

# Detecção de Concept Drifts na Série da Arrecadação ICMS do Estado de Sergipe

Arthur Fernando da Silva Santos<sup>1</sup>, Anthony Eduardo Medeiros Pantaleão<sup>1</sup>,  
Max Castor Rodrigues Junior<sup>1</sup>, Yúri Faro Dantas de Sant'Anna<sup>1</sup>

<sup>1</sup>Coordenação de Tecnologia da Informação – Centro Universitário Estácio de Sergipe  
Aracaju – SE – Brazil

{202208703208, 202304252963}@alunos.estacio.br

{max.carojun, yur.anna}@professores.estacio.br

**Abstract.** *The collection of the Tax on the Circulation of Goods and Services (ICMS) is the main source of revenue for the State of Sergipe, essential for funding public policies. Understanding variations in the time series can support strategic government decisions. This study proposed the evaluation and selection of a concept drift detection algorithm in time series, aiming to identify significant changes in the monthly ICMS revenue. Different detectors with distinct mechanisms were tested. The Bottom-up algorithm showed the best performance among the explored options, achieving a score of 5.38 in the MTR metric, which combines a low false alarm rate and fast detection. In this context, the model proves to be a promising, efficient, and viable option for real-world applications.*

**Resumo.** *A arrecadação do Imposto de Circulação de Bens e Serviços (ICMS) representa a principal fonte de receita do Estado de Sergipe, essencial para financiar políticas públicas. Compreender variações na série pode apoiar decisões estratégicas do governo. Este trabalho propôs a avaliação e seleção de um algoritmo de detecção de desvios de conceitos (concept drift) em séries temporais, visando identificar alterações significativas na arrecadação mensal do ICMS. Foram testados diferentes detectores com distintos mecanismos. O algoritmo Bottom-up apresentou o melhor desempenho entre as opções exploradas, obtendo 5,38 na métrica MTR, que combina baixa taxa de falsos alarmes e rapidez na detecção. Nesse cenário, o modelo se mostra uma opção promissora, eficiente e viável para aplicação real.*

## 1. Introdução

O Imposto sobre Circulação de Mercadorias e Prestação de Serviços (ICMS) é o maior componente tributário na arrecadação estadual.[Rezende 2009] Segundo [de Sant'Anna et al. 2024], “sua arrecadação é baseada na categoria do produto e na legislação estadual vigente, definindo a alíquota e a base de cálculo utilizadas para cada segmento econômico ou produto.”

No Estado de Sergipe o ICMS, regulamentado pela lei [Brasil 1996], corresponde a mais de 90% da receita estadual [SILVA 2013], desta forma o acompanhamento, categorização de eventos e previsões realizadas na série temporal gerada por este tributo

podem auxiliar nas tomadas de decisões [Urso et al. 2018]. Nesse contexto, a detecção de desvios poderia contribuir diretamente para a melhoria da precisão destas previsões e classificações [Lima et al. 2022].

Um desvio de conceito (concept drift), termo técnico explicado na Seção 2.1, ocorre quando a relação estatística entre as variáveis de entrada e saída muda ao longo do tempo. Tratando da arrecadação fiscal, isso pode significar, por exemplo, alterações no comportamento de consumo, políticas tributárias, incentivos fiscais ou mesmo eventos econômicos inesperados que impactam diretamente os padrões históricos de arrecadação. Ignorar esses desvios pode levar a decisões baseadas em modelos obsoletos, afetando negativamente o planejamento financeiro do Estado e a alocação eficiente de recursos públicos.

Modelos preditivos, como os utilizados para estimar arrecadação futura, são fortemente afetados por mudanças no conceito dos dados. A ocorrência de um desvio de conceito, o desempenho do modelo se degrada ao longo do tempo, os padrões históricos deixam de refletir o comportamento atual, resultando em previsões imprecisas ou enviesadas [You et al. 2021].

Em outro contexto, segundo Agrahari [Agrahari and Singh 2022], a precisão de um classificador deteriora devido ao desvio de conceito. De acordo com Hu [Hu et al. 2025], a presença de desvio de conceito geralmente provoca uma declinação significativa nas métricas de desempenho do modelo. A detecção oportuna destes desvios pode evitar a degradação do modelo, melhorando sua eficiências.

Este trabalho tem como objetivo principal analisar e selecionar o melhor algoritmo para detecção de mudanças estruturais em dados reais de arrecadação de ICMS, avaliando sua efetividade e acurácia. O restante do trabalho está organizado da seguinte forma: a Seção 2 apresenta o referencial teórico; a Seção 3 descreve a metodologia utilizada na condução do presente trabalho; a Seção 4 discute os resultados e, por fim, a Seção 5 apresenta a conclusão e são propostos trabalhos futuros.

## **2. Referencial Teórico**

Nesta seção serão abordados os conceitos necessários para compreensão e replicação dos modelos utilizados para construir este projeto.

### **2.1. Desvios de Conceito**

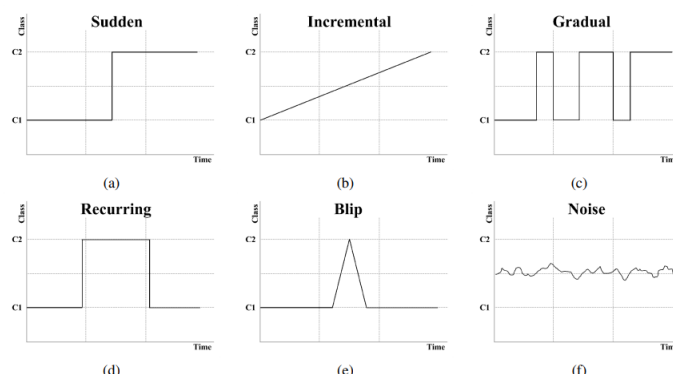
O desvio de conceito refere-se à mudança aleatória, ao longo do tempo, das propriedades estatísticas dos dados. Essas transformações são frequentemente atribuídas à influência de variáveis ocultas, cujo comportamento não pode ser diretamente observado ou mensurado [Lu et al. 2019].

Segundo [Lima et al. 2022], “Formalmente o desvio de conceito pode ser definido em qualquer cenário em que as probabilidades das variáveis variam ao longo do tempo, de modo que  $P_{t+1}(X, y) \neq P_t(X, y)$ , ou seja, a distribuição no instante  $t$  é diferente daquela no instante  $t + 1$ ”.

### **2.2. Tipos de Desvios**

De acordo com Iwashita (2018, apud Wadewale and Desai, 2015) [Iwashita and Papa 2018], os desvios podem ser classificados em: abrupto, incre-

mental, gradual, recorrente, blip e ruído, demonstrados e explicados na Figura 1.



**Figura 1. Tipos de desvios:** (a) Abrupto: o conceito muda bruscamente e sem alternância; (b) Incremental: no incremental o conceito muda lentamente ao longo do tempo; (c) Gradual: o gradual ocorre lentamente e inclui alternância na distribuição; (d) Recorrente: ocorre quando a instância do conceito desaparece temporariamente e retorna após um tempo; (e) Blip: um evento raro onde surge um conceito e logo após desaparece rapidamente; (f) Ruído: mudanças aleatórias que precisam ser filtradas

Os desvios também podem ser classificados em dois tipos diferentes: desvio de conceito real e desvio virtual. O desvio real ocorre quando há alteração na distribuição condicional das saídas, mesmo que a distribuição das variáveis de entrada permaneçam constantes. Já o virtual, geralmente está relacionado a modificações na distribuição das variáveis de entrada [Gama et al. 2014].

De acordo com [Lu et al. 2019] “Em métodos/algoritmos de detecção de desvios, um sinal de alarme é usado para indicar se o desvio de conceito ocorreu ou não no registro de tempo atual. É também um sinal para um sistema de aprendizagem se adaptar a um novo conceito.”. Ela também diz que, esse alarme com o registro temporal de detecção dos algoritmos existentes é atrasado em comparação com registro temporal de um desvio de conceito real, uma vez que a maioria dos detectores de desvios requer um número mínimo de novos dados para avaliar o status do desvio; E que o tempo de surgimento de um novo conceito ainda é vago.

### 2.3. Detecção de Desvios

A detecção de desvios, ou detecção de mudanças, é uma metodologia utilizada para identificar o momento ou intervalo de tempo em que ocorrem alterações significativas no comportamento de uma série temporal [M. and I.V. 1993]. Essa técnica possui ampla aplicação em contextos reais, como na análise financeira, bioinformática e climatologia [JAYACHANDRAN 2021].

Os métodos de detecção de pontos de mudança podem ser divididos em dois grupos principais: os online, que detectam alterações em tempo real, e os offline, que realizam a detecção após a coleta completa das amostras [TRUONG et al. 2020]. Este presente trabalho foca exclusivamente nos métodos offline, utilizando as bibliotecas Python *Ruptures*<sup>1</sup> com os algoritmos de métodos exatos Opt e PELT; os de método aproxi-

<sup>1</sup><https://centre-borelli.github.io/ruptures-docs/>

mado Window-sliding, Binary segmentation, Bottom-up e Kernel Change Points Detection; e da biblioteca *River*<sup>2</sup> com o algoritmo Page-Hinkley. Esses algoritmos são amplamente utilizados e recomendados em revisões recentes sobre o tema [Florez et al. 2025, Gama et al. 2014, Kronberg 2024, Liu et al. 2023, Yeshchenko et al. 2020].

### 2.3.1. Opt

Quando se tem um problema onde o número de desvios é conhecido e este número de pontos é fixo, sendo  $K \geq 1$ , o algoritmo Opt pode ser utilizado como solução para calcular com eficiência e detectar os desvios baseados em programação dinâmica. O Opt é baseado na seguinte observação[TRUONG et al. 2020]:

$$\begin{aligned} \min_{|T|=K} V(T, y = y_{0..T}) &= \min_{0=t_0 < t_1 < \dots < t_K < t_{K+1}=T} \sum_{k=0}^K c(y_{t_k..t_{k+1}}) \\ &= \min_{t \leq T-K} \left[ c(y_{0..t}) + \min_{t=t_0 < t_1 < \dots < t_{K-1} < t_K=T} \sum_{k=0}^{K-1} c(y_{t_k..t_{k+1}}) \right] \\ &= \min_{t \leq T-K} \left[ c(y_{0..t}) + \min_{|T|=K-1} V(T, y_{t..T}) \right] \end{aligned} \quad (1)$$

A Equação (1) expressa uma formulação recursiva da segmentação ótima de uma sequência. Ela mostra que, se conhecermos previamente as partições ótimas contendo  $K - 1$  segmentos para todos os subsinais  $Y_{t..T}$ , então é possível determinar facilmente o primeiro ponto de mudança da segmentação ótima com  $K$  segmentos.

### 2.3.2. PELT

Para um problema onde o número de desvios é desconhecido e se tem uma penalidade linear  $pen(\tau) = \beta|\tau|$ , onde  $\beta > 0$  é um suavizador de parâmetros, o algoritmo Pelt (Pruned Exact Linear Time)[Killick et al. 2012] foi criado para encontrar esta solução. Nessa abordagem, cada amostra é analisada de forma sequencial e, com base em uma regra de poda explícita, ela pode ser mantida ou descartada do conjunto de possíveis pontos de mudança. Essa regra é dada por[TRUONG et al. 2020]:

$$se \left[ \min_T V(T, y_{0..t}) + \beta|T| \right] + c(y_{t..s}) \geq \left[ \min_T V(T, y_{0..s}) + \beta|T| \right] \text{ verdadeiro,}$$

então,  $t$  não pode ser o último ponto de mudança anterior a  $T$ .

### 2.3.3. Window-sliding

O algoritmo de deslizamento de janelas, é uma alternativa aproximada e eficiente em relação aos métodos exatos. Ele opera medindo a discrepância entre duas janelas consecutivas que percorrem o sinal  $y$  de forma deslizante. Essa discrepância dada uma função de custo  $c(\cdot)$  entre dois sinais é dada por[TRUONG et al. 2020]:

$$d(y_{a..t}, y_{t..b}) = c(y_{a..b}) - c(y_{a..t}) - c(y_{t..b}) \quad (1 \leq a < t < b \leq T). \quad (2)$$

<sup>2</sup><https://riverml.xyz/dev/api/drift/PageHinkley/>

O algoritmo window-sliding mede a discrepância entre janelas adjacentes no sinal, identificando picos que indicam possíveis pontos de mudança. Sua principal vantagem é a baixa complexidade computacional e a simplicidade de implementação.

#### 2.3.4. Binary-segmentation

O algoritmo de segmentação binária, também denominado BinSeg, segundo [TRUONG et al. 2020] é conceitualmente simples e fácil de implementar, sendo uma ótima alternativa aos métodos exatos. O primeiro ponto de mudança estimado  $\hat{t}^{(1)}$  é dado por:

$$\hat{t}^{(1)} := \operatorname{argmin}_{1 \leq t < T-1} \underbrace{c(y_{0..t}) + c(y_{t..T})}_{V(T=\{t\})}.$$

Esta operação identifica o ponto de mudança que minimiza a soma dos custos. A partir desse ponto, o sinal é segmentado em duas partes na posição  $\hat{t}^{(1)}$ ; o procedimento é então repetido recursivamente em cada subsinal gerado, até que um critério de parada seja alcançado.

#### 2.3.5. Bottom-up

Este método segmenta o sinal original em vários subsinais pequenos mesclando-os sequencialmente até que restem apenas  $k$  pontos de mudança. É considerado conceitualmente simples e possui complexidade computacional linear. O algoritmo não é capaz de identificar pontos de mudança que não estejam presentes no conjunto inicial de candidatos. Além disso, nas primeiras iterações, o processo de fusão pode apresentar instabilidade. [TRUONG et al. 2020]

#### 2.3.6. KCPD

Kernel Change Points Detection (KCPD) é um algoritmo que transforma o sinal de entrada em um espaço de Hilbert de alta dimensão  $H$  por meio de uma função kernel com o objetivo de identificar deslocamentos médios. Inicialmente, são selecionados aleatoriamente índices de possíveis pontos de mudança, e um sinal bidimensional é simulado a partir de variáveis normais centradas, com matriz de covariância variável. Em seguida, os parâmetros internos do algoritmo são ajustados com base nesses dados. Por fim, aplica-se programação dinâmica para minimizar discretamente a função de contraste, retornando-se as estimativas dos pontos de mudança. [TRUONG et al. 2018]

#### 2.3.7. Page-Hinkley Test

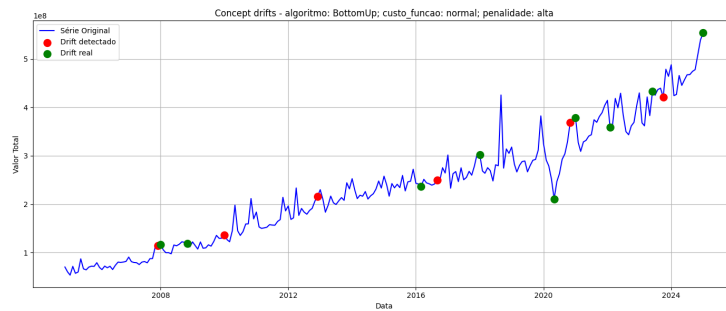
O algoritmo page-hinkley é utilizado para detectar variações, tanto aumentos quanto quedas, na média de uma sequência. Para identificar mudanças em tempo real, ele realiza dois testes simultaneamente, com base em uma variável cumulativa que representa a soma das diferenças entre os valores observados e a média calculada até o instante atual. [SEBASTIÃO and FERNANDES 2017]

### 3. Metodologia

#### 3.1. Base de Dados

Os dados mensais da arrecadação do ICMS do Estado de Sergipe, foram obtidos através da área de transparência no portal da Secretaria da Fazenda compreendendo o período de janeiro de 2005 a dezembro de 2024. Cada registro foi composto por duas colunas: Ano/Mês e Valor Total.

Segundo [Wang et al. 2011], uma das possibilidades para a determinação dos desvios de conceitos reais é o mapeamento dos mesmos através de um especialista no domínio de negócio. Desta forma, a demarcação dos pontos para esta série foi realizada por especialistas em arrecadação tributária. A escolha dos pontos foram baseadas em eventos e sazonalidades, como por exemplo: *REFIS* (Programa de Recuperação Fiscal) [Brasil 2000], e a pandemia de *COVID-19*, que provocou forte impacto na atividade econômica, levando à adoção de medidas emergenciais que afetaram diretamente a arrecadação e o comportamento dos contribuintes, justificando alterações nos dados observados[da Fonseca et al. 2021].



**Figura 2. Série da arrecadação de ICMS, comparativo entre pontos de desvio detectados e reais**

#### 3.2. Funções de Custos

A função de custo  $c(\cdot)$  atua como uma medida de homogeneidade dentro de um segmento. Sua escolha determina o tipo de mudança que o método será capaz de identificar.[TRUONG et al. 2020] Abaixo estão as principais funções utilizadas neste trabalho, todas disponíveis na biblioteca *ruptures*:

- **I1**: Detecta mudanças na mediana de um sinal. De modo geral, é um estimador robusto de uma mudança no ponto central (média, mediana, moda) de uma distribuição.
- **I2**: Detecta deslocamentos médios em um sinal.
- **normal**: Detecta mudanças simultâneas na média e na matriz de covariância de sinais gaussianos multivariados.
- **rbf (Radial Basis Function)**: Projeta os dados em um espaço de Hilbert de alta dimensão para identificar mudanças complexas na distribuição dos dados. É uma abordagem não paramétrica e versátil para detectar mudanças sutis em sequências independentes e identicamente distribuídas.
- **cosine**: Detecta mudanças na média do sinal, sendo aplicada em contextos como segmentação de textos ou música.

- **linear**: Modela os dados como regressões lineares e detecta mudanças nos coeficientes do modelo ao longo do tempo.
- **clinear**: Variante da função linear, mede o erro ao aproximar o sinal com uma spline linear.
- **rank**: Detecta mudanças gerais na distribuição de sinais multivariados, utilizando uma transformação de classificação.

A escolha da função de custo impacta diretamente a sensibilidade e a eficácia da detecção, sendo, portanto, um elemento fundamental na configuração dos algoritmos.

### 3.3. Penalidades

Quando o número de pontos de desvio de conceito são desconhecidos, uma restrição é adicionada na forma de penalidade para equilibrar o termo de ajuste. Quanto menor a penalidade, mais mudanças são detectadas, mesmo aquelas provenientes de ruído. Em contrapartida, uma penalização alta detecta apenas as alterações mais significativas, ou até mesmo nenhuma [TRUONG et al. 2020].

As penalidades foram calculadas utilizando Bayesian Information Criterion (BIC) com base na função `bic.penalty` da biblioteca *Wbs*<sup>3</sup> da linguagem R, implementada em python com a seguinte fórmula:

$$pen = k \cdot \log(n) \quad (3)$$

Onde (k) é o número multiplicador ajustável para calibrar a penalização e (n) é o total de amostras. Para esta série do estudo foram utilizados três fatores de penalização (k), low = 1.5, medium = 3 e high = 6.

### 3.4. Métricas

O desempenho do detector foi medido através destas quatro métricas MDR, MTD, MTFA e MTR, de acordo com [Lukats et al. 2024]:

1. **Missed Detection Rate (MDR)**: Mede a proporção de mudanças que não foram detectadas entre dois desvios consecutivos. Um desvio é considerado perdido se não for identificado antes do próximo ocorrer. Quanto menor o valor da MDR, melhor o desempenho do detector.
2. **Mean Time to Detection (MTD)**: Representa o tempo médio necessário para detectar corretamente um desvio, considerando apenas as detecções válidas. Detectores eficientes apresentam um MTD reduzido.
3. **Mean Time between False Alarms (MTFA)**: Avalia o intervalo médio entre falsos positivos, ou seja, detecções incorretas de mudanças após a última mudança já ter sido identificada. O MTFA não é definido se ocorrerem menos de dois falsos alarmes. Valores mais altos indicam maior robustez do detector.
4. **Mean Time Ratio (MTR)**: Combina MDR, MTD e MTFA para facilitar a comparação entre detectores. Um MTR alto indica bom desempenho (MTFA alto, MTD e MDR baixos). Sua fórmula é dada por:

$$MTR := \frac{MTFA}{MTD} \cdot (1 - MDR)$$

---

<sup>3</sup>Wild Binary Segmentation for Multiple Change-Point Detection.  
<https://www.rdocumentation.org/packages/wbs/versions/1.4>

### 3.5. Desenvolvimento

Os algoritmos foram selecionados por sua diversidade metodológica [TRUONG et al. 2018]. Todos algoritmos foram implementados em Python 3.11, utilizando as seguintes bibliotecas: *pandas*, *numpy*, *matplotlib*, *ruptures* e *river*. A implementação da metodologia deste trabalho está disponível no endereço <sup>4</sup>. Foram realizados 95 experimentos envolvendo a combinação de custos e penalidades diferentes conforme tabela 1:

Algoritmo	Funções de custo	Penalidades
BinSeg	l1, l2, rbf, clinear, ar, normal, rank	baixa, média, alta
Bottom-up	l1, l2, rbf, clinear, ar, normal, rank	baixa, média, alta
Opt	l1, l2, rbf, clinear, cosine, normal, rank	—
KCPD	linear, cosine, rbf	baixa, média, alta
Pelt	l1, l2, rbf, clinear, ar, normal, rank	baixa, média, alta
Window-sliding	l1, l2, rbf, normal, rank	baixa, média, alta
Page-Hinkley	—	—

Tabela 1. Funções de custo e penalidades utilizadas por cada algoritmo.

## 4. Resultados

A Tabela 2 exibe o melhor resultado para cada algoritmo testado:

Algoritmo	Função de Custo/Penalidade	MTR	MTFA	MTD	MDR
Bottom-up	normal / alta	<b>5,38</b>	60,00	6,20	0,44
Window-sliding	normal / baixa	4,67	32,14	6,89	<b>0,00</b>
Pelt	normal / alta	4,63	<b>70,00</b>	8,40	0,44
KCPD	linear / baixa	3,78	2,52	<b>0,67</b>	<b>0,00</b>
BinSeg	clinear / baixa	3,09	15,33	3,86	0,22
Opt	rbf / —	2,98	35,00	9,14	0,22
Page-Hinkley	simplificado	1,71	45,00	14,60	0,44

Tabela 2. Resultados das métricas para diferentes algoritmos.

O algoritmo Bottom-up com a função de custo normal e penalidade alta, obteve o melhor MTR (5,38), indicando uma boa combinação entre baixa taxa de falsos alarmes e rapidez na detecção. No entanto, sua Taxa de Detecção Perdida (MDR) foi relativamente alta (0,44), o que indica que alguns desvios reais não foram detectados. A Figura 2, que foi gerada com o algoritmo *matplotlib* <sup>5</sup>, apresenta a comparação entre os pontos detectados pelo algoritmo e os pontos reais da Arrecadação do ICMS do Estado de Sergipe no período entre 2005 a 2024.

## 5. Conclusão e Trabalhos Futuros

Este trabalho teve como objetivo investigar e comparar algoritmos de detecção de desvio de conceito, com foco na arrecadação estadual do ICMS em Sergipe. Os resultados mostraram que o objetivo inicial do trabalho foi alcançado e o método Bottom-up, mostrou ser uma opção promissora para aplicação em séries temporais deste contexto.

Este estudo contribuiu para compreensão como a escolha da função de custo e penalidade afetam diretamente a performance dos detectores. Essa análise é crucial para

<sup>4</sup><https://github.com/ArthurFer/concept-drifts-pibic>

<sup>5</sup>Matplotlib. [https://matplotlib.org/3.5.3/api/\\_as\\_gen/matplotlib.pyplot](https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot)



aplicações práticas. Ao antecipar essas alterações no comportamento da arrecadação, órgãos governamentais e gestores públicos podem tomar decisões estratégicas, como ajustes na política fiscal, planejamento orçamentário mais eficaz ou ações corretivas direcionadas. Além de permitir o reajuste de modelos preditivos e classificadores desenvolvidos para a série em questão.

Como continuidade deste trabalho, propõe-se o estudo e experimentação de testes em outros contextos a fim de confirmar a validade do modelo proposto como método auxiliar em cenários de previsão ou classificação de séries temporais.

## Referências

- [Agrahari and Singh 2022] Agrahari, S. and Singh, A. K. (2022). Concept drift detection in data stream mining : A literature review. *Journal of King Saud University - Computer and Information Sciences*, 34(10, Part B):9523–9540.
- [Brasil 1996] Brasil (1996). Lei nº 3.796, de 26 de dezembro de 1996. <https://aleselegis.al.se.leg.br/Arquivo/Documents/legislacao/html/137961996.html>. Dispõe sobre o ICMS no Estado de Sergipe.
- [Brasil 2000] Brasil (2000). Lei n.º 9.964, de 10 de abril de 2000. [https://www.planalto.gov.br/ccivil\\_03/leis/19964.htm](https://www.planalto.gov.br/ccivil_03/leis/19964.htm). Institui o Programa de Recuperação Fiscal – REFIS, destinado à regularização de débitos junto à Receita Federal e ao INSS.
- [da Fonseca et al. 2021] da Fonseca, J. L. D., Almeida, C. R. d. C., and da Silva, M. d. R. (2021). Impactos na arrecadação nos estados do nordeste na pandemia da covid-19. *Revista Conhecimento Contábil*, 11(2).
- [de Sant’Anna et al. 2024] de Sant’Anna, Y. F. D., de Farias, M. L., Júnior, M. C., Dantas, D., and Junior, M. R. (2024). Fuel classification in electronic tax documents. In *Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM*, pages 337–343. INSTICC, SciTePress.
- [Florez et al. 2025] Florez, A., Rodríguez-Moreno, I., Florez-Tapia, A. M., Artetxe, A., and Sierra, B. (2025). Lynxsight: change-point detection through different distance-based common spatial patterns. *The Journal of Supercomputing*, 81(7):858.
- [Gama et al. 2014] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, H. (2014). A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46.
- [Hu et al. 2025] Hu, L., Lu, Y., and Feng, Y. (2025). Concept drift detection based on deep neural networks and autoencoders. *Applied Sciences*, 15(6).
- [Iwashita and Papa 2018] Iwashita, A. and Papa, J. (2018). An overview on concept drift learning. *IEEE Access*, PP:1–1.
- [JAYACHANDRAN 2021] JAYACHANDRAN, B. (2021). Changepoint detection with python on sensor data in welding processes. Master of science thesis, POLITECNICO DI TORINO, Torino.
- [Killick et al. 2012] Killick, R., Fearnhead, P., and and, I. A. E. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598.

- [Kronberg 2024] Kronberg, S. (2024). Concept drift detection in document classification: An evaluation of adwin, kswin, and page hinkley using different observation variables.
- [Lima et al. 2022] Lima, M., Neto, M., Filho, T. S., and de A. Fagundes, R. A. (2022). Learning under concept drift for regression—a systematic literature review. *IEEE Access*, 10:45410–45429.
- [Liu et al. 2023] Liu, J., Yang, D., Zhang, K., Gao, H., and Li, J. (2023). Anomaly and change point detection for time series with concept drift. *World Wide Web*, 26(5):3229–3252.
- [Lu et al. 2019] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363.
- [Lukats et al. 2024] Lukats, D., Zielinski, O., Hahn, A., and Stahl, F. (2024). A benchmark and survey of fully unsupervised concept drift detectors on real-world data streams. *International Journal of Data Science and Analytics*, 19:1–31.
- [M. and I.V. 1993] M., B. and I.V., N. (1993). *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, Inc.
- [Rezende 2009] Rezende, F. (2009). *A reforma tributária e a Federação*. Fundação Getúlio Vargas.
- [SEBASTIÃO and FERNANDES 2017] SEBASTIÃO, R. and FERNANDES, J. M. (2017). Supporting the page-hinkley test with empirical mode decomposition for change detection. In *Proceedings of the International Symposium on Methodologies for Intelligent Systems*, pages 492–498, Rasdorf, Germany. Springer International Publishing.
- [SILVA 2013] SILVA, P. d. (2013). O icms como reflexo do desenvolvimento econômico dos municípios do estado de sergipe. Master’s thesis, Universidade Federal de Sergipe.
- [TRUONG et al. 2018] TRUONG, C., OUDRE, L., and VAYATIS, N. (2018). Ruptures: change point detection in python. *arXiv*, abs/1801.00826. Disponível em: <https://arxiv.org/pdf/1801.00826>. Acesso em: 21 abr. 2025.
- [TRUONG et al. 2020] TRUONG, C., OUDRE, L., and VAYATIS, N. (2020). Selective review of offline change point detection methods. *IEEE Signal Processing Magazine*, 37(6):104–116.
- [Urso et al. 2018] Urso, A., Fiannaca, A., La Rosa, M., Ravì, V., and Rizzo, R. (2018). Data mining: Classification and prediction. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, 384.
- [Wang et al. 2011] Wang, S., Schlobach, S., and Klein, M. (2011). Concept drift and how to identify it. *Journal of Web Semantics*, 9(3):247–265. Semantic Web Dynamics Semantic Web Challenge, 2010.
- [Yeshchenko et al. 2020] Yeshchenko, A., Mendling, J., Di Ciccio, C., and Polyvyanny, A. (2020). Vdd: A visual drift detection system for process mining. In *CEUR Workshop Proceedings*, pages 31–34.
- [You et al. 2021] You, X., Zhang, M., Ding, D., Feng, F., and Huang, Y. (2021). Learning to learn the future: Modeling concept drifts in time series prediction. In *Proceedings*

*of the 30th ACM International Conference on Information & Knowledge Management,*  
pages 2434–2443.