas; suporta medidas de distância Euclidianas ponderadas; simples e intuitiva;

   ii. Contras: se visualizada diretamente, parece desagradável, não parecendo nada intuitiva sua visualização;

f. **Aproximação Constante Seccional Adaptativa (APCA – Adaptative Piecewise Constant Approximation)**: generaliza PAA para permitir que os segmentos constantes seccionais tenham tamanhos arbitrários.

   i. Prós: rápida de se calcular ($O(n)$); mais eficiente que as demais técnicas (em alguns casos); suporta consultas de tamanho variável; suporta medidas de distância não-Euclidianas; suporta medidas de distância Euclidianas ponderadas; suporta consultas rápidas e exatas;

   ii. Contras: implementação complexa; se visualizada diretamente, parece desagradável, não parecendo nada intuitiva sua visualização;

g. **Aproximação Agregada Simbólica (SAX – Symbolic Aggregate approXimation)**: converte a série temporal em um alfabeto de símbolos discretos, utilizando-se assim de técnicas de indexação de cadeias de caracteres para processar os dados.

i. Prós: limita inferiormente a distância Euclidiana; reduz a dimensionalidade; reduz a numerosidade; potencialmente pode-se valer do já bem conhecido e maduro campo de pesquisa sobre cadeias de caracteres para indexação da série temporal;

ii. Contras: não é claro como se deve exatamente discretizar a série (tamanho do alfabeto, valores, formas etc.); há muitas outras propostas de técnicas usando essa mesma abordagem de discretização;

A **Figura 2.4** [Keogh, 2011] mostra um exemplo gráfico das técnicas aqui apresentadas, de como elas transformam uma série temporal para suas respectivas representações.

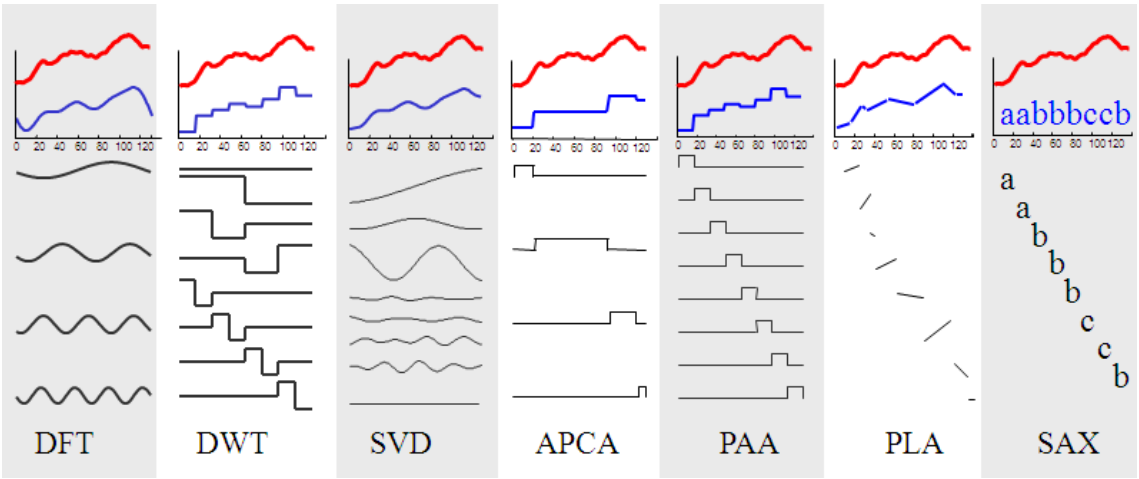

Figura 2.4. Resumo gráfico ilustrando como cada técnica apresentada representa uma série temporal [Keogh, 2011]

2.3 Utilizando Diferentes Níveis de Granularidade

Ao contrário dos métodos que procuram anomalias de um tamanho pré-especificado, o método apresentado em [Shahabi, 2000] procura por anomalias em diferentes níveis de granularidade (ou seja, dia, mês, ano). Os autores desenvolveram uma estrutura de

árvore chamada *TSA-Tree* que contém informações de tendência e de anomalias pré-computadorizadas em cada nó. A árvore cresce para uma profundidade *k* onde a granularidade diminui conforme aumenta a profundidade da árvore. Os autores utilizam a extração de características (filtros de *ondaletas* ou *wavelets*) para captar tendências e informações “surpreendentes” em cada granularidade. No entanto, os tipos de surpresas se limitam a "mudanças bruscas" nos dados que são capturados por "máximos locais".

2.4 Técnicas de Detecção de Padrões "Surpreendentes"

A técnica *InfoMiner*, apresentada em [Yang, 2001] e [Yang, 2004], detecta padrões "surpreendentes" em dados sequenciais de eventos periódicos. Assim, os dados já estão discretizados, e o período conhecido permite aos autores tratar uma única série temporal contínua como um conjunto de séries temporais menores. *InfoMiner* detecta anomalias globais por meio do cálculo do ganho de informação em cada sequência, e rotula essa sequência como anômala se o seu ganho de informação excede um certo limite. Como esta técnica é para dados de sequência de eventos, o uso deste programa em séries temporais de valores reais exigiria a discretização do conjunto de dados e comparações no espaço discretizado.

O trabalho apresentado em [Rebbapragada, 2009] introduz uma técnica para identificar padrões em dados de séries temporais utilizando um algoritmo chamado por eles como *PCAD*, que é um algoritmo baseado em agrupamento construído sobre a base do algoritmo *k-means*, mas projetado para lidar com várias séries temporais dessincronizadas a fim de encontrar correlação entre elas, gerando uma lista ordenada de ambas as anomalias globais e locais. Ela calcula o índice ou pontuação de anomalia para cada curva de luz, em relação a um conjunto de centroides produzidos por este algoritmo *k-means* modificado.

A técnica desenvolvida em [Cheng, 2009] propõe uma abordagem para a detecção e caracterização de anomalias em séries temporais multivariadas, apresentando um algoritmo para detecção de anomalias em dados de séries temporais multivariadas com ruído, empregando um método de alinhamento de núcleo de matriz para capturar as relações de dependência entre as variáveis na série. Nesta técnica, as anomalias são encontradas através da realização de um passeio aleatório transverso no grafo induzido pela matriz de núcleo alinhado.

2.5 Técnicas de Agrupamento de Dados

Em [Hautamaki, 2008], é realizado um agrupamento dos dados da série temporal no espaço Euclidiano utilizando *Random Swap* (RS) e agrupamento hierárquico aglomerativo, seguido de uma execução do *k-means* com ajuste fino para o cômputo de protótipos locais mais precisos, proporcionando uma melhor precisão de fase de agrupamento e uma melhoria para o *k-means*. Em [Rodrigues, 2008] é analisado um sistema incremental para o agrupamento de séries temporais utilizando uma técnica denominada *Online Divisive Agglomerative Clustering* (ODAC), que mantém continuamente uma hierarquia de árvore de grupos usando uma estratégia *top-down*. A qualidade do grupo é medida pelo cálculo do diâmetro do grupo, que é a maior diferença entre os objetos do mesmo grupo. Sua principal vantagem é que ele não precisa de um número predefinido de grupos alvo, executando várias vezes o algoritmo *k-means*. Sua desvantagem aparece quando a estrutura de árvore se expande e as variáveis movem-se da raiz às folhas, quando não há nenhuma certeza estatística sobre a decisão de atribuição, podendo dividir as variáveis. O cálculo de dados de alta dimensionalidade sendo processados pode representar uma desvantagem do algoritmo de agrupamento.

2.6 Conclusão

Avaliando as técnicas aqui apresentadas, após uma extensa pesquisa sobre os trabalhos correlatos, constatou-se que nenhuma das técnicas se assemelha à que é apresentada neste trabalho, conforme será apresentado no **Capítulo 3**. Essas diferenças são principalmente devidas à forma como o agrupamento da série temporal é realizado, à forma como as anomalias são identificadas em termos do comportamento geral da série, e da função de distância utilizada, que é a distância de Mahalanobis. Também identificou-se que, dentre as técnicas aqui apresentadas, a técnica HOT SAX é a que mais oferece condições de ser comparada com esta técnica aqui proposta, em função de se assemelhar em alguns aspectos, como na utilização de janelas deslizantes de tamanho fixo, na avaliação do grau de anomalia de cada subsequência, na aplicação de sua técnica a casos reais e na disponibilização de uma aplicação gráfica que permite utiliza-la sem a necessidade de reimplementá-la. Essa escolha será melhor descrita no **Capítulo 4**.

3 O ALGORITMO PROPOSTO

Neste capítulo será apresentada uma descrição formal e detalhada do algoritmo, onde será discutido cada aspecto importante do mesmo, analisando detalhadamente cada passo do algoritmo. Discutir-se-á também sobre como escolher adequadamente os dois principais parâmetros de execução do algoritmo, apresentando as principais nuances para a configuração dos mesmos.

3.1 Descrição do Algoritmo

O algoritmo, apresentado na Caixa 1, é um algoritmo de agrupamento dinâmico que, dados: (i) uma série temporal $T = \{t_1, t_2, \ldots, t_{|T|}\}$ de variáveis de valores reais indexadas no tempo em uma certa frequência e ordenadas pelo tempo, (ii) o tamanho inicial dos agrupamentos τ, e (iii) o fator de agrupamento φ, identifica um conjunto de padrões anômalos em T, $\mathbf{P} = \{\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_{|\mathbf{P}|}\}$, onde $\boldsymbol{P}_j = \{C_1, C_2, \ldots, C_{|\boldsymbol{P}j|}\}$ é um padrão anômalo de T, o qual é composto de um conjunto de grupos disjuntos $C_k = \{t_a, t_{a+1}, \ldots, t_b\}$, $1 \leq a \leq b \leq |T|$, os quais são subconjuntos ordenados de T.

Quadro 1 – Pseudo-Algoritmo do C-AMDATS

C-AMDATS (*T*, *τ*, *φ*)

```
1.  C ← CalcularGruposIniciais(T, τ);
2.  enquanto ocorrerem mudanças em C do
3.    C' ← C
4.    para i ← 1 a |T| faça
5.      Mova t_i de seu grupo em C para o grupo mais
6.      Próximo em C de acordo com f(C')
7.    fimpara
```

```
8.  fimenquanto
9.  repita
10.     Adicione C1 a P;
11.     Remova C1 de C;
12.     k ← 1;
13.     enquanto k <= |C| faça
14.         para j ← 1 a |P| faça
15.             se Pj é similar a Ck então
16.                 Adicione Ck a Pi;
17.                 Remova Ck de C;
18.             senão
19.                 k ← k + 1;
20.             fimse
21.         fimpara
22.     fimenquanto
23. até |C|=0
24. para i ← 1 a |P| faça
25.     Calcule o Índice de Anomalia r(Pi)
26. fimpara
27. OrdenarPorÍndiceDeAnomalia(P);
28. retorne P;
```

O algoritmo inicia na linha 1, computando o conjunto inicial de grupos de igual tamanho, ***C***. Neste passo, o conjunto T é dividido em um conjunto de conjuntos, ***C***, onde cada subconjunto $C_k = \{t_a, t_{a+1}, \ldots, t_b\}$ tem tamanho $|C_k| = \tau$, i.e., $b - a = \tau$ (exceto o último conjunto, nos casos em que $|T|$ não é divisível por τ). Após isso, nas linhas 2-8, o

algoritmo reconstrói ***C*** iterativamente utilizando sua cópia, ***C'*** (computado antes de cada iteração na linha 3). Para isso, nas linhas 5-6, o algoritmo utiliza $f(\boldsymbol{C'})$ para determinar qual conjunto em ***C*** é mais próximo de t_i.

A função $f(\boldsymbol{C'})$ computa a média dos segmentos de séries temporais dentro de cada grupo C'_k ou o seu centroide, $m_k = (t_a + t_{a+1} + \ldots + t_b) / (b - a)$ (veja **Figura 3.1**), e a distância, $d(t_i, m_k)$, de t_i para cada centroide m_k, para $1 \leq k \leq |\boldsymbol{C'}|$. Utilizando essas distâncias, nas linhas 5-6 o algoritmo move t_i de seu grupo atual em ***C*** para o grupo C_k, onde k é o índice do grupo C'_k cujo centroide m_k é o mais próximo de t_i de acordo com $d(.,\ .)$. Neste trabalho, a função de distância $d(.,\ .)$ escolhida foi a distância de Mahalanobis [Tou, 1974]. Essa escolha será explicada com maiores detalhes logo abaixo. Porém, com vistas a realizar comparações, a distância Euclidiana será também utilizada nos experimentos.

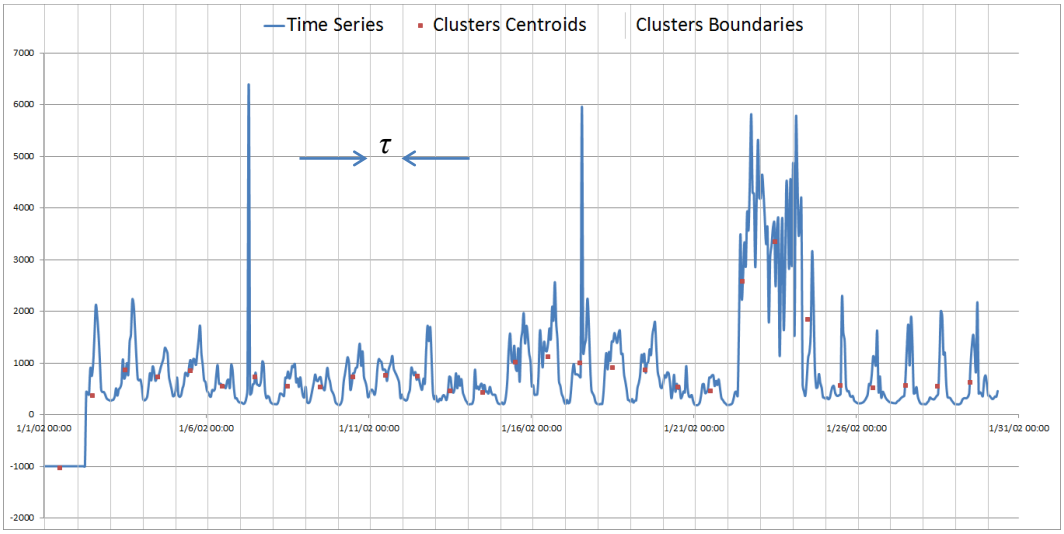

Figura 3.1. Uma série temporal T dividida em grupos C'_k de tamanho uniforme τ. Os pontos vermelhos são os centroides m_k de cada conjunto C'_k.

O laço nas linhas 2-8 termina quando nenhuma amostra t_i é movida nas linhas 4-7. Após esse laço, o conjunto de grupos ***C*** está formado em definitivo (veja **Figura 3.1**), sendo composto por grupos de amostras, C_k, que melhor agrupam as amostras de acordo com seus valores, distribuídas ao longo do eixo temporal (favor comparar **Figura 3.1**

com **Figura 3.3**; note que os tamanhos dos grupos da **Figura 3.3** não são iguais). Isso é devido à utilização da função de distância escolhida.

Em relação à função de distância, como mencionado acima, a fim de comparar os resultados da utilização da distância de Mahalanobis e seus benefícios, a distância Euclidiana, que é largamente utilizada e bem conhecida, foi escolhida e aplicada no algoritmo e nos casos de avaliação do mesmo. O uso da distância Euclidiana em algoritmos de agrupamento tende a formar grupos com formato circular, uma vez que ela não leva em conta a variância de cada dimensão do conjunto de dados (veja **Figura 3.2**). Assim, não importa se o formato do grupo é mais alongado no eixo x ou no eixo y, o formato geral do grupo sempre será circular. Entretanto, é possível que essa forma circular não seja adequada para representar o real formato do grupo. Para resolver esse problema, outra função de distância se faz necessária a fim de permitir a modelagem mais apurada do formato do grupo. Nesse contexto, a distância de Mahalanobis se torna uma boa opção porque, diferentemente da distância Euclidiana, ela leva em consideração a variância de cada dimensão simultaneamente. A **Equação** (**1** apresenta a formulação da distância de Mahalanobis [Tou,1974]:

$$d_m(x, \mu) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \qquad (1)$$

Na **Equação** (**1**, $x = (x_1, x_2, \ldots, x_n)^T$ é um variável específica do conjunto de dados, n é o número de dimensões das variáveis, $\mu = (\mu_1, \mu_2, \ldots, \mu_n)^T$ é o centroide de um determinado grupo e S é a matriz de covariância relativa àquele grupo. Em contraste com a distância Euclidiana, a aplicação da distância de Mahalanobis leva a uma forma elipsoidal. Portanto, a utilização dessa distância ao invés da Euclidiana consequentemente leva a uma medida de distância mais genérica. É importante frisar, contudo, que a utilização da distância de Mahalanobis em algoritmos de agrupamento

não é em si uma abordagem inédita, como se pode constatar em [Art, 1982] e [Singhal, 2006].

Alguém poderia observar que a utilização da matriz de covariância completa é um recurso muito complexo e por demais poderoso para modelar o problema em questão. Na verdade, se analisarmos somente a variância dos valores reais das amostras em *T*, não seria possível modelar apropriadamente o formato de cada grupo. Para ilustrarmos este aspecto, a **Figura 3.2** enfatiza as diferenças da aplicação das distâncias Euclidiana e de Mahalanobis na modelagem de um mesmo grupo hipotético, numa certa região da série temporal relativa ao Caso II.

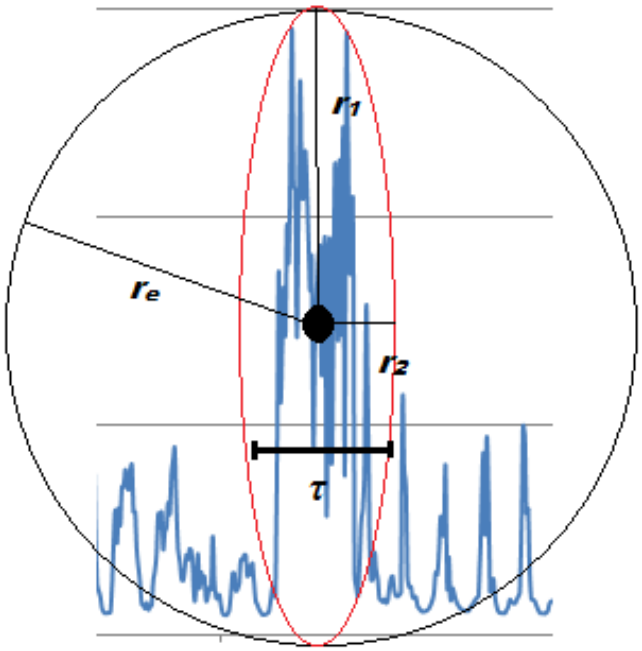

Figura 3.2. Um exemplo de uma série temporal ilustrando as diferenças entre a aplicação das distâncias Euclidiana (formando o círculo) e a de Mahalanobis (formando a elipse)

Na **Figura 3.2**, τ é o tamanho inicial dos grupos, r_e é o raio do círculo que engloba o grupo, e r_1 e r_2 correspondem aos raios da elipse que envolve o mesmo grupo. O valor de r_e é a distância Euclidiana do ponto mais distante pertencente ao grupo de seu centroide, enquanto que os valores de r_1 e r_2 foram obtidos pela aplicação da distância

de Mahalanobis. Como se pode notar, a figura geométrica que melhor engloba o grupo é a elipse, enquanto que o círculo está agrupando regiões que não pertencem ao grupo. É devido ao fato de que a função de distância de Mahalanobis leva em conta as duas dimensões simultaneamente, não separadamente. Para mostrar o impacto real de utilizar essa função distância em vez da distância euclidiana, serão apresentados os resultados da aplicação tanto a casos reais no **Capítulo 4**.

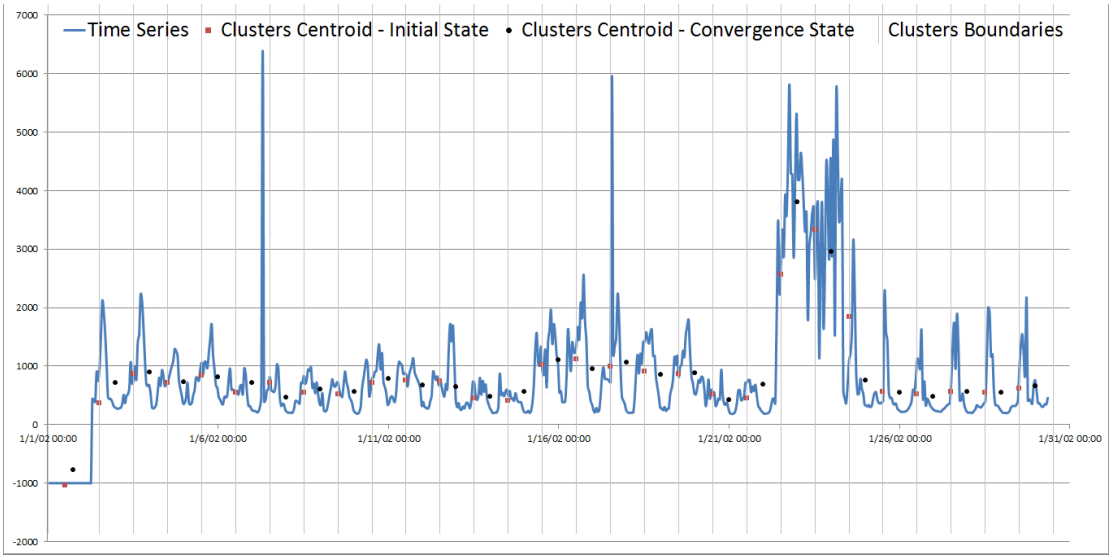

Figura 3.3. Uma série temporal *T* dividida em grupos C'_k, cada qual de tamanho variável. Os pontos vermelhos são os centroides no estado inicial do algoritmo, e os pontos pretos são os centroides após o processo iterativo descrito nas linhas 2-8

O passo seguinte (linhas 9-23) executa a tarefa de encontrar o conjunto **P** de padrões finais na série temporal *T*. Após todos os grupos terem sido encontrados, o algoritmo verifica quais grupos são semelhantes entre si. Esta similaridade é calculada utilizando o desvio padrão σ_y dos valores reais das amostras em *T*, a coordenada *y* de cada centroide e o fator de agrupamento φ. Se o módulo da diferença entre a coordenada *y* dos centroides de dois grupos é inferior ou igual a φ vezes σ_y, então estes grupos podem ser mesclados, o que significa que eles irão representar um mesmo padrão em **P**. Esta tarefa é realizada até que tudo os agrupamentos tenham sido analisados.

Na última etapa (linhas 24-27), o algoritmo realiza a detecção das anomalias. Uma anomalia é um padrão que não se assemelha com um comportamento esperado em *T*, ou seja, um padrão anômalo. Esta detecção é feita calculando o índice de anomalia para cada padrão ***P*** encontrado no passo anterior, que é calculado como a razão entre o tamanho total da série temporal e o somatório dos tamanhos dos grupos presentes em **P**. O índice de anomalia (ou classificação) *r* é uma medida de quanto ***P*** é interessante em termos de ser uma anomalia. Na sequência, o conjunto **P** é ordenado por *r* em ordem decrescente, e os padrões anormais serão aqueles com os maiores valores de índice de anomalia. Quanto maior o valor do índice de anomalia para um padrão ***P***, maior é a sua chance de ser uma anomalia em *T*. Na **Figura 3.4** é apresentado o estado final do algoritmo: todos os grupos semelhantes foram fundidos em um padrão, como afirmaram os critérios descritos acima. Três padrões foram encontrados, e de acordo com índice de anomalia, os mais anômalos cujos índices são os maiores, são os que estão em destaque na cor vermelha e verde, enquanto o padrão azul tem o menor valor de índice de anomalia.

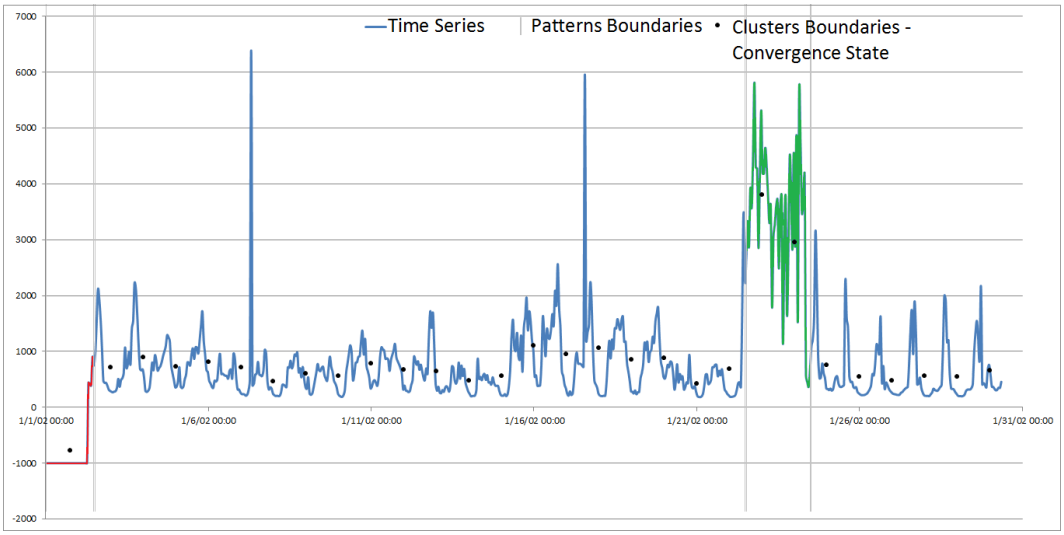

Figura 3.4. Uma série temporal *T* dividida em três padrões, ao final da execução do algoritmo. Os padrões em verde e vermelho são os mais anômalos.

Este algoritmo é baseado na ideia principal do algoritmo *k-means* [Hand, 2001]: a atribuição e a fase de atualização é basicamente a mesma que no algoritmo *k-means*. O que difere o *C-AMDATS* com o algoritmo de *k-means* são essencialmente quatro aspectos: a forma como o número de grupos é determinado em função do comportamento da série temporal, pois no *k-means* clássico esse número é informado diretamente pelo usuário; a distância de Mahalanobis é utilizada em vez da distância Euclidiana, que é geralmente utilizada em implementações do *k-means* [Hand, 2001]; e os dois últimos passos, que não estão presentes no algoritmo *k-means*. O algoritmo aqui proposto não necessita de qualquer referência ou dados de treinamento para calcular e detectar as anomalias, portanto, não há nenhuma fase de treinamento como em outras técnicas.

Com relação à sua complexidade, onde n é o número de variáveis, k é o número de grupos iniciais, e z é o número de iterações até o estado de convergência, verifica-se que:

- Na linha 1, quando a função `CalcularGruposIniciais(T, τ)` é chamada, cada amostra em T é visitada a fim de calcular o centroide inicial de cada grupo C_k. Logo, sua complexidade é $O(n)$;
- Nas linhas 2-8, a parte mais externa do laço é executada enquanto não se alcançar o estado de convergência, i.e., enquanto houver mudanças nos grupos. O laço nas linhas 4-6, interno ao primeiro, visita cada grupo C'_k a fim de calcular a matriz de covariância das amostras pertencentes a esse grupo, e em seguida calcular, para cada amostra em T, a distância de Mahalanobis até o centroide do grupo em questão. Esse passo, portanto, possui complexidade de $O(z.(k.|C_k| + k.n)) = O(zkn)$, onde $|C_k| < n$, $|C_k|$=a.n, $0 < a < 1$.

- Nas linhas 9-24, os laços, tanto o mais externo, quanto o intermediário e o mais interno, visitam somente os conjuntos de grupos $\boldsymbol{C}$ e o conjunto de padrões **P**, que iterativamente decresce o tamanho de $\boldsymbol{C}$ enquanto que incrementa o tamanho de **P**. Como não manipulam o conjunto de amostras em *T*, sua complexidade não é função de *n*, logo não impactam na complexidade geral do algoritmo em notação *O*. Mesmo assim, assumindo que o laço mais externo seja executado no máximo *k* vezes, o laço intermediário na ordem de no máximo *k* vezes, e o laço mais interno na ordem de no máximo *k* vezes (assumindo que o conjunto de padrões **P** tenha no máximo tamanho *k*), tem-se uma complexidade de $O(k^3)$;
- Nas linhas 24-26, o pior caso acontece quando |**P**| é igual a *k*, logo o laço é executado no máximo *k* vezes, tendo uma complexidade $O(k)$;
- Na linha 27, o algoritmo ordena o conjunto de padrões **P** pelo índice de anomalia de cada padrão. No pior caso, em que |**P**| é igual a *k*, e assumindo um algoritmo de ordenação de complexidade $O(n \log n)$ como o *QuickSort*, temos que nesse passo a complexidade é $O(k \log k)$.

Assim, temos que a complexidade do algoritmo é: $O(n) + O(nkz) + O(k^3) + O(k) + O(k \log k)$. Quando *n* cresce muito, os termos que não levam em conta *n* acabam por não impactarem no tempo total do algoritmo. Sendo assim, a complexidade se reduz ao termo mais complexo em função de *n*, i.e., $O(nkz)$, que é a complexidade geral do algoritmo em notação *O*.

Com relação à sua classificação, uma vez que o algoritmo é uma derivação do algoritmo *k-means*, que é um problema NP-Difícil [Mahajan, 2009], ele também pode ser classificado como um problema NP-Difícil.

No **Capítulo 2** são apresentados os trabalhos relacionados, onde se constatou não haver na literatura correlata nenhum trabalho ou algoritmo de agrupamento para detecção de anomalias em séries temporais que poderia ser até mesmo semelhante a esta técnica aqui apresentada. Existe um algoritmo de agrupamento, chamado *ISODATA* [Tou, 1974], que realiza a tarefa de mesclar e dividir os grupos durante sua execução. No entanto, a sua ideia é diferente da que foi aqui apresentada, uma vez que o algoritmo *ISODATA* leva em conta todas as dimensões dos dados para realizar as tarefas de fusão e divisão dos grupos, enquanto que a tarefa de fusão dos grupos no algoritmo *C-AMDATS* é realizada apenas uma vez, em um passo final e de um modo diferente, depois de todos os grupos terem sido encontrados, e avalia a semelhança com base em uma dimensão dos dados. Também, é importante notar que não há a tarefa de divisão de grupos no algoritmo proposto.

3.2 Definindo os Parâmetros do Algoritmo

A fim de executar o algoritmo, o usuário precisa definir basicamente dois parâmetros distintos: o *fator de agrupamento* φ e o *tamanho inicial dos grupos* τ.

O parâmetro φ é uma medida de quanto o algoritmo é sensível a alterações no comportamento normal da série temporal, uma vez que está estreitamente relacionada com o desvio padrão σ_y. Em geral, um valor de cerca de 1,0 é adequado para a maioria das aplicações. No entanto, caso seja conhecido antecipadamente, por exemplo, que os padrões anômalos na série temporal são mais sutis que o usual, seu valor pode ser diminuído ou alterado de acordo com a necessidade.

O parâmetro principal e mais importante desta técnica é o tamanho inicial dos grupos, τ. Como foi mostrado no **Capítulo 2**, há um conjunto de técnicas baseadas em segmentar a série temporal T em janelas de um comprimento pré-determinado e fixo w.

Definir o valor de *w* para essas técnicas é semelhante ao problema da definição de τ. O desafio é escolher um valor apropriado para isso, uma vez que se impõe a necessidade de se definir previamente um comprimento de janela que não seja muito grande a ponto de não conseguir enxergar a anomalia ou nem muito pequeno que não tenha um tamanho suficiente para englobar uma subsequência anômala. Assim, o usuário tem de saber, antes da execução de tais técnicas, o comprimento esperado de um padrão anômalo. Uma abordagem para lidar com este problema é definir um comprimento de janela com o mesmo tamanho de um ciclo comum da série temporal *T*, de modo que o seu comprimento possa ser conhecido antecipadamente pelo utilizador. Por exemplo, suponha uma série temporal *T* de medições de eletrocardiograma (ECG), amostrados a uma frequência determinada. Um tamanho de janela adequado seria o comprimento de um batimento cardíaco.

No entanto, pode haver casos em que o comprimento do ciclo não é conhecido, ou quando uma anomalia é de um comprimento diferente do comprimento de um ciclo. Além disso, em casos reais, não se sabe onde a anomalia se encontra na série temporal nem o seu comprimento. Assim, mesmo nos casos em que há pouco conhecimento sobre o comprimento da anomalia ou não se espera que a anomalia tenha o comprimento de um ciclo, é necessário que se possa especificar um valor correto para τ. Para esse efeito, podem ser utilizadas as informações fornecidas pela aplicação da análise espectral em *T*. Os métodos de análise espectral podem indicar periodicidade que são estacionárias, o que significa que eles não mudam muito em amplitude nem na frequência ao longo do tempo. No entanto, existem muitos casos em que as periodicidades são não estacionárias. Se este for o caso, a análise espectral pode ser ineficaz. Outro método semelhante é a análise *wavelet*. Com a análise *wavelet*, uma

série temporal pode ser estudada em múltiplas escalas (ou seja, baixas e altas frequências) simultaneamente [Hammer, 2005].

4 *C-AMDATS* – APLICAÇÕES E RESULTADOS

A fim de verificar a habilidade do algoritmo em analisar séries temporais de dados reais, esta técnica foi aplicada a alguns casos reais. Eles serão apresentados e discutidos posteriormente, assim como os resultados da aplicação do algoritmo.

Durante os testes, duas versões do algoritmo foram desenvolvidas: uma utilizando a distância Euclidiana e a outra utilizando a distância de Mahalanobis. De uma forma geral, os experimentos mostraram que a aplicação da distância de Mahalanobis levou a resultados melhores, porém demandou maior tempo de processamento de CPU devido à necessidade do algoritmo em computar a matriz de covariância para cada grupo, a cada iteração. Os resultados da aplicação de ambas as distâncias serão também apresentados.

4.1 Escolha e Comparação com Outra Técnica

Foi pesquisada outra técnica que visa detectar anomalias em séries temporais a fim de comparar com os resultados do algoritmo *C-AMDATS*. Os critérios de seleção dessa técnica foram:

- A técnica foi aplicada com sucesso na área de detecção de anomalias em series temporais?
- Necessita de dados de treinamento para encontrar as anomalias?
- Existem dados disponíveis para reprodução dos testes?
- Há informação suficiente sobre como configurar e escolher os parâmetros necessários à execução da técnica?
- Há alguma aplicação ou ferramenta disponível que permite a execução da técnica para qualquer tipo de série temporal?
- É uma pesquisa ativa na comunidade científica?

Após as atividades de pesquisa e análise, identificou-se que o algoritmo HOT SAX [Keogh, 2005] atende a esses requisitos. Na verdade, verificou-se que a abordagem HOT SAX tem uma abundância de obras que fazem referência a ela em uma ampla gama de aplicações em séries temporais, não somente na área de detecção de anomalias, que é o assunto principal do presente documento. Existe um aplicativo desenvolvido para a plataforma Windows chamado VizTree [Lin, 2004] que implementa essa técnica juntamente com uma ferramenta de visualização. A existência dessa ferramenta de execução e visualização da técnica HOT SAX possibilita executá-la sem a necessidade de reimplementá-la. Assim, aplicaram-se ambas as técnicas, *C-AMDATS* e HOT SAX, para todos os casos reais estudados neste trabalho. Além disso, gastou-se algum tempo utilizando o aplicativo VizTree a fim de compreender a sua aplicação e como configurar os parâmetros apropriadamente. Foi possível reproduzir com sucesso os casos de demonstração que acompanham a ferramenta. Deve-se ressaltar que as comparações que foram realizadas não tem a intenção de desacreditar qualquer um dos autores de ambas as abordagens.

Ao se comparar os detalhes técnicos, pode-se verificar que:

- Ambas tem complexidade $O(n)$;
- Ambas utilizam a abordagem de dividir a série temporal em janelas ou grupos de tamanho fixo;
- A técnica HOT SAX utiliza janelas deslizantes que não variam de tamanho durante sua execução, enquanto que as janelas ou grupos em *C-AMDATS* variam de posição e tamanho durante sua execução, conforme descrito na **Seção 3.1**;

- A técnica HOT SAX necessita pré-processar a série temporal através da discretização empregada pela técnica PAA (vide **Seção 2.2.1**), enquanto que *C-AMDATS* não realiza nenhum pré-processamento na série;
- Em HOT SAX, cada subsequência extraída da série, durante a fase de avaliação das janelas deslizantes, é inserida em duas estruturas de controle, uma lista e uma árvore de sufixos, como ilustrado na **Figura 2.2**, e a partir delas são identificadas as subsequências que mais se diferem das demais, apontando as regiões anômalas através da análise da frequência com que acontecem na série. Já em *C-AMDATS* cada grupo é inserido em uma lista que posteriormente será processada a fim de identificar que grupos se assemelham mais, formando os padrões, conforme explicado no **Capítulo 3**. Os padrões que tiverem os maiores índices de anomalia (vide **Capítulo 3**), ou seja, que menos se assemelharem a outros padrões, constituirão as regiões candidatas a serem anômalas.
- Conforme [Keogh, 2005], para executar HOT SAX é necessário especificar 3 parâmetros: o tamanho das janelas deslizantes, tamanho do alfabeto e a quantidade de símbolos por janela. Além disso, durante a utilização da aplicação VizTree, foi necessário especificar algumas opções a mais a fim de tirar o melhor resultado da técnica, conforme será descrito em cada caso estudado na **Seção 4.3**. Para a execução da técnica C-AMDATS, contudo, é necessário definir somente 2 parâmetros, o tamanho inicial dos grupos e o fator de agrupamento.

4.2 Descrição dos Casos Reais

Para validar os resultados da aplicação desta técnica no domínio da detecção de anomalias em séries temporais, cinco casos diferentes foram estudados. A seguir, é descrito cada um:

I. *Concentrações de ozônio troposférico em uma área metropolitana:* Este caso refere-se à medição de ozônio troposférico ao longo de três meses no ano de 2002, amostrados em frequência horária. O material foi coletado em uma área metropolitana, por um sistema de monitoramento automatizado. Referenciar-se-á a esse parâmetro como sua representação química, O_3. O seu comprimento de ciclo é de 24 horas (um ciclo diário);

II. *Concentrações de monóxido de carbono em uma área metropolitana:* refere-se à medição de monóxido de carbono durante dois meses no ano de 2002. Os dados são amostrados por hora, e também foram coletados na mesma área metropolitana do caso acima. Para referência, também será utilizada sua representação química como referência, CO. Como no caso acima, o comprimento do seu ciclo é de 24 horas;

III. *Emissões de gases de uma indústria siderúrgica:* refere-se à medição e monitoramento de um parâmetro gasoso emitido a partir de uma instalação de indústria de aço, e é amostrado por minuto. A partir de agora ele será referenciado como Parâmetro A. Este parâmetro tem um ciclo de duração de 1 hora;

IV. *Emissões de partículas de uma indústria siderúrgica:* amostrados por minuto, ele se refere ao monitoramento e medição de um tipo de partícula de uma instalação de indústria siderúrgica. Será ainda referenciado como Parâmetro B. A duração de seu ciclo é de 30 minutos;

V. *Demanda de energia em uma instalação holandesa:* este é um conjunto de dados resultados da medição do consumo de energia em um centro de pesquisa holandês para todo o ano de 1997. As anomalias estão relacionadas com semanas incomuns.

Nos primeiros quatro casos, as anomalias foram relatadas como falhas em equipamentos ou processos, e os seus dados foram previamente classificados por especialistas humanos como inválidos. Uma vez que estes dados já foram analisados por especialistas, o objetivo é avaliar se o algoritmo é capaz de identificar as mesmas anomalias que foram detectadas pelos peritos. Além disso, também foi aplicado o algoritmo *C-AMDATS* a outro caso que já foi avaliado pela abordagem HOT SAX em [Keogh, 2005], a fim de avaliar sua capacidade de identificar anomalias em outro tipo de séries temporais, sendo este o Caso V.

Nas imagens que apresentam os casos, haverá barras verticais que servirão para delimitar as regiões de fronteira entre as regiões anômalas e o restante da série.

4.3 Avaliando o Desempenho do Algoritmo

Para avaliar o desempenho do algoritmo com relação à sua capacidade de identificar os mesmos padrões anômalos identificados pelos especialistas humanos, seus resultados foram comparados aos padrões utilizando as metodologias de *recall*, *precision* e *accuracy* [Olson, 2008]. Eles são indicados como se segue:

$$precision = \frac{t_p}{t_p + f_p} \quad (2)$$

$$recall = \frac{t_p}{t_p + f_n} \quad (3)$$

$$\mathbf{accuracy} = \frac{\mathbf{t_p + t_n}}{\mathbf{t_p + f_p + t_n + f_n}} \quad (4)$$

Onde:

- t_p é o número de positivos verdadeiros, ou seja, o número de variáveis corretamente classificadas como pertencentes à classe;
- t_n é o número de negativos verdadeiros, que são as variáveis que foram corretamente classificadas como não pertencentes à classe;
- f_p é o número de positivos falsos, que são as variáveis que foram incorretamente classificadas como pertencentes à classe;
- f_n é o número de negativos falsos, que são as variáveis que foram incorretamente classificadas como não pertencentes à classe;

Para calcular estas variáveis, construiu-se uma matriz de confusão para cada abordagem, isto é, uma versão do algoritmo *C-AMDATS* usando a função de distância Euclidiana, outra versão usando a função de distância Mahalanobis, e o algoritmo HOT SAX. A **Tabela 4.1** apresenta como a matriz de confusão é construída.

Tabela 4.1. A construção da matriz de confusão

	Classe de Referência	
Classe Predita	t_p	f_p
	t_n	f_n

Recall, *precision* e *accuracy* não devem ser analisados separadamente, ao invés disso, eles devem ser considerados em conjunto na avaliação do modelo de destino. Um valor elevado de *precision* significa que todas as variáveis retornadas foram corretamente classificadas, mas é possível que se tenham perdido variáveis relevantes; enquanto que um resultado elevado de *recall* significa que todas as variáveis relevantes foram encontradas, mas pode haver uma abundância de resultados inúteis. E *accuracy* é a exatidão geral do modelo sendo considerado, que leva em conta não apenas as classificações verdadeiras positivas, mas também os valores verdadeiros negativos.

Agora serão apresentados os resultados da aplicação do algoritmo *C-AMDATS* nos cinco casos acima mencionados. Em cada caso, foram selecionadas as anomalias mais relevantes com base em seus índices de anomalia. Além disso, foram computados os resultados para ambas as funções de distância, ou seja, a de Euclides e a de Mahalanobis. Para referência, *C-AMDATS*$_M$ refere-se ao algoritmo *C-AMDATS* utilizando a função de distância de Mahalanobis, enquanto o *C-AMDATS*$_E$ irá se referir ao algoritmo *C-AMDATS* usando a função de distância Euclidiana.

Na maioria dos casos, o usuário poderia esperar que tais ferramentas de detecção de anomalias em séries temporais devessem ser capazes de indicar as subsequências anômalas em uma determinada série temporal com alguma qualidade. Valores para *recall*, *precision* e *accuracy* de 90% podem ser tão suficientemente bons quanto um valor de 100%, mas um valor de 50% pode não ser adequado, indicando que a técnica pode não ser confiável para este propósito. O objetivo nesta seção é avaliar a capacidade do algoritmo em identificar as mesmas anomalias que os especialistas humanos fizeram, já que há essa referência, em conjunto com os padrões anormais já encontrados por HOT SAX para o Caso V.

4.3.1 Caso I – Ozônio Troposférico

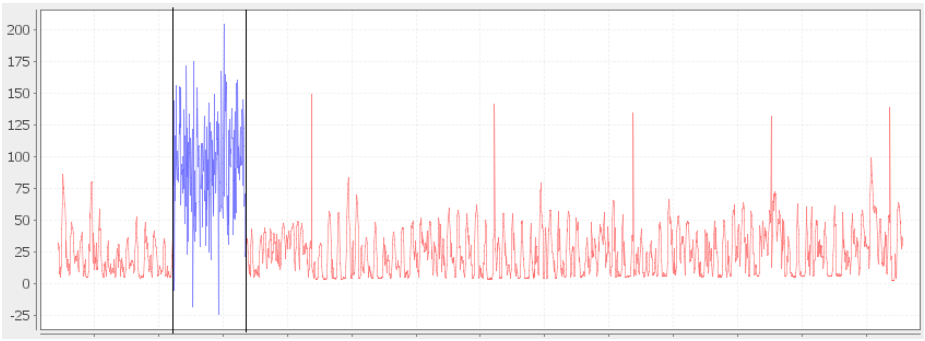

Figura 4.1. Gráfico temporal de O_3 evidenciando os padrões encontrados por *C-AMDATS*$_M$

A **Figura 4.1** mostra toda a série para este caso, destacando dois padrões principais, como resultado da aplicação da abordagem *C-AMDATS*$_M$. Baseado em seus índices de anomalia, o padrão de destaque com a cor azul, delimitado pelas barras verticais, é o padrão mais relevante em termos de ser uma anomalia. Assim, considera-se como o padrão anômalo. A seguir na **Tabela 4.2** é apresentada a matriz de confusão e os resultados para *recall*, *precision* e *accuracy*.

Tabela 4.2. Matriz de confusão do Caso I

	Padrão	**Anômalo**	*Precision*	*Recall*	*Accuracy*
C-AMDATS$_M$	190	2	0,9896	0,9948	0,9986
	2 015	1			
C-AMDATS$_D$	169	0	1,0000	0,8848	0,9900
	2 017	22			
HOT SAX	47	0	1,0000	0,2461	0,9348
	2 017	144			

Para a abordagem *C-AMDATS*, ambas as versões utilizaram os mesmos valores dos parâmetros: *tamanho iniciais dos grupos* e *fator de agrupamento* iguais a 24 horas e 1,0, respectivamente, enquanto que para HOT SAX foram definidos três parâmetros: o comprimento da janela, o número de símbolos por janela, e o tamanho do alfabeto. Foram necessárias cerca de 2 horas para que fosse possível encontrar a melhor combinação de parâmetros, e descobriu-se que um comprimento de janela de 24 horas, número de símbolos por janela de 3 e tamanho do alfabeto de 4 foi o melhor arranjo. É possível observar também que, conforme [Keogh, 2005], os melhores números para o tamanho do alfabeto são 3 ou 4. Além disso, também foi necessário definir uma opção avançada na ferramenta VizTree, chamada "No Overlapping Windows", o que levou aos melhores resultados experimentados com esta técnica.

Os resultados mostraram que, em geral, todas as abordagens conseguiram apontar a região da anomalia. No entanto, é evidente que a abordagem HOT SAX teve um mau resultado de *recall*, enquanto que *C-AMDATS$_M$* apresentou-se consideravelmente melhor. Na **Figura 4.2**, em comparação com a **Figura 4.1**, é apresentado um resultado

gráfico da aplicação da técnica HOT SAX, onde se evidencia visualmente a diferença entre os resultados de ambas as técnicas.

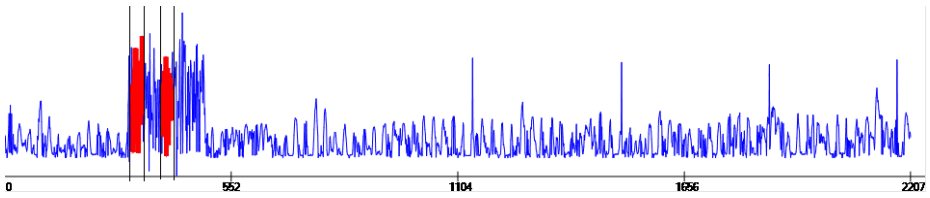

Figura 4.2. Um gráfico temporal com a mesma série de O_3 mostrando, em vermelho, a região anômala encontrada pela técnica HOT SAX

O padrão vermelho foi extraído do ramo "ccb", e é o ramo que mais corresponde à região anômala real.

4.3.2 Caso II – Monóxido de Carbono

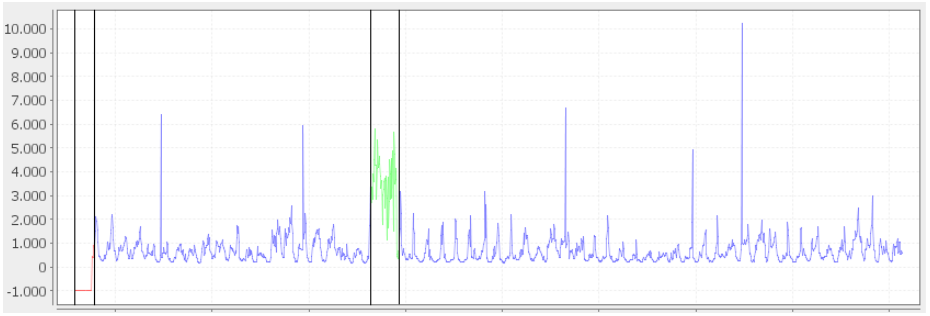

Figura 4.3. Gráfico temporal de CO, evidenciando os padrões encontrados por *C-AMDATS*$_M$

A **Figura 4.3** mostra o resultado da abordagem *C-AMDATS*$_M$ para este caso. Três padrões principais se destacam: o vermelho, o verde e o azul, em ordem de índice de anomalia, que levou a selecionar os padrões destacados com cor vermelha e verde como os padrões anômalos, sendo estes delimitados pelas barras verticais. Na

Tabela 4.3 é apresentada a matriz de confusão para este caso.

Tabela 4.3. Matriz de confusão para o Caso II

	Padrão	**Anômalo**	*Precision*	*Recall*	*Accuracy*
C-AMDATS$_M$	82	2	0,9762	0,9111	0,9931
	1 348	8			
C-AMDATS$_D$	66	7	0,9041	0,7333	0,9785
	1 343	24			
SAX	40	8	0,8333	0,4444	0,9597
	1 342	50			

Os valores dos parâmetros para *C-AMDATS* foram: *tamanho inicial dos grupos* de 24 horas e *fator de agrupamento* de 1,2. Para o HOT SAX, a técnica foi executada com os mesmos valores que foram utilizados no Caso I. Do mesmo modo, ambas as abordagens foram capazes de encontrar, alguns parcialmente, as regiões das anomalias. No entanto, mais uma vez a abordagem HOT SAX foi apenas capaz de dar uma pista sobre a segunda anomalia, como se pode ver na **Figura 4.4**, enquanto que a *C-AMDATS$_M$* foi capaz de dar um resultado mais satisfatório, em comparação com os outros.

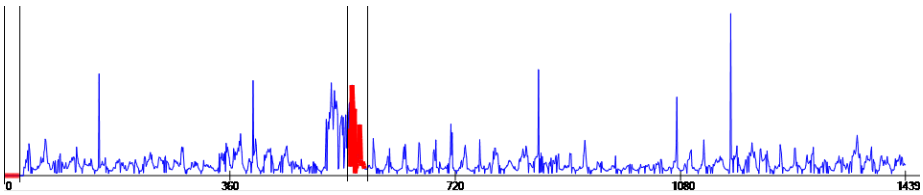

Figura 4.4. Gráfico temporal de CO onde se mostra o padrão, em vermelho, que corresponde à região anômala deste caso encontrada pela técnica HOT SAX

4.3.3 Caso III – Emissões de Gases de uma Indústria Siderúrgica

A **Figura 4.5** mostra a série temporal referente a este caso, e uma região de destaque correspondente à anomalia encontrada por *C-AMDATS*$_M$, na cor verde delimitada pelas barras verticais. Foram definidos os seguintes parâmetros para *C-AMDATS*: *tamanho inicial dos grupos* igual a 60 minutos, e *fator de agrupamento* igual a 1,0.

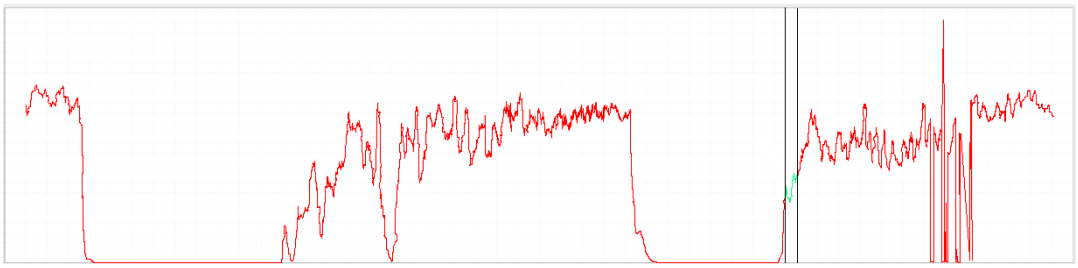

Figura 4.5. Gráfico temporal para o Parâmetro A utilizando a técnica *C-AMDATS*$_M$, evidenciando uma anomalia sutil na série

Para HOT SAX, foram escolhidos os mesmos parâmetros de configuração dos dois casos anteriores, com exceção do tamanho da janela, que foi de 60 minutos. Após sua execução, foi selecionado o padrão que apresentou melhor desempenho, o que correspondeu ao ramo "cca". A matriz de confusão para este caso é apresentada na **Tabela 4.4**.

Tabela 4.4. Matriz de Confusão para o Caso III

	Padrão	**Anômalo**	*Precision*	*Recall*	*Accuracy*
***C-AMDATS*$_M$**	50	9	0,8475	1,0000	0,9984
	5 701	0			
***C-AMDATS*$_D$**	20	39	0,3390	0,4000	0,9880
	5 671	30			
HOT SAX	0	60	0,0000	0,0000	0,9809
	5 650	50			

Este é um caso interessante. A subsequência anômala é um padrão muito sutil em relação ao restante da série. *C-AMDATS*$_M$ conseguiu identificar a anomalia, como se pode constatar quando se analisa *recall*, *precision* e *accuracy* simultaneamente. Embora a variável *accuracy* tenha apresentado valores muito bons (próximos de 1,0) em todas as abordagens avaliadas, quando se verifica mais de perto as variáveis *precision* e *recall*, podem-se ver resultados. Isso se deve ao fato de que *accuracy* leva em conta os valores de negativos verdadeiros, isto é, a capacidade do modelo em identificar o que não é uma anomalia. Como é possível observar ao analisar a matriz de confusão para este caso e a **Figura 4.6**, a abordagem HOT SAX não conseguiu detectar a anomalia em si - apesar de a região destacada estar perto da verdadeira subsequência anômala - mas conseguiu corretamente o resto da série temporal como não sendo um padrão anômalo.

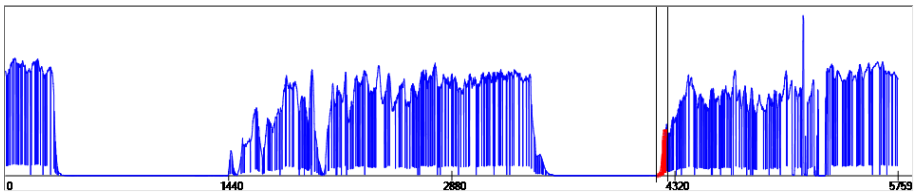

Figura 4.6. Gráfico temporal da série do Caso III, com a região em vermelho denotando a anomalia encontrada por HOT SAX

4.3.4 Caso IV – Emissões de Partículas de uma Indústria Siderúrgica

A **Figura 4.7** mostra a série temporal para este caso. A região correspondente à anomalia encontrada por *C-AMDATS*$_M$ está nas cores verde e azul, e estão delimitadas pelas barras verticais. Os valores dos parâmetros para *C-AMDATS* foram: *tamanho inicial dos grupos* igual a 30 minutos, e *fator de agrupamento* igual a 1,0.

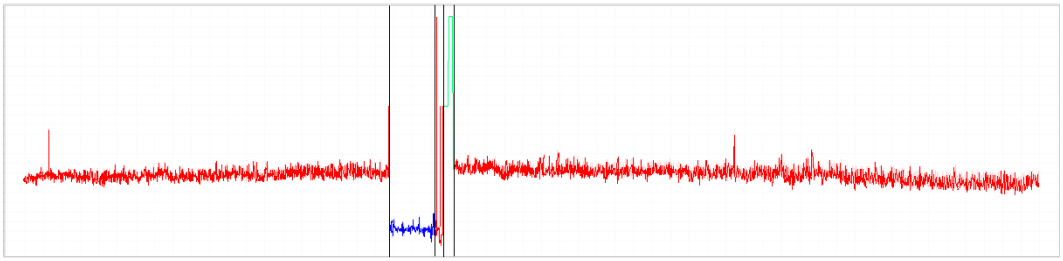

Figura 4.7. Gráfico temporal do Parâmetro B, evidenciando em azul e verde os padrões anômalos encontrados por *C-AMDATS*$_M$

Para HOT SAX, somente foi modificado o tamanho da janela de 30 minutos, mantendo os outros parâmetros com os mesmos valores utilizados nos casos anteriores. Os ramos mais significativos foram o "abd", "acd" e "dba". A matriz de confusão é apresentada na

Tabela 4.5.

Tabela 4.5. Matriz de Confusão do Caso IV

	Padrão	**Anômalo**	*Precision*	*Recall*	*Accuracy*
***C-AMDATS*$_M$**	160	20	0,8889	0,8743	0,9870
	3097	23			
***C-AMDATS*$_D$**	183	56	0,7657	1,0000	0,9830
	3061	0			
HOT SAX	35	55	0,3889	0,1913	0,9385
	3062	148			

Mais uma vez, *C-AMDATS*$_M$ apresentou os melhores resultados. *C-AMDATS*$_E$ comportou-se razoavelmente bem, enquanto que HOT SAX apresentou os piores resultados. **Figura 4.8** graficamente mostra os resultados da abordagem HOT SAX.

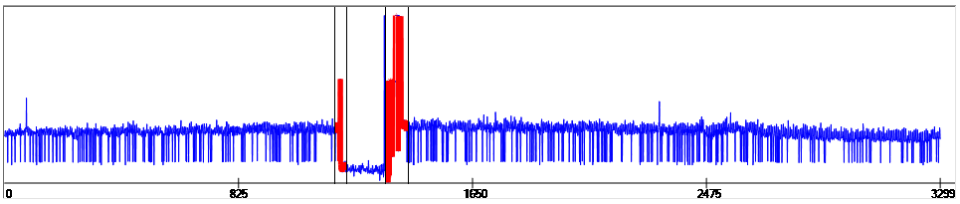

Figura 4.8. Gráfico temporal do Caso IV gerado a partir do programa VizTree, como resultado da execução da HOT SAX, onde se evidencia em vermelho os padrões anômalos encontrado pela técnica

4.3.5 Caso V – Demanda de Energia em uma Instalação Holandesa

Este caso foi avaliado por [Keogh, 2005], e [Wijk, 1999] mostra uma referência completa sobre os feriados e férias escolares do ano de 1997 na Holanda, que impactaram na demanda de energia em uma instalação holandesa, levando a comportamentos anômalos em algumas semanas. Os dados foram conseguidos a partir de [Keogh, 2011]. Como a técnica HOT SAX já foi aplicada com sucesso a este caso como descrito por [Keogh, 2005], o objetivo principal é verificar se a abordagem *C-AMDATS* também é capaz de identificar as semanas anômalas. Como não há nenhuma anotação sobre quais são exatamente as semanas anômalas, mas apenas um indício, nenhuma matriz de confusão será construída, apenas uma comparação direta será feita.

Os padrões anormais são devidos às semanas que corresponderam aos feriados e férias escolares no ano de 1997 na Holanda. A **Figura 4.9** mostra uma representação gráfica do resultado da aplicação de $C\text{-}AMDATS_M$ a esse caso. Os parâmetros *tamanho inicial dos grupos* e *fator de agrupamento* foram configurados como 5 e 0,7, respectivamente. As regiões anômalas estão delimitadas por barras verticais, conforme se pode ver na figura.

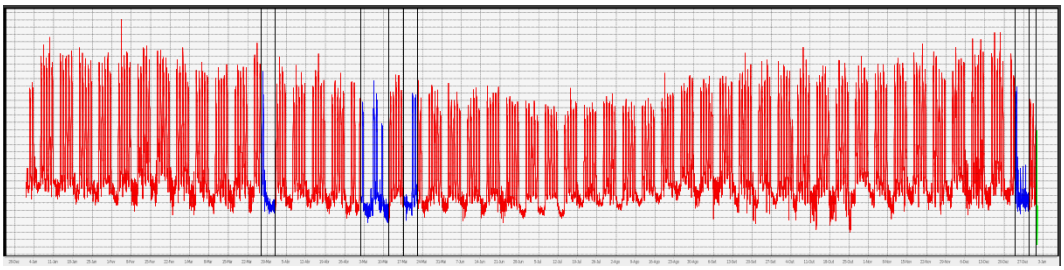

Figura 4.9. Gráfico temporal mostrando os padrões encontrados pela aplicação de *C-AMDATS*$_M$ a esse caso

As subsequências anômalas estão evidenciadas em cor azul e são apresentadas em detalhes a seguir:

- 26 de Março a 31 de Março, correspondendo ao feriado de 28 de Março a 31 de Março;
- 1º de Maio a 11 de Maio, correspondendo aos feriados de 5 de Maio, 8 de Maio, e às férias escolares de 3 de Maio a 11 de Maio;
- 17 de Maio a 22 de Maio, período relativo à semana do feriado de 19 de Maio;
- 23 de Dezembro a 29 de Dezembro, e 31 de Dezembro, relativo aos feriados de 25 e 26 de Dezembro, e às férias escolares de 21 a 31 de Dezembro.

Como se pode constatar, *C-AMDATS*$_M$ conseguiu identificar as semanas mais incomuns presentes na série. Observa-se também que, para esse caso, *C-AMDATS*$_E$ não foi capaz de identificar nenhuma das semanas incomuns.

5 CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentada uma proposta de um algoritmo para detecção de anomalias em séries temporais. Foi possível mostrar que há uma variedade de contribuições feitas para este campo de pesquisa. Identificaram-se as principais contribuições apresentadas recentemente, que foram analisadas a fim de avaliar se o algoritmo proposto neste trabalho é de fato uma nova contribuição para a comunidade científica. Verificou-se que não existe uma técnica semelhante. A seguir, foram apresentados os conceitos por trás desse trabalho, e, em seguida, foi descrita a proposta deste trabalho. Finalmente, foi aplicado o algoritmo para casos de dados reais, quando foi possível para cada caso verificar que o algoritmo apresentou bons resultados, o que mostra que ele pode ser aplicado como uma ferramenta para alavancar o trabalho dos especialistas na análise e identificação de anomalias em séries temporais. Além disso, uma comparação detalhada foi apresentada para cada caso e, em geral, todos os modelos avaliados apresentaram bons valores de *accuracy*, porém a técnica HOT SAX apresentou resultados ruins para *precision* e *recall* na maioria dos casos. Após uma análise criteriosa e precisa, conforme foi mostrado na avaliação de cada caso, *C-AMDATS* apresentou melhor desempenho no geral.

Existem vários trabalhos futuros a serem desenvolvidos a seguir:

a. Utilizar o *C-AMDATS* em um ambiente operacional, onde o algoritmo será configurado para funcionar continuamente, desempenhando as funções de análise de dados de séries temporais, procurando por padrões e enviando relatórios das análises para os especialistas. Em seguida, estes especialistas seriam capazes de verificar os resultados do algoritmo em tempo real;

b. Implementar uma função para analisar parâmetros correlatos simultaneamente a fim de encontrar anomalias correlatas. Por exemplo, ozônio e radiação solar, embora sejam parâmetros diferentes, possuem uma correlação intrínseca. Esta recomendação exigiria a criação de uma derivação do algoritmo *C-AMDATS* a ser aplicado na análise dos dados várias séries temporais na mesma execução;

c. projetar um módulo de aprendizagem de modo a permitir ao algoritmo aprender com o usuário quais são os melhores valores calculados em um determinado momento, com base em aplicações anteriores do algoritmo que foram validadas pelo usuário.

Com base nos resultados aqui apresentados, acredita-se que esse trabalho pode ser aplicado com sucesso em várias áreas deste campo de pesquisa com o intuito de melhorar a forma como os dados de séries temporais são analisados a fim de detectar anomalias.

6 REFERÊNCIAS

1. U. Rebbapragada, P. Protopapas, C.E. Brodley, C. Alcock, Finding Anomalous Periodic Time Series: An Application to Catalogs of Periodic Variable Stars, Spring Machine Learning Journal 74 (2009), pp. 281-313. DOI: 10.1007/s10994-008-5093-3;
2. E.M Knorr, R. T. Ng, Algorithms for Mining Distance-Based Outliers in Large Datasets. Em: Proceedings of the 24th International Conference on Very Large Data Bases – VLDB, VLDB International Conference, 1998, pp. 392–403;
3. S. Ramaswamy, R. Rastogi, K. Shim, Efficient Algorithms for Mining Outliers from Large Datasets. Em: SIGMOD '00: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD, 2000, pp. 427–438;
4. F. Angiulli, C. Pizzuti, Fast Outlier Detection in High Dimensional Spaces. Em: Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, 2002, pp. 15–26;
5. M. Wu, C. Jermaine, Outlier Detection by Sampling with Accuracy Guarantees. Em: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 767–772;
6. M. Breunig, H. Kriegel, R. Ng, J. Sander, LOF: Identifying Density-Based Local. Em: Proceedings of the ACM SIGMOD International Conference on Management of Data, 2000, pp. 93–104;
7. W. Jin, A. K. H. Tung, J. Han, Mining Top-N Local Outliers In Large Databases. Em: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 293–298;

8. D. Ren, B. Wang, W. Perrizo, RDF: A Density-based Outlier Detection Method Using Vertical Data Representation. Em: Proceedings of the 4th IEEE International Conference on Data Mining, 2004, pp. 503–506.
9. D. Dasgupta, S. Forrest, Novelty Detection in Time Series Data Using Ideas from Immunology. Em: Proceedings of the International Conference on Intelligent Systems, 1996, pp. 82–87. DOI: 10.1.1.57.3894;
10. E. Keogh, S. Lonardi, B. Y. Chiu, Finding Surprising Patterns in a Time Series Database in Linear Time and Space. Em: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 550–556;
11. J. Ma, S. Perkins, Online Novelty Detection on Temporal Sequences. Em: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 613–618;
12. L. Wei, N. Kumar, V. Lolla, E. Keogh, S. Lonardi, C. Ratanamahatana, Assumption-Free Anomaly Detection in Time Series. Em: SSDBM'2005: Proceedings of the 17th International Conference on Scientific and Statistical Database Management, 2005, pp. 237–240;
13. J. Yang, W. Wang, P. S. Yu, Infominer: Mining Surprising Periodic Patterns. Em: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 395–400;
14. J. Yang, W. Wang, P. S. Yu, Mining Surprising Periodic Patterns, Data Mining and Knowledge Discovery, ACM, vol. 9, 2004, pp. 189–216;
15. C. Shahabi, X. Tian, W. Zhao, TSA-Tree: A Wavelet-Based Approach to Improve the Efficiency of Multilevel Surprise and Trend Queries on Time-Series Data. Em: Proceedings of the 12th International Conference on Statistical and Scientific Database Management, 2000, pp. 55–68;

16. H. Cheng, P. Tan, C. Potter, S. Klooster: Detection and Characterization of Anomalies in Multivariate Time Series. Em: Proceedings of the 9th SIAM International Conference on Data Mining, 2009, pp. 413–424;
17. P. Norvig, S. Russel, Artificial Intelligence: A Modern Approach, Second Edition, Prentice Hall Series in Artificial Intelligence, New Jersey, 2003;
18. D. Hand, H. Mannila, P. Smyth, Principles of Data Mining, The MIT Press, Massachussets, 2001;
19. H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, E. Keogh, Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures, ACM Proceedings of the VLDB Endowment, vol. 1, no. 2, 2008, pp. 1542-1552;
20. D. Art, R. Gnanadesikan, J. R. Kettenring, Data-based Metrics for Cluster Analysis, Utilitas Mathematica, 21A, 1982, pp. 75-99;
21. J. T. Tou, R.C. Gonzalez: Pattern Recognition Principles, Addison-Wesley, Reading, MA, 1974;
22. J. B. MacQueen: Some Methods for Classification and Analysis of Multivariate Observations. Em: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297;
23. V. Hautamaki, P. Nykanen, P. Franti, Time Series Clustering by Approximate Prototypes. Em: International Conference on Pattern Recognition (ICPR), 2008, pp. 1–4;
24. P. P. Rodrigues, J. P. Pedroso, Hierarchical Clustering of Time Series Data Streams, IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 5, 2008, pp. 615-627;

25. V. Kavitha, M. Punithavalli, Clustering Time Series Data Stream – A Literature Survey, International Journal of Computer Science and Information Security, vol. 8, no. 1, 2010, pp. 289-294;

26. A. Singhal, D. E. Seborg, Clustering Multivariate Time Series Data, Journal of Chemometrics 19 (2006), pp. 427-438;

27. M. Mahajan, P. Nimbhorkar, K. Varadarajan, The Planar k-Means Problem is NP-Hard, Lecture Notes in Computer Science 5431: 274–285. DOI: 10.1007/978-3-642-00202-1_24, 2009;

28. Ø. Hammer, Paleontological Data Analysis, Wiley-Blackwell Publishing, UK, 2005;

29. E. Keogh, J. Lin, A. Fu, HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. Em: Proceedings of the 5th IEEE International Conference on Data Mining (ICDM), 2005, pp. 226–233;

30. J. Lin, E. Keogh, S. Lonardi, J. Lankford, D. Nystrom, VizTree: a Tool for Visually Mining and Monitoring Massive Time Series Databases. Em: Proceedings of the 30th International Conference in Very Large Data Bases, 2004, pp.1269-1272;

31. L. Olson, D. Delen, Advanced Data Mining Techniques, Springer, 1st edition (2008), pp. 138, ISBN 3540769161;

32. J.J. van Wijk, E.R. van Selow, Cluster and Calendar Based Visualization of Time Series Data. Em: Proceedings of IEEE Symposium on Information Visualization, 1999, pp. 4-9;

33. E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, C. A. Ratanamahatana: The UCR Time Series Classification/Clustering Homepage (2011): www.cs.ucr.edu/~eamonn/time_series_data;

34. V. Barnett e T. Lewis: Outliers in Statistical Data. John Wiley & Sons, 3ª edição, 1994.