CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

CORRESPONDÊNCIA DE DETECÇÕES A EVENTOS EM SÉRIES TEMPORAIS COM EFICIÊNCIA COMPUTACIONAL E GARANTIA DE OTIMALIDADE

Michel Siqueira Reis

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do grau de mestre.

Orientadores:
Rafaelli Coutinho e Eduardo Ogasawara

Rio de Janeiro,

29-01-2026

Correspondência de Detecções a Eventos em Séries Temporais com Eficiência Computacional e Garantia de Otimalidade

Dissertação de Mestrado em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

Michel Siqueira Reis

Aprovada por:

Presidente, Profa. Rafaelli Coutinho, D.Sc. (orientador)

Eduardo Ogasawara, D.Sc. (coorientador)

Laura Silva de Assis, D.Sc.

Rebecca Pontes Salles, D.Sc.

Rio de Janeiro,

29-01-2026

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

R375 Reis, Michel Siqueira
Correspondência de detecções a eventos em séries temporais com eficiência computacional e garantia de otimalidade / Michel Siqueira Reis. — 2026.
xiv, 68f. : il. ; enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2026.
Bibliografia: f. 64-68
Orientador: Rafaelli Coutinho
Coorientador: Eduardo Ogasawara

1. Análise de séries temporais. 2. Detecção de eventos. 3. Teoria dos grafos. 4. Algoritmos. I. Coutinho, Rafaelli. (Orient.). II. Ogasawara, Eduardo. (Coorient.). III. Título.

CDD 519.55

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

DEDICATÓRIA

Dedico este trabalho a todos que, guiados pela vontade de Jesus Cristo, empenham-se na construção de um mundo melhor.

AGRADECIMENTOS

Agradeço ao Espírito Santo de Deus por me guiar no ingresso neste mestrado, ampliar meu entendimento de maneira extraordinária e remover todos os obstáculos ao longo da jornada. Expresso minha profunda gratidão à minha mãe, que formou as primeiras bases do que sou, e à minha esposa, pelo constante incentivo e apoio. Sou igualmente grato aos meus orientadores, Rafaelli Coutinho e Eduardo Ogasawara, cuja orientação é fundamental em todas as etapas deste processo. Agradeço também aos membros do corpo docente do PPCIC pelo que me ensinaram, em especial aos que se dedicam a oferecer ensino de excelência. Por fim, reconheço e agradeço o apoio financeiro parcial do CNPq, da CAPES e da FAPERJ, que contribuem para a realização desta pesquisa.

RESUMO

Correspondência de Detecções a Eventos em Séries Temporais com Eficiência Computacional e Garantia de Otimalidade

Michel Siqueira Reis

Orientadores:
Rafaelli Coutinho e Eduardo Ogasawara

Resumo da Dissertação submetida ao Programa de Pós-graduação em Ciência da Computação do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ como parte dos requisitos necessários à obtenção do grau de mestre.

Este trabalho apresenta o SmartSoftED, uma métrica otimizada para avaliação da detecção de eventos pontuais em séries temporais. A métrica original, SoftED, introduz uma avaliação "suave" baseada em tolerância temporal, atribuindo pontuações graduais a detecções próximas de eventos reais. No entanto, sua formulação atual depende de uma abordagem gulosa que não garante otimalidade em todos os casos e exige um custo computacional quadrático, o que limita sua aplicação em ambientes de larga escala ou de processamento em tempo real. O SmartSoftED supera essas limitações ao introduzir uma estratégia que decompõe o problema em subproblemas disjuntos tratáveis: alguns podem ser resolvidos de forma eficiente e sem perda de otimalidade, enquanto outros são modelados como problemas de emparelhamento máximo ponderado em grafos bipartidos desbalanceados. Essa abordagem preserva a otimalidade das correspondências entre detecções e eventos e reduz significativamente o custo computacional. Na prática, o método alcança uma aceleração média de duas ordens de grandeza, tornando-se adequado para determinadas aplicações de grande escala e para sistemas sujeitos a restrições temporais rigorosas.

Palavras-chave: Séries temporais; Detecção de eventos; Avaliação de métricas; SoftED; SmartSoftED; Eficiência computacional; Otimalidade; Emparelhamento em grafo bipartido; Algoritmo Húngaro.

Rio de Janeiro,

29-01-2026

ABSTRACT

Matching Detections to Events in Time Series with Computational Efficiency and Optimality Guarantees

Michel Siqueira Reis

Advisors:
Rafaelli Coutinho e Eduardo Ogasawara

Abstract of Dissertation submitted to the Graduate Program in Computer Science at the Federal Center for Technological Education Celso Suckow da Fonseca (CEFET/RJ), as partial fulfillment of the requirements for obtaining the degree of Master.

This work presents SmartSoftED, an optimized metric for evaluating the detection of point events in time series. The original metric, SoftED, introduces a "soft" evaluation based on temporal tolerance, assigning gradual scores to detections that occur near actual events. However, its current formulation relies on a greedy approach that does not guarantee optimality in all cases and incurs a quadratic computational cost, limiting its applicability in large-scale or real-time processing environments. SmartSoftED overcomes these limitations by introducing a strategy that decomposes the problem into manageable, disjoint subproblems: some can be solved efficiently without loss of optimality, while others are modeled as maximum-weighted matching problems on unbalanced bipartite graphs. This approach preserves the optimality of correspondences between detections and events while significantly reducing computational cost. In practice, the method achieves an average speedup of two orders of magnitude, making it suitable for certain large-scale applications and systems with strict temporal constraints.

Key-words: Time series; Event detection; Evaluation metrics; SoftED; SmartSoftED; Computational efficiency; Optimality; Bipartite graph matching; Hungarian Algorithm.

Rio de Janeiro,

29-01-2026

Sumário

I Introdução **1**

II Referencial **4**

II.1 Detecção de Eventos em Séries Temporais 4

II.2 Métricas para avaliação da detecção de eventos em séries temporais 6

II.3 Métricas SoftED 8

II.4 Teoria dos Grafos 11

II.5 Otimização Combinatória 13

III Trabalhos Relacionados **15**

III.1 Métricas de avaliação de eventos na forma intervalar 16

III.1.1 *Composite F-Score* 17

III.1.2 Protocolo PA%K 18

III.1.3 *Balanced Adjustment* (BA) 19

III.1.4 *Precision and Recall for Time Series* 21

III.1.5 *Time-Series Aware Metrics* 23

III.1.6 *Enhanced Time-Series Aware Metrics* 25

III.1.7 Revisão das métricas intervalares 28

III.2 Métricas de avaliação de eventos na forma pontual 29

III.2.1 *The Numenta Anomaly Benchmark* (NAB) 30

III.2.2 *SoftED: Metrics for Soft Evaluation of Time Series Event Detection* 32

III.2.3 Revisão das métricas pontuais 32

IV Modelagem e Formulação do Problema de Associação entre Eventos e Detecções **35**

IV.1 Formulação do problema 35

IV.2 Algoritmo Húngaro aplicado ao problema de associações entre eventos e detecções 36

V Método **40**

V.1 Função de pertinência 40

V.2 SoftEDPar (solução ótima para associação de detecções e eventos) 41

V.3 SmartSoftED 42
V.3.1 Método principal 43
V.3.2 Análise de Complexidade Computacional 47
V.3.3 Estudo de caso do método 49

VI Avaliação experimental 53
VI.1 Experimento para verificação de manutenção de otimalidade 55
VI.2 Eficiência computacional 56
VI.2.1 Comportamento empírico da complexidade computacional nos conjuntos de dados 58
VI.2.2 Limites computacionais do método 60

VII Conclusão 62

Referências 64

Lista de Figuras

II.1 Taxonomia de eventos em séries temporais. Fonte: Adaptado de Ogasawara et al. [2025]. 5

II.2 Comparação das métricas: F1 (tradicional) e PA^{f1} (F1 ponto-ajustada). Fonte: [Sørbø and Ruocco, 2023]. 7

II.3 (a) Pertencimento a conjuntos clássicos (*crisp sets*) e (b) pertencimento a conjuntos difusos (*fuzzy sets*). Fonte: Adaptado de [Tahera et al., 2008]. 8

II.4 Comparação entre avaliação rígida e avaliação flexível da detecção de eventos [Salles et al., 2024]. 9

II.5 Como associar detecções a eventos? Adaptado de [Salles et al., 2024]. 10

II.6 Associações entre eventos e detecções. Fonte: [Reis et al., 2024]. 11

II.7 Representação da situação mencionada na Figura II.6 como um grafo. 13

III.1 Precisão, revocação e F1 de acordo com os cálculos tradicional, ponto-ajustado e segmentado. Adaptado de [Tatbul et al., 2018]. 16

III.2 F1 da métrica PA%K em função de diferentes valores de K, comparando um modelo robusto (*trained*) com um modelo fraco (*random*). Fonte: Adaptado de [Siwon et al., 2022]. 19

III.3 Visão comparativa do F1 das métricas: tradicional (F_1), ponto-ajustada (F_{1PA}), PA%K (F_{1KPA}) e BA (F_{1BA}). Fonte: Adaptado de [Bhattacharya et al., 2025]. 20

III.4 Sobreposição entre as anomalias reais (R) e as predições (P). Fonte: [Tatbul et al., 2018]. 21

III.5 Exemplo das funções de viés de posição $\delta()$ e da função de tamanho $\omega()$. Fonte: [Tatbul et al., 2018]. 22

III.6 Instâncias ambíguas e detecções parciais. Fonte: [Hwang et al., 2019]. 24

III.7 Cálculo do *Overlap Score* para anomalia a e predição p. Fonte: [Hwang et al., 2019]. 24

III.8 Exemplo de quatro possíveis casos de detecção. Fonte: [Hwang et al., 2022]. 26

III.9 Função de avaliação NAB. Fonte: [Lavin and Ahmad, 2015]. 30

IV.1 Emparelhamento em grafo bipartido completo com a representação da matriz de custo equivalente. 38

IV.2 Exemplificação das etapas do Algoritmo Húngaro. 39

V.1 Fluxograma com as etapas do método. 43
V.2 Exemplo de eventos e detecções em série sintética. 49

VI.1 Tamanho médio das séries dos conjuntos de dados. 54
VI.2 Ilustração da densidade que a soma de eventos e detecções ocupa em cada conjunto de dados, em relação ao tamanho das séries. 54
VI.3 Gráfico de coordenadas paralelas ilustrando os valores de precisão e revocação obtidos pelos métodos SoftED, SoftEDPar e SmartSoftED em cada um dos 3.071 testes experimentais realizados. 56
VI.4 Comparação do tempo de execução relativo das métricas SoftED, SoftEDPar e SmartSoftED nos diferentes conjuntos de dados. 57
VI.5 Proporção de eventos e detecções tratadas como casos simples, médios ou complexos em cada conjunto de dados. 58
VI.6 Proporção de comportamento computacional do método em cada conjunto de dados. 59
VI.7 *Heatmap* de tempo de execução da métrica em função do tamanho da série e da densidade de eventos/detecções (limite de 60 segundos). 60
VI.8 *Heatmap* de tempo de execução da métrica em função do tamanho da série e da densidade de eventos/detecções (limite de 3 segundos). 61

Lista de Tabelas

II.1 Métricas tradicionais derivadas da matriz de confusão. 7

II.2 Formalização das métricas SoftED. 10

III.1 Comparativo entre métricas intervalares considerando apenas a complexidade assintótica. *Composite F-Score* se destaca por sua granularidade híbrida. 29

III.2 Comparativo entre métricas pontuais, considerando a tolerância temporal e a complexidade assintótica. 33

V.1 Casos de associação entre eventos e detecções. 50

V.2 Passo a passo da execução do algoritmo húngaro para o caso 6. 51

V.3 Métricas calculadas SmartSoftED. 52

Lista de Acrônimos

$F1_C$	*Composite F1-Score*.
AUC-ROC	*Area Under the Receiver Operating Characteristic Curve*.
BA	*Balanced Adjustment*.
ECG	eletrocardiograma.
ETL	*Extract, Transform, Load*.
FN	Falso Negativo.
FP	Falso Positivo.
PA	*Point-Adjusted*.
TN	Verdadeiro Negativo.
TP	Verdadeiro Positivo.
TSAD	*Time Series Anomaly Detection*.
UniTED	*Unified Time Series Event Detection Repository*.

Capítulo I Introdução

Uma série temporal é uma sequência de observações registradas ao longo do tempo, permitindo a análise do comportamento de uma variável a partir de seus valores passados [Hanssens et al., 2003]. Essas séries podem apresentar mudanças significativas em determinados momentos, conhecidas como eventos [Guralnik and Srivastava, 1999]. A identificação desses eventos é fundamental, pois correspondem a fenômenos específicos em um contexto e fornecem informações críticas para a interpretação e a tomada de decisão. A detecção de eventos é realizada por meio de técnicas aplicadas ao estudo de séries temporais [Pimentel et al., 2014].

Quando um evento é rotulado em uma série temporal, sua detecção pode ser tratada como uma tarefa de classificação e, portanto, avaliada por métricas como acurácia, precisão, revocação e F1 [Lavin and Ahmad, 2015; Tatbul et al., 2018]. Essas métricas consideram qualquer desvio entre a detecção e o evento como erro absoluto e, por isso, são denominadas métricas rígidas neste estudo. Contudo, diferentemente de problemas clássicos de classificação, a avaliação de detecção de eventos exige maior flexibilidade, uma vez que identificar um evento com pequena margem de diferença temporal ainda pode ser útil em diversas aplicações. Assim, métricas rígidas não são suficientes para capturar a qualidade real dos métodos de detecção [Xu et al., 2018; Salles et al., 2024; Sørbø and Ruocco, 2023].

Como os métodos de detecção em séries temporais raramente produzem previsões perfeitas, uma boa métrica deve discernir as melhores previsões não perfeitas. Para suprir essa limitação, diversos métodos de avaliação foram propostos, cada um composto por um conjunto de métricas [Sørbø and Ruocco, 2023]. Entre esses trabalhos, destaca-se o SoftED [Salles et al., 2024], resultado de pesquisa anterior da qual o autor participou e que serve de base para o presente trabalho. Essa abordagem introduz métricas que incorporam tolerância temporal como mecanismo de suavização, permitindo avaliar detecções não perfeitas de forma mais adequada. A implementação dessa abordagem ocorre no *framework* Harbinger [Salles et al., 2020].

A metodologia SoftED exige uma etapa explícita de associação entre detecções e eventos, indispensável para o cálculo das quantidades de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos. Na implementação original, essa associação é conduzida por meio de uma estratégia gulosa, na qual cada detecção é associada, de forma sequencial, ao evento que maximiza sua pontuação segundo uma função de pertinência. Para esse fim, todos os eventos candidatos

são avaliados para cada detecção, o que acarreta, no pior caso, um custo computacional quadrático quando o número de detecções é proporcional ao número de eventos [Salles et al., 2024].

Essa abordagem não garante a otimização global das associações, uma vez que se baseia em decisões locais tomadas de forma sequencial, o que pode impedir a formação do conjunto de pares detecção–evento que maximiza a pontuação total atribuída às detecções. Além disso, a complexidade computacional quadrática da implementação limita sua aplicação em séries temporais de grande porte ou em cenários sujeitos a restrições rigorosas de tempo de execução.

A ideia central deste trabalho é reformular o processo de correspondência entre detecções e eventos da métrica SoftED como um problema de otimização. Essa reformulação posiciona o problema no contexto da otimização combinatória, permitindo explorar técnicas clássicas de emparelhamento em grafos bipartidos para a obtenção de soluções globalmente ótimas.

A estratégia proposta baseia-se em modelar apenas os trechos do problema em que há concorrência entre detecções e eventos como um emparelhamento máximo ponderado em grafos bipartidos completos desbalanceados. Diferentemente da abordagem gulosa original, essa reformulação permite introduzir garantias de otimalidade na associação entre detecções e eventos, eliminando decisões locais que podem comprometer a qualidade global da avaliação.

A partir dessa reformulação, o trabalho propõe o método híbrido SmartSoftED, que explora a estrutura do problema para decompor a correspondência em subproblemas disjuntos, resolvendo apenas os casos necessários por meio de um algoritmo de otimização, enquanto preserva soluções mais simples para os demais casos. Essa estratégia permite equilibrar qualidade das associações e custo computacional, tornando viável sua aplicação em cenários de maior escala ou de processamento em tempo real.

O método é implementado e validado experimentalmente, sendo comparado às soluções anteriores da mesma técnica (SoftED [Salles et al., 2024] e SoftEDPAR [Reis et al., 2024]) quanto à qualidade da avaliação e à eficiência de execução. Os resultados obtidos demonstram ganhos consistentes de desempenho e qualidade de avaliação, cujo alcance e limitações são analisados detalhadamente nos experimentos apresentados ao longo do trabalho.

Este trabalho está estruturado nos seguintes capítulos. O Capítulo II apresenta o referencial teórico necessário à compreensão do problema abordado, contemplando os conceitos fundamentais de séries temporais, detecção de eventos e as principais dificuldades associadas à sua avaliação. Nesse contexto, são revisadas as bases conceituais da métrica SoftED, com ênfase em suas limitações relacionadas à etapa de associação entre detecções e eventos. Em seguida, são introduzidos os fundamentos de teoria de grafos e de otimização combinatória, que permitem formalizar o problema de associação como um problema de atribuição, modelado como um emparelhamento máximo ponderado em um grafo bipartido completo desbalanceado, sustentando a abordagem proposta neste

trabalho.

O Capítulo III discute os principais trabalhos relacionados, organizando a literatura em métricas para eventos intervalares e para eventos pontuais. Essa revisão permite compreender as limitações das abordagens existentes e posicionar a proposta deste estudo no estado da arte. No Capítulo IV, apresenta-se a modelagem e a formulação do problema de associação entre eventos e detecções, estabelecendo sua definição formal e a aplicação do algoritmo húngaro ao contexto investigado. No Capítulo V, descreve-se a metodologia desenvolvida, desde a reformulação da associação entre eventos e detecções como um problema de otimização até a construção da abordagem híbrida SmartSoftED. São apresentados os princípios gerais do método, incluindo sua estrutura, baseada na decomposição do problema em subproblemas disjuntos, resolvidos com otimalidade e eficiência computacional. O Capítulo VI apresenta os experimentos conduzidos para avaliar a proposta, examinando tanto a qualidade da correspondência quanto seu desempenho computacional em diferentes cenários e conjuntos de dados. Por fim, o Capítulo VII sintetiza as contribuições do trabalho, discute suas limitações e sugere direções de pesquisa futuras.

Capítulo II Referencial

Este capítulo apresenta os conceitos fundamentais necessários à compreensão deste trabalho. Primeiro, descreve-se o que são séries temporais e como os eventos podem ser detectados e classificados. Na sequência, discutem-se as métricas de avaliação tradicionais e suas limitações quando aplicadas ao domínio das séries temporais, devido ao caráter sequencial e dependente do tempo dos dados. Em seguida, são apresentadas as métricas SoftED, que utilizam lógica difusa para incorporar tolerância temporal na associação entre detecções e eventos, resultando em um processo de avaliação menos rígido. Por fim, são definidos os conceitos fundamentais de teoria de grafos e de otimização combinatória, os quais fornecem o embasamento necessário para modelar a associação como um problema de emparelhamento em um grafo bipartido completo.

II.1 Detecção de Eventos em Séries Temporais

Uma série temporal X é uma sequência de n observações, $< x_1, x_2, x_3, \ldots, x_n >$ [Esling and Agon, 2012]. Cada observação específica da série temporal é representada por x_t, indexada no tempo por t ($t \in \{1, \ldots, n\}$). O comprimento de uma série temporal X é denotado por $|X|$, sendo $|X| = n$. As sequências que compõem uma série temporal interpretam-se como o processo da série temporal [Shumway and Stoffer, 2025], pois ela é estudada em função de seus dados passados [Hanssens et al., 2003].

Em muitos casos, essas séries apresentam mudanças significativas em determinados momentos, conhecidas como eventos [Guralnik and Srivastava, 1999]. Com o intuito de mapear os fundamentos associados à detecção de eventos, Ogasawara et al. [2025] desenvolveram uma taxonomia, ilustrada na Figura II.1. Nela, os eventos são classificados em cinco dimensões analíticas: tipos de eventos, estruturas de dados, métodos de detecção, cenários de aplicação e métricas de avaliação.

Os tipos de eventos podem ser agrupados em três categorias principais: anomalias, pontos de mudança e *motifs*. Anomalias correspondem a observações que se desviam significativamente do padrão; pontos de mudança refletem alterações relevantes na tendência ou na volatilidade da série temporal; e *motifs* são padrões recorrentes e aproximados observados na série [Ogasawara et al., 2025].

Quanto à estrutura de dados, os eventos podem ser definidos em relação à sua granularidade

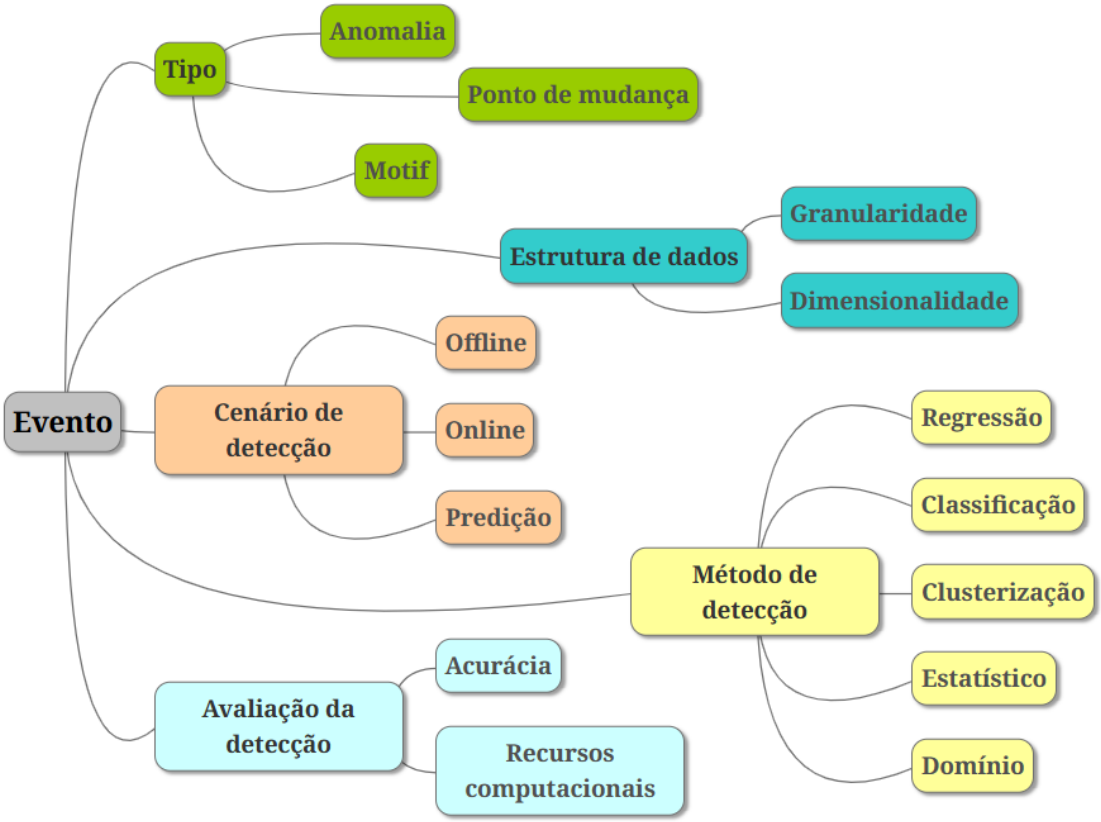


Figura II.1: Taxonomia de eventos em séries temporais. Fonte: Adaptado de Ogasawara et al. [2025].

e à sua dimensionalidade. A granularidade de um evento pode ser pontual, isto é, observações individuais que ocorrem em algum momento, ou intervalar, isto é, um conjunto de observações que ocupa um intervalo [Ariyaluran Habeeb et al., 2019]. Já a dimensionalidade indica quantas variáveis são medidas ao longo do tempo: séries univariadas medem uma variável; multivariadas, mais de uma [Ogasawara et al., 2025].

Quanto aos cenários de detecção de eventos, eles podem ser classificados em três tipos: *offline*, *online* e preditivo. A detecção *offline* ocorre após a coleta completa (ou parcial) da série temporal, permitindo uma análise retrospectiva com maior disponibilidade de recursos e menor restrição de tempo. A detecção *online* é realizada em fluxo contínuo, enquanto os dados são gerados, exigindo baixa latência e atualização frequente dos modelos para respostas imediatas. Já a detecção preditiva busca antecipar eventos futuros com base em padrões históricos, utilizando técnicas estatísticas, aprendizado de máquina ou redes neurais profundas [Ogasawara et al., 2025; Ahmad et al., 2017].

Os métodos de detecção de eventos encontrados na literatura agrupam-se em cinco categorias principais: regressão, classificação, agrupamento (*clustering*), métodos estatísticos e métodos baseados em conhecimento de domínio [Ogasawara et al., 2025]. A detecção de eventos em séries temporais é um campo em evolução, com diversos métodos disponíveis, o que torna desafiador determinar a abordagem adequada para cada domínio [Sørbø and Ruocco, 2023; Zhao, 2021].

No que se refere às técnicas de avaliação desses métodos, diversos estudos dedicam-se à análise, comparação e revisão sistemática das abordagens existentes, evidenciando a complexidade desse processo [Olteanu et al., 2023; Zhao, 2021]. Ainda não há consenso na literatura quanto às métricas mais adequadas para diferentes contextos e domínios de aplicação, e as métricas disponíveis apresentam propriedades e pressupostos distintos. Ademais, mesmo as métricas mais adotadas têm sido objeto de críticas recentes quanto à sua capacidade de avaliar os métodos de detecção [Sørbø and Ruocco, 2023].

A avaliação das detecções de eventos em séries temporais envolve medir a precisão e a eficácia na identificação de eventos com base em dados de referência. Essa tarefa é desafiadora devido à subjetividade na rotulagem dos eventos e à escassez de rótulos confiáveis em muitas aplicações. Além disso, a forma como os eventos são representados (pontuações ou rótulos) influencia diretamente a análise. Existem métodos de detecção em que cada ponto da série recebe uma pontuação de propensão, isto é, uma estimativa da probabilidade de que aquele ponto represente um evento. Em outros métodos, cada ponto é classificado com um rótulo que o classifica como típico ou anômalo [Sørbø and Ruocco, 2023; Ogasawara et al., 2025].

Apesar desses desafios, conjuntos de dados de séries temporais rotulados, com a indicação de eventos, costumam representá-los de forma binária: cada ponto temporal da série é classificado como pertencente a um evento (positivo) ou a um não-evento (negativo). Neste trabalho, considera-se exclusivamente esse tipo de representação binária.

II.2 Métricas para avaliação da detecção de eventos em séries temporais

Sob a premissa de uma representação binária, o problema de detecção de eventos em séries temporais reinterpreta-se como uma tarefa de classificação binária. Essa equivalência, por sua vez, conduz naturalmente à adoção de métricas de desempenho consagradas na literatura estatística e computacional. Tais métricas são então utilizadas para avaliar a qualidade das predições do modelo.

A matriz de confusão é a base da avaliação de classificadores binários, resultando da evolução de conceitos oriundos da estatística clássica. Sua origem remonta aos trabalhos de Karl Pearson, no início do século XX, sobre tabelas de contingência 2×2, que permitiram a estruturação sistemática de dados categóricos [Brooks et al., 2024].

Posteriormente, a distinção entre diferentes tipos de erro é formalizada pela teoria da decisão estatística, o que contribui para o desenvolvimento das categorias de falso positivo e falso negativo. Com o tempo, essas estruturas dão origem à categorização dos resultados classificatórios em quatro componentes fundamentais: verdadeiro positivo (TP), falso positivo (FP), verdadeiro negativo (TN) e falso negativo (FN) [Brooks et al., 2024].

Atualmente, diversas métricas derivam desses quatro componentes da matriz de confusão, sendo

acurácia, precisão, revocação, especificidade e F1 as mais utilizadas, cujas fórmulas podem ser observadas na Tabela II.1.

Acurácia	**Precisão**	**Revocação**	**Especificidade**	**F_1-Score**
$\frac{TP+TN}{TP+TN+FP+FN}$	$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$	$\frac{TN}{TN+FP}$	$2 \times \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}}$

Tabela II.1: Métricas tradicionais derivadas da matriz de confusão.

O principal desafio ao aplicar métricas tradicionais na avaliação de detecções de eventos em séries temporais é que elas não consideram a dimensão temporal inerente ao problema. Em casos de eventos pontuais, essas métricas desconsideram a proximidade temporal entre as detecções, tratando-as de forma binária, sem reconhecer o valor de uma detecção próxima ao evento real. Já em eventos intervalares, elas ignoram a própria extensão temporal dos eventos, avaliando apenas pontos individuais e, assim, perdendo a noção de duração e de sobreposição entre detecções e eventos.

Em resposta a essa limitação, uma das primeiras tentativas de lidar com a avaliação de eventos intervalares adotou uma formulação deliberadamente simplificada. Como exemplo, considera-se a métrica empregada por Xu et al. [2018], denominada *Point-Adjusted* (PA) (descrita em maior detalhe no Capítulo III). Nessa métrica, um evento intervalar é considerado completamente detectado desde que ao menos um de seus pontos seja identificado, independentemente da extensão temporal efetivamente coberta pela detecção. Embora tal formulação tenha o mérito de contornar, de forma imediata, a dificuldade associada à avaliação de eventos intervalares, ela tende a superestimar o desempenho dos detectores ao desconsiderar a qualidade temporal da associação.

Na Figura II.2, as esferas pretas representam instantes da série temporal sem ocorrência de eventos ou detecções, enquanto as esferas vermelhas indicam os pontos em que há eventos (na linha de rótulos) ou detecções (na linha de predições).

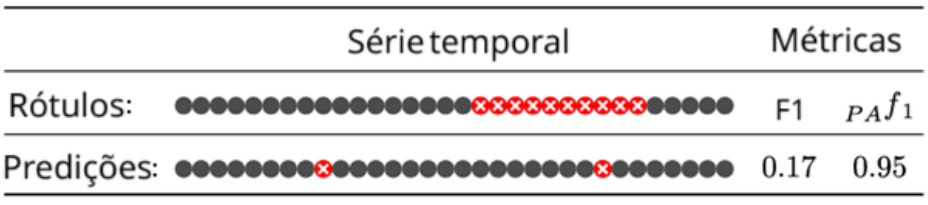


Figura II.2: Comparação das métricas: F1 (tradicional) e PA^{f1} (F1 ponto-ajustada). Fonte: [Sørbø and Ruocco, 2023].

Em resposta a essas limitações, a literatura científica apresenta métricas mais sofisticadas para a avaliação de detecções em séries temporais, voltadas a mensurar com maior rigor a precisão e a revocação. Uma revisão detalhada dessas métricas encontra-se no Capítulo III. Contudo, por serem objeto central deste estudo, as métricas SoftED [Salles et al., 2024] são apresentadas a seguir.

II.3 Métricas SoftED

As métricas SoftED utilizam princípios da computação flexível (*soft computing*) para incorporar tolerância temporal em sua avaliação. Diferentemente da computação rígida, que exige soluções exatas e determinísticas, a computação flexível admite aproximações, incertezas e graus de pertencimento, o que favorece a construção de sistemas mais adaptativos. Entre seus pilares fundamentais está a lógica difusa (*fuzzy logic*) [Tettamanzi and Tomassini, 2013]. Proposta por Lotfi Zadeh em 1965, a lógica difusa representa um avanço em relação à lógica clássica, ao permitir que a pertinência de um elemento a um conjunto seja expressa por valores contínuos entre 0 e 1, e não apenas pelos valores binários 0 ou 1 característicos de conjuntos rígidos (*crisp sets*). Essa abordagem permite modelar situações de incerteza, atribuindo diferentes graus de pertencimento em vez de uma classificação absoluta, refletindo de forma mais realista fenômenos complexos [Zadeh, 1965; Tettamanzi and Tomassini, 2013]. Esse conceito pode ser observado na Figura II.3.

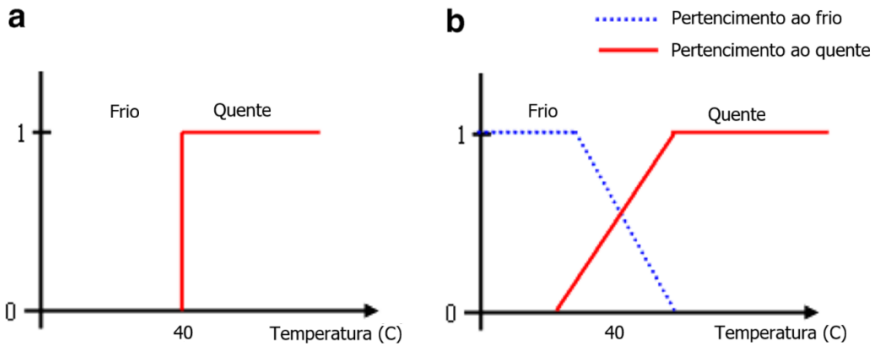


Figura II.3: (a) Pertencimento a conjuntos clássicos (*crisp sets*) e (b) pertencimento a conjuntos difusos (*fuzzy sets*). Fonte: Adaptado de [Tahera et al., 2008].

Nesse contexto, as métricas SoftED aplicam a lógica difusa como mecanismo para estender a avaliação binária tradicional, transformando-a em um processo graduado que incorpora a noção de tolerância temporal. A Figura II.4 ilustra essa aplicação: os losangos azuis representam os eventos; os círculos verdes, os eventos perfeitamente detectados (evento e detecção simultâneos); e os círculos vermelhos, detecções não perfeitas. Enquanto as métricas tradicionais avaliam as detecções de forma rígida, atribuindo apenas 0 ou 1, a abordagem flexível permite valores contínuos entre 0 e 1, ponderando o grau de pertencimento de cada detecção a um evento específico [Salles et al., 2024].

O grau de relevância de uma detecção em relação a um evento específico é determinado por uma função de pertinência triangular, utilizada na lógica difusa para modelar transições graduais e incertezas [Liu and Liu, 2001; Wu and Lee, 2007]. Considere d como o índice de uma detecção em uma série temporal, e o índice de um evento e k como um valor que representa a tolerância temporal da avaliação. A função triangular apresentada na Equação II.1 atribui uma pontuação

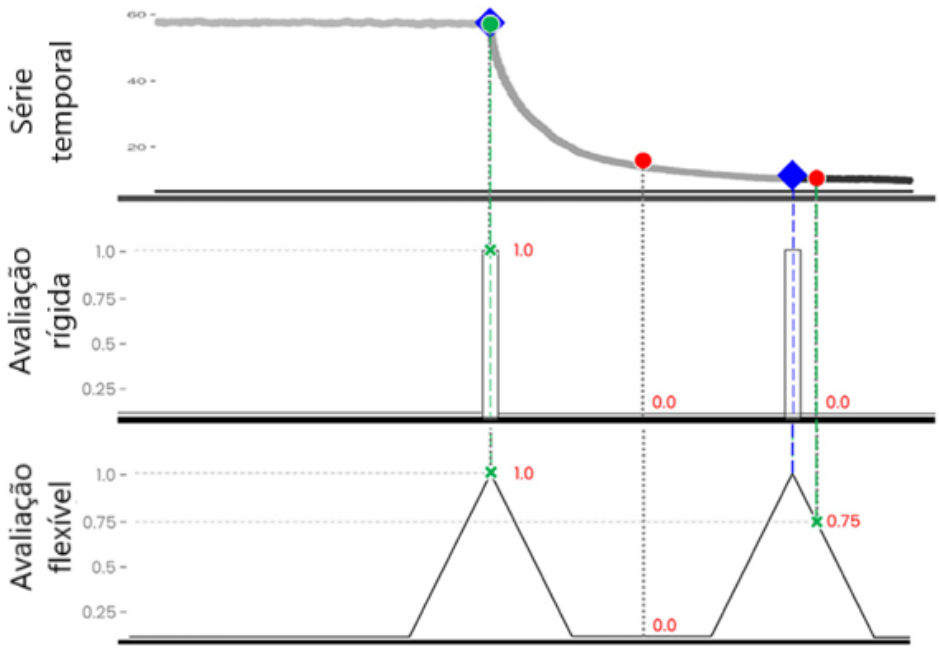


Figura II.4: Comparação entre avaliação rígida e avaliação flexível da detecção de eventos [Salles et al., 2024].

que quantifica a proximidade temporal entre uma detecção e um evento [Salles et al., 2024].

$$\mu(d, e) = \max\left(\min\left(\frac{d-(e-k)}{k}, \frac{(e+k)-d}{k}\right), 0\right) \tag{II.1}$$

Observa-se que a função assume valor máximo igual a 1 quando $d = e$, decrescendo linearmente à medida que a distância temporal $|d - e|$ aumenta, até atingir valor zero nos limites do intervalo $[e-k,\ e+k]$. Em vez de atribuir valores binários (0 ou 1) às detecções, a lógica difusa permite uma avaliação mais flexível e adaptativa no contexto da detecção de eventos.

Contudo, a introdução do parâmetro de tolerância temporal acarreta novos desafios. Na ausência da tolerância, um evento vincula-se apenas à detecção coincidente no mesmo instante. Ao incorporá-la, um evento pode associar-se a todas as detecções dentro de seu intervalo temporal, o que gera concorrência entre possíveis associações. Nesse cenário, torna-se necessário garantir regra de exclusividade, de modo que cada evento se associe a apenas uma detecção, e vice-versa.

A Figura II.5 ilustra três situações típicas de concorrência entre associações: (a) uma mesma detecção é associada a múltiplos eventos, (b) um único evento é vinculado a várias detecções ou (c) múltiplas detecções competem por múltiplos eventos.

Para preservar a integridade em relação às métricas tradicionais, adota-se uma estratégia gulosa: cada detecção associa-se ao evento que lhe confere a maior pontuação segundo as funções de pertinência. Após essa atribuição inicial, os eventos e detecções remanescentes passam pelo mesmo processo iterativamente, até que não restem associações possíveis. Na Figura II.5, em (a), a detec-

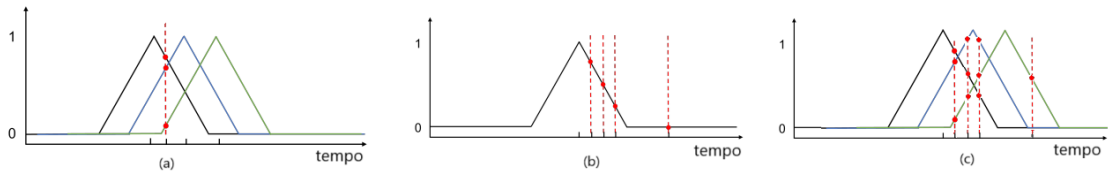


Figura II.5: Como associar detecções a eventos? Adaptado de [Salles et al., 2024].

ção vincula-se ao evento que a avalia melhor; em (b), a detecção mais bem avaliada associa-se ao único evento disponível; e em (c), as detecções mais bem pontuadas são atribuídas sucessivamente aos eventos que as avaliam de forma mais favorável. As detecções sem correspondência recebem pontuação 0.

A métrica SoftED mantém consistência com as avaliações rígidas e preserva a pontuação máxima de 1. Para isso, estabelece duas regras: (i) cada detecção recebe apenas uma pontuação, e (ii) a soma das pontuações atribuídas a um evento não pode exceder 1. Assim, mesmo que uma detecção esteja próxima de diversos eventos e gere múltiplas avaliações de pertinência, apenas o maior valor é considerado, garantindo que o total não ultrapasse o limite máximo estabelecido.

Após a obtenção da pontuação de cada detecção, cria-se um vetor com as pontuações de todas as detecções (ds). Esse vetor é utilizado para criar as versões flexíveis de TP, FP, TN e FN, a fim de calcular as demais métricas flexíveis (acurácia, revocação, precisão, etc.). Na Tabela II.2, $|X|$ denota o comprimento total da série temporal e m representa o número total de eventos na série temporal [Salles et al., 2024].

$$\mathrm{TP}_s = \sum_{i=1}^{n} ds_i \qquad \mathrm{FN}_s = m - \mathrm{TP}_s$$

$$\mathrm{FP}_s = \sum_{i=1}^{n} (1 - ds_i) \quad \mathrm{TN}_s = (|X| - m) - \mathrm{FP}_s$$

Tabela II.2: Formalização das métricas SoftED.

Depois disso, os valores de TP, FP, TN e FN são utilizados para calcular as métricas de acurácia, precisão, revocação e outras, seguindo as mesmas fórmulas das métricas tradicionais (Tabela II.1).

O processo guloso de associação entre eventos e detecções resolve o problema da associação, mas não de forma ótima. Em casos raros, como o ilustrado na situação (c) da Figura II.5, pode haver múltiplas detecções em regiões densamente povoadas por eventos, resultando em sobreposição das funções de pertinência. Nesses cenários, a abordagem gulosa pode deixar de estabelecer as melhores correspondências entre detecções e eventos.

Observe o funcionamento do mecanismo de associação gulosa na situação ilustrada na Figura II.6. Nessa abordagem, cada detecção é associada de forma independente ao evento que maximiza sua pontuação individual, caracterizando uma estratégia de otimização local. Assim, a detecção 1 vincula-se ao evento 2 (formando o par detecção–evento $[(1, 2)]$), por ser este o evento que lhe atribui a maior pontuação. Como consequência dessa decisão local, o evento 1 permanece sem detecção

associada e a detecção 2 sem evento correspondente. Ao final, as detecções 1 e 2 recebem, respectivamente, as pontuações 0,8 e 0,0. Essa estratégia, entretanto, não conduz à solução globalmente ótima, uma vez que os pares $[(1,1),(2,2)]$ resultariam em uma pontuação total de $0{,}6 + 0{,}4 = 1{,}0$, maximizando a soma das pontuações por meio de associações conjuntas mais adequadas.

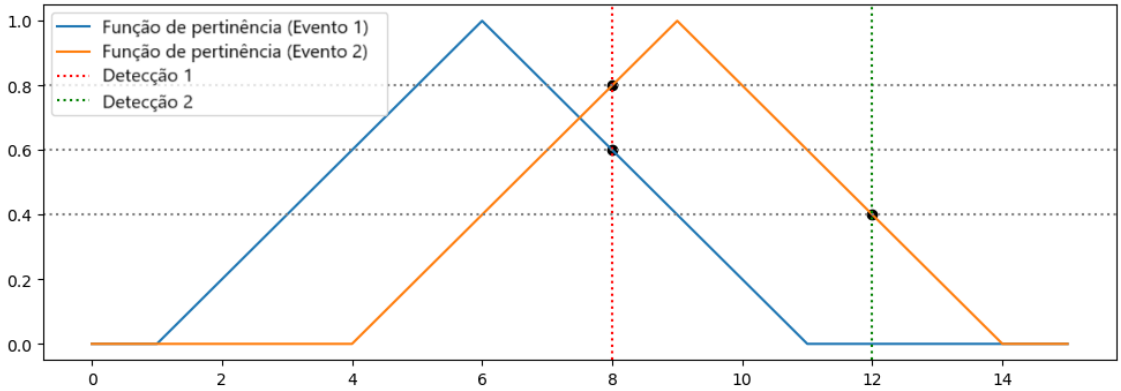


Figura II.6: Associações entre eventos e detecções. Fonte: [Reis et al., 2024].

Essa limitação evidencia a necessidade de métodos que garantam associações ótimas, capazes de refletir de forma mais fiel a qualidade real da detecção. Na maioria dos cenários práticos, os detectores de eventos em séries temporais realizam inferências a partir dos dados observados, produzindo indícios da ocorrência de eventos sem acesso direto à sua causalidade temporal exata. Como consequência, não é possível determinar de forma inequívoca se uma detecção emitida por um detector corresponde a um evento que já ocorreu ou a um evento que ainda ocorrerá. Essa ambiguidade é inerente ao problema.

Diante dessa incerteza, a busca por estabelecer uma associação globalmente ótima entre detecções e eventos mostra-se uma estratégia robusta. Nesse contexto, a maximização da soma das pontuações de associação surge como um critério natural, pois permite considerar simultaneamente todas as possíveis correspondências entre eventos e detecções, mitigando decisões subótimas decorrentes de otimizações locais. Para esse fim, o problema de associação entre eventos e detecções é reformulado no contexto da teoria dos grafos, modelando-o como um problema de emparelhamento.

II.4 Teoria dos Grafos

A Teoria dos Grafos é um ramo da matemática discreta dedicado ao estudo de estruturas denominadas *grafos*, amplamente utilizadas para modelar relações entre objetos em diversos domínios do conhecimento, como ciência da computação, engenharia, biologia, economia e ciências sociais. Desde sua introdução formal no século XVIII, com os trabalhos de Euler, os grafos tornaram-se uma ferramenta fundamental para a representação e resolução de problemas envolvendo conexões, dependências e interações entre entidades [Bollobás, 1979].

Formalmente, um grafo é definido como um par ordenado $G = (V, A)$, em que V é um conjunto finito e não vazio de vértices (ou nós) e $A \subseteq V \times V$ é um conjunto de arestas que representam relações entre pares de vértices. Cada aresta pode ser interpretada como uma ligação entre dois vértices, podendo ou não possuir direção, dependendo do tipo de grafo considerado [Bollobás, 1979].

Quando as arestas são pares ordenados, isto é, $(u, v) \neq (v, u)$, o grafo é denominado *dirigido*. Caso contrário, quando as arestas são pares não ordenados, o grafo é dito *não dirigido*. Neste trabalho, considera-se exclusivamente o caso de grafos não dirigidos, uma vez que a relação de correspondência entre os elementos modelados é simétrica [Bollobás, 1979].

Além disso, um grafo pode ser classificado como *simples* quando não possui laços (arestas que conectam um vértice a ele mesmo) nem arestas múltiplas entre o mesmo par de vértices. Grafos simples são frequentemente utilizados em problemas de modelagem abstrata, pois facilitam a análise estrutural e algorítmica [Bollobás, 1979].

Em muitos contextos, é conveniente associar valores numéricos às arestas de um grafo, representando custos, distâncias, pesos ou afinidades entre os vértices conectados. Nesse caso, o grafo é denominado *grafo ponderado*. Formalmente, define-se uma função de pesos $w : A \to \mathbb{R}$, que atribui a cada aresta um valor real. Grafos ponderados são particularmente relevantes em problemas de otimização, nos quais se busca minimizar ou maximizar a soma dos pesos das arestas selecionadas em determinada estrutura [Korte and Vygen, 2018].

Um tipo particular de grafo, denominado *grafo bipartido*, é aquele cujos vértices podem ser particionados em dois subconjuntos disjuntos U e W, de tal forma que cada aresta conecta um vértice de U a um vértice de W, e não existem arestas entre vértices de um mesmo subconjunto. Formalmente, diz-se que $G = (U \cup W, A)$, com $U \cap W = \emptyset$ e $A \subseteq U \times W$. Grafos bipartidos são utilizados para modelar problemas de atribuição ou associação entre dois conjuntos distintos de elementos [Bollobás, 1979].

No contexto dos grafos bipartidos, quando todas as possíveis conexões entre U e W estão presentes, o grafo é dito *completo*; caso contrário, é denominado *não-completo*. Quando os conjuntos U e W não têm a mesma cardinalidade, diz-se que o grafo é *desbalanceado*.

Um *emparelhamento* nesse grafo é um subconjunto de arestas $M \subseteq A$ tal que nenhum vértice pertence a mais de uma aresta em M, ou seja, cada vértice conecta-se, no máximo, a uma única aresta do emparelhamento. Se a cada aresta $(u, w) \in A$ associa-se um valor real $p(u, w) \in \mathbb{R}$, o grafo é dito *ponderado*, e o valor representa, por exemplo, a afinidade entre os vértices conectados. Um emparelhamento é *máximo* quando contém o maior número possível de arestas, e é *máximo ponderado* quando maximiza a soma dos pesos das arestas selecionadas, objetivo de diversos problemas de otimização combinatória [Korte and Vygen, 2018].

No contexto deste trabalho, os grafos são utilizados como ferramenta de modelagem para repre-

sentar o problema de associação entre detecções e eventos. O exemplo demonstrado na Figura II.6 pode ser representado pela Figura II.7, em que os vértices do conjunto à esquerda representam os eventos, os vértices do conjunto à direita representam as detecções, e as arestas entre os dois conjuntos representam os valores das possíveis associações evento-detecção (calculados pela função de pertinência correspondente).

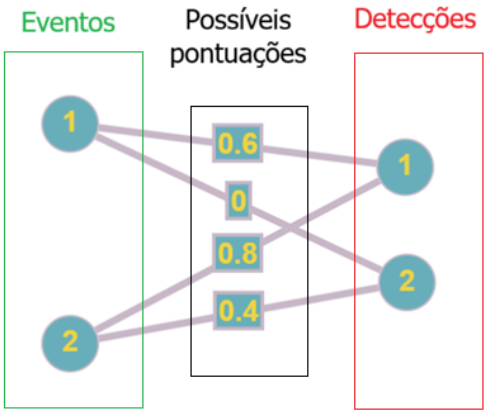


Figura II.7: Representação da situação mencionada na Figura II.6 como um grafo.

II.5 Otimização Combinatória

A otimização combinatória é uma área da matemática e da ciência da computação dedicada ao estudo de problemas nos quais se busca determinar uma solução ótima dentre um conjunto finito ou discretamente enumerável de alternativas viáveis. Diferentemente da otimização contínua, na qual as variáveis podem assumir valores em domínios contínuos, a otimização combinatória lida com estruturas discretas, frequentemente representadas por grafos, conjuntos ou permutações [Korte and Vygen, 2018].

De forma geral, um problema de otimização combinatória pode ser descrito por três componentes fundamentais: (*i*) um conjunto de soluções viáveis, definido por restrições combinatórias que limitam quais combinações de decisões são admissíveis, ou seja, respeitam o conjunto de restrições do problema; (*ii*) uma função objetivo, responsável por atribuir um valor escalar a cada solução viável; e (*iii*) um critério de otimalidade, que determina se o objetivo consiste em maximizar ou minimizar a função associada.

As restrições combinatórias introduzem dependências entre decisões individuais, de modo que a escolha de uma decisão inviabiliza outras, tornando o espaço de soluções altamente estruturado e não trivial. Entre essas restrições, destaca-se a restrição de exclusividade, que pode impor, por exemplo, que cada detecção e cada evento participem de, no máximo, uma associação. A função objetivo, por sua vez, agrega os conjuntos de contribuições associadas às decisões elementares do problema, atribuindo um valor global a cada solução viável. O critério de otimalidade consiste em selecionar,

dentre todas as associações viáveis, aquela que maximiza a soma global dessas contribuições.

No contexto deste trabalho, o problema de correspondência entre detecções e eventos é interpretado como um problema de otimização combinatória, no qual se busca determinar um conjunto ótimo de associações sob restrições de exclusividade.

Capítulo III Trabalhos Relacionados

A avaliação de detecção de anomalias em séries temporais é amplamente discutida na literatura e envolve métricas projetadas para lidar com eventos representados como pontos no tempo ou como intervalos. Para construir um panorama consistente desse campo, adotou-se a estratégia de *snowballing* a partir do estudo de revisão de Sørbø and Ruocco [2023], utilizado como fonte primária. A partir dele, exploraram-se tanto as referências citadas (*backward snowballing*) quanto os trabalhos que as citam (*forward snowballing*), o que permitiu identificar propostas consolidadas e relevantes para o problema de avaliação de detecções em séries temporais.

Ao longo da revisão, apresenta-se uma análise dos mecanismos internos dos métodos e de suas complexidades computacionais, com ênfase nos custos assintóticos. Essa análise tem como objetivo principal compreender a diversidade de comportamentos computacionais das abordagens, evidenciando que as métricas de avaliação de detecções em séries temporais podem ser significativamente mais complexas do que as métricas tradicionais.

O escopo deste capítulo concentra-se exclusivamente na detecção de eventos ou anomalias em séries temporais (*Time Series Anomaly Detection* (TSAD)), não abrangendo domínios correlatos, como a classificação, a detecção de mudanças de regime ou a detecção de anomalias em contextos espaciais ou estáticos [Sørbø and Ruocco, 2023]. Conforme discutido em Ogasawara et al. [2025], anomalias são tratadas como um tipo particular de evento cuja granularidade pode variar entre pontual e intervalar, distinção que fundamenta a separação clássica entre métricas *point-based* e *interval-based*.

Seguindo essa taxonomia, apresentam-se, inicialmente, as métricas projetadas para eventos intervalares, a maioria delas discutindo como lidam com eventos dessa natureza e, ao final, enfatizando suas limitações na avaliação de eventos pontuais. Esse aspecto é essencial para contextualizar a necessidade de métricas mais precisas nesse cenário. A discussão das métricas intervalares serve para situar o estado da arte das metodologias de avaliação, revelando também onde a complexidade computacional da metodologia SoftED se posiciona em relação a elas.

Em seguida, discute-se o movimento da literatura em direção a métricas orientadas a eventos pontuais, incluindo abordagens que lidam explicitamente com o problema de associação entre detecções e eventos, elemento central deste trabalho. Dessa forma, a revisão das métricas para eventos pontuais situa o contexto específico em que as métricas SoftED se inserem, destacando a

necessidade de melhorar sua capacidade de associar detecções a eventos pontuais de forma ótima e computacionalmente eficiente.

O objetivo deste trabalho, portanto, é propor uma evolução da família SoftED, com especial atenção ao processo de associação em aplicações de grande escala e em sistemas com restrições temporais rigorosas.

III.1 Métricas de avaliação de eventos na forma intervalar

Para superar as limitações das métricas tradicionais, dois trabalhos constituem os primeiros registros acadêmicos de uma tentativa de avaliação diferenciada para a detecção de eventos de natureza intervalar. São elas a métrica de ponto-ajustada (PA) [Xu et al., 2018] e a métrica segmentada [Hundman et al., 2018]. Ambas as abordagens podem ser visualizadas de forma intuitiva na Figura III.1.

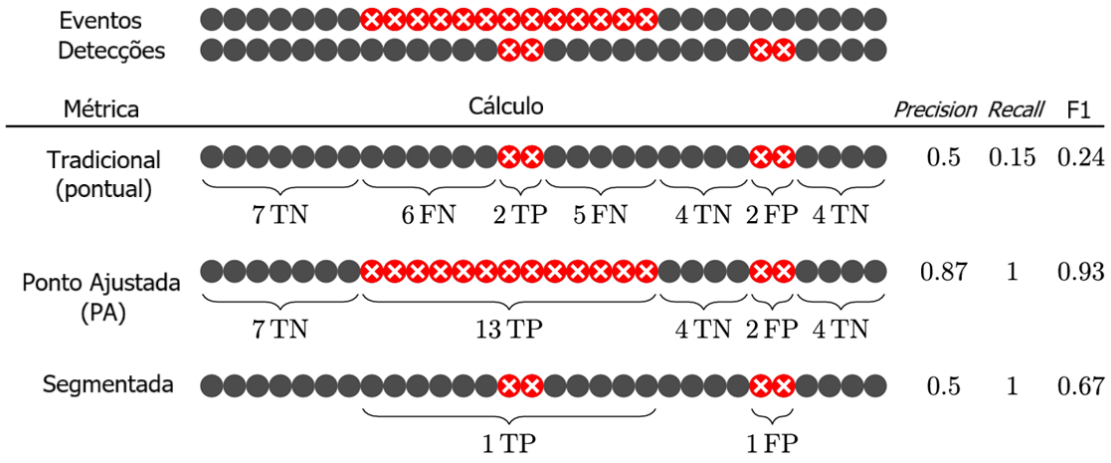


Figura III.1: Precisão, revocação e F1 de acordo com os cálculos tradicional, ponto-ajustado e segmentado. Adaptado de [Tatbul et al., 2018].

Na métrica PA, os autores apresentam um algoritmo de detecção não supervisionada de anomalias e, reconhecendo que as métricas rígidas tradicionais não capturam adequadamente a natureza temporal dos eventos, adotam a seguinte estratégia de avaliação: para cada intervalo de evento da série (positivo), sempre que ao menos um ponto de detecção ocorrer dentro desse intervalo (TP), todos os pontos de detecção correspondentes ao intervalo de evento são considerados como positivos. Assim, todo o evento é considerado perfeitamente detectado.

Após isso, aplica-se o cálculo tradicional de precisão, revocação e F1. Essa métrica tende a superestimar o desempenho dos detectores e, por isso, foi amplamente criticada na literatura, tornando-se referência crítica em trabalhos subsequentes [Audibert et al., 2020; Garg et al., 2022; Siwon et al., 2022; Doshi et al., 2022]. Apesar dessas limitações, ainda é utilizada em alguns estudos [Sørbø and Ruocco, 2023].

No trabalho de Hundman et al. [2018], os autores buscaram detectar anomalias na telemetria de espaçonaves utilizando redes LSTM e um método não paramétrico de limiar dinâmico para determinar as detecções. Para avaliá-las, adotam a métrica segmentada [Hundman et al., 2018], que considera cada intervalo de eventos ou detecções como unidade de análise, em vez de avaliar ponto a ponto. Após a detecção, os intervalos correspondentes são ligeiramente expandidos e fundidos para unir detecções próximas, e então comparados aos intervalos de eventos.

Na etapa de pontuação, registra-se um TP sempre que ao menos uma detecção sobrepõe um intervalo de evento. Múltiplas detecções do mesmo evento não aumentam essa contagem, de modo que cada intervalo contribui, no máximo, com um TP. Um FN ocorre quando um intervalo de evento não é coberto por nenhuma detecção, e um FP ocorre quando um intervalo de detecção não coincide com nenhum evento. Verdadeiros negativos (TNs) não são contabilizados. Após essa contagem intervalar, aplicam-se as definições tradicionais de precisão, revocação e F1. Tanto a métrica segmentada quanto a ponto-ajustada apresentam complexidade linear $O(N)$.

III.1.1 *Composite F-Score*

O trabalho de Garg et al. [2022] introduz o *Composite F1-Score* ($F1_C$) como métrica para avaliar a detecção de anomalias em séries temporais. Os autores afirmam que a métrica F1 tradicional mede apenas acertos em instantes isolados, sem considerar a natureza contínua dos eventos, o que leva a avaliação pessimista do desempenho. Já o $F1_{segmented}$ (F1 da métrica segmentada) consegue capturar a natureza contínua dos eventos, porém não é ideal para cálculos de precisão, pois atribui apenas um FP a uma sequência contínua de detecções incorretas.

O $F1_C$ é proposto como um meio-termo que integra as dimensões pontual e intervalar, reduzindo as limitações das abordagens isoladas. A precisão pontual ($Precision$) é extraída da métrica tradicional. A revocação intervalar ($Recall_{segmented}$) é extraída da métrica segmentada de Hundman et al. [2018]. O $F1_C$ é a média harmônica entre essas duas medidas, equilibrando ambas as perspectivas conforme demonstrado nas Equações III.1 e III.2.

$$Precision = \frac{TP}{TP + FP} \qquad Recall_{segmented} = \frac{TP_{segmented}}{TP_{segmented} + FN_{segmented}} \tag{III.1}$$

$$F1_C = \frac{2 \cdot Precision \cdot Recall_{segmented}}{Precision + Recall_{segmented}} \tag{III.2}$$

A revocação concentra-se em avaliar a capacidade do detector de identificar eventos na forma intervalar. Já a precisão preocupa-se em mensurar a proporção correta de sobreposição entre detecções e eventos. Em síntese, quando o objetivo é avaliar a capacidade de capturar eventos contínuos sem incorrer em superestimação do desempenho, o $F1_C$ se apresenta como uma formulação direta.

Apesar de estar contido nesta subseção, o $F1_C$ combina granularidades pontuais e intervalares e, portanto, é uma métrica híbrida. A implementação computacional da métrica apresenta complexidade $O(M + N)$, sendo N o tamanho da série temporal e M o número de pontos anômalos. Como $M \leq N$, a complexidade total pode ser expressa como $O(N)$.

III.1.2 Protocolo PA%K

Diversas abordagens foram propostas para contornar as limitações da métrica PA. Entre elas, Siwon et al. [2022] introduzem o protocolo PA%K. A ideia central do método é introduzir um critério adicional de cobertura. A expansão do PA sobre um intervalo anômalo só é aplicada se a proporção de pontos corretamente detectados nesse intervalo exceder um limiar K, com $K \in [0, 1]$ (por exemplo, $K = 0{,}5$ equivale a 50% de cobertura). Para $K = 0{,}5$, caso um intervalo de eventos tenha 10 pontos contínuos e ocorram 5 detecções pontuais nesse intervalo, considera-se todo o intervalo como detectado (10 TPs). Se ocorrerem apenas 4 detecções, apenas essas 4 são consideradas corretas (4 TPs).

Mais precisamente, a regra de decisão do PA%K modifica os rótulos de detecção d_t e produz novos rótulos $\hat{d}_t$ para cada intervalo anômalo S_m, conforme a Equação III.3.

$$\hat{d}_t = \begin{cases} 1, & \text{se } t \in S_m \text{ e } \dfrac{|\{t' \in S_m : d_{t'} = 1\}|}{|S_m|} > K, \\ \\ d_t, & \text{caso contrário,} \end{cases} \tag{III.3}$$

Aqui, S_m representa o m-ésimo intervalo anômalo, $d_t \in \{0, 1\}$ é a detecção original produzida pelo método no instante t, $|S_m|$ é o tamanho do intervalo, e $K \in [0, 1]$ é o limiar mínimo de cobertura exigido para que o PA%K expanda a detecção ao longo de todo o segmento. Assim, somente quando a fração de pontos detectados em S_m excede K, todos os instantes do intervalo são marcados como detecção. Nos demais casos, o valor original d_t é preservado.

Dessa forma, o PA%K funciona como uma generalização contínua entre a métrica PA e a métrica tradicional sem ajustes. Quando $K = 0$, o método se reduz ao PA clássico, bastando um ponto de detecção em um intervalo de evento para considerar o intervalo inteiro como detectado. Quando $K = 1$, somente intervalos totalmente cobertos geram pontuação máxima, reproduzindo o comportamento pontual das métricas tradicionais.

Essa parametrização permite gerar gráficos como o exemplificado pela Figura III.2. Nele, observa-se como os resultados das avaliações variam à medida que o limiar mínimo de cobertura dos eventos muda. Modelos robustos tendem a apresentar curvas de $\mathrm{F1}_{PA\%K}$ relativamente estáveis à medida que K cresce, enquanto modelos fracos exibem quedas acentuadas, muitas vezes com comportamento logarítmico.

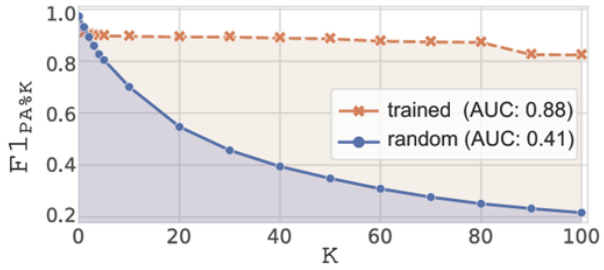


Figura III.2: F1 da métrica PA%K em função de diferentes valores de K, comparando um modelo robusto (*trained*) com um modelo fraco (*random*). Fonte: Adaptado de [Siwon et al., 2022].

Dessa forma, o PA%K reduz simultaneamente o risco de superestimação do PA e de subestimação do F1 tradicional, proporcionando uma visão mais realista da qualidade das detecções.

Os autores recomendam calcular a área sob a curva (AUC) do gráfico de $\mathrm{F1}_{PA\%K}$ ao variar K de 0 a 100 (tipicamente em incrementos de 10). Essa AUC sintetiza o comportamento global da métrica, por independer da escolha de um único limiar e por atuar como uma medida agregada de desempenho, oferecendo uma avaliação mais robusta da qualidade das detecções.

Do ponto de vista da complexidade computacional, a aplicação do PA%K a uma série de comprimento N envolve percorrer os intervalos anômalos uma única vez para computar a fração de detecções em cada S_m e, quando a cobertura excede K, propagar o rótulo ao longo do segmento. Como a soma dos comprimentos dos intervalos é limitada por N, o custo total dessa rerrotulação permanece $O(N)$.

Uma vez ajustados os rótulos $\hat{d}_t$, as contagens de TP, FP, TN e FN, bem como o cálculo do F1 associado a um valor fixo de K, podem ser obtidos em uma única passada adicional, sem alterar a ordem de grandeza do custo, que permanece linear. Caso se opte por calcular a AUC da curva $\mathrm{F1}_{PA\%K}$ variando K de 0 a 100 (tipicamente em incrementos de 10), repete-se esse procedimento 11 vezes, ou seja, $O(1)$. Sendo assim, a complexidade assintótica do método é $O(N)$.

III.1.3 *Balanced Adjustment* (BA)

A métrica *Balanced Adjustment* (BA), proposta por Bhattacharya et al. [2025], também busca mitigar o viés introduzido pela métrica PA, porém com uma estratégia de compensação. Em relação aos verdadeiros positivos (TPs), a BA se comporta como a PA: sempre que houver ao menos uma detecção dentro de um intervalo de anomalia S_m, todo o intervalo recebe crédito como detectado. A diferença está no tratamento dos falsos positivos (FPs). Para cada FP isolado, a BA define uma *ilha* temporal de intolerância ao seu redor, expandindo localmente as detecções nessa região e, assim, aumentando explicitamente a contagem de FPs.

Intuitivamente, se a métrica PA aumenta somente as detecções corretas, a métrica BA aumenta tanto as detecções corretas quanto as incorretas. A Figura III.3 apresenta uma visualização desse

mecanismo, onde as regiões em laranja destacam detecções mantidas como estão, enquanto as regiões em verde indicam instantes ajustados antes do cálculo do F1.

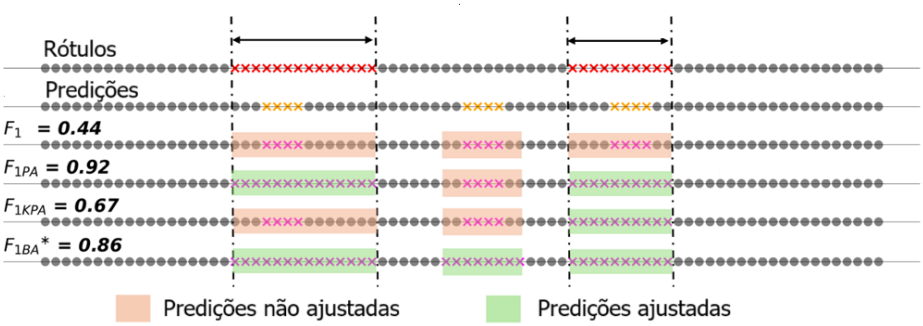


Figura III.3: Visão comparativa do F1 das métricas: tradicional (F_1), ponto-ajustada (F_{1PA}), PA%K (F_{1KPA}) e BA (F_{1BA}). Fonte: Adaptado de [Bhattacharya et al., 2025].

De forma mais precisa, seja $d_t \in \{0, 1\}$ a detecção original produzida pelo método no instante t e $y_t \in \{0, 1\}$ o rótulo verdadeiro. Para cada instante u em que ocorre um falso positivo ($d_u = 1$ e $y_u = 0$), a BA define uma ilha temporal $S_n(u)$ de largura ℓ, centrada em u, conforme demonstrado na Equação III.4.

$$S_n(u) := \big\{\, t : u - r \le t \le u + r \big\}, \qquad r = \left\lfloor \frac{\ell}{2} \right\rfloor . \tag{III.4}$$

Utilizando as ilhas $S_n(u)$ e os intervalos de anomalia S_m, a detecção ajustada pela BA, denotada por $\hat{d}_t^{BA}$, é definida pela Equação III.5 para todo instante t.

$$\hat{d}_t^{BA} = \begin{cases} 1, & \text{se } d_t = 1, \\ 1, & \text{se } t \in S_m \text{ e } \exists\, t' \in S_m \text{ tal que } d_{t'} = 1, \\ 1, & \text{se } \exists\, u \text{ tal que } t \in S_n(u), \\ 0, & \text{caso contrário,} \end{cases} \tag{III.5}$$

A partir das predições ajustadas $\hat{d}_t^{BA}$ e dos rótulos verdadeiros y_t, computam-se então as quantidades de TP, FP, TN e FN e, em seguida, o F1 associado ao protocolo BA, mantendo-se a mesma forma de cálculo do F1 tradicional.

Do ponto de vista da complexidade computacional, o protocolo BA também pode ser implementado com custo assintoticamente linear em função do tamanho da série. A identificação dos falsos positivos ($d_t = 1$ e $y_t = 0$) e a construção das ilhas $S_n(u)$ podem ser feitas em uma única passada sobre a série, na qual se marcam, para cada FP, os índices pertencentes ao intervalo de largura ℓ ao redor deles. Como o número total de posições marcadas é limitado por uma constante multiplicativa de N (mesmo na presença de sobreposições entre ilhas), o custo total de atualização

dos rótulos ajustados $\hat{d}_t^{BA}$ permanece em $O(N)$. Uma vez obtidos esses rótulos, o cálculo do F1 correspondente requer apenas uma nova varredura linear, de modo que a complexidade global da métrica BA é $O(N)$.

III.1.4 *Precision and Recall for Time Series*

O trabalho *Precision and Recall for Time Series* [Tatbul et al., 2018] surge com o intuito de estender as métricas tradicionais de precisão e revocação, do contexto da granularidade pontual, para o contexto da granularidade intervalar (Figura III.4). Consideremos que P é o conjunto de intervalos de predições (detecções), composto por cada intervalo de predição P_i, e R é o conjunto de intervalos de eventos reais, composto por cada intervalo de evento R_j.



Figura III.4: Sobreposição entre as anomalias reais (R) e as predições (P). Fonte: [Tatbul et al., 2018].

O cálculo da métrica *Range-based Precision* (Equação III.6) inicia-se com a iteração sobre todos os intervalos de predição. Para cada intervalo, calcula-se uma pontuação de precisão ($Precision_{\mathcal{T}}$). Nesse processo, cada intervalo de predição é comparado a todos os intervalos de eventos reais (N_r), a fim de identificar interseções e calcular as respectivas recompensas.

$$Precision_{\mathcal{T}}(R, P_i) = CardinalityFactor(P_i, R, \gamma) \times \sum_{j=1}^{N_r} \omega(P_i, P_i \cap R_j, \delta) \tag{III.6}$$

A métrica é composta por:

- Função de viés de posição ($\boldsymbol{\delta}$): função modificável que determina a forma de avaliar sobreposições considerando suas localizações ($\delta \in \mathbb{R},\ \delta \geq 1$).
- Função de tamanho ($\boldsymbol{\omega}$): função não modificável que determina a forma de avaliar sobreposições considerando o tamanho delas. Essa função utiliza o resultado da função de viés de posição como um de seus parâmetros ($\omega \in \mathbb{R},\ 0 \leq \omega \leq 1$).
- Função de cardinalidade ($\boldsymbol{\gamma}$): função que determina o nível de redução da pontuação da avaliação caso haja mais de um intervalo de evento real dentro do intervalo de predição em análise ($\gamma \in \mathbb{R},\ 0 \leq \gamma \leq 1$).

A função de viés de posição é escolhida pelo usuário e utilizada pela função de tamanho (Figura III.5) para calcular a pontuação de todos os intervalos de eventos reais em relação à predição em análise.

```
function ω(AnomalyRange, OverlapSet, δ)
    MyValue ← 0
    MaxValue ← 0
    AnomalyLength ← length(AnomalyRange)
    for i ← 1, AnomalyLength do
        Bias ← δ(i, AnomalyLength)
        MaxValue ← MaxValue + Bias
        if AnomalyRange[i] in OverlapSet then
            MyValue ← MyValue + Bias
    return MyValue/MaxValue
```

```
function δ(i, AnomalyLength)          ▷ Flat bias
    return 1
function δ(i, AnomalyLength) ▷ Front-end bias
    return AnomalyLength - i + 1
function δ(i, AnomalyLength) ▷ Back-end bias
    return i
function δ(i, AnomalyLength)     ▷ Middle bias
    if i ≤ AnomalyLength/2 then
        return i
    else
        return AnomalyLength - i + 1
```

Figura III.5: Exemplo das funções de viés de posição $\delta()$ e da função de tamanho $\omega()$. Fonte: [Tatbul et al., 2018].

O somatório dessas pontuações é ponderado pelo fator de cardinalidade (Equação III.7) daquela predição para obter a precisão do intervalo de predição.

$$\mathit{CardinalityFactor}(P_i, R, \gamma) = \begin{cases} 1, & \text{se } P_i \text{ sobrepõe no máximo um } R_j \in R \\ \gamma(P_i, R), & \text{caso contrário} \end{cases} \tag{III.7}$$

Invertendo-se os papéis dos intervalos de eventos reais e dos intervalos de predição, ocorre um procedimento análogo para o cálculo da métrica *Range-based Recall* (Equação III.8). A mesma função de viés de posição escolhida pelo usuário é utilizada pela função de tamanho, porém, para calcular a pontuação de todos os intervalos de predição em relação ao intervalo de evento real em análise. O somatório dessas pontuações é ponderado pelo fator de cardinalidade daquele evento real para obter a recompensa de sobreposição do intervalo de evento (Equação III.9). No cálculo da revocação, há ainda a inserção de uma recompensa de existência (Equação III.10). $\boldsymbol{\alpha}$ é um parâmetro de existência determinado pelo usuário para ponderar o grau de importância entre a mera existência de uma sobreposição e o tamanho das sobreposições no cálculo da revocação do intervalo ($\alpha \in \mathbb{R},\ 0 \leq \alpha \leq 1$).

$$\mathit{Recall}_{\mathcal{T}}(R_i, P) = \alpha \times \mathit{ExistenceReward}(R_i, P) + (1 - \alpha) \times \mathit{OverlapReward}(R_i, P) \tag{III.8}$$

$$\mathit{OverlapReward}(R_i, P) = \mathit{CardinalityFactor}(R_i, P) \times \sum_{j=1}^{N_p} \omega(R_i, R_i \cap P_j, \delta) \tag{III.9}$$

$$\mathit{ExistenceReward}(R_i, P) = \begin{cases} 1, & \text{se } \sum_{j=1}^{N_p} |R_i \cap P_j| \geq 1 \\ 0, & \text{caso contrário} \end{cases} \tag{III.10}$$

Como se observa, os processos são similares; porém, a recompensa de existência não é considerada no cálculo da precisão, pois se entende que a simples presença de um segmento de evento dentro de um segmento de detecção não contribui de modo relevante para uma métrica de precisão, o que equivale a estabelecer $\alpha = 0$. A pontuação da revocação será sempre um *trade-off* entre a recompensa de existência e a de sobreposição.

Por fim, a média da precisão de cada intervalo de predição e a média da revocação de cada intervalo de evento são calculadas para obter as métricas finais de precisão e revocação (Equação III.11).

$$\text{Precision}_{\mathcal{T}}(R, P) = \frac{\sum_{i=1}^{N_p} \text{Precision}_{\mathcal{T}}(R, P_i)}{N_p} \qquad \text{Recall}_{\mathcal{T}}(R, P) = \frac{\sum_{i=1}^{N_r} \text{Recall}_{\mathcal{T}}(R_i, P)}{N_r} \qquad \text{(III.11)}$$

A análise do código da implementação permite inferir que a complexidade computacional dos processos é da ordem de $O(R \times P)$, em que R representa o número de intervalos de eventos reais e P o número de intervalos de predições. No pior caso, R e P têm a mesma ordem de grandeza ($R \approx P$), logo, a complexidade pode ser considerada quadrática, $O(N^2)$.

III.1.5 *Time-Series Aware Metrics*

O trabalho de Hwang et al. [2019] define as métricas TaR (*Time-series Aware Recall*) e TaP (*Time-series Aware Precision*), considerando os eventos como anomalias (a) e as detecções como predições (p). O estudo afirma que as métricas tradicionais tendem a sobrevalorizar a detecção de poucas anomalias longas e a subestimar a detecção de múltiplas anomalias curtas. Para mitigar esse viés, é proposta uma avaliação que considera dois aspectos: a quantidade de anomalias distintas detectadas, denominada pontuação de detecção (TaR^d e TaP^d), e a proporção das anomalias efetivamente capturadas, denominada pontuação de porção (TaR^p e TaP^p). Por fim, os dois aspectos são combinados de forma ponderada, conforme apresentado a seguir (Equação III.15).

O estudo introduz o conceito de instâncias ambíguas (Figura III.6), pontos não rotulados da série temporal que ocorrem imediatamente após uma anomalia rotulada. Embora classificados como normais, esses pontos podem ainda estar sob influência do evento anômalo e, por isso, o método atribui a essas instâncias pontuações decrescentes em relação à anomalia anterior. Ao definir eventos e detecções como intervalos, em vez de pontos, a definição já incorpora a noção de extensão temporal. Não parece necessário introduzir intervalos adicionais além dos especificados pelos rótulos; porém, o autor argumenta que um especialista pode definir um parâmetro de tolerância (δ) após o intervalo de cada anomalia, a depender do domínio de aplicação, o que tornaria a avaliação mais acurada [Hwang et al., 2019].

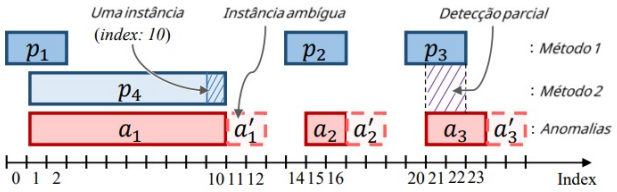


Figura III.6: Instâncias ambíguas e detecções parciais. Fonte: [Hwang et al., 2019].

Neste trabalho, o cálculo do *Overlap score* $O(a, p)$ entre cada segmento de anomalia real $a \in A$ e cada segmento de predição $p \in P$ envolve a verificação ponto a ponto da interseção entre os segmentos e a verificação dos pontos classificados como ambíguos, localizados imediatamente após as anomalias reais.

Como se observa na Figura III.7, embora a interseção entre anomalia e predição ocorra apenas nos índices 7 e 8, a função de pontuação subsequente $S(a', p)$ (Equação III.12) é calculada para os índices 9 a 12. Essa região corresponde à interseção entre o intervalo de tolerância posterior à anomalia (δ), a chamada anomalia ambígua a', e o restante da predição. O cálculo de pontuação para instâncias ambíguas utiliza uma sigmoide invertida, cuja saída diminui gradualmente à medida que a distância em relação à anomalia real aumenta. Assim, quanto mais distante a predição estiver do início de a' (índice $t_{a'}$), menor será a contribuição dela.

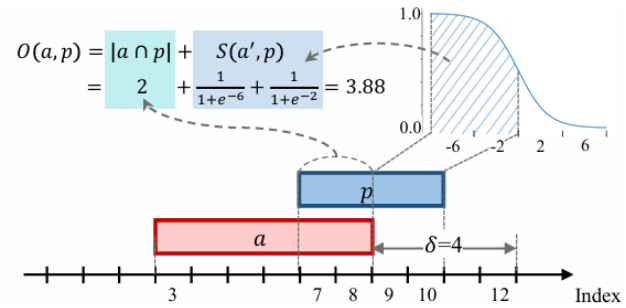


Figura III.7: Cálculo do *Overlap Score* para anomalia a e predição p. Fonte: [Hwang et al., 2019].

$$S(a', p) = \sum_{i \in (a' \cap p)} \frac{1}{1 + e^{i'}} \quad \text{, onde} \quad i' = -6 + \frac{12(i - t_{a'} - 1)}{\delta - 1} \tag{III.12}$$

Quando a pontuação $O(a, p)$ é calculada, um limiar θ (determinado pelo usuário) é utilizado para determinar se cada anomalia é corretamente detectada ou não. Uma anomalia a é considerada detectada se a soma dos valores de *overlap score* entre ela e todas as predições $p \in P$, dividida pelo tamanho da anomalia $|a|$, for maior ou igual a θ. A métrica TaR^d (Equação III.13) representa a fração de anomalias consideradas detectadas.

$$TaR^d = \frac{|A^d(\theta)|}{|A|} \quad \text{, onde} \quad A^d(\theta) = \left\{ a \in A \;\middle|\; \frac{\sum_{p \in P} O(a, p)}{|a|} \geq \theta \right\} \tag{III.13}$$

Já TaR^p (Equação III.14) calcula, para cada anomalia real $a \in A$, a fração dos pontos de a cobertos pelas predições $p \in P$, usando a pontuação de sobreposição $O(a, p)$. Essa fração é limitada superiormente a 1 por meio da função $min()$, pois o conceito de anomalia ambígua pode elevar $O(a, p)$ para valores superiores a 1. Em seguida, calcula-se a média dessa fração para todas as anomalias. Dessa forma, TaR^p busca indicar qual porção das anomalias é detectada.

$$TaR^p = \frac{1}{|A|} \sum_{a \in A} \min\left(1, \frac{\sum_{p \in P} O(a, p)}{|a|}\right) \tag{III.14}$$

A versão final da métrica TaR (Equação III.15) combina as componentes (TaR^p e TaR^d) por meio de um parâmetro de ponderação $\alpha \in [0, 1]$, definido pelo usuário:

$$TaR = \alpha \cdot TaR^d + (1 - \alpha) \cdot TaR^p \tag{III.15}$$

Um raciocínio análogo define o *Time-Series Aware Precision* (TaP), invertendo os papéis de anomalias e de predições. A métrica TaP considera, portanto, a quantidade de predições que correspondem a anomalias reais, bem como a proporção de cada predição que cobre uma anomalia verdadeira. Assim como no TaR, o TaP também se divide em TaP^d, TaP^p e uma média ponderada TaP.

O custo computacional por par (a, p) é dominado pela verificação da interseção (*overlap*), cujo tempo de execução depende do comprimento médio das anomalias e das predições (denotado por L) ou do comprimento da faixa de ambiguidade (denotado por δ). Assim, o custo do par (a, p) é da ordem de $O(L + \delta)$ no pior caso. Considerando A o número de segmentos de anomalias e P o número de segmentos de detecção, como a métrica considera todos os pares possíveis de segmentos de anomalias e de predições, a complexidade total do método de avaliação é $O(A \times P \times (L + \delta))$. Se considerarmos que, em casos comuns, os comprimentos das anomalias, predições e da faixa de ambiguidade $(L + \delta)$ podem ser limitados superiormente por uma constante pequena, pode-se considerar, para fins práticos, que a complexidade é $O(A \times P)$. No pior caso, A e P têm a mesma ordem de grandeza ($A \approx P$), logo, a complexidade pode ser considerada quadrática $O(N^2)$.

III.1.6 *Enhanced Time-Series Aware Metrics*

Posteriormente, Hwang et al. [2022] apontam que métricas que valorizam detecções parciais ou excessivas tendem a superestimar cenários imprecisos ou insuficientes, deixando de penalizar adequadamente tais predições incorretas. Conforme ilustrado na Figura III.8, apenas o caso 1 deve ser classificado como acerto. Ademais, uma métrica que reputasse o caso 2 como falha, em razão da baixa sobreposição entre a predição e a anomalia, ainda poderia não distinguir corretamente os casos 3 e 4.

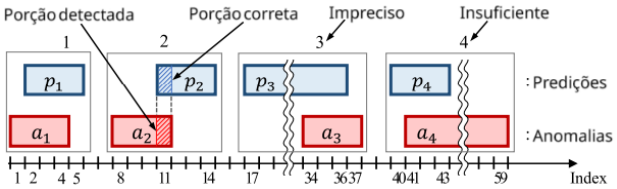


Figura III.8: Exemplo de quatro possíveis casos de detecção. Fonte: [Hwang et al., 2022].

Para mitigar essa limitação, os autores propõem o método $eTaPR$. A métrica herda os mesmos conceitos de pontuação de detecção e de porção de sua predecessora, TaPR [Hwang et al., 2019], mas incorpora também um mecanismo de referência cruzada e um esquema de ponderação específico.

O mecanismo de referência cruzada quantifica o comprimento da interseção entre as janelas de anomalia e de detecção, eliminando contribuições indevidas de casos insuficientes ou imprecisos. Para isso, busca determinar um conjunto de anomalias suficientemente detectadas (A^d) com um limiar (θ_r). Similarmente, busca determinar um conjunto de predições não excessivas (P^c) a partir de um limiar (θ_p).

O cálculo de A^d e P^c ocorre por meio de um processo iterativo de referência cruzada: inicialmente, estima-se A^d aplicando a Equação III.16 ao conjunto completo de predições P; em seguida, atualiza-se P^c por meio da Equação III.17 usando esse A^d. O processo se repete iterativamente, recalculando A^d e P^c até que ambos os conjuntos convirjam. Dessa forma, anomalias e predições associadas a casos imprecisos ou insuficientes são progressivamente excluídas dos conjuntos finais de "detectadas" e "corretas", garantindo coerência entre os limiares θ_r e θ_p.

$$A^d = \left\{ a \in A \;\middle|\; \frac{\sum_{p \in P^c} |a \cap p|}{|a|} \geq \theta_r \right\}. \tag{III.16}$$

$$P^c = \left\{ p \in P \;\middle|\; \frac{\sum_{a \in A^d} |a \cap p|}{|p|} \geq \theta_p \right\}. \tag{III.17}$$

Após isso, a pontuação de detecção é determinada pela Equação III.18 e a pontuação de porção, pela Equação III.19. Por fim, a métrica $eTaR$ é calculada de acordo com a Equação III.20.

$$s^d(a) = \begin{cases} 1, & \text{se } a \in A^d, \\ 0, & \text{caso contrário.} \end{cases} \tag{III.18}$$

$$s^p(a) = \frac{\sum_{p \in P^c} |a \cap p|}{|a|}. \tag{III.19}$$

$$\text{eTaR} = \frac{1}{|A|} \sum_{a \in A} \frac{s^d(a) + s^d(a) \times s^p(a)}{2} \tag{III.20}$$

Já para o cálculo da métrica de precisão ($eTaP$), propõe-se um esquema de ponderação para ajustar as pontuações em função da duração das janelas de detecção, impondo penalizações proporcionais ao prolongamento excessivo dos segmentos de predição (Equação III.21).

$$w_p \;=\; \frac{\sqrt{|p|}}{\sum_{q \in P} \sqrt{|q|}}. \tag{III.21}$$

Após isso, o cálculo segue uma lógica semelhante à utilizada em $eTaR$, como ilustrado na Equação III.22. Contudo, aqui as pontuações são definidas com base nas predições, de modo que $s^d(p)$ e $s^p(p)$ substituem $s^d(a)$ e $s^p(a)$, invertendo os papéis entre anomalias e predições. A principal diferença é a incorporação do esquema de ponderação, que penaliza predições excessivamente longas.

$$\text{eTaP} \;=\; \sum_{p \in P} \left(\frac{s^d(p) + s^d(p) \times s^p(p)}{2} \right) \times w_p \tag{III.22}$$

Quando se emprega a média aritmética para computar a revocação, o método tende a sobrevalorizar quaisquer detecções que ultrapassem o limiar, conferindo igual relevância à detecção parcial de segmentos, independentemente de sua extensão. Por sua vez, ao utilizar a média ponderada, cujos pesos são proporcionais à raiz quadrada do comprimento de cada segmento, para o cálculo da precisão, perde-se a equiparação previamente alcançada: segmentos mais longos e corretamente detectados exercem influência positiva sobre w_p, elevando-o artificialmente e falhando em atender ao propósito de promover uma avaliação balanceada.

Os autores afirmam que o $TaPR$ supera as limitações de complexidade impostas pelo método de Tatbul et al. [2018] e que o $eTaPR$ representa um avanço adicional. Embora a métrica de Tatbul et al. [2018] englobe múltiplos critérios de avaliação (existência, extensão, posição e cardinalidade), sua concepção permanece objetiva e clara o suficiente para quantificar o desempenho em detecções segmentadas de eventos. Em contrapartida, as métricas $TaPR$ e $eTaPR$ incorporam elementos suplementares, como os conceitos de instâncias ambíguas e o esquema de ponderação, que não se mostram essenciais.

Em relação à complexidade computacional, o método começa construindo uma matriz de anomalias e predições $M \in \mathbb{R}^{A \times P}$, em que cada entrada é preenchida com o cálculo do *overlap score* $O(a, p)$ entre cada segmento de anomalia a e predição p, exatamente conforme implementado em Hwang et al. [2019]. O custo total dessa construção permanece em $O(A \times P \times (L + \delta))$.

Em seguida, na etapa de *pruning*, o algoritmo normaliza cada linha i por $\max(a_i)$ e cada coluna j por $\max(p_j)$ para verificar, respectivamente, se $\sum_j M_{ij}/|a_i| \geq \theta_r$ e $\sum_i M_{ij}/|p_j| \geq \theta_p$. Caso contrário, a linha ou coluna inteira é zerada. No pior caso, podem ocorrer até $A + P$ iterações, cada uma custando $O(A \times P)$, totalizando $O((A + P) \times A \times P)$.

Depois, para o cálculo de $eTaR$ e $eTaP$, o algoritmo itera sobre as linhas e colunas de M não

zeradas, somando os *overlap scores* e aplicando os limiares (θ_r e θ_p), e, em seguida, calcula o score de detecção e o score de porção. Em seguida, realiza o cálculo das métricas; tudo isso ocorre em $O(A \times P)$. Assim, a complexidade total do algoritmo seria $O((A \times P \times (L+\delta)) + ((A+P) \times A \times P))$, porém, ao se considerar que $(L+\delta)$ é limitado por uma constante pequena, o termo dominante passa a ser o custo associado ao processo de *pruning*, cuja complexidade é $O((A+P) \times A \times P)$.

No melhor dos casos, quando nenhuma linha ou coluna é zerada durante o processo de *pruning*, a convergência ocorre já na primeira verificação. Nessa situação, o algoritmo executa apenas uma única passagem de somas vetoriais sobre linhas e colunas, de modo que o custo total passa a ser dominado apenas pelo preenchimento inicial da matriz de *overlap* e pelo cálculo final das métricas, resultando em $O(A \times P)$, isto é, $O(N^2)$.

Entretanto, no pior dos casos, o processo de *pruning* é longo. Caso A e P tenham a mesma ordem de grandeza ($A \approx P$), a complexidade assintótica do método é $O(N^3)$, caracterizando um comportamento cúbico.

III.1.7 Revisão das métricas intervalares

Em resumo, algumas métricas intervalares se fundamentam em determinar, de forma diferenciada, o número de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos e, depois disso, avaliar com as mesmas fórmulas das métricas tradicionais. As métricas PA [Xu et al., 2018], PA%K [Siwon et al., 2022] e BA [Bhattacharya et al., 2025] se baseiam na modificação de rótulos antes de aplicar uma avaliação tradicional. A métrica segmentada [Hundman et al., 2018] considera intervalos de eventos e de detecções como unidades. A métrica $F1_C$ [Garg et al., 2022] propõe uma combinação explícita entre a métrica pontual tradicional e a métrica segmentada (duas visões), utilizando a precisão pontual e a revocação segmentada. Em seguida, aplica um F1 híbrido, ainda que o procedimento permaneça semelhante ao das métricas tradicionais.

Outras métricas baseiam-se em avaliações de sobreposições, como Precision$_\mathcal{T}$ / Recall$_\mathcal{T}$ [Tatbul et al., 2018], TaR/TaP [Hwang et al., 2019] e $eTaR/eTaP$ [Hwang et al., 2022]. Essas métricas possuem mecanismos próprios para determinar a precisão e a revocação. Esses grupos de métricas diferem quanto ao compromisso entre a riqueza da avaliação e o custo computacional. Uma comparação da complexidade computacional de todos os métodos pode ser vista na Tabela III.1.

Cada métrica voltada à avaliação de detecções de eventos com granularidade intervalar possui características próprias, mas, no domínio ao qual se destinam, isto é, detecção de eventos intervalares, elas se mostram superiores às métricas tradicionais. Isso ocorre porque as métricas tradicionais tratam detecções e eventos de forma pontual, adotando uma abordagem ingênua para problemas de intervalo, ao ignorarem a duração e a extensão temporal dos eventos.

Embora sejam adequadas ao domínio de eventos de granularidade intervalar, essas métricas

Técnica	**Complexidade**
ponto-ajustada (PA)	$O(N)$
Segmentada ($F1_{\text{segmented}}$)	$O(N)$
PA%K	$O(N)$
BA	$O(N)$
Precision$_{\mathcal{T}}$ / Recall$_{\mathcal{T}}$	$O(N^2)$
TaR / TaP	$O(N^2)$
eTaR / eTaP	$O(N^3)$
Composite F-Score ($F1_C$)	$O(N)$

Tabela III.1: Comparativo entre métricas intervalares considerando apenas a complexidade assintótica. *Composite F-Score* se destaca por sua granularidade híbrida.

mostram-se inadequadas para a avaliação de eventos pontuais, pois não foram projetadas considerando as particularidades desse tipo de ocorrência. No contexto de eventos pontuais, a proximidade temporal entre detecções e eventos torna-se um aspecto essencial. Nesse cenário, mecanismos de tolerância temporal surgem como uma forma de aprimorar a avaliação, incorporando pequenas discrepâncias temporais como parte natural do processo de comparação entre eventos e detecções.

A seguir, discutem-se as principais métricas propostas especificamente para a avaliação de eventos pontuais, destacando como elas abordam esse desafio e de que maneira diferem das métricas orientadas a intervalos.

III.2 Métricas de avaliação de eventos na forma pontual

As métricas pontuais preservam a granularidade original dos rótulos, tratando eventos e detecções como ocorrências instantâneas no tempo. Nesse contexto, a avaliação é conduzida por meio de contagens ponto a ponto, o que permite o uso direto das métricas tradicionais de classificação. Não surpreende, portanto, que diversos trabalhos na literatura [Liu et al., 2013; Kravchik and Shabtai, 2018; Li et al., 2019; Abdulaal et al., 2021] adotem exclusivamente medidas clássicas, como acurácia, precisão, revocação e F1, ou mesmo métricas derivadas dessas, como a *Area Under the Receiver Operating Characteristic Curve* (AUC-ROC) [Spackman, 1989; Hanley and McNeil, 1983].

Entretanto, embora essas métricas capturem corretamente relações locais entre rótulos verdadeiros e previstos, não consideram propriedades estruturais de séries temporais, como a proximidade entre detecções e eventos. Assim, uma detecção que ocorra poucos instantes antes ou depois do evento real é avaliada da mesma forma que uma detecção completamente deslocada no tempo. Para contornar essa limitação, algumas propostas mantêm a granularidade pontual, mas introduzem mecanismos adicionais para lidar com tolerâncias temporais. As métricas NAB (*Numenta Anomaly Benchmark*) [Lavin and Ahmad, 2015] e SoftED [Salles et al., 2024] seguem essa abordagem, incorporando janelas de tolerância que suavizam o impacto de pequenos deslocamentos temporais entre eventos e detecções.

III.2.1 *The Numenta Anomaly Benchmark* (NAB)

A métrica NAB [Lavin and Ahmad, 2015] é desenvolvida com o objetivo de avaliar algoritmos de detecção de eventos em dados de fluxo contínuo (*streaming*), em contexto em que as pontuações de anomalia têm a forma de pontuações de propensão. O método realiza a binarização das pontuações de anomalia geradas pelos detectores. Cada valor contínuo associado a um instante da série temporal é comparado a um limiar (*threshold*), de modo que pontuações iguais ou superiores ao limiar são consideradas detecções, enquanto valores inferiores ao limiar são tratados como ausência de detecção. A partir daí, pode-se trabalhar com detecções binárias.

A métrica adota um viés de avaliação que privilegia detecções precoces. Para isso, o método estabelece uma janela temporal à frente do evento e considera apenas a primeira detecção ocorrida nela como válida. Na Figura III.9, podemos observar detecções fora da janela temporal (pontos vermelhos) e dentro dela (pontos verdes).

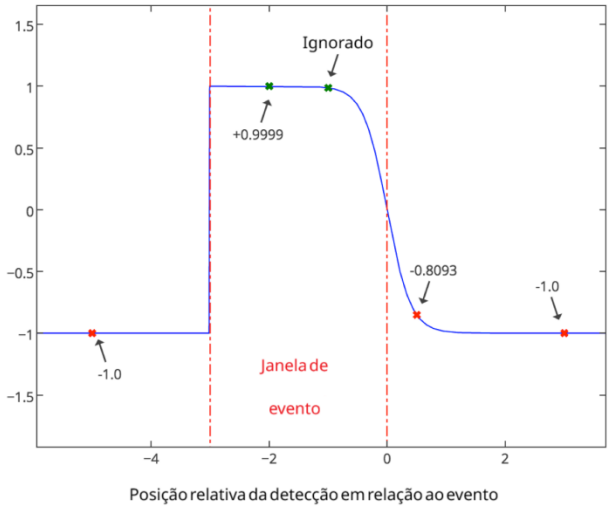


Figura III.9: Função de avaliação NAB. Fonte: [Lavin and Ahmad, 2015].

Essa métrica inclui três *perfis de aplicação*: um é o perfil padrão, outro penaliza falsos positivos (priorizando precisão) e outro penaliza falsos negativos (priorizando revocação). Os valores $A_{TP}, A_{FP}, A_{FN}, A_{TN}$ são pesos aplicados aos cálculos da métrica, definidos por:

$$A_{TP}, A_{TN} \in [0, 1] \quad \text{e} \quad A_{FP}, A_{FN} \in [-1, 0]. \tag{III.23}$$

Os autores reconhecem que a aplicação desses pesos é arbitrária, mas explicam que o propósito é adaptar-se a diferentes demandas de diferentes domínios.

A função de avaliação do NAB, definida na Equação III.24, é aplicada a cada detecção da série, na qual x corresponde à posição relativa da detecção em relação à posição de cada evento (no canto direito da janela de evento).

$$s(x) = (A_{TP} - A_{FP}) \left(\frac{1}{1 + e^{5x}} \right) - 1 \tag{III.24}$$

Após essa etapa, as detecções são filtradas: se houver múltiplas detecções dentro de uma janela, apenas a primeira é considerada, enquanto as demais são apagadas. Dessa forma, o conjunto de detecções efetivamente computado ($D_{filtered}$) é composto por todas as detecções que ocorrem fora das janelas, contabilizadas como falsos positivos, e por zero ou uma detecção em cada janela. A pontuação total (S) da série pode ser obtida pela Equação III.25. Ela é a soma das pontuações das detecções, sendo f_d o número de janelas sem detecções.

$$S = \left(\sum_{d \in D_{\text{filtered}}} s(x) \right) + A_{\text{FN}}\, f_d \tag{III.25}$$

Após isso, a pontuação S é normalizada por meio da Equação III.26, sendo S_{nulo} a pontuação para zero detecções e S_{perfeito} a pontuação para um detector perfeito, em que todos os eventos são perfeitamente detectados e não há falsos positivos.

$$S_{normalizado} = 100 \cdot \frac{S - S_{\text{nulo}}}{S_{\text{perfeito}} - S_{\text{nulo}}} \tag{III.26}$$

O fator 100 é utilizado apenas para escalar a pontuação normalizada para uma faixa percentual, na qual 0 e 100 representam, respectivamente, os desempenhos de um detector nulo e de um detector perfeito.

Quanto à avaliação da complexidade computacional, primeiramente, as janelas de anomalia são definidas, estabelecendo os intervalos temporais em que um evento anômalo é considerado válido. Antes da etapa de avaliação, o módulo valida essas janelas, garantindo que não haja sobreposição entre elas. Caso sejam identificados intervalos que se cruzam, o processo é interrompido e um erro é lançado, exigindo a correção manual dos rótulos. Considerando E como o número de eventos, essa etapa ocorre em $O(E)$.

Durante a avaliação, o NAB percorre a série temporal de forma sequencial, associando a cada instante a respectiva pontuação. Nesse processo, mantém em memória a lista predefinida de janelas de anomalia e, ao atingir o início de uma delas, passa a avaliar as detecções apenas em relação àquele intervalo específico. Ao alcançar o fim da janela, o algoritmo finaliza a análise desse evento e passa para a próxima janela, segmentando a avaliação até o fim da série. Essa etapa ocorre em $O(D)$, onde D é o número de detecções. Em seguida, a filtragem de detecções e o cálculo da pontuação final e da pontuação normalizada são $O(D)$. A complexidade computacional do método é $O(D+E)$. Como, no pior caso, D e E têm a mesma ordem de magnitude, tem-se $O(D + E) = O(N)$, em que N representa o tamanho do problema.

III.2.2 *SoftED: Metrics for Soft Evaluation of Time Series Event Detection*

As explicações relativas às métricas SoftED [Salles et al., 2024] já foram apresentadas (Seção II.3); aqui analisam-se detalhes da implementação do algoritmo, bem como sua complexidade computacional.

O algoritmo inicia recebendo duas sequências binárias: uma representando as detecções realizadas e outra, os eventos reais. Primeiramente, são extraídos os índices correspondentes às posições das detecções e dos eventos. Essa etapa é $O(|X|)$, sendo X o tamanho da série.

Na etapa seguinte, constrói-se uma matriz de pertinência Mu de dimensões $D \times E$, calculando, para cada detecção $D[i]$ e evento $E[j]$, a pontuação determinada pela função triangular de pertinência relativa a $E[j]$. Como essa operação requer iterar sobre todas as combinações possíveis, sua complexidade é $O(D \times E)$.

Posteriormente, o algoritmo identifica, para cada detecção, o conjunto de eventos com maior pertinência e, inversamente, para cada evento, as detecções associadas. Ambas as operações envolvem novamente a varredura da matriz construída anteriormente, linha a linha e coluna a coluna, o que gera complexidade adicional de ordem $O(D \times E)$.

Por fim, o algoritmo determina, para cada evento, a detecção mais próxima e calcula as pontuações finais de eventos e detecções, com base nas associações já estabelecidas. Essas operações também têm complexidade $O(D \times E)$. Portanto, a complexidade total do algoritmo é $O(D \times E)$. Como D e E têm a mesma ordem de grandeza ($D \approx E$), a complexidade pode ser considerada quadrática, isto é, $O(N^2)$.

III.2.3 Revisão das métricas pontuais

No grupo das métricas pontuais, a precisão, a revocação e o F1 tradicional, bem como as medidas derivadas (acurácia, precisão e revocação), não consideram tolerância temporal em sua análise. Elas realizam contagens diretas de TP, FP, TN e FN, podendo ser computadas em $O(N)$. A AUC-ROC também opera ponto a ponto, mas requer a ordenação dos escores de predição para a construção da curva ROC, elevando a complexidade para $O(N \log N)$.

O método NAB [Lavin and Ahmad, 2015] introduz tolerância temporal em suas avaliações por meio de janelas temporais anteriores aos eventos pontuais e considera apenas a primeira detecção ocorrida no intervalo dessas janelas como válida, descartando as demais. Detecções externas às janelas são consideradas falsos positivos. A pontuação atribuída a cada detecção é determinada por uma função sigmoidal ajustada por pesos específicos, o que permite diferentes perfis de penalização de acordo com o domínio de aplicação. O processo de avaliação percorre a série temporal de forma sequencial, verificando a ativação das janelas e aplicando a função de pontuação apenas à detecção relevante em cada uma delas. Tanto a filtragem de múltiplas detecções quanto o cálculo

da pontuação final são lineares em função do número de detecções. Dessa forma, a métrica mantém um custo computacional linear, com complexidade total $O(N)$.

Já o método SoftED [Salles et al., 2024] introduz tolerância temporal em suas avaliações por meio de janelas temporais em torno dos eventos pontuais e realiza associações para determinar a que evento cada detecção pertence. A pontuação atribuída a cada detecção é determinada por uma função de pertinência baseada em lógica difusa. Essa pertinência graduada permite que múltiplas detecções disputem um mesmo evento, o que gera situações de concorrência entre possíveis associações. A métrica exige a construção de uma matriz de pertinência de dimensões $D \times E$, a partir da qual são realizadas associações gulosas entre eventos e detecções. Isso resulta em um comportamento quadrático $O(N^2)$ quando o número de detecções D e o de eventos E estão da mesma ordem.

O comparativo das métricas pontuais pode ser visto na Tabela III.2.

Técnica	**Tolerância temporal**	**Complexidade**
F1 tradicional	Não	$O(N)$
AUC-ROC	Não	$O(N \log N)$
NAB	Sim	$O(N)$
SoftED	Sim	$O(N^2)$

Tabela III.2: Comparativo entre métricas pontuais, considerando a tolerância temporal e a complexidade assintótica.

Diferenciais e limitações das métricas SoftED

Diante do exposto, sabemos que NAB e SoftED superam as métricas tradicionais nas avaliações de eventos pontuais ao incorporarem tolerância temporal em seus métodos. Em termos de comportamento, contudo, as duas métricas diferem substancialmente. A NAB privilegia detecções antecipadas, atribuindo maior crédito às que ocorrem antes do evento, mas não recompensa detecções exatas (simultâneas ao evento) nem as posteriores. Já o SoftED atribui pontuação máxima às detecções exatas e é capaz de avaliar detecções anteriores e posteriores, graduando suavemente a relevância conforme a distância temporal entre a detecção e o evento.

Cabe ressaltar, ainda, que as associações entre detecções e eventos em SoftED são sempre guiadas pela proximidade temporal, refletindo explicitamente o grau de pertinência entre cada par detecção–evento. Já a NAB adota uma estratégia distinta: considera apenas a primeira detecção que ocorre dentro da janela temporal do evento e ignora as demais, sem realizar um processo de associação estruturado que busque distinguir, de fato, qual detecção corresponde a qual evento.

Essa distinção também é evidenciada na avaliação experimental conduzida no próprio trabalho que introduz o SoftED. Ao comparar quantitativamente SoftED e NAB, os autores observam que o NAB pode incorporar níveis de tolerância mais altos do que o SoftED em diversos cenários, pois seu grau de tolerância depende diretamente do tamanho das *anomaly windows* definidas automati-

camente, que variam conforme a série temporal e podem assumir amplitudes muito distintas entre bases de dados [Salles et al., 2024]. Consequentemente, essa característica dificulta comparações sob condições padronizadas, podendo gerar janelas demasiadamente largas ou estreitas para determinadas aplicações, além de reduzir a interpretabilidade dos resultados para especialistas de domínio. Em contrapartida, o SoftED fornece um mecanismo controlável de tolerância por meio do parâmetro k, permitindo que analistas definam explicitamente o nível de tolerância temporal desejado e interpretem mais diretamente os impactos dessa escolha na pontuação final [Salles et al., 2024].

Além disso, na análise qualitativa apresentada por Salles et al. [2024], o SoftED demonstrou maior consistência em diferentes cenários de avaliação, especialmente por favorecer a seleção de detectores que conciliam proximidade temporal com redução de falsos positivos e falsos negativos. Segundo os autores, em cinco dos seis cenários avaliados com especialistas, a métrica baseada em F1 sob a abordagem SoftED foi a que mais corroborou a escolha do detector considerado mais adequado, sendo inclusive a única a sustentar a recomendação em quatro cenários distintos [Salles et al., 2024]. Embora o NAB tenha se destacado em um dos cenários, os autores discutem que parte dessa preferência esteve associada à interpretação de detecções anteriores como previsões, um aspecto que pode introduzir vieses de interpretação se não houver distinção clara entre detecção com atraso e antecipação real do evento [Salles et al., 2024].

Em síntese, o SoftED mostra-se mais abrangente do que o NAB na avaliação de eventos de granularidade pontual, ao aplicar tolerâncias temporais anterior e posterior ao evento, ao realizar associações mais conscientes entre eventos e detecções e ao graduar suas pontuações com base na proximidade temporal.

Entretanto, a métrica ainda apresenta oportunidades de aprimoramento. Do ponto de vista computacional, o NAB mantém vantagem ao operar com custo quase-linear, enquanto o SoftED apresenta maior complexidade. No que diz respeito ao processo de associação, embora supere o NAB, o SoftED não garante otimalidade nas correspondências entre eventos e detecções. Assim, tanto a busca por associações ótimas quanto a melhoria da eficiência computacional constituem o foco central das contribuições deste trabalho.

Capítulo IV Modelagem e Formulação do Problema de Associação entre Eventos e Detecções

Neste capítulo, o problema de associação entre eventos e detecções no método SoftED é formulado como um problema de emparelhamento máximo ponderado em um grafo bipartido completo e desbalanceado, no contexto de otimização combinatória. A partir dessa modelagem, apresenta-se o Algoritmo Húngaro como uma solução clássica e amplamente adotada para a resolução desse tipo de problema.

IV.1 Formulação do problema

Sejam $D = \{d_1, \ldots, d_{|D|}\}$ o conjunto de detecções pontuais e $E = \{e_1, \ldots, e_{|E|}\}$ o conjunto de eventos pontuais. Seja ainda $k > 0$ o parâmetro de tolerância temporal empregado pelo SoftED. Esse método define uma função de afinidade local $\mu(d, e) \in [0, 1]$, que quantifica o grau de compatibilidade entre uma detecção d e um evento e em função de sua proximidade temporal.

A partir dessa definição, o problema é modelado por meio de um grafo bipartido ponderado $G = (V, A)$, em que o conjunto de vértices é dado por $V = D \cup E$, com $D \cap E = \emptyset$, e o conjunto de arestas é definido por $A = D \times E$. Cada aresta $(d, e) \in A$ recebe um peso $w_{de} = \mu(d, e)$, correspondente à afinidade atribuída pelo SoftED. Os conjuntos D e E podem ser estendidos com elementos artificiais (*dummy*) quando necessário, de modo que o número de detecções seja igual ao número de eventos ($|D| = |E|$), balanceando o grafo bipartido. As associações envolvendo tais elementos artificiais recebem peso zero, assim como os pares inviáveis, correspondentes às associações fora da janela temporal definida por k.

A associação entre detecções e eventos é formulada como um problema de otimização combinatória, no qual se busca selecionar associações um-para-um que maximizem a soma das afinidades globais. Para esse fim, introduz-se a variável de decisão binária x_{de}, que assume valor 1 caso a detecção d seja associada ao evento e, e o valor 0 caso contrário.

A função objetivo definida na Equação IV.1 maximiza a soma das afinidades das associações selecionadas, onde w_{de} representa a afinidade entre a detecção d e o evento e.

$$\max \sum_{d \in D} \sum_{e \in E} w_{de}\, x_{de}. \tag{IV.1}$$

O problema de atribuição proposto nesta pesquisa está sujeito a restrições de exclusividade. A Equação IV.2 garante que cada detecção $d \in D$ seja atribuída a exatamente um evento $e \in E$, enquanto a Equação IV.3 estabelece a condição simétrica, assegurando que cada evento $e \in E$ seja atribuído a exatamente uma detecção $d \in D$. Por fim, a Equação IV.4 define o domínio das variáveis de decisão, impondo que x_{de} seja binária, isto é, $x_{de} = 1$ quando a associação entre d e e é selecionada e $x_{de} = 0$ caso contrário. As associações ($x_{de} = 1$) cujo peso é zero ($w_{de} = 0$) representam casos que não contribuem para a função objetivo, incluindo as associações de pares inviáveis e as associações com elementos artificiais.

$$\sum_{e \in E} x_{de} = 1, \quad \forall d \in D, \tag{IV.2}$$

$$\sum_{d \in D} x_{de} = 1, \quad \forall e \in E, \tag{IV.3}$$

$$x_{de} \in \{0, 1\}, \quad \forall d \in D,\, e \in E. \tag{IV.4}$$

A formulação definida pelas Equações (IV.1)–(IV.4) caracteriza um problema de emparelhamento máximo ponderado em um grafo bipartido completo, no qual se busca uma correspondência ótima sob restrições de exclusividade.

IV.2 Algoritmo Húngaro aplicado ao problema de associações entre eventos e detecções

O problema apresentado na Seção IV.1 pode ser representado por meio de uma matriz de afinidades $W \in [0, 1]^{|D| \times |E|}$, construída diretamente a partir da função de pertinência do SoftED, de modo que cada entrada W_{ij} corresponde ao valor $\mu(d_i, e_j)$ atribuído à associação entre a detecção d_i e o evento e_j. Quando necessário, os conjuntos D e E podem ser estendidos com elementos artificiais (*dummy*) para tornar a matriz quadrada, sendo que associações envolvendo tais elementos (assim como pares inviáveis fora da janela temporal definida por k) recebem afinidade nula e, portanto, não contribuem para o valor da função objetivo. Como a formulação maximiza a soma das afinidades, associações com afinidade nula tendem a ser selecionadas apenas quando necessário, isto é, quando não há alternativas com pesos positivos.

Embora a formulação apresentada seja naturalmente expressa como um problema de maximização de afinidades, algoritmos clássicos de atribuição são usualmente definidos como problemas de minimização de custo. Essa diferença é resolvida por meio de uma transformação adequada da matriz de afinidades em uma matriz de custos, preservando a ordem relativa das soluções. Dessa forma, a modelagem aqui estabelecida fornece a base necessária para a aplicação de algoritmos

clássicos de emparelhamento.

De forma geral, os algoritmos destinados à resolução do problema de emparelhamento máximo ponderado em grafos bipartidos podem ser organizados em três grandes classes. A primeira compreende métodos baseados em caminhos aumentantes de custo mínimo (*shortest augmenting paths*) [Hung and Rom, 1980; Jonker and Volgenant, 1987]. A segunda engloba métodos do tipo primal-dual, entre os quais se destaca o Algoritmo Húngaro proposto por Kuhn [1955], bem como abordagens baseadas em fluxo máximo, como os algoritmos de Ford and Fulkerson [1963] e Edmonds and Karp [1972]. Por fim, a terceira classe inclui métodos baseados em programação linear, que variam desde o método Simplex geral até versões especializadas, como o Simplex de Transporte, explorando a estrutura particular do problema [Barr et al., 1977; Goldfarb, 1985].

As implementações mais eficientes dessas abordagens apresentam, no pior caso (para instâncias de dimensão $n \times n$), complexidade computacional da ordem de $O(N^3)$. Dado que todas as alternativas consideradas possuem complexidade assintótica equivalente, neste trabalho opta-se pela utilização do Algoritmo Húngaro, por se tratar de um método consolidado e amplamente estudado, além de contar com implementações robustas na linguagem em que o artefato SoftED foi desenvolvido. Essa escolha favorece a integração, a manutenção e a reprodutibilidade dos experimentos, sem prejuízo da qualidade da solução obtida. A seguir, apresenta-se uma descrição do Algoritmo Húngaro e de seus princípios fundamentais, contextualizando sua aplicação ao problema de associação entre eventos e detecções abordado neste trabalho.

O Algoritmo Húngaro, também conhecido como Algoritmo de Kuhn-Munkres, é uma técnica de otimização combinatória que resolve o problema de atribuição de forma eficiente. Esse problema consiste em associar elementos de dois conjuntos disjuntos de modo a minimizar o custo total ou maximizar o lucro total das associações [Kuhn, 1955].

No contexto do problema de associação entre detecções e eventos definido na Seção IV.1, o Algoritmo Húngaro opera sobre a matriz de custos obtida a partir da matriz de afinidades do método SoftED. A matriz de custos é definida como o oposto aditivo da matriz de afinidades, isto é, $C = -W$. Nessa representação, cada linha da matriz corresponde a uma detecção d_i, cada coluna corresponde a um evento e_j, e cada entrada C_{ij} representa o valor negativado da associação entre d_i e e_j, derivado da transformação da afinidade temporal $\mu(d_i, e_j)$. A Figura IV.1 ilustra um grafo bipartido completo e sua representação matricial para um exemplo simplificado de associações entre detecções (D1, D2 e D3) e eventos (E1, E2 e E3).

Após sua execução, o algoritmo identifica um conjunto de associações que minimiza o custo total. Essa solução corresponde à seleção globalmente ótima de pares detecção-evento que maximizam a afinidade temporal definida pelo método SoftED sob as restrições de exclusividade impostas ao problema. A Figura IV.2 exemplifica cada passo deste processo.

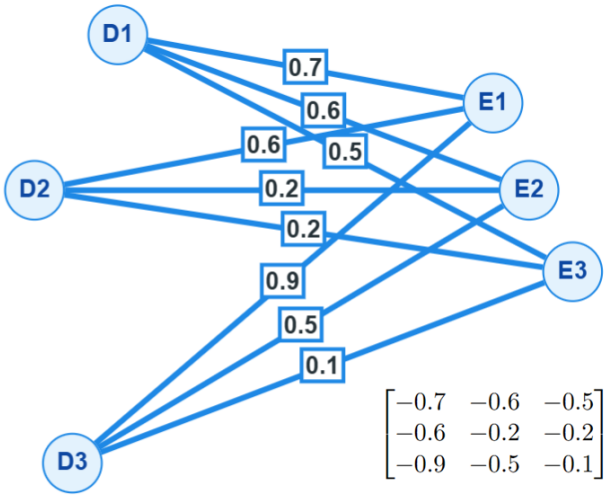


Figura IV.1: Emparelhamento em grafo bipartido completo com a representação da matriz de custo equivalente.

1. **Criação da matriz:** Quando os dois conjuntos têm o mesmo tamanho n, constrói-se uma matriz quadrada de dimensões $n \times n$, em que cada célula representa a associação entre uma detecção (linhas) e um evento (colunas). Caso os conjuntos tenham tamanhos diferentes, utiliza-se o tamanho do maior conjunto como referência para a construção da matriz $n \times n$. As posições correspondentes a associações inexistentes (devido à menor quantidade de elementos em um dos conjuntos) são preenchidas com 0. Dessa forma, garante-se que a matriz utilizada na otimização é sempre quadrada.

2. **Subtração das linhas:** Para cada linha da matriz, correspondente a uma detecção, identifica-se o menor valor e subtrai-se esse valor de todos os elementos da própria linha. Essa operação garante que cada linha contenha ao menos um zero, indicando a posição de menor custo (ou maior afinidade) nela.

3. **Subtração das colunas:** Para cada coluna da matriz, correspondente a um evento, identifica-se o menor valor e subtrai-se-o de todos os elementos da própria coluna. Assim como na etapa anterior, o objetivo é garantir que cada coluna contenha ao menos um zero, indicando a posição de menor custo nela.

4. **Cobertura de zeros:** Verifica-se se é possível associar cada detecção a um evento distinto utilizando apenas as posições com valor zero. Para isso, cobrem-se todos os zeros da matriz com o menor número possível de linhas e colunas. Se a quantidade total de linhas e colunas utilizadas nessa cobertura for igual a n, então uma atribuição ótima é encontrada. Caso contrário, segue-se para a próxima etapa do algoritmo.

5. **Ajuste da matriz:** Nesta etapa, busca-se identificar novas associações ótimas (elementos

não cobertos). Determina-se o menor valor não coberto na matriz (valor 0.2 no caso citado) e subtrai-se esse valor dos elementos não cobertos. Em seguida, adiciona-se o mesmo valor aos elementos cobertos duas vezes para preservar a coerência das transformações matriciais. O procedimento retorna à etapa de cobertura de zeros, estabelecendo um ciclo iterativo que se repete até a convergência para uma solução ótima de atribuição.

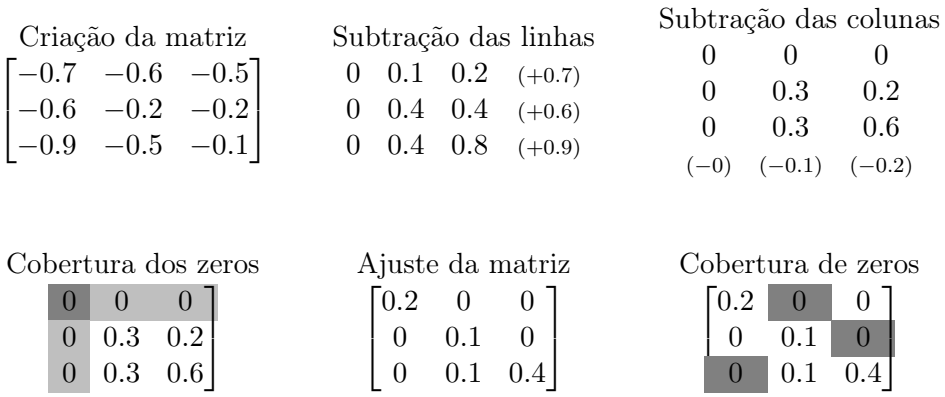


Figura IV.2: Exemplificação das etapas do Algoritmo Húngaro.

Ao final do processo ilustrado na Figura IV.2, obtém-se a atribuição ótima $[(1,2),(2,3),(3,1)]$, correspondente às associações $D1-E2$, $D2-E3$ e $D3-E1$ com pontuações 0.6, 0.2 e 0.9 respectivamente.

A complexidade computacional do Algoritmo Húngaro é $O(N^3)$, o que o torna viável para instâncias de dimensão moderada. Contudo, em cenários de larga escala ou sob restrições de tempo de execução, essa complexidade pode tornar-se um fator limitante.

Este trabalho busca resolver o problema inicial de associação entre detecções e eventos e, ao mesmo tempo, contornar essa limitação de complexidade computacional, segmentando o problema original em subproblemas disjuntos com três níveis de complexidade: simples, médio e complexo. A abordagem aplica o Algoritmo Húngaro apenas aos subproblemas complexos, o que viabiliza sua utilização em ambientes com restrições de desempenho.

Capítulo V Método

Inicialmente, observa-se que a avaliação de detecções de eventos pode ser realizada por meio das métricas flexíveis implementadas em SoftED. Contudo, o método de associação entre detecções e eventos adotado pelo SoftED não garante soluções ótimas. Nota-se que, esse problema de associação pode ser estruturado no paradigma da teoria dos grafos e compreendido como um problema de emparelhamento em grafos bipartidos [Bollobás, 1979]. O Algoritmo Húngaro [Kuhn, 1955] constitui uma abordagem eficiente para o problema de emparelhamento em grafos bipartidos, capaz de resolvê-lo de forma ótima. Tendo estas questões em vista, realiza-se um trabalho [Reis et al., 2024], apresentado nas Seções V.1 e V.2, que estrutura todo o problema nesse paradigma e o soluciona.

Como a complexidade computacional da nova solução é elevada, o método é revisado para reduzi-la. A intenção é viabilizar sua aplicação em contextos de larga escala ou em cenários com restrições de execução em tempo real. O resultado é uma estratégia híbrida, apresentada na Seção V.3, que divide o problema em subproblemas disjuntos e os classifica em casos simples, médios e complexos. Os casos simples são resolvidos por meio de associações diretas; os casos médios, por uma varredura linear; e os casos complexos, como problemas de atribuição, isto é, emparelhamentos em grafos bipartidos.

V.1 Função de pertinência

O cálculo da pontuação de cada detecção é realizado pela função de pertinência `detection_score`, que segue a mesma estrutura empregada em SoftED, conforme apresentado no Algoritmo 1. A função recebe como entradas o índice da detecção (d), o índice do evento (e) e o tamanho da janela temporal (k).

Algoritmo 1 Cálculo de `detection_score`(d, e, k)

1: **Função** *detection_score*(d, e, k)
2: **retorne** $\max\left(\min\left(\frac{d-(e-k)}{k}, \frac{(e+k)-d}{k}\right), 0\right)$

V.2 SoftEDPar (solução ótima para associação de detecções e eventos)

O método SoftEDPar [Reis et al., 2024] surge como uma melhoria do método SoftED [Salles et al., 2024], alterando apenas o mecanismo de associação entre detecções e eventos. Diferentemente do método SoftED, que realiza uma varredura de complexidade quadrática e adota associações gulosas, o método SoftEDPar emprega uma abordagem baseada na teoria dos grafos. Nele, todo o problema é estruturado como um emparelhamento máximo ponderado em um grafo bipartido completo desbalanceado. A implementação desse mecanismo de associação está descrita no Algoritmo 2.

Algoritmo 2 Associação ótima entre detecções e eventos (SoftEDPar)

```
1:  Função complex_cases_association(D, E, k)
2:      n ← length(D)
3:      m ← length(E)
4:      Inicialize matriz Mu de tamanho n × m
5:      para j ← 1 até m faça
6:          para i ← 1 até n faça
7:              Mu[i, j] ← detection_score(D[i], E[j], k)
8:      associationMatrix ← HungarianSolver(−Mu)
9:      scores ← Mu[associationMatrix.pairs]
10:     retorne scores
```

O algoritmo recebe os índices das detecções (D), os índices dos eventos (E) e o valor da janela temporal (k). Nas linhas 2 e 3, extraem-se o número de detecções (n) e o número de eventos (m). Em seguida, nas linhas 4 a 7, constrói-se a matriz de pertinência Mu, com dimensões $|D| \times |E|$, em que cada elemento Mu_{ij} representa a pontuação calculada pela função triangular de pertinência, indicando a proximidade entre a detecção $D[i]$ e o evento $E[j]$. Essa relação pode ser representada por um grafo bipartido $G = (D, E, A)$, cujas arestas $(d_i, e_j) \in A$ têm pesos definidos pelos valores de pertinência Mu_{ij}. Como essa operação requer iterar sobre todas as combinações possíveis, sua complexidade é $O(D \times E)$.

Um *matching* em G é definido como um subconjunto de arestas $P \subseteq A$ tal que cada vértice em D esteja conectado a, no máximo, um vértice em E, e vice-versa. Dada a matriz de pertinência Mu, o objetivo é encontrar um emparelhamento que maximize a soma dos valores das arestas selecionadas. Para resolver esse problema, utiliza-se o Algoritmo Húngaro (linha 8).

Como se observa na Seção IV.2, o Algoritmo Húngaro busca o emparelhamento de custo mínimo em uma matriz de custos C. Define-se uma matriz C pela negação dos valores da matriz Mu e aplica-se o algoritmo. Como resultado, obtém-se o subconjunto ótimo de arestas P, que representa os pares ideais entre detecções e eventos. Os emparelhamentos formados são ótimos [Kuhn, 1955]. Após estas etapas, os índices dos pares ótimos (i, j) são utilizados para acessar as pontuações correspondentes Mu_{ij}, e esses valores são armazenadas no vetor `scores` (linha 9), que compõe o resultado final.

A complexidade computacional dessa implementação é da ordem de $O((\max(|D|, |E|))^3)$. Consequentemente, a abordagem SoftEDPar apresenta complexidade cúbica, pois opera sobre toda a série sem qualquer forma de decomposição.

V.3 SmartSoftED

A partir dos resultados do SoftEDPar, identifica-se uma estratégia central para melhorar o desempenho computacional: reduzir o tamanho da entrada fornecida ao Algoritmo de Kuhn-Munkres (também conhecido como Algoritmo Húngaro). Esse algoritmo possui complexidade $O(N^3)$, o que torna seu uso direto caro em entradas muito grandes. Conclui-se que não é necessário estruturar todo o problema como um emparelhamento em um grafos bipartido.

Dependendo da configuração das localizações dos eventos e detecções e do tamanho da janela de tolerância temporal, surgem diferentes cenários, que podem ser tratados como três casos distintos:

- ***Casos simples.*** Ocorrem quando há apenas uma detecção dentro da área de pertinência de um evento. Nesse caso, aplica-se uma associação direta com complexidade constante $O(1)$.
- ***Casos médios.*** Ocorrem quando há apenas uma detecção dentro da área de pertinência de mais de um evento, ou quando há várias detecções dentro da área de pertinência de apenas um evento. Nesses cenários, realiza-se uma varredura que encontra a melhor associação com complexidade linear $O(N)$.
- ***Casos complexos.*** Ocorrem quando múltiplas detecções incidem em regiões em que funções de pertinência de eventos se sobrepõem, gerando conflitos de associação. Nesses cenários, a abordagem adotada em SoftEDPar mostra-se mais apropriada. A complexidade é cúbica, $O(N^3)$.

A proposta consiste em dividir o problema de associação entre detecções e eventos em subproblemas disjuntos por meio da identificação prévia de cada um dos três casos.

Quando a associação pode ser realizada por regras determinísticas, isto é, em casos simples, aplica-se um mapeamento direto de custo constante (O(1)). Nos casos resolvidos por procedimentos lineares, isto é, casos médios, utiliza-se uma estratégia intermediária capaz de obter a correspondência ótima em tempo $O(N)$. Por fim, quando nenhuma dessas abordagens é aplicável, isto é, em casos complexos, a instância é tratada como um grafo bipartido completo ponderado e resolvida por meio de otimização combinatória pelo Algoritmo Húngaro, em $O(N^3)$.

Essa decomposição assegura que apenas as instâncias genuinamente complexas sejam submetidas à etapa de emparelhamento bipartido, reduzindo substancialmente o custo computacional global.

V.3.1 Método principal

O funcionamento geral do método é apresentado no fluxograma da Figura V.1.

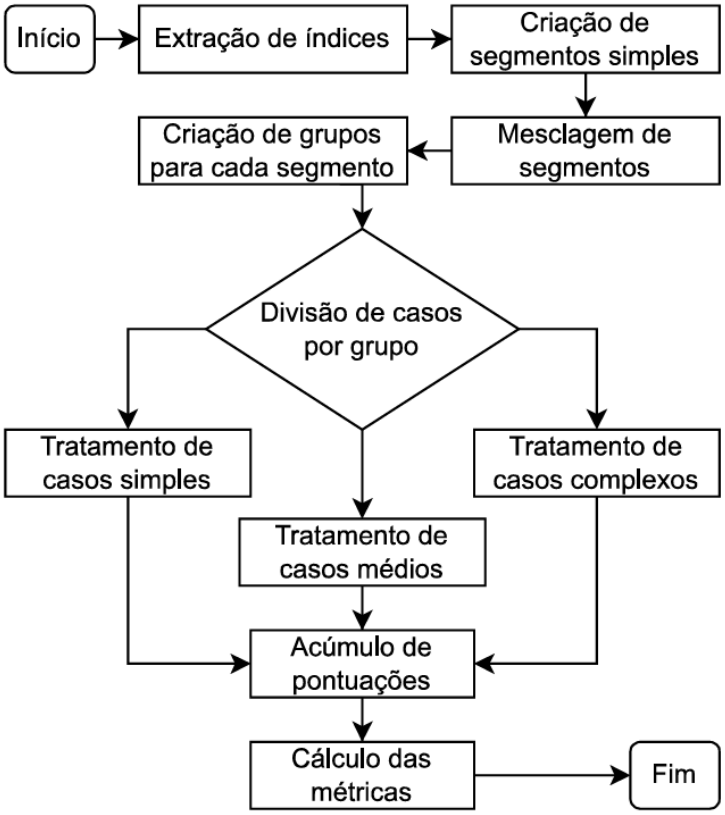


Figura V.1: Fluxograma com as etapas do método.

O processo inicia-se com a extração dos índices de eventos e de detecções. Em seguida, cada índice de evento serve como ponto central para a criação de *segmentos simples*, definidos pelos limites $[e - k, e + k]$. Esses segmentos são analisados em ordem crescente, verificando se o segmento atual apresenta sobreposição com o imediatamente seguinte. Sempre que houver interseção, os segmentos são mesclados, gerando novos segmentos que englobam todas as regiões de possível conflito entre funções de pertinência.

Após a mesclagem, cada segmento dá origem a um *grupo*, composto pelos subconjuntos de eventos e de detecções localizados dentro de seus limites. Cada grupo é classificado, segundo sua cardinalidade, como casos simples, médios ou complexos, e é encaminhado ao procedimento de associação correspondente. Por fim, as pontuações de todos os grupos são agregadas, permitindo o cálculo final da métrica. A formulação completa do método principal é apresentada no Algoritmo 3.

O método desenvolvido recebe como entrada dois vetores booleanos: `detection`, que indica os instantes em que ocorre uma detecção, e `event`, que indica os instantes em que ocorre um evento. Além desses vetores, a função também utiliza o parâmetro k, que corresponde ao tamanho da janela temporal. Como saída, o método retorna um vetor de pontuações S_d, que quantifica a qualidade

Algoritmo 3 Cálculo de $smart_soft_scores$ para avaliação suave de detecções

Função $smart_soft_scores(detection, event, k)$
  $D \leftarrow \{i \mid detection[i] = \text{TRUE}\}$ ▷ índices de detecção
  $E \leftarrow \{j \mid event[j] = \text{TRUE}\}$ ▷ índices de eventos
  $n \leftarrow length(D)$; $m \leftarrow length(E)$
  $segments \leftarrow \big[\, [e-k,\, e+k] \;\forall e \in E \,\big]$ ▷ janelas de tolerância
  $merged_segments \leftarrow merge_segments(segments)$
  $groups \leftarrow \emptyset$
  para cada $seg \in merged_segments$ **faça**
    $D_{seg} \leftarrow \{d \in D \mid seg.inf \leq d \leq seg.sup\}$
    $E_{seg} \leftarrow \{e \in E \mid seg.inf \leq e \leq seg.sup\}$
    $groups \leftarrow groups \cup \{(D_{seg}, E_{seg})\}$
  $S_d \leftarrow$ vetor de zeros de tamanho $length(D)$
  $p \leftarrow 1$ ▷ ponteiro para o vetor de pontuações S_d
  para cada $(D_{seg}, E_{seg}) \in groups$ **faça**
    $n \leftarrow length(D_{seg})$; $m \leftarrow length(E_{seg})$
    se $n = 1$ **e** $m = 1$ **então** ▷ caso simples
      $S_d[p] \leftarrow detection_score(D_{seg}[1], E_{seg}[1], k)$
      $p \leftarrow p + 1$
    senão se $n > 1$ **e** $m = 1$ **então** ▷ caso médio 1
      $vals \leftarrow \{detection_score(d, E_{seg}[1], k) \mid d \in D_{seg}\}$
      $S_d[p] \leftarrow \max(vals)$; $p \leftarrow p + 1$
    senão se $n = 1$ **e** $m > 1$ **então** ▷ caso médio 2
      $vals \leftarrow \{detection_score(D_{seg}[1], e, k) \mid e \in E_{seg}\}$
      $S_d[p] \leftarrow \max(vals)$; $p \leftarrow p + 1$
    senão se $n > 1$ **e** $m > 1$ **então** ▷ caso complexo
      $scores \leftarrow complex_cases_association(D_{seg}, E_{seg}, k)$
      para cada $s \in scores$ **faça**
        $S_d[p] \leftarrow s$; $p \leftarrow p + 1$
  retorne S_d

Procedimento $merge_segments(segments)$
  Ordene $segments$ por limite inferior
  $merged \leftarrow \emptyset$
  $current \leftarrow segments[1]$
  para $i \leftarrow 2$ **até** $nrow(segments)$ **faça**
    $next \leftarrow segments[i]$
    se $next.inf \leq current.sup$ **então**
      $current.sup \leftarrow \max(current.sup,\, next.sup)$ ▷ mescla sobreposição
    senão
      Acrescente $current$ em $merged$; $current \leftarrow next$
  Acrescente $current$ em $merged$
  retorne $merged$

das associações entre detecções e eventos, permitindo o cálculo da métrica posteriormente, de forma análoga ao SoftED [Salles et al., 2024].

Extração de índices e criação de segmentos

A função inicia extraindo os índices nos quais ocorrem detecções e eventos (linhas 2 e 3). Isso permite trabalhar apenas com os pontos relevantes dos vetores de entrada, evitando iterar sobre toda a série temporal. Considere D o vetor de índices das detecções e E o vetor de índices dos eventos. Para cada evento, constrói-se um intervalo que se estende de $(e-k)$ a $(e+k)$, definindo uma janela de tolerância temporal em torno de cada evento (linha 5). Essa etapa cria um conjunto de segmentos de tempo que delimita as regiões em que uma detecção pode ser associada a um evento.

Mesclagem de segmentos

Considerando a possibilidade de sobreposição, utiliza-se (linha 6) a função de mesclagem de segmentos: `merge_segments` (linhas 30 a 41). Essa função percorre a lista de segmentos em ordem crescente (linhas 34) e agrupa aqueles que se sobrepõem (linhas 36 e 37), formando um novo segmento, a união dos segmentos sobrepostos. Para reduzir a complexidade computacional, essa mesclagem é realizada por meio de uma varredura linear do primeiro ao último segmento.

A mesclagem ocorre da seguinte maneira: inicialmente, o primeiro segmento é definido como segmento ativo chamado: `current` (linha 33). A cada passo, o próximo segmento (`next`) é usado como comparação. Então é verificado se o limite inferior do próximo segmento é menor ou igual ao limite superior do segmento ativo (linha 36). Em caso afirmativo, o limite superior do segmento ativo é atualizado com o limite superior do segmento seguinte (linha 37). Caso contrário, o segmento ativo é finalizado e adicionado a uma nova lista de segmentos já processados. Em seguida, o próximo segmento passa a ser o novo segmento ativo, reiniciando o processo (linha 39). O procedimento é executado até que todos os segmentos sejam processados. O resultado obtido é fundamental, pois permite decompor o problema em subproblemas disjuntos.

Criação de grupos para cada segmento

Após a mesclagem de todos os segmentos sobrepostos, procede-se à etapa de criação de grupos (linhas 7 a 11). Para cada segmento resultante (linha 8), é construído um grupo (linha 11) que consiste em duas estruturas: um vetor contendo os índices das detecções pertencentes ao segmento, denotado por D_{seg} (linha 9), e outro vetor contendo os índices dos eventos associados ao mesmo segmento, denotado por E_{seg} (linha 10).

Cada grupo formado corresponde a um subproblema independente de associação entre detecções e eventos. Em particular, os grupos originados a partir de segmentos de sobreposição identificam as regiões da série temporal nas quais a associação entre eventos e detecções tende a apresentar maior complexidade.

Divisão de casos por grupo

Uma vez formados os grupos, o algoritmo itera sobre cada par $(D_{\text{seg}}, E_{\text{seg}}) \in$ `groups` (linha 14). Em seguida, define-se $n = |D_{\text{seg}}|$ como o número de detecções no grupo e $m = |E_{\text{seg}}|$ como o número de eventos (linha 15).

Cada grupo é então classificado em uma de quatro configurações possíveis, de acordo com as cardinalidades de D_{seg} e E_{seg}, sendo cada configuração encaminhada a um procedimento de associação específico. Quando $n = 1$ e $m = 1$, o grupo contém uma única detecção e um único evento, caracterizando o caso simples (linha 16). Quando $n > 1$ e $m = 1$, há múltiplas detecções associadas a um único evento, configurando o caso médio 1 (linha 19). Quando $n = 1$ e $m > 1$, há um único detecção para múltiplos eventos, caracterizando o caso médio 2 (linha 22). Por fim, quando $n > 1$ e $m > 1$, o grupo apresenta múltiplas detecções e múltiplos eventos, correspondendo ao caso complexo (linha 25).

Tratamento dos casos simples (um evento e uma detecção)

Nesse cenário, a associação é direta e o custo é constante, pois corresponde à chamada única da função `detection_score` para o par formado por $D_{\text{seg}}[1]$ e $E_{\text{seg}}[1]$. A pontuação obtida é armazenada em S_d (linha 17) e o ponteiro p é atualizado para a próxima posição livre (linha 18).

Tratamento dos casos médios tipo 1 (um evento e múltiplas detecções)

Nesse cenário, há um único evento e várias detecções dentro da mesma janela de tolerância. Cada detecção em D_{seg} é avaliada em relação ao evento único em $E_{\text{seg}}[1]$ por meio da função `detection_score` (linha 20), e a maior pontuação entre elas é selecionada (linha 21). O custo desse processo é linear no número de detecções do grupo.

Tratamento dos casos médios tipo 2 (múltiplos eventos e uma detecção)

Nesse cenário, há uma única detecção e vários eventos na mesma região de pertinência. A detecção em $D_{\text{seg}}[1]$ é avaliada em relação a cada evento em E_{seg} utilizando a função `detection_score` (linha 23), e, novamente, seleciona-se a maior pontuação (linha 24). Assim como no caso médio do tipo 1, o custo é linear em função do número de eventos do grupo.

Tratamento dos casos complexos (múltiplas detecções e múltiplos eventos)

Nos casos complexos, há múltiplas detecções e múltiplos eventos em uma mesma região de pertinência. Nessa configuração, podem ocorrer conflitos de associação, pois uma detecção pode estar próxima de vários eventos e, analogamente, um evento pode estar próximo de várias de-

tecções, o que impossibilita a definição de um par ótimo por regras locais simples. Devido a esse fator, todas as detecções em D_{seg} e todos os eventos em E_{seg} são processados pela função `complex_cases_association` (linha 26), que emprega o procedimento de otimização combinatória introduzido em SoftEDPar [Reis et al., 2024].

Observe que somente esses segmentos complexos da série são submetidos ao processo de otimização cúbica, em contraste com SoftEDPar, no qual toda a série era tratada dessa forma. O procedimento segue a mesma lógica da formulação de SoftEDPar, implementada no Algoritmo 2, o qual é explicado a seguir.

Primeiro, constrói-se uma matriz de pontuações **Mu** de dimensões $|D_{\text{seg}}| \times |E_{\text{seg}}|$ (linha 4), em que cada entrada $\mathbf{Mu}[i, j]$ representa a afinidade entre a i-ésima detecção de D_{seg} e o j-ésimo evento de E_{seg}, calculada por `detection_score` (linha 7). Em seguida, aplica-se o Algoritmo Húngaro (via `HungarianSolver`) à matriz $-\mathbf{Mu}$, obtendo-se o emparelhamento ótimo que maximiza a soma global das pontuações (linha 8), conforme discutido na Seção IV.2.

Os valores de **Mu** correspondentes aos pares escolhidos são então extraídos e armazenados no vetor `scores`, que contém as pontuações associadas às detecções desse grupo (linha 9). Em seguida, os valores de `scores` são inseridos no vetor global de pontuações S_d (voltando ao Algoritmo 3, linha 27 a 28). O custo dessa etapa é cúbico, porém limitado ao tamanho do grupo, isto é, ao número de detecções e eventos considerados nessa associação.

Acúmulo de pontuações e cálculo das métricas

Conforme a lista de grupos é percorrida linearmente, cada caso é resolvido de acordo com sua configuração. As pontuações das associações são armazenadas no vetor S_d até que os resultados de todas as pontuações sejam consolidados. Por fim, esse vetor é retornado pela função `smart_soft_scores` e utilizado para criar as versões flexíveis de TP, FP, TN e FN, conforme demonstrado anteriormente na Tabela II.2.

V.3.2 Análise de Complexidade Computacional

A análise da complexidade computacional do método considera os principais blocos do algoritmo. Na etapa de extração de índices (linhas 2 e 3), as operações utilizadas têm complexidade $O(|X|)$, onde X é o tamanho da série. Se D e E representam o número de detecções e de eventos, respectivamente, a etapa de criação dos segmentos tem custo $O(E)$. Após a ordenação inicial dos segmentos, a etapa de mesclagem de segmentos realiza-se em uma única passagem sobre eles, o que também resulta em $O(E)$.

Para cada segmento mesclado, realiza-se uma filtragem para extrair os subconjuntos D_{seg} e E_{seg}, que contêm, respectivamente, as detecções e os eventos dentro dos limites do intervalo considerado.

Ainda que essa operação se repita para cada grupo, os intervalos não se sobrepõem. Por isso, cada índice de D e de E é processado, no máximo, uma vez ao longo de toda a execução. Assim, o custo agregado permanece linear, com complexidade $O(D + E)$.

Considere g o número total de grupos formados após o particionamento, sendo:

- $g_{simples}$: número de grupos classificados como casos simples (um único evento e uma única detecção);
- g_{medio1}: número de grupos classificados como casos médios em que um único evento pode ser associado a mais de uma detecção;
- g_{medio2}: número de grupos classificados como casos médios em que vários eventos podem ser associados a uma única detecção;
- $g_{complexo}$: número de grupos classificados como casos complexos (múltiplos eventos e múltiplas detecções).

Assim, $g = g_{simples} + g_{medio1} + g_{medio2} + g_{complexo}$.

Em cada grupo simples, a associação entre detecção e evento é direta e ocorre por meio da função `detection_score`, de custo constante ($O(1)$). Portanto, a complexidade total dos grupos simples é $O(g_{simples})$. Cada grupo médio é resolvido por meio de uma varredura linear que aplica a função `detection_score` a cada possível associação e, em seguida, outra varredura linear que encontra o maior valor. O grupo médio 1 realiza a varredura das detecções; logo, a complexidade total para os grupos médios 1 é $O(g_{medio1} \times d)$, em que d representa o número de detecções em cada grupo desse tipo. O grupo médio 2 faz a varredura sobre os eventos, logo, a complexidade total para os grupos médios 2 é $O(g_{medio2} \times e)$, em que e representa o número de eventos em cada grupo desse tipo.

Em cada grupo complexo, com d detecções e e eventos, duas operações são realizadas. Primeiro, ocorre a construção da matriz de pontuações **Mu**, com custo $O(d \times e)$. Depois, aplica-se o Algoritmo Húngaro para resolver o problema de alocação ótima entre detecções e eventos, com complexidade $O(\max(d, e)^3)$. Assim, a complexidade total associada a cada grupo complexo é $O((d \times e) + \max(d, e)^3)$. A complexidade total do algoritmo pode ser observada na Equação V.1.

$$O\left(D + E + g_{simples} + \sum_{g_{medio1}} d + \sum_{g_{medio2}} e + \sum_{g_{complexo}} \max(d, e)^3\right) \tag{V.1}$$

Essa análise mostra que, na ausência de casos complexos ($g_{complexo} = 0$), a métrica apresenta comportamento linear. Como os segmentos não se sobrepõem, cada detecção e cada evento pertencem exatamente a um único grupo, garantindo que todos os elementos de D e E sejam processados apenas uma vez. Assim, a complexidade global reduz-se para $O(D + E)$, conforme evidenciado na Equação V.2.

$$O\left(D + E + g_{simples} + \sum_{g_{medio1}} d + \sum_{g_{medio2}} e\right) = O(D + E). \tag{V.2}$$

No pior caso, a tolerância temporal de todos os eventos da série se sobrepõe, resultando na formação de um único grande segmento, no qual $g_{\text{simples}} = g_{\text{medio1}} = g_{\text{medio2}} = 0$ e $g_{\text{complexo}} = 1$. Nessa situação, a matriz **Mu** passa a conter todas as detecções e eventos, e o método SmartSoftED passa a operar com complexidade cúbica, de forma análoga ao comportamento do SoftEDPar [Reis et al., 2024].

Em cenários usuais, a área de abrangência das janelas temporais é pequena em relação ao intervalo entre ocorrências de eventos, o que faz com que os casos simples e médios sejam muito mais numerosos do que os casos complexos. Como consequência, o custo médio de processamento é substancialmente reduzido.

V.3.3 Estudo de caso do método

Apresenta-se, nesta seção, um estudo de caso do método utilizando uma série sintética, ilustrada na Figura V.2. Essa série reúne diferentes tipos de ocorrências de eventos e detecções, abrangendo diversas possibilidades de associação entre eles. Os números de 1 a 49 representam os índices da série. Os retângulos verdes representam eventos, os vermelhos detecções. A linha verde representa o espaço de cobertura da janela temporal. O objetivo é ilustrar o funcionamento do método de forma holística. Para este estudo, o valor da variável de tolerância temporal (k) foi definido como 3.

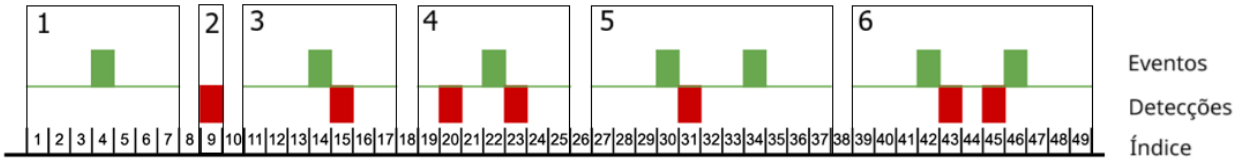


Figura V.2: Exemplo de eventos e detecções em série sintética.

Os seis casos ocorrem em sequência e estão definidos na Tabela V.1. Cada linha representa um caso, com sua respectiva descrição, a abordagem que o algoritmo adota e a pontuação obtida. Conforme já mostrado na Tabela II.2, os valores das métricas (TP, FP, TN, FN) são todos derivados do vetor de pontuações das detecções ds. O caso 1 (FN) e o caso 2 (FP) são ignorados, pois um evento não detectado e uma detecção fora das regiões temporais de eventos não geram pontuação. Ignorá-los também evita processamento desnecessário, reduzindo o custo computacional da avaliação.

O caso 3 produz uma pontuação decorrente da associação direta. O caso 4 produz a pontuação correspondente à associação da detecção mais próxima do evento único e, no caso 5, a pontuação correspondente à associação do evento mais próximo à única detecção. O caso 6 produz um vetor de

pontuações com as 2 melhores associações possíveis. Ao fim, todos esses valores são concatenados em um único vetor de pontuações, utilizado para o cálculo das métricas.

Caso	Descrição	Abordagem	Pontuação
1	Uma região de evento sem detecção	Região ignorada	-
2	Uma detecção fora de região de evento	Detecção ignorada	-
3	Uma região de evento com uma detecção	Caso simples	$0.\overline{6}$
4	Uma região de evento com duas detecções	Caso médio 1	$0.\overline{6}$
5	Duas regiões sobrepostas de evento com uma detecção	Caso médio 2	$0.\overline{6}$
6	Duas regiões sobrepostas de evento com duas detecções	Caso complexo	$[0.\overline{6}, 0.\overline{6}]$

Tabela V.1: Casos de associação entre eventos e detecções.

Os casos 1 e 2 não acrescentam nenhum valor ao vetor de pontuações. O caso 3 é tratado como um caso simples e executa a função `detection_score` com $d = 15$ e $e = 14$.

$$\begin{aligned}
\texttt{detection_score}(15, 14, 3) &= \max\left(\min\left(\frac{15-(14-3)}{3}, \frac{(14+3)-15}{3}\right), 0\right) \\
&= \max\left(\min\left(\frac{15-11}{3}, \frac{17-15}{3}\right), 0\right) \\
&= \max\left(\min\left(\frac{4}{3}, \frac{2}{3}\right), 0\right) \\
&= \max\left(\frac{2}{3}, 0\right) \\
&= \frac{2}{3} = 0.\overline{6}.
\end{aligned}$$

O caso 4 é tratado como caso médio 1 (com $e = 22$ e $d \in \{20, 23\}$):

$$\max\{\texttt{detection_score}(20, 22, 3), \texttt{detection_score}(23, 22, 3)\} = 0.\overline{6}$$

O caso 5 é considerado médio 2 (com $d = 31$ e $e \in \{30, 34\}$):

$$\max\{\texttt{detection_score}(31, 30, 3), \texttt{detection_score}(31, 34, 3)\} = 0.\overline{6}$$

O caso 6 é tratado como complexo. A função `complex_cases_association` é aplicada à lista de detecções D e à lista de eventos E. A matriz Mu é inicializada, e cada um de seus elementos $Mu[i, j]$ é preenchido pelos resultados da função `detection_score` aplicada a cada par (i, j).

$$\texttt{detection_score}(1, 1, 3) = \max\left\{\min\left(\tfrac{43-(42-3)}{3}, \tfrac{(42+3)-43}{3}\right), 0\right\} = \max\left\{\min\left(\tfrac{4}{3}, \tfrac{2}{3}\right), 0\right\} = 0.\overline{6}$$

$$\texttt{detection_score}(2,1,3) = \max\left\{\min\left(\frac{43-(46-3)}{3},\ \frac{(46+3)-43}{3}\right),0\right\} = \max\{\min(0,\ 2),0\} = 0$$

$$\texttt{detection_score}(1,2,3) = \max\left\{\min\left(\frac{45-(42-3)}{3},\ \frac{(42+3)-45}{3}\right),0\right\} = \max\{\min(2,\ 0),0\} = 0$$

$$\texttt{detection_score}(2,2,3) = \max\left\{\min\left(\frac{45-(46-3)}{3},\ \frac{(46+3)-45}{3}\right),0\right\} = \max\left\{\min\left(\frac{2}{3},\ \frac{4}{3}\right),0\right\} = 0.\overline{6}$$

Depois, essa matriz é utilizada como entrada para o Algoritmo Húngaro. Nas linhas ou colunas sem associação (matrizes retangulares), o próprio algoritmo cria linhas ou colunas extras com valores iguais a 0 antes de efetuar as operações. Após esse preenchimento, obtém-se a matriz que serve de base para a etapa de otimização.

Caso 6

	42	46
43	$0.\overline{6}$	0
45	0	$0.\overline{6}$

Após isso, executa-se o Algoritmo Húngaro para encontrar as atribuições ótimas. A Tabela V.2 demonstra como o processo ocorre no caso 6.

Inversão de sinal	Subtração das linhas	Subtração das colunas	Cobertura dos zeros
$\begin{bmatrix} -0.\overline{6} & 0 \\ 0 & -0.\overline{6} \end{bmatrix}$	$\begin{matrix} 0 & 0.\overline{6} & (-(-0.\overline{6})) \\ 0.\overline{6} & 0 & (-(-0.\overline{6})) \end{matrix}$	$\begin{matrix} 0 & 0.\overline{6} \\ 0.\overline{6} & 0 \\ (-0) & (-0) \end{matrix}$	$\begin{bmatrix} 0 & 0.\overline{6} \\ 0.\overline{6} & 0 \end{bmatrix}$

Tabela V.2: Passo a passo da execução do algoritmo húngaro para o caso 6.

Como se vê, os pares de índices que criam as correspondências detecção-evento selecionados para o caso 6 são [43,42] e [45,46], contendo as associações ótimas de valor $0.\overline{6}$ cada. As associações estão ilustradas na matriz abaixo:

Caso 6

	42	46
43	$0.\overline{6}$	0
45	0	$0.\overline{6}$

Ao final do processo, todas essas pontuações são concatenadas em um único vetor (ds). Esse vetor é utilizado para o cálculo das métricas, seguindo o mesmo procedimento das métricas SoftED originais [Salles et al., 2024]. A formalização desse processo foi apresentada anteriormente na Tabela II.2. A Tabela V.3 apresenta as métricas finais calculadas neste estudo de caso.

Métrica	**Valor**
TP_s	$3.33\overline{3}$
FP_s	$1.66\overline{6}$
FN_s	$3.66\overline{6}$
TN_s	$40.33\overline{3}$
Precisão (P)	$0.66\overline{6}$
Revocação (R)	0.476
F_1	$0.55\overline{5}$

Tabela V.3: Métricas calculadas SmartSoftED.

Capítulo VI Avaliação experimental

Neste capítulo, são apresentados os resultados dos testes das métricas SoftED [Salles et al., 2024], SoftEDPar [Reis et al., 2024] e SmartSoftED, realizados sobre conjuntos de dados integrados em um repositório chamado *Unified Time Series Event Detection Repository* (UniTED) [Lima et al., 2025]. O UniTED é um repositório unificado de séries temporais rotuladas para detecção de eventos (com foco em anomalias) que padroniza o acesso, os metadados e os processos de *Extract, Transform, Load* (ETL).

Entre os principais conjuntos de dados utilizados, destaca-se UCI 3W (*Oil Well*), composto por séries multivariadas de poços de petróleo com ocorrência rara de eventos indesejáveis; trata-se de um conjunto de dados realista e público, com diferentes tipos de eventos e variáveis de processo, disponível no *UCI Machine Learning Repository*. O UCR Anomaly Archive consiste em uma coleção de aproximadamente 250 séries univariadas rotuladas com anomalias, criada para mitigar limitações comuns em *benchmarks*, como a densidade irreal de anomalias e a trivialidade das detecções, e é amplamente utilizada em comparações sistemáticas. O MIT-BIH Arrhythmia Database contém sinais de eletrocardiograma (ECG) rotulados (amostrados a 360 Hz), com anotações detalhadas. O conjunto Yahoo Webscope S5 (*Yahoo Labs*) reúne quatro subgrupos, sendo o *A1* composto por dados reais (como tráfego *web*) e os *A2–A4* por séries sintéticas que incorporam variações contextuais e mudanças de tendência, constituindo um *benchmark* clássico para detecção de anomalias em séries temporais. O NAB (*Numenta Anomaly Benchmark*) contém cerca de 50 arquivos com séries reais e artificiais provenientes de diversos domínios, como *Cloudwatch*, *AdExchange*, tráfego e redes sociais, e é acompanhado de um protocolo de pontuação voltado a cenários de detecção em fluxo contínuo (*streaming/real-time*). Por fim, o conjunto GECCO (*Industrial Challenge* 2018) contém dados multivariados de sensores ambientais e operacionais de sistemas de tratamento de água, com eventos anotados e utilizados em competições industriais de detecção *online*.

A Figura VI.1 apresenta um gráfico de barras com o tamanho médio das séries para cada conjunto de dados, enquanto a Figura VI.2 mostra diversos *boxplots* com a densidade de eventos e detecções em relação ao tamanho das séries. Em conjunto, essas figuras permitem construir uma intuição sobre como a combinação entre o comprimento das séries e a densidade de eventos e detecções pode influenciar o desempenho computacional do método.

Pode-se observar que os conjuntos MIT e GECCO apresentam as maiores séries temporais,

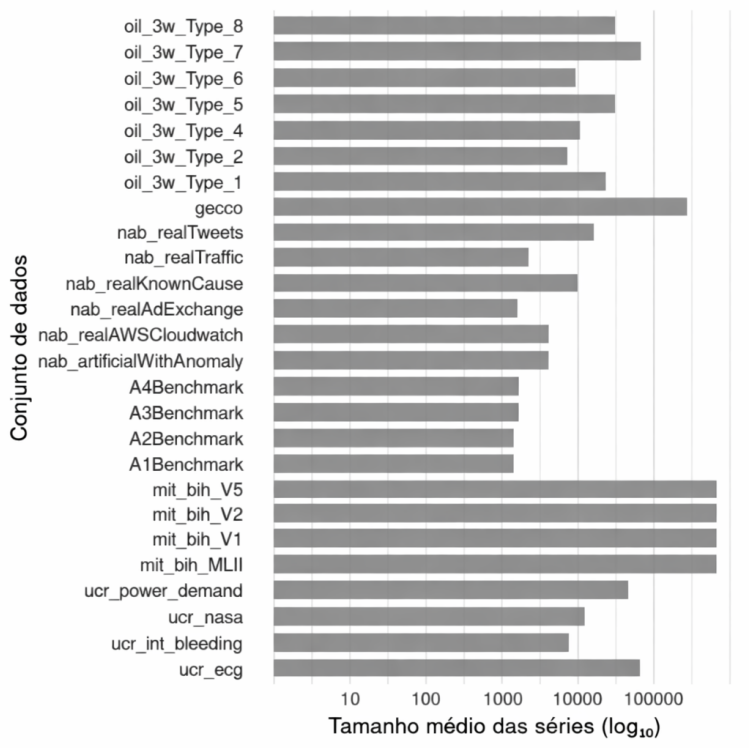


Figura VI.1: Tamanho médio das séries dos conjuntos de dados.

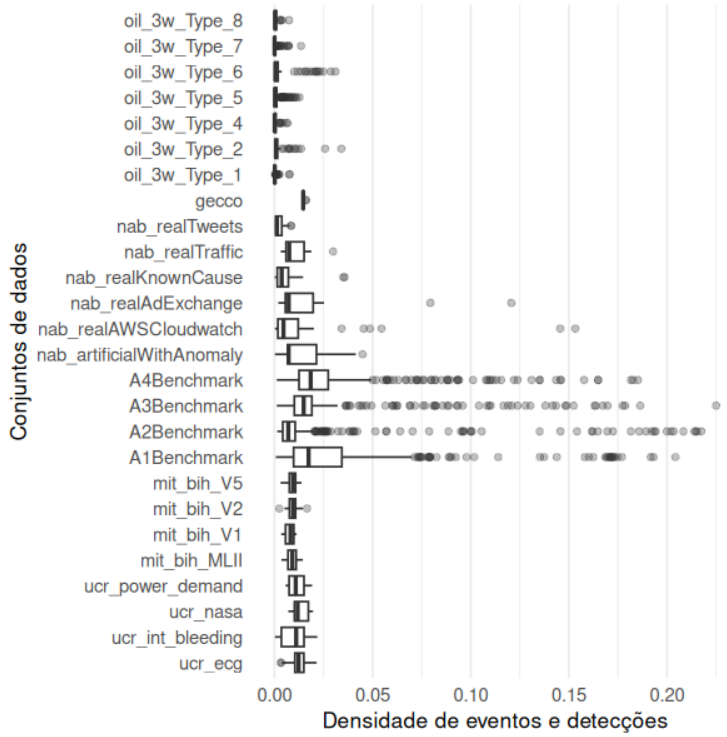


Figura VI.2: Ilustração da densidade que a soma de eventos e detecções ocupa em cada conjunto de dados, em relação ao tamanho das séries.

enquanto o conjunto *Yahoo Labs* apresenta séries temporais menores. Com relação à densidade da ocorrência de eventos e detecções, o conjunto *Yahoo Labs* apresenta maior densidade e maior variabilidade de densidades do que todos os outros conjuntos de dados. Percebe-se também que, na

maioria dos casos, a densidade de eventos e de detecções é inferior a 2,5% do tamanho das séries.

Os experimentos foram conduzidos empregando cinco algoritmos de detecção de eventos (REMD, FBIAD, ARIMA, ELM e LSTM), de modo a contemplar tanto abordagens clássicas quanto modelos baseados em aprendizagem profunda. Por limitações computacionais, o algoritmo REMD não foi aplicado aos conjuntos de dados de maior porte (UCI, UCR, MIT e GECCO). Adicionalmente, o método ELM não pôde ser executado no conjunto UCI devido a falhas técnicas recorrentes durante a execução experimental. Ao todo, foram realizados 3.071 testes experimentais. No total, o tempo computacional despendido foi de aproximadamente 20 horas, contemplando a execução dos detectores, o armazenamento dos resultados em disco para posterior reutilização e a avaliação das detecções pelos diferentes métodos considerados (SoftED, SoftEDPar e SmartSoftED).

VI.1 Experimento para verificação de manutenção de otimalidade

Nesta seção, realiza-se uma verificação empírica da manutenção da otimalidade da técnica proposta, com o objetivo de assegurar que os ganhos previamente alcançados pela melhoria introduzida em SoftEDPar [Reis et al., 2024] sejam integralmente preservados. A Figura VI.3 apresenta dois gráficos de coordenadas paralelas contendo os valores de precisão e de revocação obtidos para cada teste nas três métricas avaliadas.

Cada linha nos gráficos de coordenadas paralelas corresponde à conexão dos valores de um teste experimental individual para as três métricas (SoftED, SoftEDPar e SmartSoftED), totalizando 3.071 instâncias avaliadas. Os resultados de precisão estão à esquerda, enquanto os de revocação, à direita. Dessa forma, diferenças entre métodos manifestam-se como desvios verticais ou inclinações distintas entre os eixos correspondentes.

Observa-se que não há desvio vertical entre os valores de SoftEDPar e SmartSoftED, indicando que, para todos os testes realizados, as pontuações de precisão e revocação obtidas por esses dois métodos são idênticas. Esse comportamento evidencia que a reformulação proposta não altera os resultados da avaliação, preservando integralmente as medidas de precisão e revocação e, consequentemente, a otimalidade previamente alcançada pelo SoftEDPar.

Por outro lado, nota-se a presença de variações entre SoftED e SoftEDPar/SmartSoftED, com deslocamentos ascendentes em algumas linhas, os quais refletem ganhos de precisão ou de revocação decorrentes da introdução de associações ótimas. Assim, a figura confirma, de forma robusta, por meio de evidência quantitativa em larga escala, que o SmartSoftED mantém os mesmos resultados ótimos do SoftEDPar, ao mesmo tempo em que preserva as melhorias em relação ao SoftED.

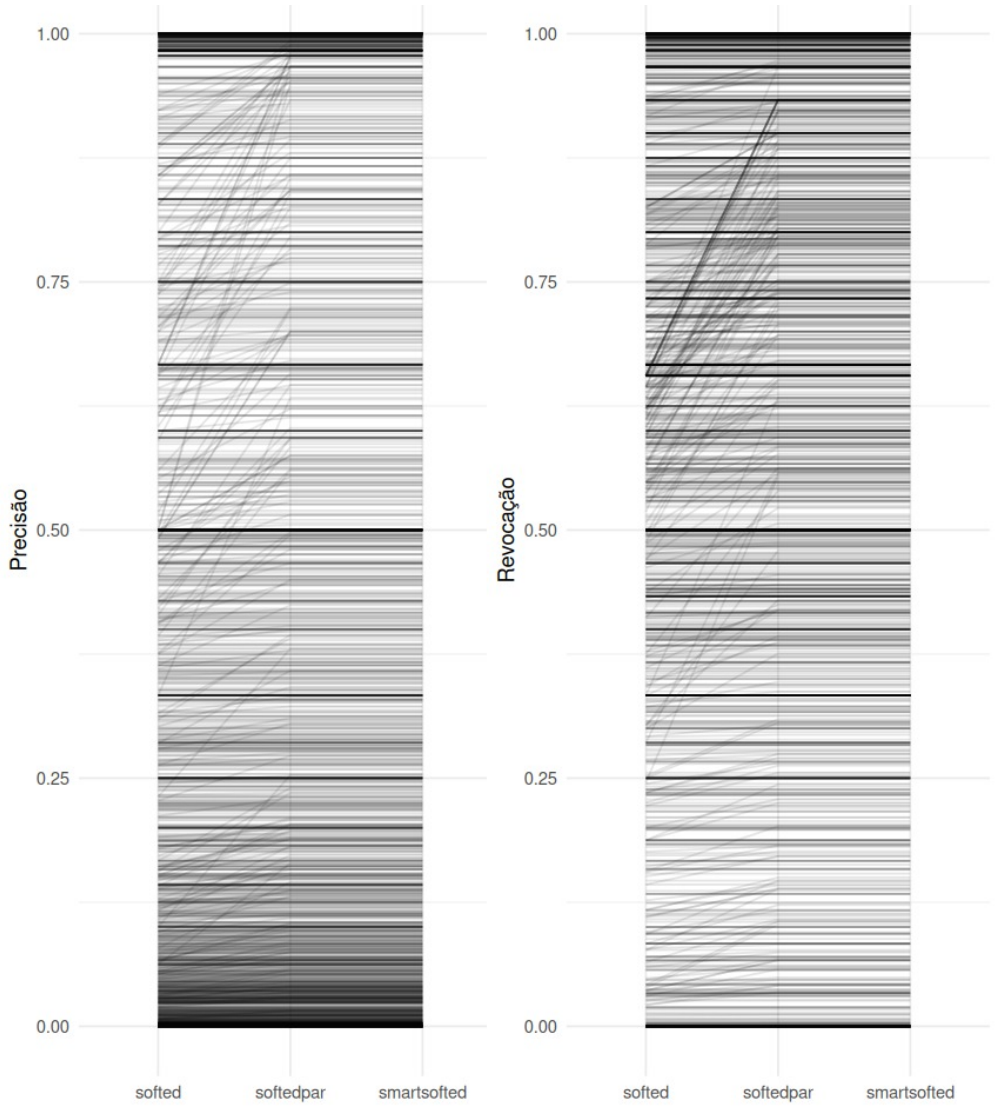


Figura VI.3: Gráfico de coordenadas paralelas ilustrando os valores de precisão e revocação obtidos pelos métodos SoftED, SoftEDPar e SmartSoftED em cada um dos 3.071 testes experimentais realizados.

VI.2 Eficiência computacional

Do ponto de vista do desempenho computacional, espera-se que o SmartSoftED proporcione ganhos significativos, viabilizando sua aplicação tanto em conjuntos de dados de maior porte quanto em cenários de avaliação em tempo real. A Figura VI.4 apresenta uma comparação do tempo computacional das métricas SoftED, SoftEDPar e SmartSoftED, expressa em termos relativos. Para cada conjunto de dados, o tempo total de execução dos três métodos foi medido e normalizado em relação ao método mais rápido. Nesse contexto, o método com menor tempo de execução assume valor unitário (1), enquanto os demais são representados por fatores multiplicativos que indicam o quanto são mais lentos em comparação ao método mais eficiente.

Os resultados indicam que o SmartSoftED apresenta ganhos computacionais substanciais na maior parte dos conjuntos avaliados, sendo em média 226 vezes mais rápido do que o SoftED e

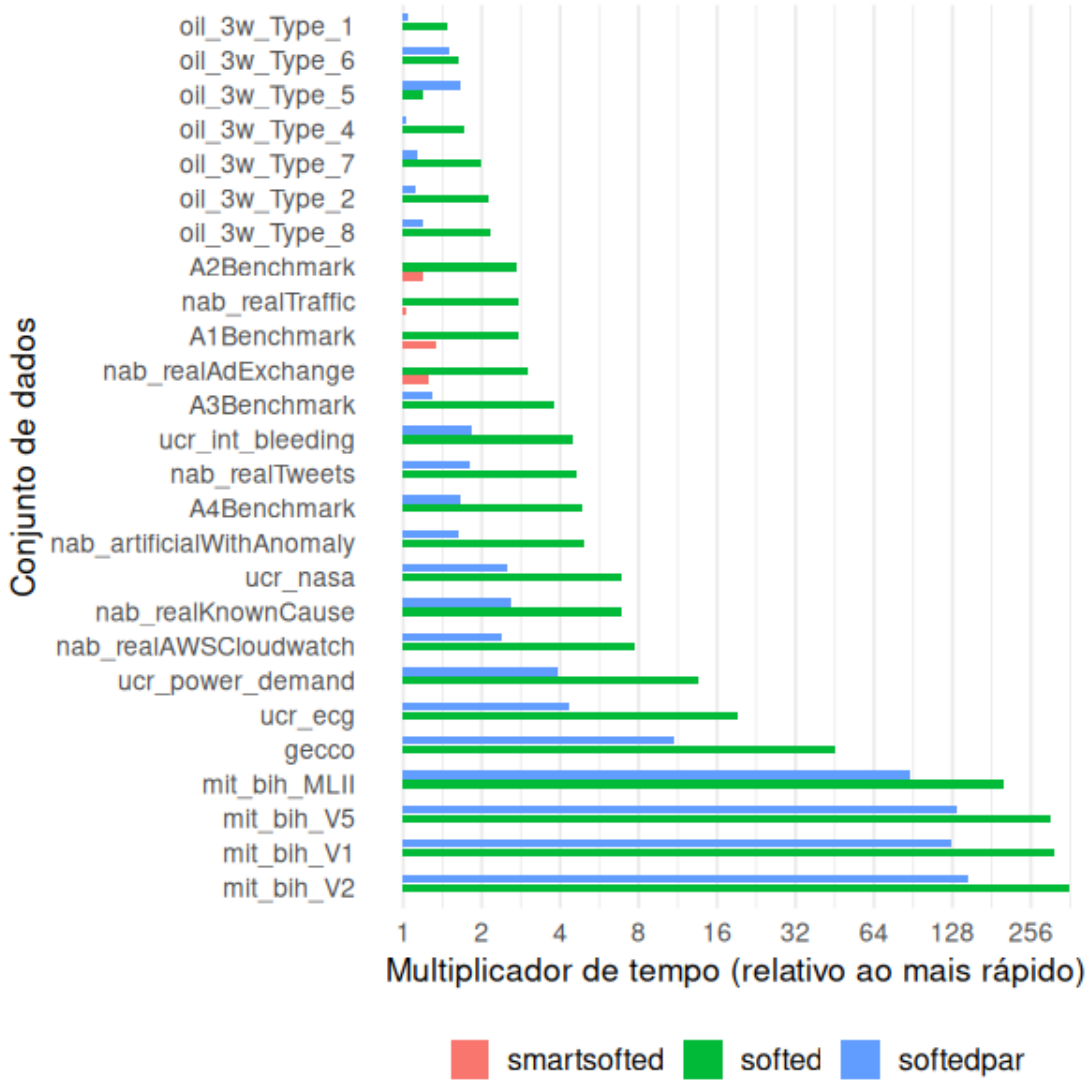


Figura VI.4: Comparação do tempo de execução relativo das métricas SoftED, SoftEDPar e SmartSoftED nos diferentes conjuntos de dados.

94 vezes mais rápido do que o SoftEDPar. A melhoria de desempenho torna-se mais evidente em conjuntos de dados maiores. Observaram-se apenas quatro casos em que o SmartSoftED não apresentou o melhor desempenho (situações onde a barra vermelha aparece), ficando abaixo do desempenho do SoftEDPar. Nessas situações, tratava-se de conjuntos de dados pequenos, com poucas detecções e eventos, em que o tempo médio de avaliação foi inferior a 0,1 segundos.

A Figura VI.5 apresenta as proporções de casos simples, médios e complexos em cada conjunto de dados, permitindo visualizar como a composição estrutural das séries influencia o tempo de execução do método.

Observa-se que, para um dado conjunto de dados, quanto maior o comprimento das séries e menor a incidência de casos classificados como complexos, maiores são as acelerações obtidas pelo método. No conjunto MIT, por exemplo, o SmartSoftED chegou a alcançar ganhos superiores a 256× em relação ao SoftED (Figura VI.4, conjuntos de dados: mit_bih_V5, mit_bih_V1 e mit_bih_V2).

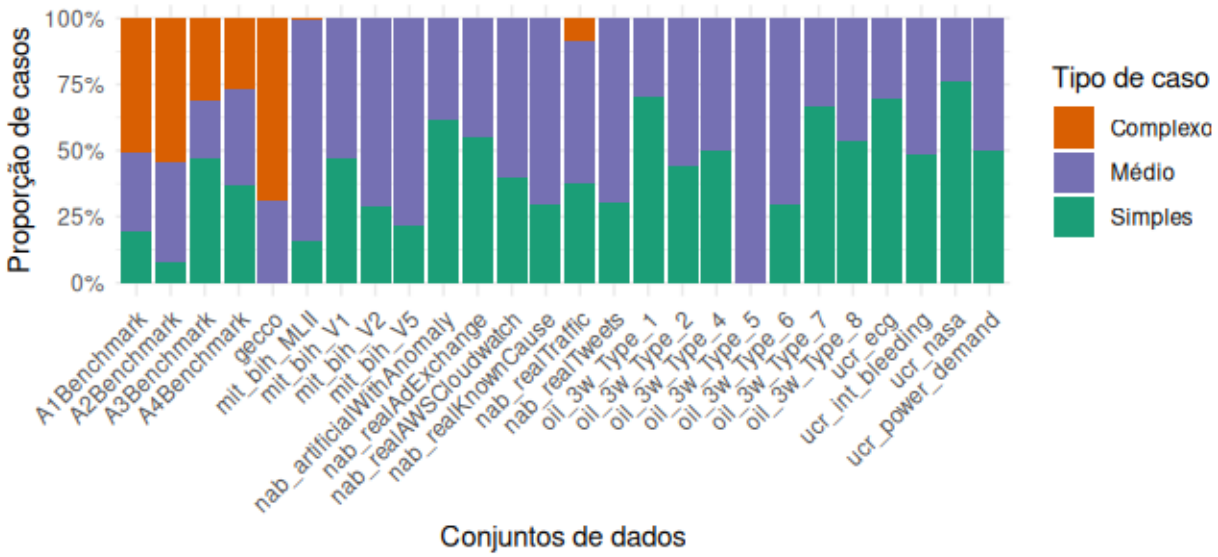


Figura VI.5: Proporção de eventos e detecções tratadas como casos simples, médios ou complexos em cada conjunto de dados.

Esse comportamento evidencia a escalabilidade do SmartSoftED e confirma a expectativa de que a introdução de mecanismos mais eficientes de associação entre detecções e eventos torna a métrica mais adequada para a avaliação de grandes bases de séries temporais.

VI.2.1 Comportamento empírico da complexidade computacional nos conjuntos de dados

A complexidade computacional do método foi apresentada na Seção V.3.2; no entanto, buscou-se determinar o comportamento empírico da complexidade computacional nos conjuntos de dados testados. Verificou-se que, para que o algoritmo apresente complexidade assintoticamente linear, é suficiente que o custo cúbico acumulado de todos os casos complexos seja limitado, em ordem de grandeza, por $D + E$. Seja $\mathcal{G}_{\text{complexo}}$ o conjunto de grupos classificados como complexos, e d_g e e_g o número de detecções e de eventos no grupo g, respectivamente; para garantir que o crescimento dos subproblemas cúbicos não domine o custo total, é necessário que a Equação VI.1 seja satisfeita.

$$\sum_{g \in \mathcal{G}_{\text{complexo}}} \max\left(d_g, e_g\right)^3 \;=\; O(D + E). \tag{VI.1}$$

Como forma prática de mensurar esse comportamento em cada instância, foi criada uma variável booleana chamada `linear_behavior` (Equação VI.2), que verifica se o custo cúbico acumulado de todos os casos complexos é menor ou igual à soma do número total de eventos e de detecções. Quando essa condição é satisfeita, considera-se que o comportamento observado está contido no regime linear esperado.

$$\text{linear_behavior} \;=\; \left(\sum_{g \in \mathcal{G}_{\text{complexo}}} \max\left(d_g, e_g\right)^3 \;\leq\; D + E \right). \tag{VI.2}$$

A Figura VI.6 apresenta, por conjunto de dados, a proporção de casos em que o algoritmo apresenta comportamento linear ou não. É possível observar que o conjunto de dados GECCO não apresenta comportamento linear em nenhum dos casos. Isso se deve ao fato de que os eventos desse conjunto de dados são intervalares, o que aciona muitos casos complexos (de complexidade cúbica). Ainda assim, na Figura VI.4 podemos observar que, nesse conjunto de dados, o SmartSoftED superou o SoftED em 48× e o SoftEDPar em 11×. Apesar da ausência de comportamento linear, a Figura VI.5 indica que, em GECCO, pouco mais de 25% das associações foram tratadas como casos médios, o que explica os ganhos de desempenho observados.

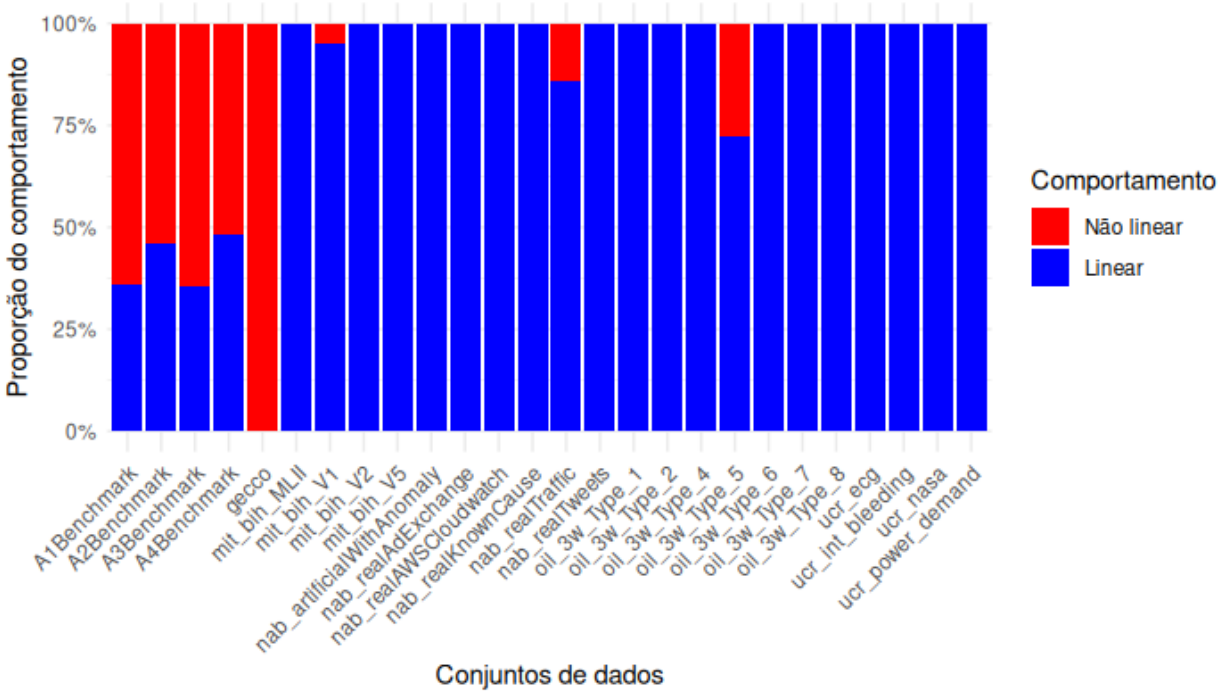


Figura VI.6: Proporção de comportamento computacional do método em cada conjunto de dados.

Também é possível notar ao observar em associação as Figuras VI.2 e VI.5 que os conjuntos de dados com maior proporção de eventos e detecções, como os do *Yahoo Labs* (A1Benckmark, A2Benckmark, A3Benckmark, A4Benckmark) tendem a apresentar mais ocorrências de casos complexos e, consequentemente, a comportar-se de forma não linear.

Ao todo, o algoritmo se comportou de forma linear em 62% dos testes, o que representa um ganho considerável em relação a SoftED e SoftEDPar, que se comportam com complexidade quadrática e cúbica, respectivamente, em 100% dos casos.

VI.2.2 Limites computacionais do método

Com relação ao desempenho computacional, busca-se compreender não apenas como a complexidade do método se manifesta em cenários reais, mas também identificar seus limites práticos, isto é, em quais configurações o método se torna computacionalmente inviável.

Para isso, classes sintéticas foram definidas a partir de combinações entre o tamanho da série n (de 10^2 a 10^8) e a densidade ϕ de eventos/detecções (de 0,1% a 25,6%). Para cada par (n, ϕ), gerou-se o número correspondente de eventos: $n_{\text{ev}} = \lfloor n \times \phi \rfloor$.

Os eventos foram distribuídos ao longo da série de forma uniforme, isto é, seus índices foram amostrados aleatoriamente com igual probabilidade no conjunto $\{1, \ldots, n\}$, resultando em instâncias que não favorecem nenhuma região temporal específica.

As detecções, por sua vez, foram geradas como aproximações ruidosas desses eventos. Para cada evento, aplicou-se um deslocamento aleatório pertencente à janela simétrica $[-J, J]$, com $J = 60$, produzindo uma distribuição concentrada em torno dos eventos e que simula o comportamento típico de detectores cujo erro é limitado. A escolha de $J = 60$ relaciona-se ao valor da janela de tolerância temporal utilizada pelo SmartSoftED ($k = 30$), de modo a construir cenários em que aproximadamente metade das detecções cai na região de pertinência do evento.

Na Figura VI.7, observa-se um *heatmap* que relaciona o tempo de execução da métrica a diferentes combinações de tamanho da série e proporção de eventos/detecções. A região azul-escura corresponde a tempos de execução desprezíveis (próximos de zero), enquanto áreas progressivamente mais claras indicam o aumento do tempo de execução até atingir o limite superior de 60 segundos (em amarelo). Esse padrão visual permite intuir que há um limite prático de aplicabilidade da métrica para séries maiores que 10^7. Para séries menores que esse limite, a proporção de eventos e de detecções também pode inviabilizar o processamento em menos de 60 segundos.

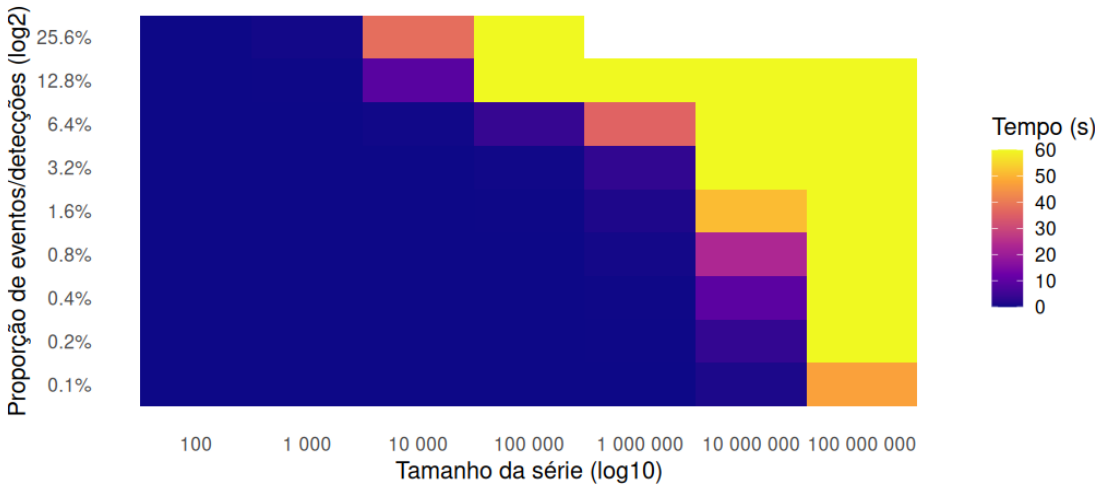


Figura VI.7: *Heatmap* de tempo de execução da métrica em função do tamanho da série e da densidade de eventos/detecções (limite de 60 segundos).

Para o contexto de aplicações de tempo real, adotou-se um limite operacional de 3 segundos. Nesse cenário, avaliou-se em quais combinações de tamanho da série e densidade de eventos a métrica permanece dentro desse intervalo, produzindo um segundo *heatmap* (Figura VI.8) que evidencia a região de viabilidade prática para sistemas que demandam resposta imediata. Há um limite prático de aplicabilidade da métrica para séries maiores que 10^6. Para séries menores que esse limite, a proporção de eventos e de detecções também pode inviabilizar o processamento em menos de 3 segundos.

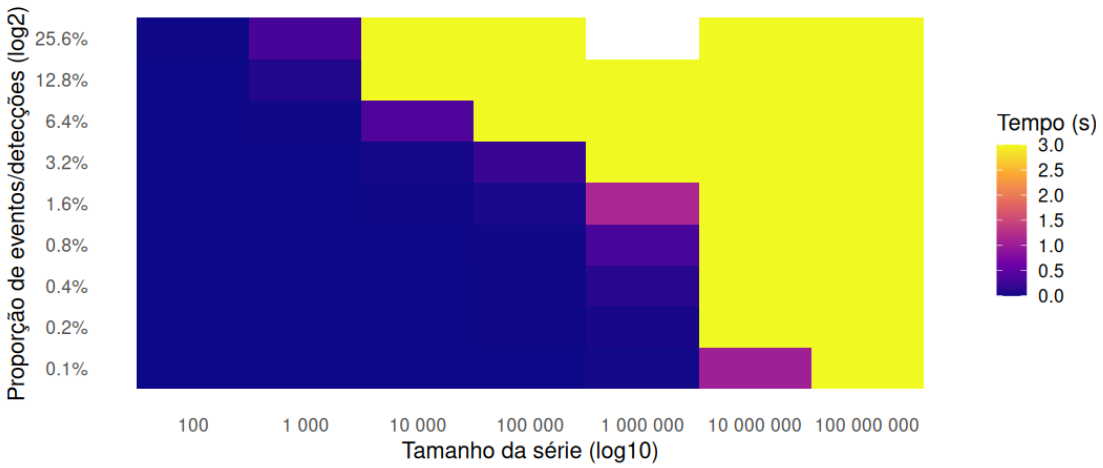


Figura VI.8: *Heatmap* de tempo de execução da métrica em função do tamanho da série e da densidade de eventos/detecções (limite de 3 segundos).

Capítulo VII Conclusão

Este trabalho reformulou o problema de correspondência entre detecções e eventos em séries temporais como um emparelhamento máximo ponderado em grafo bipartido completo desbalanceado, propondo inicialmente o método *SoftEDPar* e, em seguida, o método híbrido *SmartSoftED*. A nova formulação permitiu substituir a associação gulosa do *SoftED* por associações ótimas, ao mesmo tempo em que introduziu uma estratégia de segmentação baseada em janelas de tolerância que separa o problema em subcasos simples, médios e complexos. Dessa forma, apenas os trechos realmente ambíguos da série precisam ser resolvidos via Algoritmo Húngaro, enquanto os demais são tratados por funções diretas de pertinência ou varreduras lineares, reduzindo sensivelmente o custo computacional sem abrir mão da otimalidade.

Os experimentos realizados sobre o repositório UniTED com diferentes detectores de eventos mostraram que o *SmartSoftED* preserva exatamente as mesmas pontuações de precisão e revocação obtidas pelo *SoftEDPar*, confirmando que o particionamento em subproblemas não introduz perdas de qualidade na avaliação. Em termos de desempenho, a métrica apresentou ganhos significativos em relação tanto ao *SoftED* quanto ao *SoftEDPar*, alcançando em média acelerações de duas ordens de grandeza e obtendo comportamento empiricamente linear em cerca de 62% dos testes.

Os experimentos com séries sintéticas permitiram investigar os limites práticos de aplicabilidade do *SmartSoftED* em função do tamanho da série e da densidade de eventos/detecções. Os *heatmaps* resultantes atuam como um guia visual, permitindo identificar, a partir da combinação entre tamanho das séries e proporção de eventos e detecções, em quais cenários o método permanece viável ou se torna computacionalmente restritivo. Foram contemplados tanto cenários de grande escala quanto situações em que o sistema está sujeito a restrições temporais rigorosas.

Apesar dos avanços, algumas limitações permanecem, pois são inerentes à metodologia. Em situações caracterizadas por janelas de tolerância muito amplas e alta densidade de eventos e detecções, os segmentos classificados como complexos tornam-se predominantes e o custo total converge para o regime cúbico, reduzindo os ganhos em relação a uma solução plenamente baseada no Algoritmo Húngaro.

Outra limitação é que a métrica foi concebida para associar eventos pontuais a detecções pontuais, não sendo adequada ao cenário de eventos intervalares. A tentativa de tratar um intervalo como um conjunto consecutivo de pontos acionaria, de forma recorrente, os casos complexos, tornando

a técnica ineficiente e inadequada para esse tipo de dado. Para o contexto intervalar, as métricas apresentadas na Seção III.1 representam o estado da arte e são as mais apropriadas.

Como continuidade desta pesquisa, sugere-se investigar abordagens baseadas em computação paralela para acelerar o processamento dos segmentos classificados como complexos. Embora o *SmartSoftED* reduza substancialmente a necessidade de resolver subproblemas cúbicos, esses segmentos ainda podem se tornar dominantes em cenários adversos, de modo que a exploração de paralelismo surge como uma estratégia promissora para ampliar ainda mais a escalabilidade da métrica. Adicionalmente, propõe-se investigar métodos ótimos alternativos para emparelhamento em grafos bipartidos que, embora apresentem complexidade assintótica equivalente à do Algoritmo Húngaro no pior caso, possam oferecer melhor desempenho médio nas instâncias do problema tratado.

Referências

Abdulaal, A., Liu, Z., and Lancewicki, T. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 2485–2494, New York, NY, USA. Association for Computing Machinery, 2021. https://doi.org/10.1145/3447548.3467174

Ahmad, S., Lavin, A., Purdy, S., and Agha, Z. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017. https://doi.org/https://doi.org/10.1016/j.neucom.2017.04.070

Ariyaluran Habeeb, R. A., Nasaruddin, F., Gani, A., Targio Hashem, I. A., Ahmed, E., and Imran, M. Real-time big data processing for anomaly detection: A survey. *International Journal of Information Management*, 45:289–307, 2019. https://doi.org/https://doi.org/10.1016/j.ijinfomgt.2018.08.006

Audibert, J., Michiardi, P., Guyard, F., Marti, S., and Zuluaga, M. A. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3395–3404, New York, NY, USA. Association for Computing Machinery, 2020. https://doi.org/10.1145/3394486.3403392

Barr, R. S., Glover, F., and Klingman, D. The alternating basis algorithm for assignment problems. *Math. Program.*, 13(1):1–13, 1977. https://doi.org/10.1007/BF01584319

Bhattacharya, D., Mukherjee, S., Kamanchi, C., Ekambaram, V., Jati, A., and Dayama, P. Towards unbiased evaluation of time-series anomaly detector. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2025. https://doi.org/10.1109/ICASSP49660.2025.10890568

Bollobás, B. *Graph Theory*, volume 63 of *Graduate Texts in Mathematics*. Springer, New York, NY, 1979.

Brooks, H., Flora, M., and Baldwin, M. A rose by any other name: On basic scores from the 2×2 table and the plethora of names attached to them. *Artificial Intelligence for the Earth Systems*, 3, 2024. https://doi.org/10.1175/AIES-D-23-0104.1

Doshi, K., Abudalou, S., and Yilmaz, Y. Reward once, penalize once: Rectifying time series anomaly detection. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022. https://doi.org/10.1109/IJCNN55064.2022.9891913

Edmonds, J. and Karp, R. M. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19:248–264, 1972. https://doi.org/10.1145/321694.321699

Esling, P. and Agon, C. Time-series data mining. *ACM Comput. Surv.*, 45(1), 2012. https://doi.org/10.1145/2379776.2379788

Ford, L. R. and Fulkerson, D. R. *Flows in Networks*. Princeton University Press, 1963.

Garg, A., Zhang, W., Samaran, J., Savitha, R., and Foo, C.-S. An evaluation of anomaly detection and diagnosis in multivariate time series. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2508–2517, 2022. https://doi.org/10.1109/TNNLS.2021.3105827

Goldfarb, D. Efficient dual simplex algorithms for the assignment problem. *Math. Program.*, 33(2):187–203, 1985. https://doi.org/10.1007/BF01582245

Guralnik, V. and Srivastava, J. Event Detection from Time Series Data. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 33–42, New York, NY, USA. ACM, 1999. https://doi.org/10.1145/312129.312190

Hanley, J. A. and McNeil, B. J. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843, 1983. https://doi.org/10.1148/radiology.148.3.6878708

Hanssens, D. M., Parsons, L. J., and Schultz, R. L. *Market Response Models: Econometric and Time Series Analysis*. Springer, Boston, Mass., 2nd edition, 2003.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Soderstrom, T. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 387–395, New York, NY, USA. Association for Computing Machinery, 2018. https://doi.org/10.1145/3219819.3219845

Hung, M. S. and Rom, W. O. Solving the assignment problem by relaxation. *Operations Research*, 28:969–982, 1980. https://doi.org/10.1287/opre.28.4.969

Hwang, W.-S., Yun, J.-H., Kim, J., and Kim, H. C. Time-series aware precision and recall for anomaly detection: Considering variety of detection result and addressing ambiguous labeling. In

Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, page 2241–2244, New York, NY, USA. Association for Computing Machinery, 2019. https://doi.org/10.1145/3357384.3358118

Hwang, W.-S., Yun, J.-H., Kim, J., and Min, B. G. "Do you know existing accuracy metrics overrate time-series anomaly detections?". *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 403–412, 2022. https://doi.org/10.1145/3477314.3507024

Jonker, R. and Volgenant, A. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987. https://doi.org/10.1007/bf02278710

Korte, B. and Vygen, J. *Combinatorial Optimization*. Springer Berlin Heidelberg, 2018.

Kravchik, M. and Shabtai, A. Detecting cyber attacks in industrial control systems using convolutional neural networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, CPS-SPC '18, page 72–83, New York, NY, USA. Association for Computing Machinery, 2018. https://doi.org/10.1145/3264888.3264896

Kuhn, H. W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. https://doi.org/10.1002/nav.3800020109

Lavin, A. and Ahmad, S. Evaluating real-time anomaly detection algorithms – the numenta anomaly benchmark. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 38–44, 2015. https://doi.org/10.1109/ICMLA.2015.141

Li, D., Chen, D., Goh, J., and Ng, S.-k. Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series, 2019.

Lima, J., Castro, H., Oliveira, L., Paixão, E., Baroni, L., Salles, R., Vargas, R., and Ogasawara, E. United: A unified time series event detection repository. In *Anais do XIX Brazilian e-Science Workshop*, pages 1–8, Porto Alegre, RS, Brasil. SBC, 2025. https://doi.org/10.5753/bresci.2025.247972

Liu, S., Yamada, M., Collier, N., and Sugiyama, M. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83, 2013. https://doi.org/10.1016/j.neunet.2013.01.012

Liu, S.-C. and Liu, S.-Y. An efficient expert system for air compressor troubleshooting. *Expert Systems*, 18:203–214, 2001. https://doi.org/10.1111/1468-0394.00175

Ogasawara, E., Salles, R., Porto, F., and Pacitti, E. *Event Detection in Time Series*. Springer Nature Switzerland, 2025.

Olteanu, M., Rossi, F., and Yger, F. Meta-survey on outlier and anomaly detection. *Neurocomputing*, 555:126634, 2023. https://doi.org/https://doi.org/10.1016/j.neucom.2023.126634

Pimentel, M. A., Clifton, D. A., Clifton, L., and Tarassenko, L. A review of novelty detection. *Signal Processing*, 99:215 – 249, 2014. https://doi.org/10.1016/j.sigpro.2013.12.026

Reis, M., Salles, R., Xexéo, G., Coutinho, R., and Ogasawara, E. Matching detections to events in time series. *2024: Anais do Simpósio Brasileiro de Banco de Dados*, pages 785–791, 2024. https://doi.org/10.5753/sbbd.2024.243275

Salles, R., Escobar, L., Baroni, L., Zorrilla, R., Ziviani, A., Kreischer, V., Delicato, F., Pires, P. F., Maia, L., Coutinho, R., Assis, L., and Ogasawara, E. Harbinger: Um framework para integração e análise de métodos de detecção de eventos em séries temporais. In *Anais do Simpósio Brasileiro de Banco de Dados (SBBD)*, pages 73–84. SBC, 2020. https://doi.org/10.5753/sbbd.2020.13626

Salles, R., Lima, J., Reis, M., Coutinho, R., Pacitti, E., Masseglia, F., Akbarinia, R., Chen, C., Garibaldi, J., Porto, F., and Ogasawara, E. Softed: Metrics for soft evaluation of time series event detection. *Computers & Industrial Engineering*, 198:110728, 2024. https://doi.org/https://doi.org/10.1016/j.cie.2024.110728

Shumway, R. H. and Stoffer, D. S. *Time series analysis and its applications : with R examples*. Springer Cham, 2025.

Siwon, K., Choi, K., Choi, H.-S., Lee, B., and Yoon, S. Towards a rigorous evaluation of time-series anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36:7194–7201, 2022. https://doi.org/10.1609/aaai.v36i7.20680

Spackman, K. A. SIGNAL DETECTION THEORY: VALUABLE TOOLS FOR EVALUATING INDUCTIVE LEARNING. In Segre, A. M., editor, *Proceedings of the Sixth International Workshop on Machine Learning*, pages 160–163. Morgan Kaufmann, San Francisco (CA), 1989.

Sørbø, S. and Ruocco, M. Navigating the metric maze: a taxonomy of evaluation metrics for anomaly detection in time series. *Data Mining and Knowledge Discovery*, 38:1027–1068, 2023. https://doi.org/10.1007/s10618-023-00988-8

Tahera, K., Ibrahim, R., and Lochert, P. A fuzzy logic approach for dealing with qualitative quality characteristics of a process. *Expert Systems with Applications*, 34(4):2630–2638, 2008. https://doi.org/https://doi.org/10.1016/j.eswa.2007.05.025

Tatbul, N., Lee, T. J., Zdonik, S., Alam, M., and Gottschlich, J. Precision and recall for time series. In *Advances in Neural Information Processing Systems*, volume 2018-December, pages 1920 – 1930, 2018.

Tettamanzi, A. and Tomassini, M. *Soft Computing: Integrating Evolutionary, Neural, and Fuzzy Systems*. Springer Science & Business Media, 2013.

Wu, W.-W. and Lee, Y.-T. Developing global managers' competencies using the fuzzy dematel method. *Expert Systems with Applications*, 32:499–507, 2007. https://doi.org/10.1016/j.eswa.2005.12.005

Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., Chen, J., Wang, Z., and Qiao, H. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 187–196, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee, 2018. https://doi.org/10.1145/3178876.3185996

Zadeh, L. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965. https://doi.org/https://doi.org/10.1016/S0019-9958(65)90241-X

Zhao, L. Event prediction in the big data era: A systematic survey. *ACM Comput. Surv.*, 54(5), 2021. https://doi.org/10.1145/3450287

TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO/DIVULGAÇÃO DE DOCUMENTO ELETRÔNICO NO CATÁLOGO DO SISTEMA DE BIBLIOTECAS DO CEFET/RJ E NO REPOSITÓRIO INSTITUCIONAL

Na qualidade de titular dos direitos de autor da publicação abaixo citada, de acordo com a Lei nº 9610/98, autorizo, a partir da presente data, o CEFET/RJ, a disponibilizar gratuitamente, sem ressarcimento de direitos autorais, conforme permissões assinadas abaixo, no Catálogo Online do Sistema de Bibliotecas e no Repositório Institucional do Cefet/RJ, ou qualquer outro formato de disseminação da informação que a instituição implemente, no formato PDF, para fins de leitura, impressão e/ou download pela internet, a título de divulgação da produção científica gerada por esta Instituição.

1- Tipo de trabalho: [ ] Projeto Final (Graduação) [ ] Especialização
[x] Dissertação [ ] Tese

2- Identificação do trabalho/autor(es):

Curso: Mestrado em Ciência da Computação
Título: Correspondência de Detecções a Eventos em Séries Temporais com Eficiência Computacional e Garantia de Otimalidade
Autor(es): Michel Siqueira Reis
Orientador: Rafaelli de Carvalho Coutinho
Co-orientador: Eduardo Soares Ogasawara

3- Informações de acesso ao trabalho:

Este trabalho é confidencial? [ ] Sim [X] Não
Pode ser liberado para publicação? [X] Total [ ] Parcial [ ] Não
Justifique:_____
_____
Em caso de publicação parcial, assinale as permissões:
[ ] Sumário
[ ] Capítulos. Especifique: _____
[ ] Bibliografia
[ ] Outras partes do trabalho: _____

Rio de Janeiro, _____/_____/_____

gov.br Documento assinado digitalmente
MICHEL SIQUEIRA REIS
Data: 28/01/2026 15:45:21-0300
Verifique em https://validar.iti.gov.br

_____
Assinatura do(s) autor(es)

gov.br Documento assinado digitalmente
RAFAELLI DE CARVALHO COUTINHO
Data: 28/01/2026 17:51:26-0300
Verifique em https://validar.iti.gov.br

_____
Assinatura do orientador

gov.br Documento assinado digitalmente
EDUARDO SOARES OGASAWARA
Data: 29/01/2026 11:45:41-0300
Verifique em https://validar.iti.gov.br

_____
Assinatura do co-orientador