

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica

Daniel da Silva Diogo Lara

Estratégias não supervisionadas para
identificação automática de *drifts*
incrementais

Belo Horizonte

2025

Daniel da Silva Diogo Lara

**Estratégias não supervisionadas para identificação
automática de *drifts* incrementais**

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Doutor em Engenharia Elétrica.

Orientador: Walmir Matos Caminhas
Coorientador: Cristiano Leite de Castro

Belo Horizonte

2025

L318e	<p>Lara, Daniel da Silva Diogo. Estratégias não supervisionadas para identificação automática de <i>drifts</i> incrementais [recurso eletrônico] / Daniel da Silva Diogo Lara. – 2025. 1 recurso online (98 f. : il., color.) : pdf.</p> <p>Orientador: Walmir Matos Caminhas. Coorientador: Cristiano Leite de Castro.</p> <p>Tese (doutorado) – Universidade Federal de Minas Gerais, Escola de Engenharia.</p> <p>Inclui bibliografia.</p> <p>1. Engenharia elétrica – Teses. 2. Brasil – Indústrias – Inovações tecnológicas – Teses. 3. Mudanças – Teses. 4. Radar – Detecção automática – Teses. I. Caminhas, Walmir Matos. II. Castro, Cristiano Leite de. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.</p>
	CDU: 621.3(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS

Escola de Engenharia

COLEGIADO DO CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FOLHA DE APROVAÇÃO

"Estratégias Não Supervisionadas Para Identificação Automática de Drifts Incrementais"

Daniel da Silva Diogo Lara

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 15 de dezembro de 2025.

Por:

Prof. Dr. Walmir Matos Caminhas
DELT (UFMG) - Orientador

Prof. Dr. Cristiano Leite de Castro
DEE (UFMG) - Coorientador

Prof. Dr. Frederico Gadelha Guimarães
DCC (UFMG)

Prof. Dr. Prof. Dr. Rodrigo Amador Coelho
(IFNMG)

Prof. Dr. Marcos Flávio S. V. D'Angelo
DCC (Unimontes)

Dr. Pedro Henrique Silva Coutinho
(UERJ)



Documento assinado eletronicamente por **Walmir Matos Caminhas, Presidente de comissão**, em 15/12/2025, às 16:31, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Cristiano Leite de Castro, Professor do Magistério Superior**, em 15/12/2025, às 17:38, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Rodrigo Amador Coelho, Usuário Externo**, em 16/12/2025, às 15:56, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Pedro Henrique Silva Coutinho, Usuário Externo**, em 16/12/2025, às 18:17, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Marcos Flávio Silveira Vasconcelos DAngelo, Usuário Externo**, em 28/12/2025, às 19:48, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Frederico Gadelha Guimaraes, Professor do Magistério Superior**, em 29/12/2025, às 17:02, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador 4824526 e o código CRC **D67CD93A**.

Dedico este trabalho aos meus pais, esposa e filhos

Agradecimentos

Este trabalho é o resultado da contribuição de cada professor que, ao longo dos anos, pôde compartilhar comigo um pouco do seu conhecimento. Em especial, agradeço aos professores Walmir Caminhas e Cristiano Leite de Castro pelo tempo e dedicação compartilhados, não somente neste trabalho, mas também em disciplinas cursadas. Também agradeço ao colega Rodrigo Amador Coelho pelas inúmeras discussões a respeito deste conteúdo.

Agradeço ainda a todos os colegas da Petrobras que contribuíram de alguma forma com esse trabalho, seja com uma dica, um conhecimento compartilhado ou um cenário de desafio encontrado nas plantas em que atuamos. Em especial devo citar os nomes de alguns que contribuíram grandemente para que esse projeto se tornasse realidade: Márcio Henrique Freire, Donisete Magri (gerentes que viabilizaram), Leonardo Catão de Oliveira e Marcelo Bustamante pelas constantes trocas e contribuições com relação aos processos e técnicas.

Por fim, e não menos importante, expresso minha profunda gratidão à minha esposa, Lorena Lara, e aos meus filhos, Daniel Júnior e Antonella Lara, pelo amor, paciência e compreensão ao longo desta jornada. Seu apoio incondicional foi essencial para que eu pudesse me dedicar a este trabalho, mesmo nos momentos de ausência e desafios. A presença, o carinho e a força que encontrei em cada um deles foram fundamentais para a concretização desta conquista.

*“Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)*

Resumo

Sistemas industriais estão sujeitos a diferentes tipos de alterações em seus pontos de operação, decorrentes tanto de ações deliberadas, como a modificação de um *setpoint*, quanto de falhas, degradação de equipamentos ou condições anormais da planta. Tais mudanças podem representar novos estados válidos, oportunidades de otimização ou indicar desvios que comprometem o desempenho e a segurança do processo. Diante desse cenário, torna-se imprescindível o emprego de mecanismos automáticos para a detecção de mudanças, uma vez que a supervisão humana contínua se mostra inviável em sistemas com grande número de variáveis. Essa limitação é ainda mais evidente em processos industriais, nos quais as séries temporais são tipicamente de alta dimensionalidade, apresentam comportamento não estacionário e são frequentemente afetadas por ruído e pela presença de *outliers*. Neste contexto, este trabalho propõe duas metodologias não supervisionadas e robustas a ruído para detecção automática de mudanças incrementais em processos industriais. A primeira, denominada **NR-EIDDM** (*Noise-Resistant Equal-Intensities Drift Detection Method*), baseia-se no particionamento n -dimensional do espaço de entrada e na análise estatística de histogramas com densidades iguais, obtidos por meio dos algoritmos DBSCAN, responsável pela filtragem de ruídos e *outliers*, e *K-Means* com restrições, que define partições de igual densidade. Em condições estacionárias, espera-se que as distribuições apresentem intensidades semelhantes, sendo desvios significativos identificados pela aplicação do teste qui-quadrado entre histogramas sucessivos. A segunda metodologia, denominada **UDrAE** (*Unsupervised Drift detection via Denoising AutoEncoder reconstruction loss*), emprega um *Denoising AutoEncoder* capaz de modelar o comportamento normal do sistema enquanto atenua ruídos de medição. Alterações na distribuição de entrada são inferidas por meio do monitoramento da função de perda de reconstrução, cujos aumentos significativos indicam possíveis *drifts* de conceito. Os métodos foram avaliados com bases sintéticas e reais, incluindo conjuntos gaussianos com níveis variados de ruído e simulações do Sistema de Tanques Interativos (STI) do Laboratório de Validação de Sistemas (LVAS/PPGEE-UFMG). Para validação em contexto industrial, utilizaram-se bases de dados reais não rotuladas provenientes de sistemas da Petrobras, permitindo avaliar a aplicabilidade das propostas em cenários complexos. A análise dos resultados, com base em métricas de precisão, *F1-score* e atraso médio de detecção, demonstrou que ambas as abordagens identificam de forma eficaz a maioria das mudanças nos diferentes conjuntos de dados, mantendo desempenho competitivo e robusto em comparação com métodos de referência da literatura, como QuantTree, QT-EWMA, SPLN, OCDD, D3 e IKS.

Palavras-chave: *drift*, detecção de mudanças de conceito, processos industriais, *autoencoder*, histogramas, DBSCAN, *constrained-kmeans*, *Denoising AutoEncoder*.

Abstract

Industrial systems are subject to different types of changes in their operating points, arising both from deliberate actions, such as *setpoint* adjustments, and from faults, equipment degradation, or abnormal plant conditions. Such changes may represent new valid operating states, opportunities for optimization, or deviations that compromise process performance and safety. In this context, the adoption of automatic change detection mechanisms becomes imperative, since continuous human supervision is unfeasible in systems with a large number of variables. This limitation is even more pronounced in industrial processes, where time series are typically high-dimensional, exhibit non-stationary behavior, and are frequently affected by measurement noise and the presence of *outliers*. Within this framework, this work proposes two unsupervised and noise-robust methodologies for the automatic detection of incremental changes in industrial processes. The first method, termed **NR-EIDDM** (*Noise-Resistant Equal-Intensities Drift Detection Method*), is based on an n -dimensional partitioning of the input space and on the statistical analysis of equal-density histograms, obtained through the DBSCAN algorithm, responsible for filtering noise and *outliers*, and a constrained K -Means algorithm, which defines partitions with equal density. Under stationary conditions, the distributions are expected to exhibit similar intensities, with significant deviations being identified through the application of the chi-square test between successive histograms. The second methodology, referred to as **UDrAE** (*Unsupervised Drift detection via Denoising AutoEncoder reconstruction loss*), employs a *Denoising AutoEncoder* capable of modeling the normal behavior of the system while attenuating measurement noise. Changes in the input distribution are inferred by monitoring the reconstruction loss function, whose significant increases indicate potential concept *drifts*. The proposed methods were evaluated using both synthetic and real datasets, including Gaussian distributions with varying noise levels and simulations of the Interactive Tank System (ITS) from the System Validation Laboratory (LVA/PPGEE-UFG). For validation in an industrial context, real unlabeled datasets from Petrobras systems were employed, allowing the assessment of the applicability of the proposed approaches in complex real-world scenarios. The analysis of the results, based on precision, $F1$ -score, and average detection delay metrics, demonstrated that both approaches effectively identify the majority of changes across different datasets, while maintaining competitive and robust performance when compared to reference methods from the literature, such as QuantTree, QT-EWMA, SPL, OCDD, D3, and IKS.

Keywords: concept drift; concept drift detection, industrial processes, autoencoder, histograms, DBSCAN, constrained k-means, Denoising AutoEncoder.

Lista de Ilustrações

Figura 1 – Tipos de Mudanças. Fonte: Lu et al. [2020].	20
Figura 2 – <i>Framework</i> apresentado por Lu et al. [2020] para detecção de <i>drift</i> . Fonte: Adaptados de Lu et al. [2020].	28
Figura 3 – Análise quantitativa apresentada por Iwashita and Papa [2018] para publicações relacionadas a detecção de <i>drift</i> . Fonte: Adaptado de Iwashita and Papa [2018].	30
Figura 4 – Método apresentado por Souza et al. [2020] para detecção de <i>drift</i> . Fonte: Adaptado de Souza et al. [2020].	31
Figura 5 – Exemplo de funcionamento do DBSCAN para $\gamma = 4$ - Adaptado da internet.	44
Figura 6 – Exemplo de utilização do método do cotovelo, em destaque a determinação de ε tendo $\gamma = 5$	44
Figura 7 – Arquitetura do método proposto.	46
Figura 8 – Exemplo de filtragem e criação de <i>subclusters</i>	47
Figura 9 – Esquemático simplificado para a estratégia proposta.	52
Figura 10 – Arquitetura do <i>AutoEncoder</i> proposto	55
Figura 11 – Exemplo do resultado apresentado pelo <i>AutoEncoder</i> implementado	56
Figura 12 – Esquemático simplificado para arquitetura proposta	57
Figura 13 – Sistema de Tanques Interativos. Fonte: Adaptado de Caminhas et al. [1997].	64
Figura 14 – Exemplo de base gaussiana gerada	68
Figura 15 – Exemplo de simulação de entupimento entre TQ-1 e TQ-2 a um fator de 0.28.	70

Lista de Tabelas

Tabela 1 – Síntese comparativa das principais características dos métodos propostos	61
Tabela 2 – Tipos de falhas no Sistema de Tanques Interativos.	67
Tabela 3 – Detalhamento dos conjuntos de dados obtidos da base de Gaussianas. .	68
Tabela 4 – Detalhamento dos conjuntos de dados simulados a partir do modelo do STI.	69
Tabela 5 – Distribuição das amostras nos conjuntos de dados de sensores de gás. .	70
Tabela 6 – 3W - Conjuntos de dados selecionados.	71
Tabela 7 – Avaliação de desempenho do método NR-EIDDM para os dados gaussianos.	73
Tabela 8 – Avaliação do método UDrAE para o conjunto de bases gaussianas. . . .	74
Tabela 9 – Avaliação de desempenho do método NR-EIDDM para os dados do STI.	74
Tabela 10 – Avaliação do método UDrAE para as bases STI.	75
Tabela 11 – Resultados do método NR-EIDDM para a os conjuntos de dados de Sensores de Gás.	75
Tabela 12 – Resultados do método UDrAE para a os conjuntos de dados de Sensores de Gás.	76
Tabela 13 – 3W (NR-EIDDM) - Resultados combinados com o classificador.	76
Tabela 14 – 3W (UDrAE) - Resultados combinados com o classificador.	76
Tabela 15 – Comparação de métricas para o conjunto de dados T85.	77
Tabela 16 – Comparação de métricas para o conjunto de dados T85.	77
Tabela 17 – Bases de dados Gaussianas - Comparação da precisão entre os métodos QT-EWMA, QT, SPLL, NR-EIDDM, OCDD, D3, IKS e UDrAE. . . .	78
Tabela 18 – Bases de dados Gaussianas - Comparação do <i>F1-score</i> entre os métodos QT-EWMA, QT, SPLL, NR-EIDDM, OCDD, D3, IKS e UDrAE. . . .	79
Tabela 19 – Bases de dados Gaussianas — Comparação do atraso médio entre os métodos QT-EWMA, QT, SPLL, NR-EIDDM, OCDD, D3, IKS e UDrAE.	79
Tabela 20 – Bases de dados do STI — Comparação da precisão entre os métodos NR-EIDDM, OCDD, D3, UDrAE e SPLL.	80
Tabela 21 – Bases de dados do STI — Comparação do <i>F1-score</i> entre os métodos NR-EIDDM, OCDD, D3, UDrAE e SPLL.	80

Tabela 22 – Comparação do atraso médio entre os métodos NR-EIDDM, OCDD, D3, UDrAE e SPLL para os conjuntos de dados do STI.	81
Tabela 23 – Comparação da precisão entre os métodos D3, IKS, NR-EIDDM e UDrAE para os conjuntos de dados de sensores de gás.	81
Tabela 24 – Comparação do F1-score entre os métodos D3, IKS, NR-EIDDM e UDrAE para os conjuntos de dados de sensores de gás.	81
Tabela 25 – Comparação do atraso médio entre os métodos D3, IKS, NR-EIDDM e UDrAE para os conjuntos de dados de sensores de gás.	82
Tabela 26 – 3W — Comparação da Precisão entre o <i>Baseline</i> e os métodos OCDD, D3, IKS, NR-EIDDM e UDrAE.	82
Tabela 27 – 3W — Comparação do F1-score entre o <i>Baseline</i> e os métodos em Adaptado (OCDD, D3, IKS, NR-EIDDM e UDrAE).	82
Tabela 28 – Comparação de métricas para o conjunto de dados T85.	83

Lista de Algoritmos

1	Algoritmo NR-EIDDM	49
2	Função Realiza Partição Inicial	49
3	Função Atualiza Janelas	50
4	Algoritmo UDrAE	57
5	Função de Treinamento do DAE	58
6	Função de limiarização da perda	58

Glossário

batch Lote, Batelada. Pacote com quantidade limitada de dados. [30](#)

bootstrapping O Bootstrapping é uma técnica estatística não paramétrica computacionalmente intensiva de reamostragem que tem como finalidade obter informações de características da distribuição de alguma variável aleatória. [28](#)

denoising autoencoders *Denoising autoencoders* são redes neurais projetadas para aprendizado não supervisionado, com o objetivo de filtrar dados de entrada por meio de uma representação compacta e reconstruí-los a partir dessa representação. Eles são amplamente utilizados para tarefas como redução de dimensionalidade, remoção de ruído e detecção de anomalias.. [32](#)

drifts Veja *drift*. [25](#), [27](#), [29](#), [67–69](#)

drift O *concept drift* ou somente *drift* é uma mudança implícita nos dados que leva os modelos a ficarem defasados. [8](#), [10](#), [14](#), [23–34](#), [67](#), [68](#), [77](#)

log-likelihood Verossimilhança logarítmica. [32](#)

online Processamento em tempo real. [30](#), [33](#), [34](#)

outliers Ponto fora da curva, dado espúrio. [29](#), [36](#), [39](#)

Acrônimos

AUC *Area Under the ROC Curve.* [33](#)

DDM *Drift Detection Method.* [28](#), [29](#)

DELM *Dynamic Extreme Learning Machine.* [29](#)

EDDM *Early Drift Detection Method.* [28](#), [29](#)

FW-DDM *Fuzzy Windowing Drift Detection Method.* [29](#)

GMA *Geometric Moving Average .* [29](#)

HDDM *Hoeffding Drift Detection Method.* [29](#)

LLDD *Learning with Local Drift Detection.* [28](#)

Sumário

1	Introdução	19
1.1	Motivação	23
1.2	Definição do problema	24
1.3	Objetivos do trabalho	25
1.4	Contribuições	25
1.4.1	Publicações	26
1.5	Organização do Trabalho	26
2	Estado da Arte	27
2.1	Abordagens Exploradas na Literatura	27
2.1.1	Métodos Não Supervisionados	29
2.1.1.1	IBDD (<i>Image-Based Drift Detector</i>)	30
2.1.1.2	IKS (<i>Incremental Kolmogorov-Smirnov Test</i>)	31
2.1.1.3	NM-DDM (<i>Nonparametric Multidimensional Drift Detection Method</i>)	32
2.1.1.4	<i>Plover</i>	32
2.1.1.5	3D (<i>Discriminative Drift Detector</i>)	33
2.1.1.6	QuantTree	33
2.1.1.7	QT-EWMA (<i>QuantTree Exponentially Weighted Moving Average</i>)	34
2.1.1.8	SPLL (<i>Semiparametric log-likelihood criterion</i>)	34
2.1.1.9	EI-Kmeans(<i>Equal Intensity Kmeans</i>)	35
2.1.1.10	OCDD (<i>One-Class Drift Detector</i>)	35
2.2	Considerações	36
3	Detectors de mudança de conceito propostos	38
3.1	Método para Detecção de Mudanças Incrementais Robusto a Ruído baseado em agrupamento de dados	39
3.1.1	DBSCAN - <i>Density-Based Spatial Clustering of Applications with Noise</i>	40
3.1.2	<i>K-Means</i> com restrições (<i>Constrained K-Means</i>)	41
3.1.3	Método Proposto - <i>Noise Resistent Equal Intensities Drift Detection Method</i> (NR-EIDDM)	43
3.1.4	Análise de Complexidade de Tempo	48
3.1.5	Discussão sobre a Avaliação de Parâmetros Externos	50
3.1.6	Considerações	51
3.2	Método para Detecção de Mudanças Incrementais baseado em <i>Denoising AutoEncoder</i> (DAE) e monitoração dos resíduos	51

3.2.1	Método Proposto - <i>Unsupervised Drift detection via Denoising AutoEncoder reconstruction loss</i> (UDraE)	52
3.2.2	Caracterização da Mudança	53
3.2.3	<i>Denoising AutoEncoder</i> Proposto	53
3.2.4	Limiarização	55
3.2.5	Algoritmo	57
3.2.6	Discussão sobre a Avaliação de Parâmetro Externo	59
3.2.7	Análise de Complexidade	59
3.2.8	Considerações	60
3.3	Síntese dos métodos propostos	60
4	Experimentos e Resultados	62
4.1	Desafios e Observações na Avaliação com Bases Reais e Artificiais	62
4.2	Sistema de Tanques Interativos - STI	63
4.2.1	Modelagem conceitual	64
4.2.2	Modelagem com falhas e considerações práticas	65
4.3	Bases Usadas na Avaliação	66
4.3.1	Bases Artificiais	67
4.3.1.1	Gaussianas	67
4.3.1.2	STI Simulado	68
4.3.2	Bases Reais	69
4.3.2.1	Conjunto de Dados “ <i>Gas Sensor Array</i> ”	69
4.3.2.2	3W	71
4.3.3	T85 - Petrobras	71
4.4	Métricas de Avaliação	72
4.5	Experimentos	73
4.5.1	Gaussianas (NR-EIDDM)	73
4.5.2	Gaussianas (UDrAE)	74
4.5.3	STI Simulado (NR-EIDDM)	74
4.5.4	STI Simulado (UDrAE)	75
4.5.5	Bases de Sensores de Gás (NR-EIDDM)	75
4.5.6	Bases de Sensores de Gás (UDrAE)	75
4.5.7	3W (NR-EIDDM)	76
4.5.8	3W (UDrAE)	76
4.5.9	T85 (NR-EIDDM)	77
4.5.10	T85 (UDrAE)	77
4.6	Comparações	77
4.6.1	Gaussianas	78
4.6.2	STI Simulado	78
4.6.3	Base de Sensores de Gás	79

4.6.4	3W	80
4.6.5	T85	80
4.7	Considerações	81
5	Conclusão e Trabalhos Futuros	85
5.1	Síntese das Contribuições	85
5.2	Vantagens e Características Observadas	86
5.3	Limitações e Hipóteses de Projeto	87
5.4	Trabalhos Futuros	88
5.5	Considerações Finais	89
	Referências	90

Capítulo 1

Introdução

A natureza de um fluxo de dados pode se modificar ao longo do tempo. Quando isso ocorre, geralmente está associada uma perda de desempenho e degradação de modelos preditivos que assumem uma relação estática entre entradas e saídas [Gama et al., 2014, Lu et al., 2020, Bayram et al., 2022]. Em sistemas dinâmicos monitorados por séries temporais, é comum que diferentes comportamentos se manifestem em distintos pontos de operação. A identificação de mudanças nesses pontos é essencial para garantir a eficiência e o desempenho dos modelos, sendo também crucial para a segurança em contextos nos quais os dados são utilizados para o monitoramento de processos industriais. Essas transições podem ocorrer de diversas maneiras [Ghomeshi et al., 2019, Lu et al., 2020, Yuan et al., 2022]:

- **Abrupta** — Ocorre quando há uma mudança súbita e significativa no conceito subjacente aos dados, de modo que o novo conceito substitui completamente o anterior em um intervalo de tempo muito curto. Esse tipo de mudança é comum em situações de falhas de componentes, trocas abruptas de configuração, paradas de emergência ou eventos anômalos de processo. A principal característica desse tipo de mudança é a descontinuidade temporal, que provoca uma variação abrupta nas estatísticas do processo (média, variância, correlação entre variáveis, etc.). Por esse motivo, é mais facilmente identificável por métodos baseados em limiares ou estatísticas de desvio.
- **Gradual** — Nesse tipo de mudança, os conceitos antigo e novo coexistem durante um período de transição, alternando-se de forma intermitente e com frequências variáveis. Inicialmente, o conceito antigo predomina, mas sua frequência diminui à medida que o novo conceito se consolida. Essa coexistência pode ocorrer, por exemplo, durante o desgaste progressivo de sensores, em que falhas de leitura tornam-se cada vez mais frequentes até que o dispositivo deixe de fornecer dados.
- **Incremental** — Representa a forma mais sutil e difícil de caracterizar das mudanças

de conceito. Ocorre quando o conceito é incrementalmente substituído por outro, através de uma sequência contínua de estados intermediários, sem fronteiras claramente definidas entre o antigo e o novo regime. Em termos estatísticos, cada nova observação apresenta pequenas variações em relação à anterior, frequentemente sem significância isolada, mas que, ao longo do tempo, acumulam-se em uma transformação substancial do processo. Essa característica torna a mudança incremental a mais desafiadora para detecção automática, especialmente na presença de ruídos e *outliers*, que podem mascarar a tendência real. Um exemplo típico ocorre na degradação de uma placa de orifício, onde o desgaste incremental altera lentamente as medições de vazão e pressão, produzindo uma alteração quase imperceptível até que os desvios se tornem significativos.

- **Recorrente** — Ocorre quando um conceito previamente observado reaparece após determinado período de tempo, o que é comum em sistemas sazonais, cíclicos ou sujeitos a regimes operacionais repetitivos. Nesses casos, o modelo deve ser capaz de reconhecer que o conceito já foi aprendido anteriormente, evitando reclassificá-lo como uma nova mudança. Um exemplo clássico ocorre em processos com padrões de demanda periódicos, como o volume de entregas de alimentos durante dias úteis e finais de semana, em que há alternância cíclica entre diferentes regimes de operação. A principal dificuldade associada a esse tipo de mudança é distinguir entre uma recorrência legítima e uma nova configuração estruturalmente semelhante, exigindo abordagens com memória temporal ou mecanismos de reutilização de modelos.

A Figura 1 ilustra os diferentes tipos descritos.

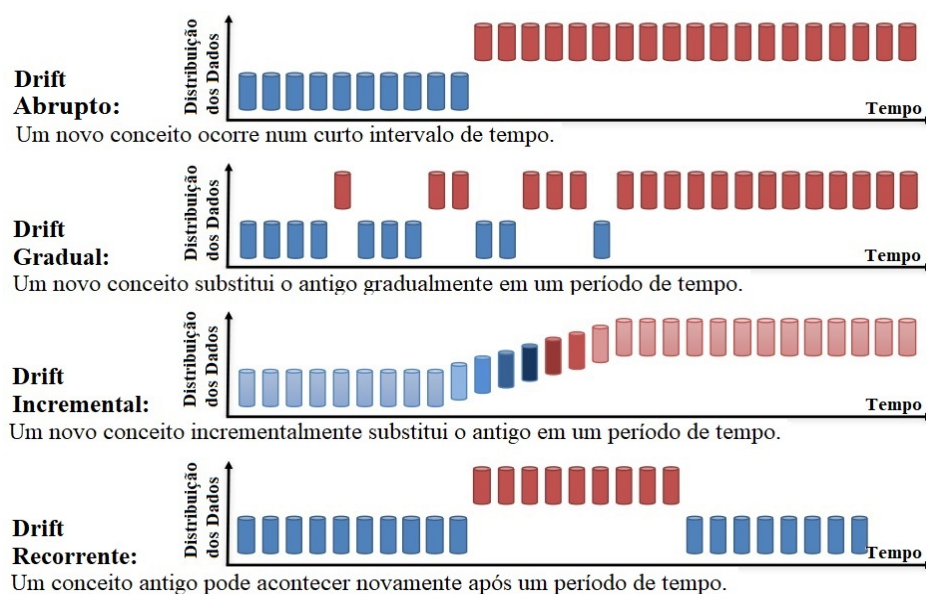


Figura 1 – Tipos de Mudanças. Fonte: Lu et al. [2020].

Conceitos previamente conhecidos podem apresentar propriedades estatísticas bem definidas e catalogadas. No entanto, em sistemas reais, essas características raramente permanecem estáticas por longos períodos de tempo. Na prática, fatores como variações físicas, alterações nas condições de operação ou interações com outros sistemas podem modificar o comportamento do processo, levando-o a operar em diferentes regimes. Esses novos regimes podem ser desejados e previamente conhecidos (e, portanto, catalogados), indesejados mas conhecidos, ou ainda completamente desconhecidos e não catalogados.

A identificação de mudanças de regimes pode ser realizada por meio de métodos automáticos de identificação de mudanças de conceito. De forma mais específica, no contexto industrial, sistemas dinâmicos estão especialmente sujeitos à natureza não estacionária de seus parâmetros [Sun et al., 2020, Precup et al., 2015]. Os dados provenientes desses sistemas são frequentemente contaminados por ruídos e *outliers*, o que compromete a estabilidade e a confiabilidade das medições. Fatores como a transmissão de dados por longas distâncias, a presença de imprecisões nos instrumentos de medição e a coleta de dados em ambientes sujeitos a variações de campo eletromagnético são exemplos recorrentes que explicam essa natureza ruidosa dos sinais observados [Li et al., 2019].

Alguns autores classificam essas perturbações como meras variações sensoriais, argumentando que apenas alterações decorrentes diretamente do processo devem ser consideradas como mudanças reais [Kadlec et al., 2009]. No entanto, sob a perspectiva de métodos baseados em dados, qualquer desvio significativo nos valores monitorados pode ser interpretado como um indicativo de anomalia ou de mudança em relação ao comportamento anteriormente observado. Assim, mesmo variações atribuídas a imperfeições nos sensores não devem ser negligenciadas, dado seu impacto potencial na interpretação e desempenho dos modelos preditivos.

Todos os tipos de mudanças podem ser observadas em ambientes industriais. No entanto, mudanças de conceito abruptas, como falhas em equipamentos, erros operacionais ou interrupções na comunicação, tendem a ser mais facilmente detectáveis devido à sua natureza repentina e evidente [Gama et al., 2014]. Esse caráter abrupto também pode se manifestar em mudanças graduais ou recorrentes, especialmente quando ocorrem transições rápidas entre estados previamente conhecidos.

Além disso, são comuns mudanças incrementais, típicas de processos de degradação progressiva, como o desgaste natural de componentes ou o acúmulo gradual de efeitos em variáveis do processo. Tais mudanças são particularmente desafiadoras para os sistemas de detecção automática. Como destacado por Guo et al. [2022], alterações incrementais costumam ser mais difíceis de identificar automaticamente em comparação com mudanças abruptas ou recorrentes, devido à sua natureza sutil e contínua. Na prática, é comum que modelos preditivos utilizados em inferências de variáveis de processos industriais necessitem de reconfiguração ou retreinamento periódico para lidar com essas alterações

lentas, garantindo assim a manutenção da acurácia e da estabilidade operacional do sistema [Lu and Chiang, 2018, Mittermaier et al., 2025].

O intervalo de retreinamento dos modelos é fortemente dependente das características do processo e da natureza das variáveis envolvidas [Lu and Chiang, 2018]. No entanto, na maioria dos casos, identificar precisamente o momento em que o sistema migra de forma incremental de uma região de operação para outra é uma tarefa desafiadora. Frequentemente, essa detecção ocorre de forma reativa, com intervenção do engenheiro de processo apenas após a manifestação de efeitos da mudança. Essa abordagem reativa pode gerar prejuízos financeiros, pode representar riscos operacionais relevantes à integridade da planta ou até mesmo perdas de oportunidades de otimização não identificadas a tempo.

Cabe destacar que a nova condição de operação nem sempre é indesejada. Pelo contrário, ela pode corresponder a uma situação mais vantajosa, com maior potencial de lucro, melhores condições de otimização ou margens de segurança ampliadas para as variáveis de processo. Entretanto, se essas mudanças não forem corretamente identificadas, tais oportunidades permanecem inexploradas.

Nesse contexto, é importante ressaltar que, embora falhas representem mudanças de conceito, nem toda mudança está associada a uma falha. Mudanças de conceito representam um fenômeno mais amplo do que somente falhas [Gama et al., 2014]. O reconhecimento oportuno dessas transições operacionais é, portanto, fundamental para garantir desempenho, segurança e eficiência no contexto industrial. Essa distinção é crucial em aplicações industriais, onde a capacidade de detectar e adaptar-se às mudanças, mesmo que não sejam falhas, pode aumentar significativamente a robustez e a confiabilidade dos sistemas de monitoramento.

A alteração no ambiente das variáveis de entrada, que resulta em mudanças nas propriedades estatísticas das variáveis-alvo, é conhecida na literatura como *concept drift* ou simplesmente *drift* [Lu et al., 2020]. A identificação precisa do momento em que um *drift* se inicia é essencial para a tomada de decisão eficaz em ambientes com fluxos contínuos de dados [Sobhani and Beigy, 2011]. Em diversos trabalhos da área, os termos *concept drift detection* e *change detection* são utilizados de forma intercambiável. Por esse motivo, neste trabalho, os termos "mudança" e "*drift*" podem ser utilizados como sinônimos. Também podem ser usados como sinônimos os termos "conceito" e "*concept*" associados a mudança e "*drift*".

Este estudo propõe duas abordagens voltadas à detecção de mudanças de conceito incrementais em processos industriais, com ênfase na robustez frente a observações atípicas do processo (ruídos e *outliers*). Tais observações, embora não constituam erros propriamente ditos, desviam-se dos padrões reais de operação e podem comprometer a acurácia dos modelos ao elevar a taxa de falsos positivos. A primeira proposta, denominada **NR-EIDDM** (*Noise-Resistant Equal-Intensities Drift Detection Method*), baseia-se no

particionamento n -dimensional do espaço de entrada e na comparação estatística de histogramas de igual densidade, obtidos por meio da filtragem de ruído e *outliers* via DBSCAN e da definição de partições balanceadas por *K-Means* com restrições. Sob condições estacionárias, espera-se a manutenção de intensidades semelhantes entre distribuições sucessivas, sendo desvios relevantes identificados pelo teste qui-quadrado. A segunda metodologia, denominada **UDrAE** (*Unsupervised Drift detection via Denoising AutoEncoder reconstruction loss*), emprega um *Denoising AutoEncoder* para modelar o comportamento normal do sistema e mitigar ruídos de medição. Ambas demonstraram robustez na identificação de *drifts* incrementais, uma vez que, frequentemente, essas mudanças apresentam variações de baixa magnitude em relação às oscilações naturais induzidas pelo ruído do processo. Nesse contexto, a resistência a observações atípicas torna-se um aspecto fundamental, pois permite distinguir alterações genuínas nos dados de processo de flutuações espúrias. Como consequência, a acurácia na detecção de *drifts* incrementais é aprimorada, inclusive em cenários industriais reais.

1.1 Motivação

O volume de dados gerado por processos e sistemas tem crescido exponencialmente nos últimos anos [Gama et al., 2014]. Esse crescimento é impulsionado por avanços tecnológicos como o desenvolvimento de soluções de armazenamento cada vez mais acessíveis, a ampliação da capacidade de transmissão de dados em tempo real e a crescente integração online dos processos industriais. No entanto, a mera disponibilidade dessa massa de dados não garante sua utilidade: é necessário usá-la para tomadas de decisão em tempo hábil. Para isso, modelos computacionais são construídos com o propósito de correlacionar dados históricos com conhecimentos previamente adquiridos em etapas anteriores do desenvolvimento.

Contudo, essa tarefa torna-se especialmente desafiadora quando a natureza dos dados é não estacionária. A presença de mudanças nos padrões estatísticos dos dados compromete a validade dos modelos previamente ajustados. A comunidade científica tem dedicado esforços consideráveis para lidar com esse problema. Quando ocorre uma mudança, ou *drift*, no comportamento do processo, o conhecimento estatístico incorporado nos modelos precisa ser atualizado. Identificar precisamente o momento em que um modelo deixa de ser confiável representa um dos grandes desafios enfrentados pela área de inteligência computacional. Lu et al. [2016a], Liu et al. [2017a], Coelho and de Castro [2020], Bayram et al. [2022], Gulcan and Can [2023], Barros and Santos [2018] e Coelho et al. [2023] são exemplos de esforços na pesquisa deste tema.

Concept drift incremental tem sido foco de pesquisa em alguns problemas práticos [Lu et al., 2020]. Entretanto, a maior parte das pesquisas tem seu foco em quando

o *drift* ocorreu, e não em como ou quais amostras fazem parte da mudança. Um maior entendimento da mudança pode facilitar tanto na adaptação quanto na comparação dos eventos. Identificar quando o *drift* ocorreu é a função essencial que todo método de detecção deve ter. Basicamente, esses métodos retornam o *timestamp* do momento de ocorrência da mudança. Entretanto, alguns poucos métodos trazem também alguma informação de como o *drift* ocorreu [Dasu et al., 2006, Lu et al., 2016b, 2014, Kifer et al., 2004]. Geralmente essa informação está relacionada com uma métrica de severidade da mudança. Desta forma, é possível quantificar a similaridade entre os conceitos antigo e novo. Já a duração do *drift* é uma indicação da região de transição entre os conceitos. Alguns métodos, além de evidenciar o quando, também apresentam quais amostras fazem parte desta região de transição [Dasu et al., 2006, Gama and Castillo, 2006, Liu et al., 2017a]. Um maior detalhamento de como a mudança ocorre e sua duração podem ser interessantes também no sentido de entender melhor o comportamento dos sistemas e agregar mais inteligência aos modelos. Isso faz com que a abordagem deixe de ser reativa somente no momento em que aconteceu o *drift* partindo para uma abordagem proativa. À medida que se compreende melhor a natureza e o mecanismo das mudanças, torna-se possível prever alterações estatísticas nas variáveis-alvo envolvidas, ou até mesmo identificar transformações nos dados de entrada que eliminem a necessidade de qualquer adaptação no modelo previamente treinado.

Dada a natureza dinâmica dos processos industriais, a identificação de diferentes pontos de operação pode se tornar um grande desafio. Esses pontos podem surgir a partir de mudanças graduais, inserção de diferentes pontos de referência, existência de alguma característica sazonal do processo ou até mesmo o desgaste lento e natural dos equipamentos. Associada a isso, existe a natureza complexa de monitoração que pode ocorrer por meio de muitas variáveis de processo, sendo umas mais sensíveis a determinados eventos que outras. Um ser humano, mesmo que muito atento e bem treinado, ainda pode não perceber determinadas tendências diante da complexidade das variáveis a serem observadas. Existe também a possibilidade de um cenário em que a mudança ocorre de forma que seja percebida somente por meio da análise de um determinado número de variáveis. Neste caso, cada variável individualmente pode estar dentro dos limiares e tolerâncias aceitáveis, porém o conjunto pode apresentar propriedades estatísticas diferentes. Diante disso, este trabalho investiga detectores não supervisionados robustos a dados espúrios e adequados para detectar mudanças incrementais em dados industriais

1.2 Definição do problema

Uma definição de *Concept drift* é apresentada por Liu et al. [2020a] de acordo com o que se segue: considere um espaço topológico, espaço de características, denotado como $\mathcal{X} \subseteq \mathbb{R}^d$, onde d é a dimensionalidade do espaço de características. Uma tupla

(X, y) denota uma instância observada em que $X \in \mathcal{X}$. X é também definido como vetor de características e $y \in \{y_1, \dots, y_c\}$ é a classe ou estado da planta. O subscrito c indica o número de classes. Um *stream* de dados pode ser definido como uma sequência de instâncias denotada por \mathcal{D} . Um *Concept drift* é dito existente se, para duas janelas de tempo T_1 e $T_2 \in \mathcal{D}$ distintas, a probabilidade conjunta de X e y são diferentes, ou seja, $p_{T_1}(X, y) \neq p_{T_2}(X, y)$.

A probabilidade conjunta é definida por $p(X, y) = p(y|X).p(X)$. Existem três formas de se alterar o valor da probabilidade conjunta. A primeira é quando somente $p(X)$ se altera enquanto $p(y|X)$ permanece constante, neste caso, a literatura define esse cenário como *drift* virtual. A segunda forma é quando $p(X)$ permanece constante e somente $p(y|X)$ se altera, neste cenário, o *drift* é definido como real. Por fim, existe a mistura dos cenários quando todas as probabilidades se alteram. Quando a última ocorre, existem *drifts* real e virtual ao mesmo tempo. De acordo com Pesaranghader et al. [2018], na prática, um *drift* virtual geralmente ocorre em conjunto com um *drift* real. Vale observar que métodos de detecção não supervisionados são capazes de perceber alterações somente em $p(X)$, uma vez que não têm acesso a y .

Desta forma, o problema de pesquisa que este trabalho busca estudar se resume em possíveis respostas para a pergunta: **como localizar temporalmente, de forma automática, um ou mais pontos de mudanças (*concept drift*) incrementais de um processo industrial sujeito a ruído por meio da análise de suas variáveis monitoradas?** Existem algumas abordagens e metodologias que procuram lidar com este problema. Este trabalho apresenta algumas das principais dessas metodologias além de propor outras duas estratégias.

1.3 Objetivos do trabalho

A proposta apresentada endereça o problema de identificação de *drifts* incrementais sob ruído. Apesar de poder ser empregada em mudanças graduais e abruptas, este não é o foco principal.

1.4 Contribuições

Grande parte dos trabalhos publicados na literatura trata o problema de identificação automática *drifts* sob a ótica de mudanças abruptas. Estudos como o de Iwashita and Papa [2018] evidenciam uma expressiva discrepância entre as abordagens supervisionadas e não supervisionadas, mostrando que apenas cerca de 3% das publicações existentes empregam métodos não supervisionados. Quando se busca aplicar técnicas de detecção de mudanças em dados reais, surge ainda o desafio adicional da presença de ruídos e

pontos atípicos, que podem comprometer a acurácia e a estabilidade das soluções. Diante desses obstáculos, a principal contribuição deste trabalho consiste na proposta de dois métodos não supervisionados, robustos a ruído, capazes de identificar automaticamente *drifts* incrementais em processos industriais.

1.4.1 Publicações

[1] - D. S. D. Lara, R. A. Coelho, C. L. Castro, and W. M. Caminhas. An unsupervised noise-resistant method for detection of incremental drifts. *Knowledge and Information Systems*, 2025a. doi: 10.1007/s10115-025-02592-2. (Publicado)¹

[2] - D. S. D. Lara, R. A. Coelho, C. L. Castro, and W. M. Caminhas. Unsupervised drift detection via denoising autoencoder reconstruction loss. *IEEE Access*, 2025b. (Em Revisão)

1.5 Organização do Trabalho

Este trabalho está organizado em cinco capítulos. O Capítulo 1 apresentou a introdução, contextualizando o problema de pesquisa, seus objetivos e contribuições. O Capítulo 2 apresenta a revisão da literatura, abordando os principais conceitos e métodos relacionados à detecção de *concept drift*. O Capítulo 3 descreve e detalha os métodos propostos, incluindo seus fundamentos e etapas de implementação. O Capítulo 4 apresenta os resultados experimentais e as análises obtidas a partir da aplicação dos métodos propostos em diferentes bases de dados. Por fim, o Capítulo 5 apresenta as conclusões do trabalho e discute possibilidades para trabalhos futuros.

¹ <https://link.springer.com/article/10.1007/s10115-025-02592-2>

Capítulo 2

Estado da Arte

Este capítulo apresenta uma revisão dos trabalhos mais citados da área nos últimos anos apresentando os métodos e soluções para o problema de identificação de *drifts*. Essa é uma área bastante vasta e, por esse motivo, foi realizado um esforço para dar um enfoque maior nos trabalhos não supervisionados que de alguma forma usa estratégias similares às propostas.

2.1 Abordagens Exploradas na Literatura

Há um grande esforço da comunidade científica, nos últimos anos, procurando meios eficazes de tratar o problema de detecção de *concept drift* em uma grande variedade de cenários. Alguns trabalhos de revisão que resumem de forma clara esses esforços são: Krawczyk et al. [2017], Ramírez-Gallego et al. [2017], Hinder et al. [2024], Lukats et al. [2025], Gama [2012], de Andrade Silva et al. [2013], Agrahari and Singh [2022], Lu et al. [2020] e Gemaque et al. [2020]. Em especial, Lu et al. [2020] apresentaram as principais estratégias que métodos de detecção de *drifts* geralmente seguem. A Figura 2 detalha isso. Um grande número de trabalhos de alguma maneira interage com as etapas mostradas para a identificação, adaptação ou correção. Entretanto, todos os *surveys* evidenciaram também que grande parte dos esforços estão focados na detecção de *drifts* abruptos. Por último, é observável o fato do grande número destas investigações que lidam com aprendizagem supervisionada. Iwashita and Papa [2018] apresentaram de forma quantitativa o volume de artigos da área que lida com aprendizagem supervisionada comparado com aqueles que lidam com aprendizagem não supervisionada.

Seguindo a estratégia de identificação de *drift* apresentado na Figura 2, um número considerável de métodos está baseado na tentativa de detectar o *drift* por meio da identificação da variação da distribuição estatística dos dados. Os testes de hipóteses para se identificar essa variação mudam, porém a ideia por trás é a mesma. Procuram-se criar modelos a partir de duas janelas de dados, sendo uma antes e outra após o *drift*. É então

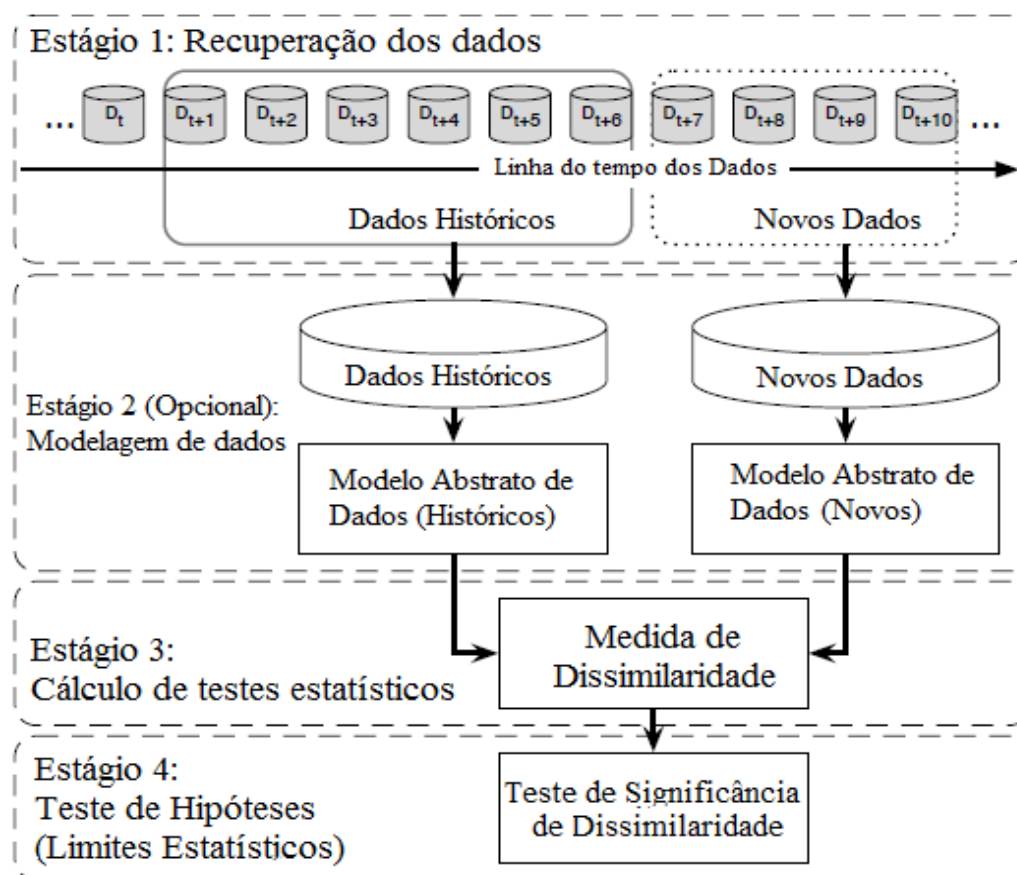


Figura 2 – *Framework* apresentado por Lu et al. [2020] para detecção de *drift*. Fonte: Adaptados de Lu et al. [2020].

realizada uma medida de dissimilaridade seguida por um teste estatístico de significância. Alguns métodos que usam esta metodologia são Alippi and Roveri [2008] e Zhang et al. [2017] que procuraram estimar novas distribuições dos testes estatísticos para adaptar o modelo. Já Bu et al. [2018] e Dasu et al. [2006] focaram sua abordagem em soluções por meio de *bootstrapping*. Lu et al. [2014] concentrou seus trabalhos em testes de permutação enquanto que Blanco et al. [2015] se focaram na identificação de limites baseados na desigualdade de Hoeffding. D'Angelo et al. [2011] apresentaram uma estratégia interessante baseada em agrupamento *fuzzy* capaz de transformar qualquer distribuição original de uma série temporal em outra cuja a distribuição modelada é a beta. Na sequência é aplicado o algoritmo de *Metropolis – Hastings* para identificar o ponto de mudança.

Apesar de inúmeros métodos já terem sido apresentados, um dos mais referenciados ainda é supervisionado. Esse método é o *Drift Detection Method (DDM)* apresentado por Gama et al. [2004]. O DDM definiu o nível de alerta e o nível de *drift* para a detecção de *concept drift*. A partir deste, inúmeras variações supervisionadas similares surgiram. Ainda em Lu et al. [2020], são apresentadas algumas destas variações que foram *Learning with Local Drift Detection (LLDD)* [Gama and Castillo, 2006], *Early Drift Detection Method (EDDM)* [Baena-Garcia et al., 2006], *Drift Detection Method based on*

the Hoeffding's inequality (HDDM) [Blanco et al., 2015], Fuzzy Windowing Drift Detection Method (FW-DDM) [Liu et al., 2017b] e Dynamic Extreme Learning Machine (DELM) [Xu and Wang, 2017].

Um fator também relevante é o fato de que poucos dos trabalhos encontrados são dedicados unicamente para processos industriais. Sun et al. [2020] apresentaram uma revisão de alguns destes métodos. A realidade dos processos industriais apresenta desafios que dificultam bastante o bom desempenho de grande parte dos métodos já citados. Frequentemente, dados industriais são contaminados por valores espúrios. Estudos no sentido de avaliar a robustez destes métodos com relação a ruídos também são escassos. A título de exemplo, o trabalho descrito em Jaramillo-Valbuena et al. [2017] procurou endereçar esse desafio por meio da avaliação do desempenho de 4 métodos conceituados até sua época no que diz respeito à detecção de *drift* com relação a ruídos e *outliers* (avaliou os resultados de acordo com a acurácia e Yonden's J). Os métodos avaliados foram DDM, EDDM, GMA e Exponentially Weighted Moving Average Chart Detection. EDDM foi o que apresentou melhor desempenho. EDDM surgiu com o intuito de resolver uma deficiência do DDM para detectar *drifts* incrementais.

O DDM e suas variações consistem em treinar um modelo para determinado conceito e, a partir de uma inferência de probabilidade para resultados futuros, determinar se o sistema sofreu uma mudança. A ideia central por trás desses métodos é que, para um modelo adequadamente treinado, espera-se que novos exemplos sejam previstos com uma taxa de acerto consistente com aquela observada durante o treinamento. Quando o modelo passa a apresentar um aumento significativo na taxa de erro, isso pode indicar que o sistema sofreu alguma mudança em sua dinâmica. Essa divergência entre o comportamento esperado e o observado é frequentemente um indício de *concept drift*, exigindo reavaliação ou adaptação do modelo. A principal diferença entre o DDM e o EDDM reside no fato de que o DDM apenas conta um número absoluto de erros para decidir se o sistema sofreu um *drift* enquanto que o EDDM leva em consideração o número de erros e a distância temporal em que os erros ocorreram.

2.1.1 Métodos Não Supervisionados

No estado da arte, os métodos de aprendizado e detecção de *concept drift* em ambientes de tempo real têm sido amplamente associados a abordagens não supervisionadas. Essa estratégia é particularmente atrativa no contexto industrial, onde a rotulação contínua dos dados é inviável devido ao alto custo, à complexidade dos processos e à ausência de supervisão humana. Em geral, esses métodos transformam os dados em representações que destacam alterações na distribuição estatística, permitindo a identificação de mudanças de conceito sem depender de informações rotuladas. Embora essa estratégia esteja se tornando cada vez mais presente na literatura, ainda não corresponde à maior parte das abordagens

existentes. [Iwashita and Papa \[2018\]](#) evidenciaram que aproximadamente 85% dos artigos publicados estão tratando de métodos supervisionados, 12% semi-supervisionados e somente 3% não supervisionados. Este trabalho também apresentou outras estatísticas tais como as bases mais usadas em *papers* que tratam o problema, percentual de abordagens multi-classe ou binária, aprendizagem *online* ou *batch* e os métodos mais usados com maior frequência em comparações nos artigos. A Figura 3 apresenta alguns dos resultados relacionados pelos autores.

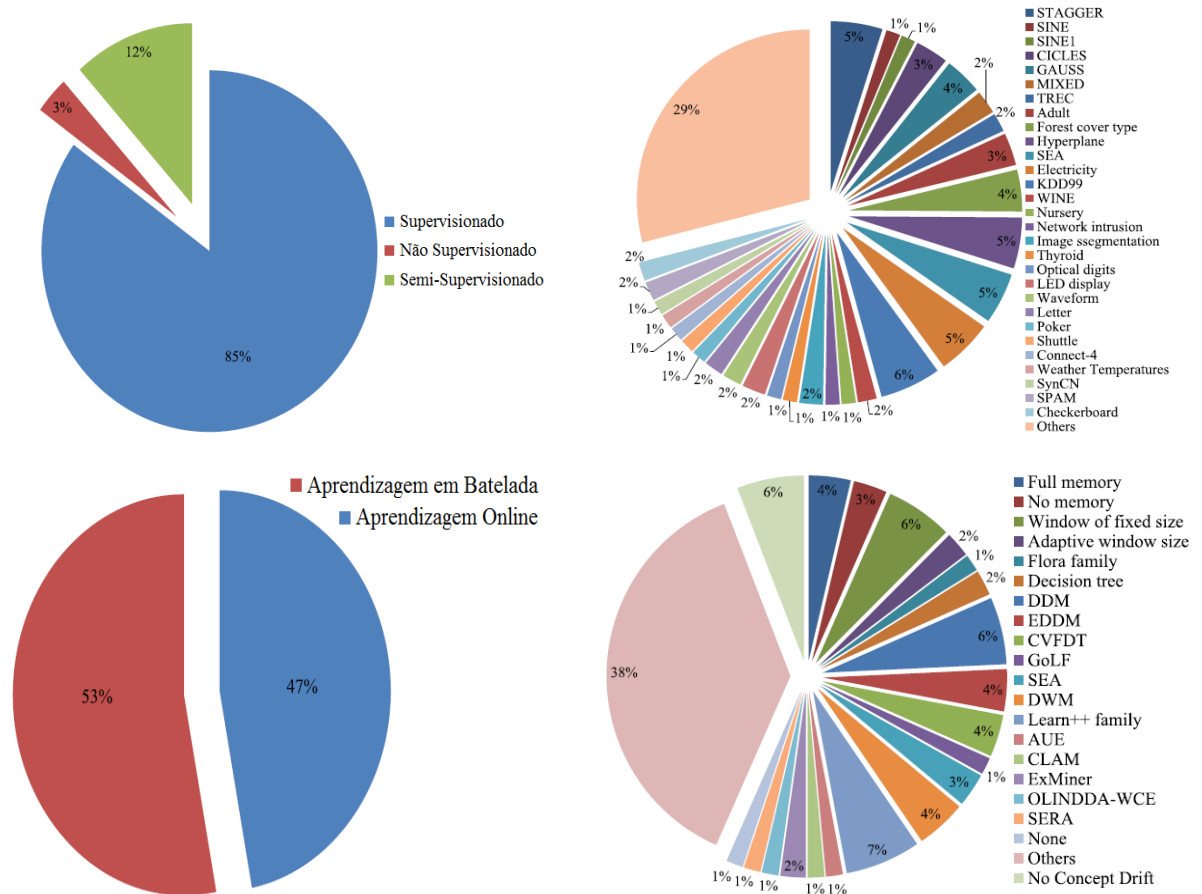


Figura 3 – Análise quantitativa apresentada por [Iwashita and Papa \[2018\]](#) para publicações relacionadas a detecção de *drift*. Fonte: Adaptado de [Iwashita and Papa \[2018\]](#).

Em muitas aplicações industriais, é desejável detectar o *concept drift* tão logo ele ocorra, permitindo uma resposta rápida do sistema. A seguir, são apresentadas algumas estratégias representativas de detecção de *drifts* de forma não supervisionada, com ênfase em suas capacidades de operar em cenários dinâmicos.

2.1.1.1 IBDD (*Image-Based Drift Detector*)

IBDD foi apresentado por [Souza et al. \[2020\]](#). Trata-se de uma abordagem de detecção não supervisionada baseada em processamento de imagens. Basicamente, a ideia consiste em, para cada amostra, empilhar todos os valores normalizados das características

(dimensões) em uma coluna da imagem. Vários exemplos sequenciais formam uma imagem. A resolução é o número de características *versus* número de amostras. A Figura 4 apresenta um exemplo de imagem formada desta maneira.

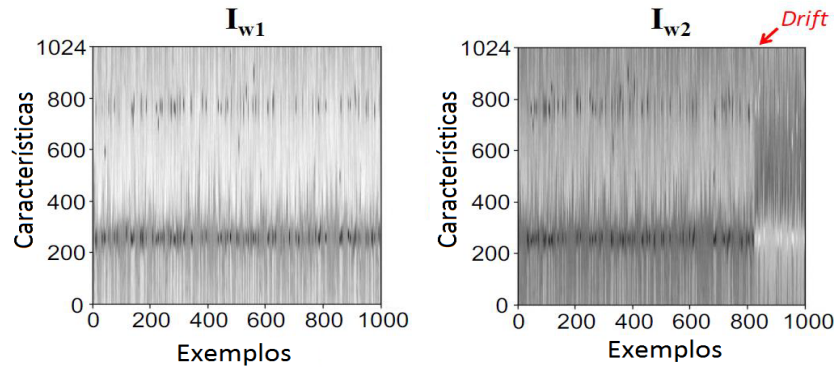


Figura 4 – Método apresentado por Souza et al. [2020] para detecção de *drift*. Fonte: Adaptado de Souza et al. [2020].

A detecção da mudança ocorre por meio do erro quadrático médio calculado para a janela avaliada em comparação com a janela de referência. É interessante observar que, desta maneira, para este exemplo específico, é possível identificar visualmente o *drift*.

2.1.1.2 IKS (*Incremental Kolmogorov-Smirnov Test*)

O trabalho apresentado por dos Reis et al. [2016] traz uma abordagem online não supervisionada usando IKS (*Incremental Kolmogorov-Smirnov Test*). A sugestão dos autores foi aplicar o teste KS (*Kolmogorov-Smirnov Test*) individualmente em cada atributo. O objetivo era reduzir o uso de testes multivariados e de funções de mapeamento. Isso simplifica o método e traz ganhos em processamento. Segundo os autores, a detecção da mudança a partir de um único atributo pode ser suficiente para a detecção de um *drift* geral.

Para cada atributo, o IKS mantém duas janelas de tamanho J . Uma delas é fixa contendo os dados usados para o treinamento do modelo mais recente. A outra é chamada de janela de detecção. Esta última é deslizante e contém os dados do fluxo de entrada. À medida que cada instância é disponibilizada, ela é classificada e o teste KS é executado comparando a distribuição dos dados na janela de referência com a janela de detecção. Se a hipótese nula de que as distribuições são idênticas é rejeitada, uma mudança é sinalizada. Caso contrário, a janela é deslocada e uma nova instância é obtida a partir do fluxo.

Foram propostas três possíveis reações para o caso de detecção de *drift*:

- Treinar um novo modelo com os dados da janela deslizante para rótulos reais. Neste caso, o *drift* é identificado de forma não supervisionada, porém o classificador

utilizado nos experimentos é supervisionado (por esta razão a necessidade de rótulos verdadeiros).

- Aplicar uma transformação AB (Transformar uma variável do intervalo A para o intervalo B) que siga os dois critérios a seguir: (a) aplicar uma transformação linear nos dados de referência, de forma a fazer com que os atributos envolvidos no *drift* tenham média e desvio padrão idênticos àquelas amostras nos dados de detecção. (b) Aplicar o teste KS novamente. Se o teste KS rejeitar a hipótese nula, um novo treinamento é realizado (neste caso também são necessários rótulos reais). É esperado que tal transformação leve as amostras de referência para uma distribuição que apresente as mesmas características da distribuição da janela mais recente. Neste caso, se o teste KS não rejeitar a hipótese nula após a transformação, é possível usar o mesmo classificador do item anterior usando os mesmos rótulos.
- Implementação de uma árvore de decisão modificada baseada nas amostras que causaram o *drift*. Quando uma mudança é identificada, rótulos verdadeiros são demandados somente para as instâncias que atingem as folhas com um atributo que disparou o *drift*.

2.1.1.3 NM-DDM (*Nonparametric Multidimensional Drift Detection Method*)

O NM-DDM foi proposto por [Mustafa et al. \[2017\]](#). O método proposto consiste no uso de *denoising autoencoders* e uma detecção de mudança baseada em *log-likelihood random walk*. A busca por novos conceitos ocorre no espaço latente do *denoising auto encoder* chamado pelos autores de *deep features*. A identificação do *drift* reside no cálculo da razão da verossimilhança logarítmica entre dados de duas janelas, uma antes e outra após o *concept drift*. A monitoração de possíveis pontos de *drift* é multidimensional. Já o cálculo da verossimilhança logarítmica é realizado para cada atributo e, se a maior razão entre as duas janelas é maior que um determinado limiar, o *drift* é identificado. Na sequência, as amostras da segunda janela são usadas para o treinamento de um novo modelo. Instâncias com baixa confiança são enviadas para o rótulo manual. Instâncias com alto valor de confiança são rotuladas com o valor previsto.

2.1.1.4 *Plover*

A técnica Plover [[de Mello et al., 2019](#)] apresentou uma análise teórica sobre fundamentos e garantias para detecção não supervisionada de *drift*. Segundo os autores, enquanto métodos supervisionados são suportados pela teoria estatística, métodos não supervisionados dependem de métricas internas ou externas que não seguem garantias similares. Medidas internas, tais como distâncias para os centroides, métricas para deter-

minação do quão compacto é o *cluster* ou médias e variâncias são computadas a partir dos dados.

Plover segue uma abordagem similar a outros métodos baseados em janelas fixas e deslizantes já presentes na literatura. A metodologia sugere o uso de duas janelas para comparação porém com funções mais genéricas para o cálculo da divergência. As funções para o cálculo da divergência entre as janelas devem atender basicamente dois pré-requisitos. A função deve ser selecionada de forma que seja independente dos dados de entrada. Eles devem ser independentes e identicamente distribuídos. Essa função pode ser média, variância, densidade espectral, etc. O método é genérico de forma a permitir a escolha das métricas de acordo com o problema. Este trabalho trouxe um arcabouço teórico para justificar o uso de tais estratégias, além de evidenciar a natureza dos métodos não supervisionados.

2.1.1.5 3D (*Discriminative Drift Detector*)

DDD ou 3D é um método não supervisionado *online* proposto por Gözüağık et al. [2019] que procura identificar o *drift* por meio da AUC (*area under the curve*). Os autores propuseram uma janela de tamanho fixo deslizante contendo basicamente dois conjuntos: o antigo e o novo. Um classificador simples é treinado para distinguir entre esses dois conjuntos. O *drift* é detectado de acordo com o desempenho desse classificador medido por meio da AUC. Esse processo é repetido sempre que chegam novos exemplos. A janela contendo os exemplos possui duas partes, uma com dados antigos e outra com dados recentes. O método propõe rotular os antigos com 0 e os novos com 1. Executa-se um classificador nestes dados e o desempenho da AUC é usado para avaliar se houve um *drift*.

2.1.1.6 QuantTree

O QuantTree é um método não supervisionado para detecção de mudanças, baseado em histogramas, proposto por Boracchi et al. [2018]. A principal ideia da abordagem é particionar o espaço de entrada em *bins* contendo densidade uniforme de amostras. Um aspecto notável demonstrado pelos autores é que, sob certas condições, as probabilidades associadas a esses *bins* tornam-se independentes da distribuição original dos dados. Foi demonstrado que essa propriedade também se aplica a todos os métodos de particionamento baseados em histogramas.

Essa característica tem implicações práticas importantes: ela permite que os limiares dos testes estatísticos utilizados para detectar mudanças possam ser definidos numericamente, com base em dados sintéticos gerados sob controle da taxa de falso positivo. Em outras palavras, **a robustez da etapa de avaliação torna-se independente da distribuição real dos dados de entrada**, o que é particularmente vantajoso em cenários não supervisionados.

De acordo com Liu et al. [2020a], métodos baseados em árvores que exploram essa ideia têm obtido resultados promissores para detecção de *drift* em modo *batch*. O QuantTree, em particular, adota uma estratégia de particionamento que busca atenuar o impacto da maldição da dimensionalidade ao construir regiões de igual densidade sem depender de uma discretização regular.

A detecção é realizada por meio do teste qui-quadrado de Pearson, aplicado entre duas janelas deslizantes de histogramas. A sinalização do *drift* ocorre com base em valores tabelados do teste, possibilitando uma inferência estatística direta.

Uma limitação do QuantTree é que, por utilizar partições hierárquicas baseadas em árvores, a construção dos *bins* não leva em consideração a estrutura espacial dos dados. Essa abordagem pode reduzir a interpretabilidade das regiões particionadas e, em alguns casos, gerar zonas cegas onde mudanças sutis no conceito passam despercebidas.

2.1.1.7 QT-EWMA (*QuantTree Exponentially Weighted Moving Average*)

QT-EWMA, apresetado por Frittoli et al. [2021], define uma estatística, baseada no particionamento do espaço de entrada implementado no QuantTree [Boracchi et al., 2018], que controla a taxa de alarmes falsos (falso positivos) por meio do controle de um indicador definido como (*Average Run Length* - ARL_0). Basicamente, o método trabalha com detecção *online* em *datastreams* multivariados baseado numa avaliação estatística que monitora as probabilidades dos *bins* num histograma QuantTree. Também define um esquema de Monte Carlo para calcular os limiares de detecção objetivando o controle da taxa de falso positivos. Usam, para isso, o resultado apresentado por Boracchi et al. [2018] de que os testes estatísticos realizados nas partições são independentes da distribuição original dos exemplos de entrada.

2.1.1.8 SPLL (*Semiparametric log-likelihood criterion*)

O SPLL, proposto por Kuncheva [2011], é um método não supervisionado para detecção de *drifts* em ambientes *online*, baseado em um arcabouço estatístico que compara duas janelas de dados deslizantes, denotadas por W_1 e W_2 . Os autores destacam que o método combina vantagens de duas abordagens previamente estabelecidas: uma baseada na divergência de Kullback-Leibler e outra no teste estatístico de Hotelling t^2 .

A proposta do SPLL se diferencia por apresentar melhor desempenho e maior simplicidade quando os dados são previamente normalizados. O princípio central dos métodos comparados consiste em estimar as distribuições associadas às duas janelas e disparar um alarme de mudança sempre que o teste estatístico indicar que os dados de uma janela provêm de uma distribuição significativamente distinta da outra.

Uma contribuição adicional do SPLL é a adoção de uma escala logarítmica para

o tratamento dos dados. Os resultados indicaram que, para dados não normalizados, o desempenho do SPLL é comparável ao dos métodos anteriores; entretanto, quando os dados são normalizados, o SPLL supera as abordagens concorrentes em precisão e estabilidade.

2.1.1.9 EI-Kmeans (*Equal Intensity Kmeans*)

EI-Kmeans é um método não supervisionado para detecção de mudanças baseado em histogramas, proposto por Liu et al. [2020a]. A técnica consiste em uma adaptação do algoritmo K-means convencional, com o objetivo de gerar partições que contenham densidades iguais de amostras. Para isso, os autores propõem uma estratégia de inicialização que garante que cada *bin* (ou partição) abranja aproximadamente o mesmo número de pontos. Estudos anteriores relataram que *bins* construídos com densidade uniforme apresentaram resultados superiores Liu et al. [2020a, 2018], Boracchi et al. [2017].

O principal diferencial do EI-Kmeans em relação a outros métodos baseados em histogramas está no fato de considerar a estrutura espacial dos dados ao definir os *bins*. Essa característica confere ao método uma maior capacidade de interpretação, já que as regiões particionadas refletem melhor a distribuição real das amostras no espaço de entrada.

Após a definição dos *bins*, são calculadas as densidades de referência durante a fase de treinamento. Em seguida, o método compara dois histogramas ao longo do tempo: um fixo, estabelecido com base nos dados de treinamento, e outro construído de forma contínua à medida que novas amostras são recebidas em tempo real. A comparação entre essas distribuições é feita por meio do teste qui-quadrado de Pearson, cuja principal vantagem, segundo os autores deste trabalho e de trabalhos semelhantes, é ser independente da distribuição estatística subjacente dos dados, tornando o procedimento robusto e aplicável a diferentes cenários.

2.1.1.10 OCDD (*One-Class Drift Detector*)

OCDD (*One-Class Drift Detector*), proposto por Gözüağık and Can [2021], é um método não supervisionado voltado para a detecção de *concept drift* em fluxos de dados fundamentado em técnicas de modelagem de fronteiras de classe única. A proposta baseia-se na ideia de que, em cenários onde apenas dados estacionários estão disponíveis durante o treinamento, é possível modelar a distribuição por meio de uma máquina de vetor de suporte de classe única (*One-Class SVM*) e monitorar mudanças em sua capacidade de generalização ao longo do tempo.

O OCDD parte do pressuposto de que, após o treinamento de um modelo de detecção de *drifts* (normalmente um *One-Class SVM* ou outra técnica de fronteira), mudanças conceituais nos dados podem ser detectadas ao observar alterações sistemáticas nos erros de reconstrução ou rejeição. Em particular, o método monitora estatísticas

dos escores de decisão fornecidos pelo modelo ao processar novas amostras. Caso seja observada uma mudança consistente nos padrões de aceitação/rejeição, isto é, um aumento significativo na quantidade de amostras consideradas como fora da fronteira normal, o sistema sinaliza um possível *drift*. Os autores demonstraram, por meio de experimentos em conjuntos de dados sintéticos e reais, que o OCDD apresenta desempenho competitivo frente a outras abordagens não supervisionadas de detecção de mudanças.

2.2 Considerações

Diversas tendências e lacunas puderam ser identificadas ao longo desta revisão. Observa-se que ainda há amplo espaço para o avanço de abordagens voltadas especificamente a processos industriais, os quais apresentam características próprias, em geral, ausência de rótulos, forte presença de ruído e variações incrementais ao longo do tempo. Grande parte dos métodos encontrados na literatura concentra-se na detecção de *drifts* abruptos, ou busca abranger contextos genéricos de dados não estruturados, como fala e texto. Em contrapartida, as mudanças incrementais, mais representativas de fenômenos industriais contínuos, permanecem pouco exploradas. Conforme apontado por [Iwashita and Papa \[2018\]](#), há uma discrepância significativa entre o número de métodos supervisionados e não supervisionados disponíveis na literatura.

Outro aspecto crítico refere-se à robustez dos algoritmos diante de dados atípicos (*outliers*) e ruídos, uma condição recorrente em ambientes industriais reais. Poucos estudos abordam de forma sistemática o impacto dessas perturbações ou propõem estratégias específicas de mitigação. Além disso, persiste uma lacuna importante quanto à interpretabilidade das mudanças detectadas: além de identificar o instante de ocorrência do *drift*, é desejável que o método seja capaz de indicar quais amostras contribuíram para a mudança e qual a intensidade associada a ela, fornecendo assim suporte analítico à decisão de reconfiguração de modelos.

Embora muitos métodos apresentem avanços em adaptabilidade e sensibilidade estatística, ainda enfrentam limitações relevantes quando aplicados a fluxos industriais ruidosos e não rotulados. Abordagens como IBDD e IKS mantêm custo computacional tolerável, mas perdem desempenho em cenários com dimensões elevadas, em particular, QT-EWMA, QuantTree e SPLL retornam apenas o instante estimado de mudança. Métodos baseados em probabilidade local ou agrupamento, como SPLL e EI-KMeans, oferecem maior flexibilidade, porém degradam sob alta variabilidade do ruído. Já técnicas mais recentes, como OCDD, ainda dependem consideravelmente de calibração manual de limiares e parâmetros (No caso específico do OCDD, seus autores sugerem uma lista de parâmetros a depender dos cenários). Assim, verifica-se que nenhuma dessas soluções concilia, de forma robusta, detecção não supervisionada, robustez a ruído e estabilidade temporal, requisitos

essenciais para aplicações industriais contínuas. Diante desse panorama, o capítulo seguinte apresenta as metodologias propostas, concebidas para superar essas limitações.

Capítulo 3

Detectores de mudança de conceito propostos

Este trabalho propõe dois métodos não supervisionados, orientados a séries temporais oriundas de processos industriais, desenvolvidos especificamente para lidar com alguns desafios descritos na subseção 2.2. Ambos os métodos buscam oferecer robustez frente a pontos atípicos, garantir sensibilidade a mudanças incrementais na distribuição dos dados sem causar alarmes falsos, aumentando a confiança nas detecções.

O NR-EIDDM (*Noise-Resistant Equal Intensities Drift Detection Method*) combina um agrupamento adequado de dados seguido por um particionamento de forma que histogramas por partição possam ser calculados. A principal estratégia desta abordagem é mapear, no espaço n -dimensional, as regiões mais comuns para os pontos de operação. É realizada, então, uma monitoração destas regiões por meio da comparação de histogramas de treinamento e operação. Diferenças estatísticas nos histogramas indicam uma alteração nestes espaços e, conseqüentemente, uma possível mudança.

Já o **UDrAE** (*Unsupervised Drift detection via Denoising AutoEncoder reconstruction loss*) emprega técnicas de aprendizagem de máquina associadas à degradação do modelo para indicar uma mudança. É utilizado um *Denoising AutoEncoder* (DAE) em conjunto com uma avaliação de erro de reconstrução para modelar a estrutura dos dados e, conseqüentemente, identificar mudanças.

Ambas as propostas buscam preencher lacunas identificadas na literatura, mantendo, simultaneamente, viabilidade computacional para aplicação em ambientes industriais reais. As seções seguintes apresentam os fundamentos teóricos e os detalhes de implementação de cada abordagem.

3.1 Método para Detecção de Mudanças Incrementais Robusto a Ruído baseado em agrupamento de dados

O NR-EIDDM (*Noise-Resistant Equal Intensities Drift Detection Method*) é um método não supervisionado voltado para a detecção de mudanças de conceito incrementais em séries temporais de processos industriais. A proposta combina elementos do DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) [Ester et al., 1996] e do *constrained K-Means* [Bradley et al., 2000] com o objetivo de filtrar pontos atípicos e particionar o espaço de entrada de forma controlada e robusta.

O nome “*Noise-Resistant Equal Intensities Drift Detection Method*” sintetiza de maneira clara tanto a proposta quanto as principais contribuições do algoritmo. O termo “*Drift Detection Method*” evidencia o objetivo central de identificar mudanças na distribuição dos dados ao longo do tempo, aspecto essencial em fluxos contínuos e processos dinâmicos. O qualificativo “*Noise-Resistant*” destaca a robustez do método diante de dados espúrios e *outliers*, reduzindo a ocorrência de falsos positivos e aumentando a confiabilidade da detecção em cenários industriais reais. Por sua vez, a expressão “*Equal Intensities*” remete à técnica estatística empregada, na qual os dados são particionados em *bins* de intensidade uniforme (*equal-intensity binning*), o que possibilita comparações mais equilibradas entre distribuições mesmo em situações com escalas irregulares ou densidades variáveis. Assim, o nome escolhido procura comunicar de forma direta e precisa tanto a finalidade quanto a inovação do método, ressaltando sua robustez e aplicabilidade em contextos desafiadores de monitoramento de dados.

O DBSCAN é reconhecido por sua capacidade de formar agrupamentos de densidade variável e, sobretudo, por sua resiliência a pontos atípicos, característica fundamental em ambientes industriais. Já o *K-Means* com restrições (*constrained K-Means*) [Bradley et al., 2000] é utilizado para ajustar o agrupamento às características desejadas, permitindo a criação de partições mais homogêneas e controladas. Essa combinação visa garantir que os histogramas resultantes das partições tenham densidades uniformes. Apesar de não apresentar uma justificativa teórica, estudos anteriores [Boracchi et al., 2017, Liu et al., 2018, Yonekawa et al., 2022] afirmam que esquemas de particionamento com histogramas de densidade uniforme são superiores com base em avaliações experimentais.

Durante o treinamento, o espaço de entrada é particionado em regiões, ou *bins*, que armazenam histogramas com intensidades balanceadas entre si. Em outras palavras, os histogramas são construídos de forma que cada região contenha aproximadamente o mesmo número de amostras, promovendo comparações mais sensíveis a mudanças na distribuição original.

A detecção da mudança de conceito é então conduzida por meio do teste estatístico chi-quadrado de Pearson [Pearson, 1900], que compara a distribuição de frequências entre duas janelas de tempo: uma de referência (treinamento) e outra de monitoramento. A rejeição da hipótese nula indica que os dados da janela atual provêm de uma distribuição diferente, caracterizando, assim, um indício de *concept drift*.

3.1.1 DBSCAN - *Density-Based Spatial Clustering of Applications with Noise*

O DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) [Ester et al., 1996] é um algoritmo de agrupamento referenciado em vários trabalhos devido à sua capacidade de identificar grupos com formatos e tamanhos arbitrários, mesmo em conjuntos de dados com presença significativa de ruído [Khan et al., 2018]. Uma de suas principais vantagens é que não exige a definição prévia do número de grupos [Arya and Sikka, 2014], o que o torna especialmente atrativo para aplicações em ambientes industriais, onde a estrutura dos dados pode ser dinâmica e desconhecida a priori.

A ideia central por trás do DBSCAN é identificar agrupamentos com base na densidade local dos dados. De forma simplificada, um ponto é considerado pertencente a um grupo se a sua vizinhança contiver um número mínimo de pontos dentro de um raio pré-definido. Em outras palavras, a região ao redor de cada ponto deve apresentar uma densidade suficiente para caracterizar a presença de uma estrutura de agrupamento.

Para compreender adequadamente o funcionamento do algoritmo, algumas definições são necessárias. Seja D o conjunto de dados a ser particionado e $b \in D$ um ponto qualquer.

Definição 1: (ε - Vizinhança) A vizinhança de um ponto a com raio ε é chamada de ε -Vizinhança de a e é dada por: $N_\varepsilon(a) = \{b \in D \mid \text{dist}(a, b) < \varepsilon\}$.

Definição 2: (Ponto central) Se a vizinhança de um ponto contém ao menos um número mínimo de pontos, $MinPts$, então aquele ponto é chamado interno ou ponto central.

Definição 3: (Ponto de borda) Se a vizinhança de um ponto, $N_\varepsilon(a)$, contém menos que $MinPts$ mas contém algum ponto central, então o aquele ponto a é chamado de ponto de borda.

Definição 4: (Alcance direto por densidade) Um ponto a é alcançável por densidade diretamente do ponto b , se a está na vizinhança ε de b , e b é um ponto central.

Definição 5: (Alcance por densidade) Um ponto a é alcançável por densidade a partir do ponto b com respeito a ε e $MinPts$ em um conjunto D se existe uma cadeia de pontos $\{a_1, \dots, a_n\}$, tais que $a_1 = b$ e $a_n = a$ sendo a_{i+1} alcançável por densidade

diretamente de a_i com respeito a ε e $MinPts$, para $1 \leq i \leq n, a_i \in D$. Há portanto um fechamento transitivo do alcance por densidade.

Definição 6: (Conexão por densidade) Um objeto a é conectado por densidade ao objeto b com respeito a ε e $MinPts$ em um conjunto de objetos, D , se existe um objeto r em D tal que ambos a e b são alcançáveis por densidade do objeto r com respeito a ε e $MinPts$.

Definição 7: (Cluster DBSCAN) Um cluster C com respeito a ε e $MinPts$ é um conjunto não vazio de D satisfazendo as seguintes condições:

(Maximilidade) $\forall a, b \in D$: se $a \in C$, e b é alcançável por densidade a partir de a com respeito a ε e $MinPts$, então $b \in C$.

(Conectividade) $\forall a, b \in C$, a é conectado por densidade a b com respeito a ε e $MinPts$.

Um cluster DBSCAN é o conjunto de pontos conectados por densidade que é máximo com respeito a alcançabilidade por densidade. Além disso, um cluster DBSCAN é inequivocamente determinado por qualquer de seus centros [Ester et al., 1996].

Definição 8:(Ruído): Sejam C_1, C_2, \dots, C_k clusters do conjunto de dados D com respeito a ε e $MinPts$. Se um ponto a não pertence a nenhum destes k clusters, ele é um ruído. Ruídos são pontos que não são diretamente alcançados por algum ponto central.

Com essa modelagem, é possível realizar uma implementação que, para ε e $MinPts$ determinados, seja possível encontrar clusters em um conjunto de dados n -dimensionais D .

3.1.2 K -Means com restrições (*Constrained K-Means*)

O método K -Means com restrições (*Constrained K-Means*) [Bradley et al., 2000] é uma variação semi-supervisionada do K -Means original [MacQueen, 1965]. Basicamente, o método com restrições permite a incorporação de algumas informações conhecidas *a priori*. Desta forma, é possível limitar valores mínimos e máximos para as quantidades de elementos de cada grupo, por exemplo. O algoritmo original do K -Means toma como entrada um conjunto de dados D e o número de grupos k e fornece na saída um particionamento dos seus padrões em k grupos. Na maioria das aplicações, o valor de k é conhecido e os valores iniciais para os centros dos grupos são tomados aleatoriamente dentre os padrões do conjunto de dados [Teixeira et al., 2011]. Entretanto, o K -Means encontra as melhores opções locais para grupos. Isso significa que ele encontra um ótimo local de forma que a soma das distâncias de norma-2 entre cada ponto e o centro do grupo mais próximo é minimizada. O algoritmo padrão implementa duas etapas simples:

Etapas de atribuição: Atribui pontos para os grupos por meio do centroide mais próximo especificado:

$$P_j^{(t)} = \{X_i : \|X_i - C_j^{(t)}\| \leq \|X_i - C_{j^*}^{(t)}\| \forall j^* = 1, \dots, k\} \quad (3.1)$$

Onde X_i denota os pontos num espaço n -dimensional, C_j denota os centroides dos grupos e $P_j^{(t)}$ representa o conjunto de pontos pertencentes ao cluster j em um dado instante.

Etapa de atualização: Calcula a média para cada cluster a fim de obter os novos centroides:

$$C_j^{(t+1)} = \frac{1}{|P_j^{(t)}|} \sum_{X_i \in P_j^{(t)}} X_i \quad (3.2)$$

Essas etapas são executadas repetidamente até que as localizações dos centroides não sofram alterações adicionais. É garantida a minimização das distâncias euclidianas de cada ponto para o centro do grupo mais próximo. Diversas variantes do algoritmo *K-Means* foram propostas ao longo dos anos, visando melhorar o desempenho em alguns contextos ou aprimorar alguma característica para contextos específicos [Arthur and Vassilvitskii, 2006, Fränti and Kivijärvi, 2000, Pelleg et al., 2000, Likas et al., 2003]. Uma dessas variações é o *K-Means* com restrições (*constrained K-Means*) [Bradley et al., 2000], que se destaca por incorporar restrições explícitas no processo de agrupamento. Essa abordagem mantém a estrutura geral do *K-Means* convencional, mas substitui a etapa de atribuição por um processo baseado em técnicas de otimização e programação linear. Com isso, torna-se possível impor restrições como, por exemplo, um número mínimo ou máximo de pontos por grupo, representado por parâmetros como τ_h , controlando diretamente a composição dos grupos formados. Desta forma, o algoritmo funciona da seguinte maneira:

Dados m pontos em \mathbb{R}^n , um número mínimo de pontos $\tau_h \geq 0$, $h = 1, \dots, k$ e centros $C_1^{(t)}, C_2^{(t)}, \dots, C_k^{(t)}$ na iteração t , compute $C_1^{(t+1)}, C_2^{(t+1)}, \dots, C_k^{(t+1)}$ na iteração $t + 1$ de acordo com o que segue:

Etapa de atribuição: Seja $T_{i,h}^t$ uma solução para o seguinte problema de programação linear com $C_h^{(t)}$ fixo:

$$\text{minimize}_T \sum_{i=1}^n \sum_{h=1}^k T_{i,h} \cdot \left(\frac{1}{2} \|X_i - C_h^{(t)}\|^2\right) \quad (3.3)$$

$$\text{sujeito a } \sum_{i=1}^n T_{i,h} \geq \tau_h, h = 1, \dots, k \quad (3.4)$$

$$\sum_{h=1}^k T_{i,h} = 1, i = 1, \dots, n \quad (3.5)$$

$$T_{i,h} \geq 0, i = 1, \dots, n, h = 1, \dots, k \quad (3.6)$$

Etapa de atualização: Atualize $C_h^{(t+1)}$ da seguinte forma:

$$C_h^{(t+1)} = \begin{cases} \frac{\sum_{i=1}^m T_{i,h}^{(t)} X_i}{\sum_{i=1}^m T_{i,h}^{(t)}}, \text{ se } \sum_{i=1}^m T_{i,h}^{(t)} > 0 \\ C_h^{(t)}, \text{ caso contrário} \end{cases} \quad (3.7)$$

Note que $T_{i,h}$ é uma variável de seleção assumindo o valor 1 se o ponto X_i é mais próximo do centro C_h e zero caso contrário.

Essa estratégia é menos propensa a encontrar ótimos locais mesmo para os casos que se permitem restrições mais relaxadas, tornando o algoritmo mais próximo do *K-Means* original [Bradley et al., 2000]. Segundo Malinen and Fränti [2014], enquanto a complexidade de tempo da etapa de atribuição do *K-Means* convencional é $O(k.n)$, a complexidade de tempo da etapa de atribuição para o *K-Means* com restrições é $O(k^{3.5}n^{3.5})$, sendo k o número de grupos e n o número de exemplos.

3.1.3 Método Proposto - *Noise Resistent Equal Intensities Drift Detection Method* (NR-EIDDM)

Uma vez estabelecidos os conceitos fundamentais para a construção do método proposto, esta seção descreve, em detalhes, a estratégia adotada para a detecção de *drifts*.

A estratégia realiza o particionamento do espaço de entrada por meio de um processo de aprendizado conduzido sobre um conjunto inicial de amostras. Após a aquisição dessa amostragem, o algoritmo DBSCAN [Ester et al., 1996] é aplicado com o objetivo de identificar e remover pontos atípicos do conjunto de dados. Essa filtragem inicial contribui para uma estrutura de partição mais consistente, tornando-a menos sensível a ruídos e garantindo maior robustez às análises subsequentes.

O DBSCAN requer dois parâmetros principais: ε (epsilon) e γ (também referido como *MinPts*). Esses parâmetros influenciam diretamente a sensibilidade do algoritmo: ε define a distância máxima entre dois pontos para que sejam considerados vizinhos, enquanto γ especifica o número mínimo de pontos necessários para formar uma região densa considerada um cluster.

A seleção desses parâmetros é realizada automaticamente, conforme estratégia definida pelo método. A Figura 5 ilustra o funcionamento do DBSCAN, destacando a separação entre pontos pertencentes aos grupos e aqueles considerados atípicos.

Note que o ponto A e os demais vermelhos são pontos centrais. A área ao redor destes pontos em um raio ε contém pelo menos 4 outros pontos (incluindo o mesmo). Eles formam um único agrupamento porque todos os pontos são mutuamente alcançáveis por densidade, ou seja, cada ponto pode ser conectado a qualquer outro por uma cadeia de vizinhança dentro do raio estabelecido. Os pontos B e C não são centrais porém são alcançáveis por A e assim pertencem ao mesmo grupo. Já o ponto N é um ponto atípico por não ser um ponto central e muito menos diretamente alcançável a partir de qualquer outro ponto do grupo.

O valor de ε deve ser determinado a partir do problema a ser tratado. Em

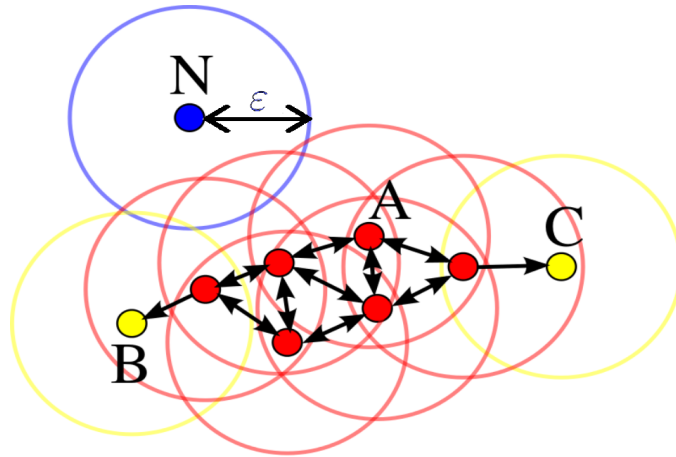


Figura 5 – Exemplo de funcionamento do DBSCAN para $\gamma = 4$ - Adaptado da internet.

concordância com o resultado apresentado por Sander et al. [1998], no algoritmo proposto, γ recebe o dobro do número de características da base decrescido de uma unidade. Já a determinação de ϵ exige uma análise mais detalhada. A escolha é feita por meio da plotagem de um gráfico de k -distâncias médias ordenadas para os k vizinhos mais próximos. O método proposto consiste em computar as distâncias dos k mais próximos vizinhos em uma matriz de pontos. A ideia é calcular a média das distâncias de cada ponto para os seus k vizinhos mais próximos. O Valor de k deve ser γ ($k = \gamma$). Essas k -distâncias médias são plotadas em ordem crescente. O objetivo é encontrar o ponto de maior curvatura ("cotovelo da curva"). Este ponto corresponde ao ponto ótimo para o parâmetro ϵ . Apesar de serem sugeridos automaticamente, esses parâmetros continuam liberados para que o usuário possa fazer um ajuste fino, caso necessário. A Figura 6 apresenta um exemplo para um ponto ótimo, em destaque, para ϵ próximo de 0.15 em uma base de aproximadamente 5500 exemplos. O Algoritmo 1 detalha essa atribuição.

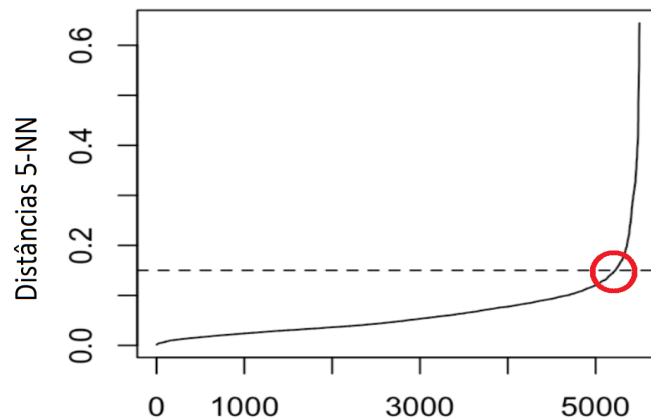


Figura 6 – Exemplo de utilização do método do cotovelo, em destaque a determinação de ϵ tendo $\gamma = 5$

O algoritmo DBSCAN tem como função identificar subconjuntos disjuntos de dados que formam agrupamentos coerentes, ao mesmo tempo em que isola os pontos

atípicos, que não pertencem a nenhum grupo. Assim, seja U o conjunto total de dados no espaço \mathbb{R}^d . O DBSCAN é capaz de produzir dois subconjuntos: S e N , tais que

$$U = \left(\bigcup_{k=1}^l S_k \right) \cup N, \quad \text{com } S_i \cap S_j = \emptyset, \quad S_i \cap N = \emptyset \quad \text{para todo } i \neq j,$$

onde S_k representa os agrupamentos válidos e N corresponde ao conjunto de pontos atípicos (ruído e *outliers*). O objetivo desta etapa é eliminar as regiões espúrias N e definir as regiões relevantes para o particionamento do espaço.

Em seguida, utiliza-se o algoritmo *K-Means* com restrições para localizar os centros dos agrupamentos $\{C_k\}_{k=1}^s$ em \mathbb{R}^d , de forma a minimizar a soma das distâncias Euclidianas (norma-2) entre cada ponto X_i e o centro C_k mais próximo, respeitando certas restrições de cardinalidade. As restrições impõem que:

$$e \leq |\tau_h| \leq s, \quad \text{para todo } h = 1, \dots, k,$$

onde $|\tau_h|$ denota o número de pontos atribuídos ao agrupamento h , e e e s são os limites inferior e superior, respectivamente, para o tamanho de cada grupo.

O dimensionamento de cada partição é realizado com base na definição do número de *subclusters* (número de *bins* - parâmetro do algoritmo) no menor grupo S_i retornado pelo DBSCAN. Definindo e e s adequadamente (valores idênticos, por exemplo), utiliza-se o *K-Means* com restrições para estabelecer partições de forma que cada *bin* possua o mesmo número de elementos.

Sejam l o número total de partições desejadas e S_i o menor conjunto entre os agrupamentos $\{S_k\}_{k=1}^l$ identificados pelo DBSCAN. Assim, o tamanho ideal de cada partição pode ser definido por:

$$C_{\text{size}} = \frac{|S_i|}{l}.$$

Esse valor C_{size} é utilizado como parâmetro no algoritmo *K-Means* com restrições, determinando a cardinalidade esperada para cada partição.

Com o valor de C_{size} fixado, o *K-Means* com restrições é aplicado a todos os grupos S_k resultantes do DBSCAN. Cada *subcluster* gerado por este processo define uma região no espaço de entrada correspondente a uma partição balanceada. Dessa forma, durante a etapa de treinamento, todos os *subclusters* são construídos com o objetivo de produzir histogramas com densidades uniformes.

Cada *subcluster* resultante representa uma partição balanceada do espaço de entrada, garantindo que todas as regiões tenham densidades similares. Essa homogeneidade, de acordo com referências práticas de trabalhos anteriores [Boracchi et al., 2017, Liu et al., 2018, Yonekawa et al., 2022], é importante para a construção de histogramas uniformes. Assim, durante a fase de treinamento, o modelo é preparado para detectar mudanças com

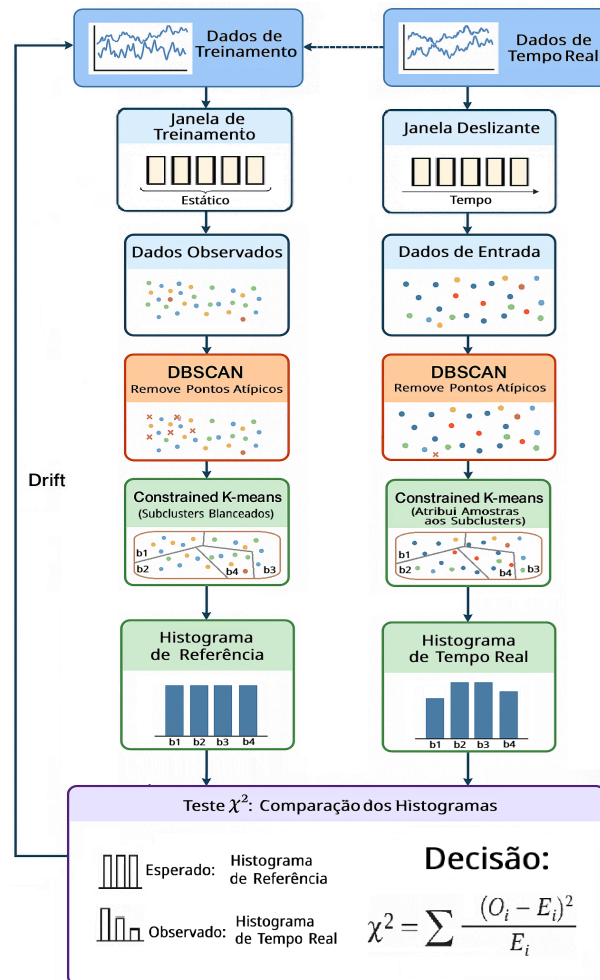


Figura 7 – Arquitetura do método proposto.

maior sensibilidade e robustez, uma vez que a comparação entre regiões torna-se mais estável e estatisticamente significativa.

Nas etapas de avaliação, a detecção de mudanças ocorre de forma *online*. Para cada nova amostra adquirida, a observação mais antiga da janela de detecção é descartada, permitindo uma atualização contínua das densidades observadas nas regiões previamente definidas. Espera-se que, em condições de regime permanente, as densidades dos histogramas dessas regiões permaneçam estáveis e similares às observadas durante o treinamento. Em outras palavras, os dados devem continuar se concentrando nas mesmas regiões com a mesma intensidade. Quando essa condição não é satisfeita, assume-se a ocorrência de um *drift*. A Figura 7 apresenta um esquemático para a arquitetura proposta.

A Figura 8 ilustra didaticamente os resultados obtidos pelo método proposto em uma base bidimensional. A Figura 8a mostra os dados de treinamento originais. Já a Figura 8b apresenta os resultados após a filtragem realizada pelo algoritmo DBSCAN, seguida do particionamento obtido via *K-Means* com restrições. Os pontos marcados com cruzes vermelhas representam os centróides de cada *subcluster*. Nota-se que os pontos destacados na Figura 8a foram corretamente identificados como atípicos e, portanto, não

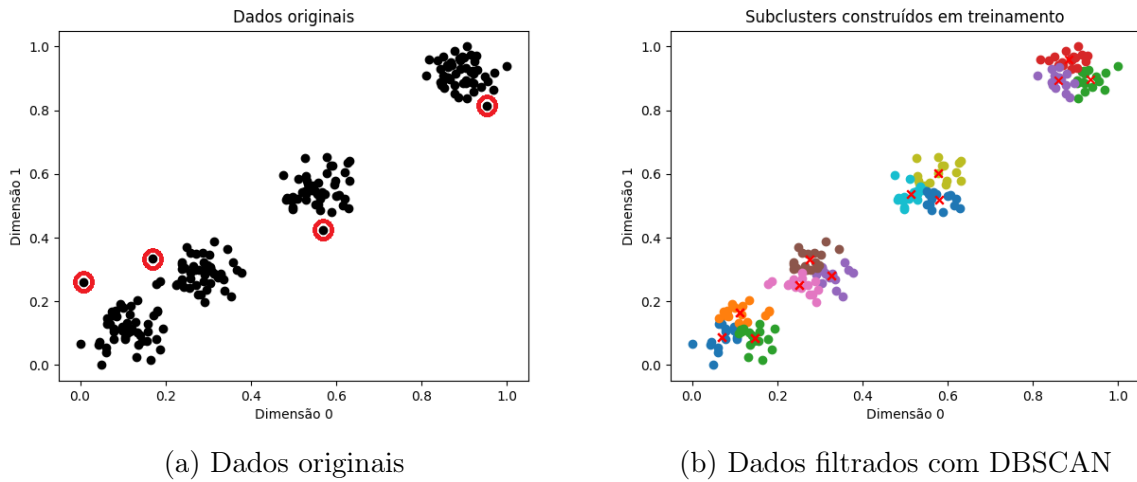


Figura 8 – Exemplo de filtragem e criação de *subclusters*

considerados na construção dos *subclusters* da Figura 8b. Cada *subcluster* gerado contém exatamente o mesmo número de amostras.

A similaridade entre os histogramas por região é avaliada por meio do teste chi-quadrado (χ^2 test) [Pearson, 1900]. Esse teste estatístico apresenta a vantagem de ter o custo computacional $O(\eta)$ (com η sendo o número de amostras da janela de treinamento) e de ser independente da distribuição original dos dados e da dimensionalidade do espaço amostral [Boracchi et al., 2017]. No método proposto, duas janelas são definidas: w_1 e w_2 . A janela w_1 é fixa e serve como referência, construída a partir dos particionamentos balanceados definidos durante o treinamento. Por outro lado, a janela w_2 é deslizante e atualizada continuamente conforme novas amostras chegam ao sistema. A cada nova observação, a amostra mais antiga em w_2 é descartada, mantendo o tamanho da janela constante. Em seguida, o teste χ^2 é aplicado para comparar os histogramas de w_1 e w_2 .

Sempre que um novo estado emerge, os pontos tendem a migrar para regiões diferentes no espaço de entrada, o que resulta em alterações nos histogramas. Isso geralmente leva à rejeição da hipótese nula no teste χ^2 , cuja premissa é de que não há *drift*. Essa rejeição ocorre devido ao resultado do teste chi-quadrado ser maior que o tabelado para o grau de liberdade definido pelo número de partições. Assim, a rejeição dessa hipótese indica a ocorrência de uma mudança de conceito.

O Algoritmo 1 descreve todas as etapas do procedimento NR-EIDDM. Primeiramente, executado apenas uma vez, o procedimento carrega a janela de treinamento, define os parâmetros do algoritmo e realiza o treinamento inicial (Linhas 2–8). Em seguida, executado a cada nova amostra, entra-se no laço principal, que atualiza o histograma em tempo real (janela deslizante), calcula a estatística χ^2 , testa a hipótese nula de ausência de deriva e aciona o retreinamento quando necessário (Linhas 10–20). Essencialmente, o laço principal realiza todo o processamento em tempo real. Na ausência de *drift*, ele

normaliza a amostra recebida, atualiza o histograma em tempo real e calcula a estatística χ^2 . Quando uma mudança é detectada, o procedimento aciona o retreinamento para atualizar o histograma de referência (Linhas 16–17).

O Algoritmo 2 detalha a fase de treinamento do NR-EIDDM, executada na inicialização e durante eventos de mudança: ele filtra ruídos e *outliers*, particiona o espaço de características e constrói o histograma de referência de intensidade igual (densidade uniforme). O funcionamento ocorre da seguinte forma: o DBSCAN identifica as regiões densas e remove ruídos e *outliers* (Linhas 2–3); em seguida, o K-Means com restrições divide cada região em subconjuntos de igual cardinalidade, a partir dos quais o histograma de referência é montado (Linhas 7–15). Como todos os *bins* possuem o mesmo tamanho na fase de treinamento, espera-se que esse histograma seja (quase) plano. Na prática, ele é um vetor indexado por *bin*, em que cada entrada armazena a contagem de amostras correspondente.

O Algoritmo 3 detalha a fase do processamento em tempo real, no qual é mantido um histograma de janela deslizante atualizado a cada ciclo do laço principal: inicialmente, verifica-se se a amostra recebida é ruído ou *outlier* (Linha 2). Caso seja mantida, a janela é avançada e o histograma é atualizado incrementando o *bin* da nova amostra e decrementando o *bin* da amostra removida (Linhas 3–7). À medida que o fluxo evolui, o histograma em tempo real pode desviar-se do histograma de referência plano, possibilitando a posterior comparação χ^2 e a decisão de mudança.

3.1.4 Análise de Complexidade de Tempo

O algoritmo é baseado em dois algoritmos de agrupamento sequenciais e em um teste χ^2 . A complexidade de tempo no pior caso para o DBSCAN é $O(n^2 \cdot d)$ Schubert et al. [2017], considerando distâncias euclidianas. O K-Means com restrições possui complexidade de tempo $O(nd \cdot 2^{\bar{O}(k/\epsilon)})$ Bhattacharya et al. [2018]. Além disso, inicialmente é necessário calcular o KNN para determinar o valor de ϵ . A complexidade de tempo do KNN é $O(n)$ Wu et al. [2002]. Por fim, é executado o teste χ^2 , cuja complexidade de tempo é $O(k)$. Assim, a complexidade total do algoritmo, no pior caso em que há necessidade de re-treinamento para cada amostra, é $O(n^2 \cdot d) + O(nd \cdot 2^{\bar{O}(k/\epsilon)}) + O(n) + O(k)$, onde n é o tamanho da janela, d é o número de atributos, k é o número de *bins* e ϵ é o erro do K-Means com restrições. Os três primeiros termos são responsáveis pelo tempo de execução no pior caso; contudo, eles só contribuem durante o processo de treinamento. Quando o re-treinamento não é necessário, o algoritmo apenas avalia as janelas e executa o teste χ^2 , cuja complexidade é $O(k)$.

Algoritmo 1: Algoritmo NR-EIDDM

Input:

- 1: Numero de amostras usadas na etapa de aprendizagem, η
- 2: Stream de dados, D de dimensão d
- 3: DBSCAN epsilon, ε
- 4: DBSCAN min de pontos, γ
- 5: Número de bins, n_{bins}

Output:

- 1: Posições de drifts, $drifts$

```

1 Function Main():
2    $k \leftarrow$  Número de atributos dos exemplos
3    $\gamma \leftarrow 2 * k - 1$ 
4   Calcule a distância entre cada amostra e seus  $\gamma$  vizinhos mais próximos usando  $knn$ 
5   Plote as distâncias ordenadas
6    $\varepsilon \leftarrow$  Distância no ponto de máxima curvatura
7    $X \leftarrow$  Carregue e normalize os exemplos iniciais  $\eta$  de  $D$ 
8   Histogramas construídos, centros  $\leftarrow$  Realize_Initial_Partition( $X, \varepsilon, \gamma, n_{bins}$ )
9    $X_{atual} \leftarrow$  Última amostra em  $X$ 
10  foreach  $X_{new} \in D$  do
11    Normalize  $X_{new}$ 
12    if  $Distancia(X_{new}, X_{atual}) > \omega$  then
13      Histogramas em tempo real  $\leftarrow$  Update_Window( $X_{new}, X, \varepsilon, centros$ )
14      Calcule  $\chi^2$  para os histogramas construídos e histogramas de tempo real
15      if Hipótese nula é rejeitada then
16        Histogramas construídos  $\leftarrow$  Realize_Initial_Partition( $X, \varepsilon, \gamma, n_{bins}$ )
17        Anexe( $drifts, index(X_{new})$ )
18      end
19       $X_{atual} \leftarrow X_{new}$ 
20    end
21  end
22  return  $drifts$ 
23 End Main

```

Algoritmo 2: Função Realiza Partição Inicial

Input:

- 1: Amostras da janela de aprendizado, X de dimensão d
- 2: DBSCAN epsilon, ε
- 3: DBSCAN min de pontos, γ
- 4: Número de bins, n_{bins}

Output:

- 1: Histogramas e Centros dos *Subclusters*, $histBineCenters$

```

1 Function Realize_Initial_Partition( $X, \varepsilon, \gamma, n_{bins}$ ):
2    $Clusters \leftarrow DBSCAN(X, \varepsilon, \gamma)$ 
3    $C_m \leftarrow \min\{Clusters\}$ 
4    $binSize \leftarrow |C_m| / n_{bins}$ 
5    $histBin \leftarrow \emptyset$ 
6    $Centers \leftarrow \emptyset$ 
7   foreach  $C \in Clusters$  do
8      $C_n \leftarrow |C| / binSize$ 
9      $clusterCenters, kmclusters = KMeansConstrained(C_n, binSize)$ 
10    foreach  $C_{km} \in kmclusters$  do
11       $clusterBins \leftarrow \text{append}(|C_{km}|)$ 
12    end
13     $histBin \leftarrow \text{append}(clusterBins)$ 
14     $Centers \leftarrow \text{append}(clusterCenters)$ 
15  end
16  return  $\{histBin, Centers\}$ 
17 End Realize_Initial_Partition

```

Algoritmo 3: Função Atualiza Janelas

Input:

- 1: Nova amostra obtida *online*, X_{new}
- 2: Janela de dados, X de dimensão d
- 3: DBSCAN epsilon, ϵ
- 4: Centros dos *subclusters*, $centers$

Output:

- 1: Janela de dados atualizada e Histogramas de tempo real, X e $realHist$

```

1 Function Update_Window( $X_{new}, X, \epsilon, centers$ ):
2   if euclideanDistance( $X_{new}, centers$ ) <  $\epsilon$  then
3     Inclua  $X_{new}$  na janela  $X$ 
4     Delete o último exemplo da janela  $X$ 
5     Atualize os histogramas de tempo real;
6     return { $X$ , Histogramas de tempo real}
7   end
8 End Update_Window

```

3.1.5 Discussão sobre a Avaliação de Parâmetros Externos

Os únicos parâmetros externos necessários para o método proposto são o número de partições (n_{bins}) e o tamanho da janela de treinamento. O número de partições está diretamente relacionado aos graus de liberdade do teste qui-quadrado (χ^2). Quanto maior o número de partições, maior será o valor crítico do teste para um mesmo nível de significância. Dessa forma, o número de partições pode ser utilizado como uma estratégia para controlar a sensibilidade do método e gerenciar a taxa de falsos positivos.

A Lei dos Grandes Números [Casella and Berger, 2002] reforça o princípio de que, à medida que o tamanho da janela de amostragem aumenta, a média amostral converge progressivamente para a média verdadeira da população. Em outras palavras, com um maior número de amostras, as estimativas estatísticas tornam-se mais precisas e representam melhor a distribuição real dos dados. Isso implica que, quanto maior for a janela de observação e, conseqüentemente, mais representativas forem as partições das regiões de maior concentração amostral, mais confiáveis serão os resultados. Por outro lado, janelas menores possibilitam a detecção mais rápida de mudanças. O desafio, portanto, consiste em encontrar um equilíbrio entre representatividade e capacidade de detecção precoce.

Além disso, o teste chi-quadrado de Pearson (χ^2) deve ser aplicado sobre um conjunto amostral suficientemente grande. Segundo o Teorema Central do Limite [Casella and Berger, 2002], a distribuição χ^2 é a soma de variáveis aleatórias independentes com média e variância finitas, convergindo para uma distribuição normal à medida que o número de observações O_i aumenta [Liu et al., 2020b]. Para fins práticos, Parameswaran [1979] afirma que, para $O_i > 50$ e $E_i > 5$, a distribuição dos testes estimados se aproxima suficientemente da normal, tornando aceitável desprezar pequenas discrepâncias. Portanto, é necessário garantir que o tamanho da janela seja superior a 50 e que cada partição contenha mais de 5 amostras.

Ainda assim, a metodologia depende de um processo amostral que represente

adequadamente a distribuição dos dados. Recomenda-se, portanto, utilizar janelas e números de partições pelo menos uma ordem de magnitude acima dos limiares teóricos. A atuação eficaz do DBSCAN na filtragem de pontos atípicos na etapa anterior permite um ajuste mais restritivo sobre o número de partições.

3.1.6 Considerações

O algoritmo do método proposto (NR-EIDDM) apresenta uma forma parametrizada para lidar com mudanças incrementais num espaço multidimensional de variáveis ruidosas. Esse é um cenário bastante comum em processos industriais. A escolha do número de partições adequadas, a depender do problema, seguido do teste chi-quadrado se apresentou como uma alternativa simples e robusta para a identificação de mudanças. A variação do número de partições por *cluster* permite o ajuste fino na detecção de mudanças.

3.2 Método para Detecção de Mudanças Incrementais baseado em *Denoising AutoEncoder*(DAE) e monitoração dos resíduos

Outra estratégia proposta, denominada **UDrAE** (*Unsupervised Drift detection via Denoising AutoEncoder reconstruction loss*), é uma técnica para detecção de mudanças de conceito em séries temporais de processos industriais. Sua operação é não supervisionada e está estruturada em três componentes principais: (i) um *denoising autoencoder* (DAE) adaptativo treinado para reconstruir dados do processo; (ii) um mecanismo de janela deslizante para predição da entrada; e (iii) uma estratégia de detecção baseada em monitoração da função custo. Trata-se de uma abordagem não supervisionada robusta a ruídos também desenhada para lidar com mudanças de conceito incrementais.

O nome “*Unsupervised Drift detection via Denoising AutoEncoder reconstruction loss*” expressa de forma direta o propósito e a base técnica do método. O termo “*Drift detection*” indica o objetivo central: identificar mudanças de conceito em fluxos de dados ao longo do tempo. O qualificativo “*Unsupervised*” ressalta que a detecção é realizada sem necessidade de rótulos ou supervisão externa, característica essencial em cenários industriais e aplicações reais em que anotações manuais são inviáveis. A parte “*via Denoising AutoEncoder*” especifica a técnica adotada: utiliza-se um *autoencoder* com capacidade de reconstruir os dados mesmo na presença de ruídos, de modo a aprender representações robustas e captar desvios estruturais no comportamento do processo. Por fim, a expressão “*reconstruction loss*” descreve a métrica utilizada como sinal de detecção: o erro de reconstrução, quando aumenta significativamente, indica que o modelo treinado deixou de representar adequadamente a distribuição atual dos dados,

sinalizando a ocorrência de uma mudança. Assim, o nome escolhido procura sintetizar tanto a abordagem não supervisionada quanto a inovação de empregar um erro de reconstrução de *autoencoders* de ruído como critério sensível e robusto para detectar mudanças em séries temporais.

Na maioria das vezes, uma mudança de conceito refere-se a uma mudança desconhecida e latente na relação subjacente entre as variáveis de entrada e saída. Devido à sua capacidade de aprender mapeamentos não lineares e à sua robustez ao ruído, um *Denoising AutoEncoder* se apresenta como uma opção adequada para modelar essa relação.

Essa proposta combina um *Denoising AutoEncoder* com o monitoramento da perda de reconstrução para detectar mudanças no comportamento dos dados. O método proposto adota um esquema de processamento composto por duas etapas principais: o treinamento de um *Denoising Autoencoder* (DAE) e a posterior monitoração de sua degradação de desempenho por meio de uma janela deslizante. Em cada instante de tempo, uma janela de tamanho fixo contendo as amostras mais recentes é utilizada como entrada para o mecanismo de detecção de mudanças. Durante a fase de treinamento, é possível estimar um valor de referência para a função de perda do DAE. Já em operação, a função de perda é continuamente reavaliada na janela deslizante e comparada com o valor obtido no treinamento. Diferenças estatisticamente significativas entre esses valores indicam a possibilidade de ocorrência de *concept drift*.

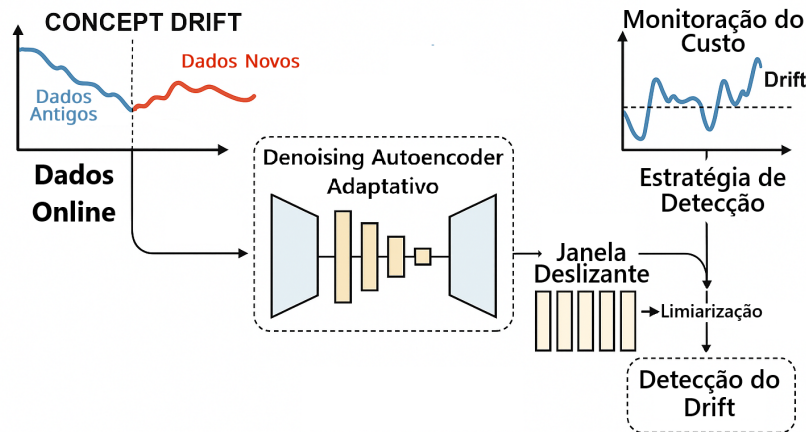


Figura 9 – Esquemático simplificado para a estratégia proposta.

3.2.1 Método Proposto - *Unsupervised Drift detection via Denoising AutoEncoder reconstruction loss (UDraE)*

Considere um processo industrial descrito pela Equação (3.8), em um cenário sem mudanças, onde \mathbf{T} representa os dados de entrada, $G(\mathbf{T})$ captura a verdadeira relação entre entrada e saída, e r denota ruído:

$$U = G(\mathbf{T}) + r \quad (3.8)$$

Quando ocorre uma mudança de conceito, um fator interno ou externo altera essa relação. O novo comportamento do processo pode então ser representado pela Equação (3.9):

$$U = G(\mathbf{T}) + F(\mathbf{T}) + r \quad (3.9)$$

Nessa formulação, $F(\mathbf{T})$ representa um fator desconhecido associado à mudança no processo que altera o mapeamento original $G(\mathbf{T})$. Um DAE, treinado em um ambiente estacionário, aprende a reconstruir a função $G(\mathbf{T})$. No entanto, diante de uma alteração na distribuição dos dados, o DAE tende a continuar reproduzindo o padrão previamente aprendido, resultando em um aumento no resíduo entre as entradas e as saídas do modelo. Esse aumento do erro de reconstrução funciona como um indicador de *concept drift*. Adicionalmente, a magnitude do erro absoluto médio (MAE, do inglês *Mean Absolute Error*) fornece uma medida quantitativa da severidade da mudança, permitindo comparar a intensidade de diferentes eventos ao longo do tempo.

3.2.2 Caracterização da Mudança

Buscando uma maior explicabilidade da mudança, o método foi pensado para que, por meio da análise dos dados, apresentasse uma boa caracterização da mudança. Desta forma, o UDrAE possibilita a caracterização da mudança por meio da identificação de:

- **Quando** ela ocorre — ao identificar o instante temporal em que o erro absoluto médio (MAE) ultrapassa o limiar de detecção;
- **Duração** em que ela ocorre — ao isolar as amostras, dentro da janela deslizante, que apresentam altos erros de reconstrução;
- **Como** ela ocorre — ao avaliar a sua severidade e sua evolução por meio da magnitude e da persistência dos valores de MAE.

3.2.3 Denoising AutoEncoder Proposto

Um *Denoising AutoEncoder* é uma extensão dos *AutoEncoders* tradicionais construído para aprender representações robustas a partir de dados corrompidos. Dados de entrada $\mathbf{x} \in \mathbb{R}^d$ são artificialmente corrompidos por um processo estocástico, produzindo uma versão ruidosa $\tilde{\mathbf{x}}$. O *encoder* mapeia $\tilde{\mathbf{x}}$ para uma representação latente $\mathbf{z} \in \mathbb{R}^k$, e o *decoder* tenta reconstruir a entrada original \mathbf{x} a partir de \mathbf{z} .

$$\mathbf{z} = f_{\theta_e}(\tilde{\mathbf{x}}) = \sigma_e(\mathbf{W}_e \tilde{\mathbf{x}} + \mathbf{b}_e) \quad (3.10)$$

$$\hat{\mathbf{x}} = f_{\theta_d}(\mathbf{z}) = \sigma_d(\mathbf{W}_d \mathbf{z} + \mathbf{b}_d) \quad (3.11)$$

onde $\sigma_e(\cdot)$ e $\sigma_d(\cdot)$ são funções de ativação (e.g., *tanh* ou *ReLU*), \mathbf{W}_e e \mathbf{W}_d são os pesos do *encoder* e *decoder*, respectivamente, e \mathbf{b}_e , \mathbf{b}_d são os vetores de bias.

A função custo $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}})$ pode combinar o erro de reconstrução com termos de regularização L1 aplicados aos pesos e às ativações da camada latente:

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{d} \sum_{i=1}^d (x_i - \hat{x}_i)^2 + \lambda_w (\|\mathbf{W}_e\|_1 + \|\mathbf{W}_d\|_1) + \lambda_a \|\mathbf{z}\|_1 \quad (3.12)$$

onde:

- $\|\cdot\|_1$ denota a norma L1 (soma dos valores absolutos),
- λ_w é o hiperparâmetro de regularização dos pesos,
- λ_a é o hiperparâmetro de regularização das ativações.

A regularização L1 nos pesos promove esparsidade nas conexões sinápticas, enquanto a regularização nas ativações força o modelo a ativar seletivamente apenas alguns neurônios na representação latente, favorecendo uma codificação mais interpretável e robusta.

Um dos principais desafios no projeto de um *AutoEncoder* está na definição adequada da dimensionalidade do espaço latente \mathbf{Z} . Essa dimensão deve ser suficientemente reduzida para evitar sobreajuste, mas ainda assim elástica o bastante para capturar a estrutura intrínseca dos dados. Em geral, espera-se que o espaço latente tenha dimensionalidade inferior à do espaço de entrada, exceto em casos específicos.

Com o objetivo de atender a esses requisitos, [Ng et al. \[2011\]](#) recomendam o uso de regularização para promover esparsidade tanto nos pesos quanto nas ativações dos neurônios, além da definição de uma arquitetura compacta. Complementarmente, [Ashfahani et al. \[2020\]](#) propuseram uma abordagem adaptativa baseada na análise de viés e variância da camada latente, permitindo o ajuste dinâmico do número de neurônios. De forma simplificada, a estratégia considera que um viés elevado indica capacidade insuficiente do modelo, enquanto uma variância alta sugere sobreajuste.

Neste trabalho, propõe-se um *Denoising AutoEncoder* que combina as ideias centrais dessas duas abordagens. É aplicado o método de Análise de Componentes Principais (PCA) [Pearson \[1901\]](#) sobre os dados de treinamento, com o objetivo de identificar o número mínimo de componentes necessárias para representar 99% da variância. Esse número define a quantidade de neurônios na camada latente. Buscando atender a adaptação dinâmica

e ao mesmo tempo controlando viés e variância, aplica-se regularização L1 durante o treinamento para favorecer representações mais esparsas.

O treinamento é conduzido com o uso de parada antecipada [Montavon et al., 2012], a fim de evitar o sobreajuste. Ao final do processo, espera-se obter um *Denoising AutoEncoder* que incorpore as características desejadas, conforme descritas por Ng et al. [2011]. A rede proposta possui 3 camadas sendo: (i) uma camada inicial com a dimensão dos dados; (ii) uma camada oculta com a dimensão determinada pela análise do PCA e (iii) uma camada de saída que apresenta dimensionalidade idêntica à dos dados de entrada. A Figura 10 apresenta a arquitetura da rede proposta. A função de ativação utilizada em todos os neurônios é a tangente hiperbólica. A arquitetura foi treinada com otimizador Adam. A Figura 11 ilustra um exemplo de resultado obtido com essa abordagem em um cenário de duas dimensões com presença de mudança no conceito.

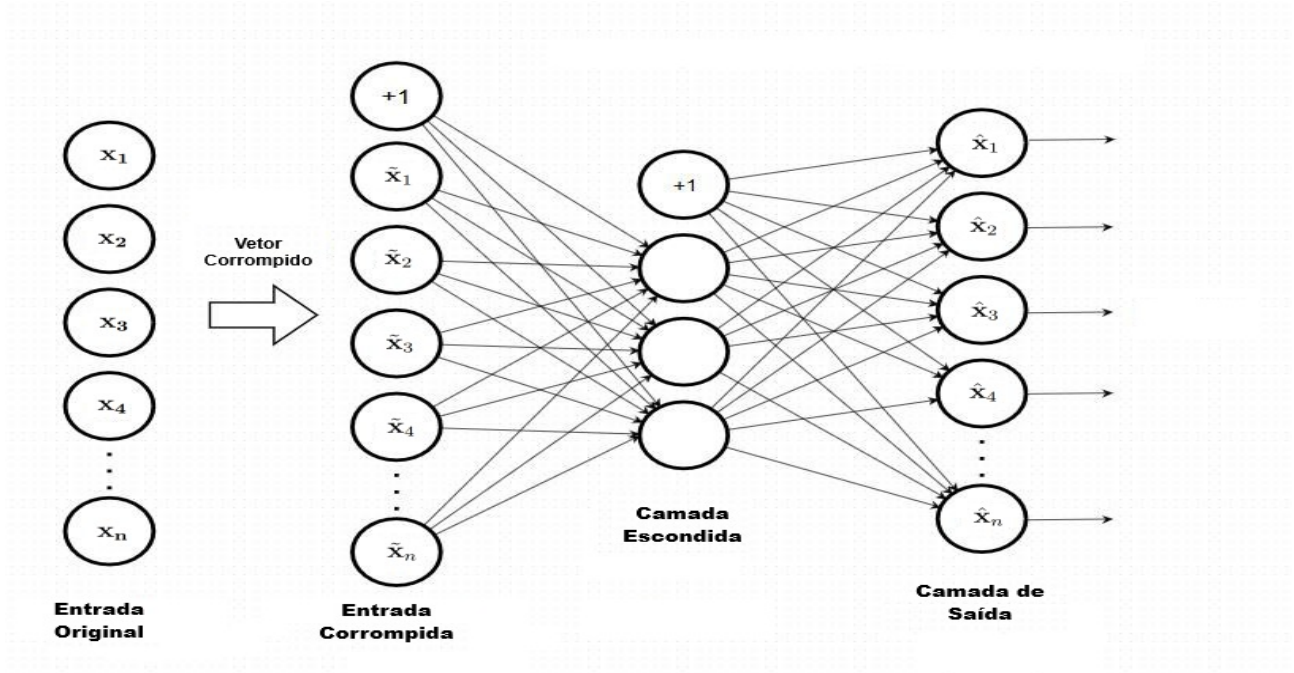


Figura 10 – Arquitetura do *AutoEncoder* proposto

3.2.4 Limiarização

O limiar de detecção de mudanças é estimado imediatamente após o treinamento, utilizando uma janela pós-treinamento cujo tamanho é igual ao da janela de treinamento. Esse limiar é calculado com base nos valores de erro absoluto médio (MAE) observados no pós-treinamento. Especificamente, o limiar de decisão é definido como:

$$\theta = \mu_{\text{pos}} + 7\sigma_{\text{pos}} \quad (3.13)$$

onde μ_{pos} e σ_{pos} representam, respectivamente, a média e o desvio padrão dos valores de erro absoluto médio (MAE) observados na janela pós-treinamento. Esse limiar fornece um

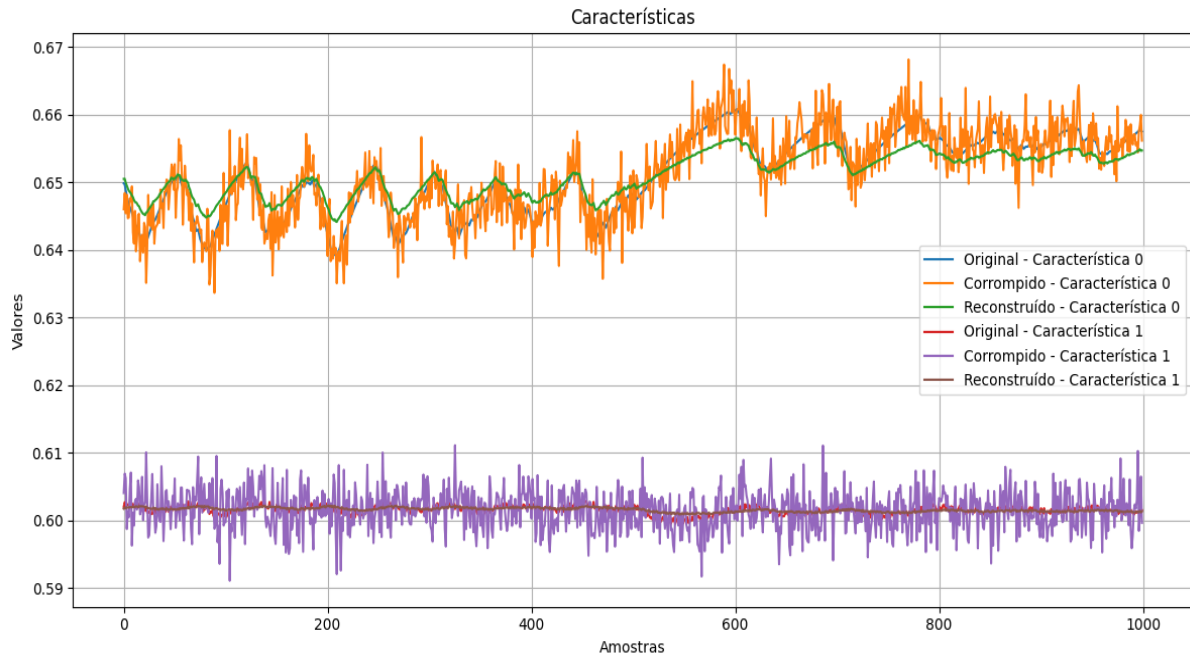


Figura 11 – Exemplo do resultado apresentado pelo *AutoEncoder* implementado

critério robusto e adaptativo, que leva em consideração a variabilidade natural no erro de reconstrução, ao mesmo tempo em que se mantém sensível a mudanças significativas na distribuição dos dados.

Buscando uma implementação que seja mais genérica possível sem se preocupar com a normalidade dos dados, a escolha de j desvios padrões foi de acordo com a desigualdade de Chebyshev. Formalmente, a desigualdade de Chebyshev afirma que, para qualquer distribuição com média μ e desvio padrão σ :

$$P(|X - \mu| \geq j\sigma) \leq \frac{1}{j^2} \quad (3.14)$$

onde P denota probabilidade, μ denota a média e σ o desvio padrão. Aplicando essa desigualdade com $j = 7$, temos:

$$P(|X - \mu| \geq 7\sigma) \leq \frac{1}{49} \approx 0,02 \quad (3.15)$$

Dessa forma, a desigualdade de Chebyshev estabelece um limite superior para a probabilidade de que um dado se desvie da média em mais de j desvios padrão, independentemente do tipo distribuição original, desde que esta possua variância finita. Ou seja, ao utilizar um limiar definido como $\mu + 7\sigma$, garante-se que, sob qualquer distribuição (não necessariamente normal), no máximo $\approx 2\%$ dos dados estarão acima desse limiar.

3.2.5 Algoritmo

O Algoritmo 4 apresenta a implementação das ideias propostas. O Algoritmo 5 detalha a construção e o treinamento do DAE, enquanto o Algoritmo 6 descreve a definição do limiar da métrica de monitoramento.

Algoritmo 4: Algoritmo UDrAE

Input:

- 1: Tamanho da janela de dados X , η
- 2: Fluxo de Dados, $\mathcal{D} \in \mathbb{R}^{n \times d}$

Output:

- 1: Posições de Drift, Drifts

```

1 Function Main():
2    $X \leftarrow \eta$  amostras de  $\mathcal{D}$ 
3   Normalize  $X$ 
4    $\text{DAE} \leftarrow \text{TreinaDAE}(X, 0.03)$ 
5    $\text{MAE\_Limiar} \leftarrow \text{ObtemParametros}(\mathcal{D}, X, \eta, \text{DAE})$ 
6    $\text{MAE\_Intensidades} \leftarrow []$ 
7    $\text{Drifts} \leftarrow []$ 
8   foreach  $x_{new} \in \mathcal{D}$  do
9     Normalize  $x_{new}$ 
10    Descarte o elemento mais antigo de  $X$ 
11    Acrescente  $x_{new}$  a  $X$ 
12     $\hat{X} \leftarrow \text{DAE.predict}(X)$ 
13     $\text{MAE} \leftarrow \frac{1}{\eta} \sum_{i=1}^{\eta} |x_i - \hat{x}_i|$ 
14    if  $(\text{MAE}/\text{MAE\_Limiar}) > 1$  then
15       $\text{onDrift} \leftarrow \text{Verdadeiro}$ 
16      Acrescente a posição da amostra em  $\text{Drifts}$ 
17      Acrescente  $\text{MAE}$  em  $\text{MAE\_Intensidades}$ 
18      Continue
19    end
20    if  $\text{onDrift}$  then
21      # A região de mudança terminou e demanda novo treinamento
22
23       $\text{DAE} \leftarrow \text{TreinaDAE}(X, N_f)$ 
24       $\text{MAE\_Limiar} \leftarrow \text{ObtemParametros}(\mathcal{D}, X, \eta, \text{DAE})$ 
25       $\text{onDrift} \leftarrow \text{Falso}$ 
26
27    end
28  end
29  return  $\text{drifts}$ 
30 End Main

```

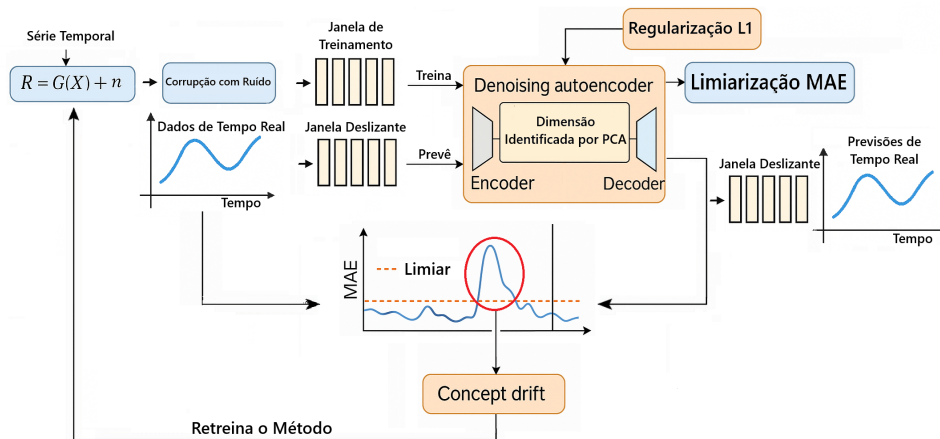


Figura 12 – Esquemático simplificado para arquitetura proposta

3.2.6 Discussão sobre a Avaliação de Parâmetro Externo

O cenário ideal para métodos não supervisionados envolve a minimização do número de parâmetros ajustáveis. No caso perfeito, não seriam necessários parâmetros definidos pelo usuário. No entanto, isso raramente é alcançável na prática.

O método proposto requer um parâmetro externo: o tamanho da janela deslizante, denotado por η . Esse parâmetro influencia diretamente tanto o treinamento do AutoEncoder quanto os valores típicos da janela de MAE (Erro Absoluto Médio). Uma janela deslizante maior geralmente melhora a acurácia das previsões da rede neural. No entanto, também pode aumentar a latência de detecção, pois isso causaria uma maior variância na janela de MAE.

O principal desafio está em encontrar um equilíbrio: selecionar um tamanho de janela que seja suficientemente representativo para apoiar previsões precisas, ao mesmo tempo em que permita a detecção oportuna de mudanças. Em aplicações práticas, uma possível estratégia para ajustar esse parâmetro é induzir intencionalmente uma mudança no sistema, quando viável, e determinar empiricamente o tamanho de janela ideal para uma detecção rápida.

3.2.7 Análise de Complexidade

O método consiste na criação e treinamento de um *Denoising AutoEncoder* seguido por uma etapa de definição de limiar. Para a criação e treinamento, o modelo considerado é um *AutoEncoder* de três camadas totalmente conectadas: uma camada de entrada de dimensão (d), uma camada escondida de dimensão (r) (definida a partir da redução de dimensionalidade via PCA) e uma camada de saída também de dimensão (d). O treinamento é realizado em *batches* de tamanho (B) até completar uma época com (η) amostras.

A etapa de *forward* envolve duas multiplicações matriz-vetor: entrada (d) \times camada escondida (r) seguida por camada escondida (r) \times saída (d). Somando as operações, o custo por amostra é da ordem: $O(dr)$. Para um *batch* de tamanho (B), tem-se $O(Bdr)$, e para uma época completa, $O(\eta dr)$. Já o *backpropagation* envolve a propagação dos gradientes e o cálculo dos gradientes dos pesos. Cada uma dessas operações tem custo assintótico $O(Bdr)$. Como há múltiplas operações desse tipo, o custo por *batch* é proporcional a uma constante vezes $O(Bdr)$. Portanto, em ordem de grandeza, o custo também é $O(Bdr)$. Em uma época completa, o custo total é $O(\eta dr)$, ou seja, a mesma ordem da inferência, embora com uma constante multiplicativa maior.

O cálculo do MAE é definido de acordo com a equação 3.16. Trata-se portanto de uma operação $O(\eta)$, onde η é o número de amostras usadas no cálculo.

$$\text{MAE} = \frac{1}{\eta} \sum_{i=1}^{\eta} |y_i - \hat{y}_i| \quad (3.16)$$

Desta forma, o custo computacional total, no pior dos casos, supondo um retreinamento da rede para cada amostra (cenário bastante improvável) é $O(\eta dr) + O(\eta) = O(\eta(dr + 1)) = O(\eta dr)$.

3.2.8 Considerações

O método *UDrAE* propõe uma abordagem não supervisionada e adaptativa para detecção de mudanças de conceito em séries temporais industriais sujeitas a ruído, combinando *AutoEncoders* com uma estratégia baseada em janelas deslizantes. A utilização de um *Denoising AutoEncoder* permite modelar relações não lineares e robustas entre as variáveis de entrada, mesmo em ambientes com dados corrompidos. A atualização contínua da janela e a reavaliação do modelo após a detecção de mudanças garantem uma resposta a novos regimes de dados. A reconfiguração do modelo a partir da nova janela após a detecção de uma mudança garante a adaptação do modelo ao novo padrão de dados. Dessa forma, o *UDrAE* se mostra uma solução promissora e eficiente para monitoramento contínuo de processos dinâmicos em ambientes industriais e aplicações em tempo real.

3.3 Síntese dos métodos propostos

Com o intuito de consolidar as principais características dos detectores desenvolvidos neste trabalho, a Tabela 1 apresenta uma comparação direta entre o *Noise-Resistant Equal-Intensities Drift Detection Method* (NR-EIDDM) e o *Unsupervised Drift detection via Denoising AutoEncoder reconstruction loss* (UDrAE). Ambos foram concebidos para lidar com fluxos de dados industriais não rotulados com mudanças incrementais sujeitos a pontos atípicos, porém cada um adota princípios distintos de modelagem, o primeiro baseado em reconstrução neural e o segundo em análise estatística de densidades. Essa complementaridade permite cobrir diferentes naturezas da mudança, buscando equilibrar robustez, interpretabilidade e custo computacional.

Em conjunto, os dois métodos oferecem uma estrutura abrangente para monitoramento de mudanças em processos industriais, atuando de forma complementar: enquanto o UDrAE aprende representações contínuas de regimes operacionais, o NR-EIDDM quantifica variações estatísticas discretas entre janelas sucessivas de dados.

Dessa forma, os métodos propostos abordam o problema da detecção não supervisionada de mudanças de conceito em fluxos ruidosos. O capítulo seguinte apresenta os resultados experimentais obtidos com ambas as metodologias, avaliando seu desempenho

Tabela 1 – Síntese comparativa das principais características dos métodos propostos

Aspecto	UDrAE	NR-EIDDM
Abordagem	Autoencoder de remoção de ruído com limiar estatístico (Chebyshev)	Análise de densidades iguais via DBSCAN + <i>Constrained K-Means</i> e teste χ^2
Tipo de dado	Fluxos contínuos	Fluxos discretizados ou segmentados
Supervisão	Não supervisionado	Não supervisionado
Robustez ao ruído	Alta, pela injeção controlada de ruído gaussiano	Alta, pela filtragem do DBSCAN
Interpretação	Global, baseada no erro médio de reconstrução	Local, baseada em regiões ou <i>bins</i> alterados
Atualização após detecção	Recalibração do limiar e re-treino do <i>autoencoder</i>	Redefinição das partições de referência e histogramas
Custo computacional	Moderada	Baixa
Entregável	<i>Timestamp</i> , região e intensidade do <i>drift</i>	<i>Timestamp</i> e região do <i>drift</i>
Vantagem principal	Sensibilidade a mudanças incrementais sob ruído; maior interpretabilidade	Média interpretabilidade e custo computacional compatível com aplicações industriais
Limitação principal	Custo computacional elevado quando o PCA é ineficiente	Dificuldade com dados esparsos, não estruturados e desbalanceados

em bases sintéticas e industriais, bem como sua comparação com métodos de referência da literatura.

Capítulo 4

Experimentos e Resultados

Este capítulo descreve a metodologia experimental adotada, bem como os principais resultados obtidos. Os experimentos foram conduzidos com conjuntos de dados artificiais e reais. No caso das bases artificiais, buscou-se configurar um cenário desafiador e representativo das condições observadas em sistemas reais. Para os dados reais, como a localização exata das mudanças conceituais é desconhecida, foi adotada uma estratégia indireta [Yu et al., 2019, Coelho et al., 2023] para identificação de mudanças baseada na aplicação de um regressor ou classificador, conforme a natureza do problema.

Entre os conjuntos artificiais utilizados, destaca-se a implementação de um modelo representativo do sistema de tanques interativos do Laboratório de Validação de Sistemas (LVAS), o que possibilitou a geração de dados sintéticos baseados em um processo físico real. Adicionalmente, foram construídos conjuntos de dados gaussianos com diferentes níveis de ruído e *drifts* incrementais em distintas velocidades. Entre os conjuntos reais, ressalta-se o uso de medições provenientes de poços de petróleo da Petrobras, bem como de dados empregados na inferência de propriedades do Diesel na Refinaria Gabriel Passos (Petrobras). Detalhes complementares sobre os experimentos e os demais conjuntos de dados são apresentados nas seções subsequentes.

4.1 Desafios e Observações na Avaliação com Bases Reais e Artificiais

A avaliação de métodos de detecção de *concept drift* apresenta desafios distintos dependendo dos dados utilizados. Em bases de dados **artificiais**, nas quais as posições exatas dos *drifts* são conhecidas previamente por construção, é possível realizar uma avaliação rigorosa e direta por meio de métricas como precisão, revocação e *F1-score*. Essa abordagem permite quantificar com clareza os acertos (verdadeiros positivos), as detecções equivocadas (falsos positivos) e as omissões (falsos negativos), resultando em uma análise mais objetiva e controlada da eficácia do método de detecção.

No entanto, em **bases de dados reais**, a situação é significativamente mais complexa. As mudanças de conceito, quando ocorrem, não possuem marcações explícitas e podem ser sutis, graduais ou até mesmo recorrentes. Isso dificulta ou inviabiliza a identificação precisa de falsos positivos e falsos negativos. Consequentemente, a avaliação direta por métricas tradicionais de classificação binária torna-se impraticável. Como alternativa, pode-se utilizar métricas indiretas, como a **acurácia global do sistema de aprendizado**, antes e após a detecção de *drift*, para inferir a qualidade do detector.

Esse tipo de avaliação, embora prática, pode induzir conclusões equivocadas. Métodos de detecção que produzem grande número de falsos positivos, ou seja, que detectam mudanças inexistentes, tendem a acionar frequentemente o retreinamento do modelo de predição. Isso pode resultar em uma melhoria artificial da acurácia preditiva, mascarando o desempenho deficiente do detector de *drift*. Em outras palavras, um método pode parecer eficaz sob a ótica da acurácia global, mesmo sendo um mau detector, pois aciona retreinamentos desnecessários com frequência, mantendo o modelo atualizado mais por sorte do que por mérito real de detecção [Yu et al., 2019].

Neste contexto, visando isolar a eficiência dos métodos de detecção de *drift*, uma estratégia adequada consiste em configurar, em um cenário comparativo, todos os detectores para identificarem o mesmo número de mudanças em uma dada base de dados. Dessa forma, utilizando o mesmo classificador ou regressor com configurações idênticas, a acurácia global tende a refletir predominantemente o desempenho do detector de *drifts*, minimizando a influência de outras variáveis envolvidas.

Portanto, a **análise crítica da métrica utilizada** é essencial para a avaliação justa de detectores de *drift*. Em ambientes reais, a ausência de *ground truth* explícito para os pontos de mudança exige cuidado adicional na interpretação dos resultados.

4.2 Sistema de Tanques Interativos - STI

O Sistema de Tanques Interativos do Laboratório de Validação de Sistemas - LVAS foi utilizado em vários trabalhos do PPGEE da UFMG e está descrito em praticamente todos eles [Caminhas et al., 1997, Braga, 1994, Jota et al., 1995, Pereira, 1996]. Basicamente é constituído por três tanques interconectados (TQ-1, TQ-2 e TQ-3), duas bombas (BA-1 e BA-2), um conjunto de instrumentos e válvulas (FCV-1, FCV-2, FCV-3, FCV-4, FCV-5). O tanque TQ-1 se encontra num nível abaixo dos demais. TQ-2 e TQ-3 estão no mesmo nível. O controle é realizado por meio de um PLC e um microcomputador executando um software de supervisão. Dados do processo podem ser coletados por meio dos instrumentos da planta. A vazão de entrada no tanque TQ-2 (q_i), a vazão de saída do tanque TQ-3 (q_o) e os níveis (h_1 , h_2 e h_3) podem ser medidos por meio da instrumentação. A Figura 13 apresenta um esquemático do sistema.

Na modelagem proposta para identificação de *drifts*, o objetivo foi controlar o nível do tanque TQ-3 (h_3) em conjunto com a vazão de saída (q_0) e causar desvios lentos por meio de vazamentos, entupimentos e falhas. Desta forma, tornou-se possível obter as bases simuladas para avaliação do método proposto. O ciclo de operação do STI consiste no bombeamento da água do tanque TQ-1 para o tanque TQ-2 por meio da bomba BA-1. A água flui para o tanque TQ-3 através do princípio dos vasos comunicantes. A intensidade deste acoplamento é controlada por meio das válvulas FV-3 e FV-4 (a válvula FV-5 fica permanentemente fechada durante o funcionamento normal). O ciclo se completa com o bombeamento da água do tanque TQ-3 de volta para o TQ-1 por meio da bomba BA-2. O controle do nível no tanque TQ-3 é feito por meio da atuação na válvula FCV-1. FCV-1 controla a vazão de entrada de TQ-2 que, conseqüentemente controla a vazão de entrada em TQ-3. Por fim, a vazão de saída de TQ-3 é controlada por meio da atuação na válvula FCV-2 [Caminhas et al., 1997].

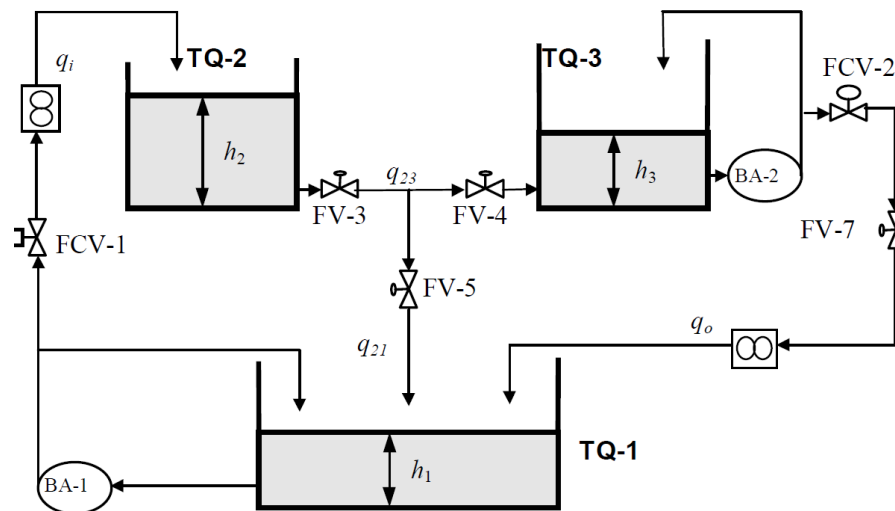


Figura 13 – Sistema de Tanques Interativos. Fonte: Adaptado de Caminhas et al. [1997].

4.2.1 Modelagem conceitual

O STI também foi modelado matematicamente nos trabalhos citados. A modelagem foi realizada por meio do balanço de massa entre os tanques TQ-2 e TQ-3 utilizando-se a Lei de Bernoulli [Braga, 1994]. Tomando $x_1 = h_2$, $x_2 = h_3$, $y_1 = h_3$, $y_2 = q_0$ e $y_3 = q_i$ em que h_2 é o nível do tanque TQ-2, h_3 é o nível do tanque TQ-3, q_0 é a vazão de saída de TQ-3 e q_i é a vazão de entrada no tanque TQ-2. Tem-se modelo conceitual do sistema

dado por:

$$\begin{cases} \dot{x}_1 = \frac{1}{A} [q_i(R_1, u_1, p_1) - q_{23}(x_1, x_2)] \\ \dot{x}_2 = \frac{1}{A} [q_{23}(x_1, x_2) - q_o(R_2, u_2, p_2)] \end{cases}$$

$$\begin{cases} y_1 = x_2 \\ y_2 = q_o(R_2, u_2, p_2) \\ y_3 = q_i(R_1, u_1, p_1) \end{cases} \quad (4.1)$$

onde,

$$\begin{cases} q_i(R_1, u_1, p_1) = Q_{imax} \cdot q_f(R_1, u_1, p_1) \\ q_o(R_2, u_2, p_2) = Q_{omax} \cdot q_f(R_2, u_2, p_2) \\ q_{23} = \frac{\sqrt{x_1 - x_2}}{R_h} \end{cases}$$

De acordo com [Pereira \[1996\]](#), q_f é definido como:

$$q_f(R_i, u_i, p_i) = \frac{1}{\sqrt{1 + \left(\frac{1}{R^{2(u_i-1)} - 1} \right) \cdot p_i}} \quad (4.2)$$

sendo:

$q_o(R_2, u_2, p_2)$: Vazão de saída do tanque TQ-3 em m^3/s ;

$q_i(R_1, u_1, p_1)$: Vazão de entrada do tanque TQ-2 em m^3/s ;

$q_{23}(x_1, x_2)$: Vazão entre os tanques TQ2 e TQ-3 em m^3/s ;

$q_f(R_i, u_i, p_i)$: Vazão fracionária da válvula "i";

Q_{imax} : Vazão máxima de entrada do tanque TQ-2 em m^3/s ;

Q_{omax} : Vazão máxima de saída do tanque TQ-3 em m^3/s ;

A : Área da base dos tanques TQ-2 e TQ-3 em m^2 ;

R_1 : Representa o alcance da válvula FCV-1;

R_2 : Representa o alcance da válvula FCV-2;

p_1 : Razão entre a queda de pressão mínima e máxima da válvula FCV-1;

p_2 : Razão entre a queda de pressão mínima e máxima da válvula FCV-2;

R_h : Resistência hidráulica do duto entre os tanques;

u_1 : Sinal de controle fracionário da válvula FCV-1;

u_2 : Sinal de controle fracionário da válvula FCV-2;

4.2.2 Modelagem com falhas e considerações práticas

[Caminhas et al. \[1997\]](#) modelaram um conjunto falhas para o STI. Foram inseridos modelos de falhas para a instrumentação, agarramento de válvulas, entupimentos nos dutos e desligamento de bombas. Também foi inserida a natureza ruidosa das medições.

Com essa modelagem mais completa, tornou-se possível uma simulação realista da planta trazendo a liberdade de simulação com bastante precisão. Desta forma, o modelo geral para o Sistema de Tanques Interativos é descrito pelo conjunto de equações a seguir:

$$\begin{aligned}
& \begin{cases} \dot{x}_1 = \frac{1}{A} [q_i^f(R_1, u_1, p_1) - q_{23}^f(x_1, x_2) - \frac{\sqrt{x_1}}{R_{hv21}} \cdot k_{fv23}] \\ \dot{x}_2 = \frac{1}{A} [q_{23}^f(x_1, x_2) - q_o^f(R_2, u_2, p_2, x_2) - \frac{\sqrt{x_2}}{R_{hv31}} \cdot k_{fv23}] \end{cases} \\
& \begin{cases} y_1 = k_{fy1} \cdot x_2 + r_{y1} \\ y_2 = k_{fy2} \cdot q_o^f(R_2, u_2, p_2, x_2) + r_{y2} \\ y_3 = k_{fy3} \cdot q_i^f(R_1, u_1, p_1) + r_{y3} \end{cases} \\
\text{onde,} & \begin{cases} q_i^f(R_1, u_1, p_1) = Q_{imax} \cdot \bar{k}_{fe12} \cdot \bar{k}_{fba1} \cdot [\bar{k}_{fcv1} \cdot q_f(R_1, u_1, p_1) + k_{fcv1} \cdot q_{a1}] \\ q_o^f(R_2, u_2, p_2, x_2) = Q_{omax} \cdot \bar{k}_{fe31} \cdot \left[\bar{k}_{fba2} \cdot (\bar{k}_{fcv2} \cdot q_f(R_2, u_2, p_2) + k_{fcv2} \cdot q_{a2}) - k_{fba2} \cdot K_{b2} \cdot \frac{\sqrt{x_2 - h_{ba2}}}{Q_{omax} \cdot R_{hb31}} \cdot R_2^{u_2 - 1} \right] \\ q_{23}^f = \bar{k}_{fe23} \cdot \frac{\sqrt{x_1 - x_2}}{R_h} \\ q_f(R_i, u_i, p_i) = \frac{1}{\sqrt{1 + \left(\frac{1}{R^2(u_i - 1) - 1} \right) \cdot p_i}} \end{cases} \quad (4.3)
\end{aligned}$$

Foram inseridos sinais aleatórios de distribuição uniforme e média nula para simular o cenário ruidoso de um ambiente real. Essas variáveis são r_{y1} , r_{y2} e r_{y3} cujas amplitudes máximas são $a_1 = 0,075\%x_{2max}$, $a_2 = 0,25\%Q_{omax}$ e $a_3 = 0,5\%Q_{imax}$ respectivamente. Também foram inseridas algumas condições de intertravamento da seguinte forma:

$$k_{fba1} = \begin{cases} 1 & \text{se } x_1 > x_{1max} \text{ ou se ocorrer falha na bomba BA-1} \\ 0 & \text{se } x_1 \leq x_{1max} \text{ e não ocorrer falha na bomba BA-1} \end{cases} \quad (4.4)$$

$$k_{fba2} = \begin{cases} 1 & \text{se } x_2 \leq 0.04 \cdot x_{2max} \text{ ou se ocorrer falha na bomba BA-2} \\ 0 & \text{se } x_2 > 0.04 \cdot x_{2max} \text{ e não ocorrer falha na bomba BA-2} \end{cases} \quad (4.5)$$

$$K_{b2} = \begin{cases} 0 & \text{se } x_2 \leq h_{ba2} \\ 1 & \text{se } x_2 > h_{ba2} \end{cases} \quad (4.6)$$

A Tabela 2 apresenta as falhas modeladas, seus respectivos coeficientes e os possíveis valores para cada. A constante h_{ba2} representa a altura do ponto de tomada de água da bomba BA-2.

4.3 Bases Usadas na Avaliação

Diversos métodos reportados na literatura apresentam bom desempenho na detecção de mudanças abruptas, por essa razão, o foco principal deste trabalho concentrou-se na seleção de bases adequadas para a identificação de *drifts* incrementais.

Tabela 2 – Tipos de falhas no Sistema de Tanques Interativos.

Tipo de Falha	Parâmetro Indicador da Falha	Valores
Agarramento da válvula CV-1	k_{fcv1}	{0,1}
Agarramento da válvula CV-2	k_{fcv2}	{0,1}
Desligamento da bomba BA-1	k_{fba1}	{0,1}
Desligamento da bomba BA-2	k_{fba2}	{0,1}
Entupimento no duto entre TQ-1 e TQ-2	k_{fe12}	[0,1]
Entupimento no duto entre TQ-3 e TQ-1	k_{fe31}	[0,1]
Entupimento no duto entre TQ-2 e TQ-3	k_{fe23}	[0,1]
Vazamento no duto entre TQ-2 e TQ-3	k_{fv23}	[0,1]
Falha no sensor de nível de TQ-3	k_{fy1}	{0,1}
Falha no sensor de vazão de saída de TQ-3	k_{fy2}	{0,1}
Falha no sensor de vazão de entrada de TQ-2	k_{fy3}	{0,1}

Nas bases sintéticas, foram gerados dados com distribuição gaussiana controlada, de modo a permitir a caracterização incremental das mudanças. As bases simuladas foram obtidas a partir da simulação do sistema STI, permitindo reproduzir condições dinâmicas realistas, porém com total controle dos parâmetros de variação. Já as bases reais incluem tanto conjuntos de dados disponíveis no repositório UCI quanto bases industriais provenientes de plantas reais da Petrobras, possibilitando a avaliação do método em condições industriais autênticas.

4.3.1 Bases Artificiais

Para todas as bases sintéticas gaussianas, foram gerados 40000 exemplos por *dataset* com *drifts* incrementais entre as amostras 9500 e 10500, 19500 e 20500, 29500 e 30500. Para as bases simuladas oriundas do STI simulado, também foram gerados 40000 exemplos *dataset*, porém os *drifts* ficaram entre as amostras 9500 e 12500, 19500 e 22500, 29500 e 32500.

4.3.1.1 Gaussianas

Foram gerados conjuntos de dados com distribuição gaussiana. Cada conjunto contém 15 atributos. Inicialmente, é definido um estado estacionário composto por exemplos com média aleatória no intervalo $[0, 20]$ e desvio padrão unitário. Foram produzidas bases combinando três velocidades de mudança, três níveis de ruído e dois percentuais de ruído para cada combinação. As velocidades de mudança correspondem a incrementos positivos ou negativos na média dos exemplos durante a região de *drift*.

Os níveis de ruído indicam o grau de amplificação do desvio padrão aplicado às amostras ruidosas, enquanto os percentuais de ruído representam a proporção de exemplos

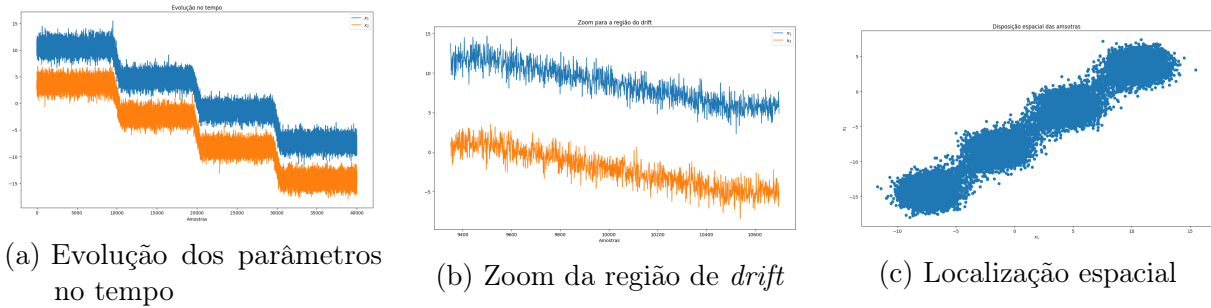


Figura 14 – Exemplo de base gaussiana gerada

corrompidos por ruído proveniente de uma distribuição gaussiana adicional, cuja média aleatória varia no intervalo $[0, 20]$ e cuja variância assume valores entre $[1, 3]$. Além disso, foi gerada uma base puramente gaussiana, sem amplificação ou injeção adicional de ruído, utilizada como referência de comportamento ideal.

A movimentação lenta (*drift* incremental) foi induzida a partir do deslocamento incremental dos centros das distribuições gaussianas. A Tabela 3 apresenta a combinação dos parâmetros empregados na geração das bases, enquanto a Figura 14 ilustra um exemplo bidimensional, no qual se observam quatro regiões estacionárias e três transições conceituais. A figura também inclui um detalhamento ampliado da região de mudança, evidenciando o processo de transição entre os conceitos.

Tabela 3 – Detalhamento dos conjuntos de dados obtidos da base de Gaussianas.

Base	% de Ruído	Nível de ruído	Velocidade da Mudança
data_0_noise.csv	0	0	0.06
data_01_l1_s06_noise.csv	1	1	0.06
data_01_l1_s12_noise.csv	1	1	0.12
data_01_l15_s06_noise.csv	1	2	0.06
data_01_l15_s12_noise.csv	1	2	0.12
data_01_l20_s06_noise.csv	1	3	0.06
data_01_l20_s12_noise.csv	1	3	0.12
data_03_l1_s06_noise.csv	3	1	0.06
data_03_l1_s12_noise.csv	3	1	0.12
data_03_l15_s06_noise.csv	3	2	0.06
data_03_l15_s12_noise.csv	3	2	0.12
data_03_l20_s06_noise.csv	3	3	0.06
data_03_l20_s12_noise.csv	3	3	0.12
data_06_l1_s06_noise.csv	6	1	0.06
data_06_l1_s12_noise.csv	6	1	0.12
data_06_l15_s06_noise.csv	6	2	0.06
data_06_l15_s12_noise.csv	6	2	0.12
data_06_l20_s06_noise.csv	6	3	0.06
data_06_l20_s12_noise.csv	6	3	0.12

4.3.1.2 STI Simulado

Foi implementado o modelo que contempla falhas e considerações práticas para o STI de acordo com o apresentado por Caminhas et al. [1997]. Foram propostos cenários contendo *drifts* incrementais por meio da simulação de entupimentos e vazamentos. Os

cenários que eram mais propícios à geração de *drifts* incrementais estão relacionados com entupimentos e vazamentos.

Entretanto, como o sistema tem dinâmica lenta, algumas variáveis demoram para refletir alterações bruscas como falhas de bombas, válvulas e sensores. Normalmente essas falhas refletem rapidamente nos sinais de controle. Porém as variáveis de nível e vazão acabam sendo mais lentas, salvo para aquelas falhas de bomba e válvula que afetam diretamente determinada vazão. Por este motivo, procurou-se escolher uma combinação de variáveis que representasse um conjunto interessante para os cenários simulados. Foram escolhidas as variáveis de nível do tanque 2, nível do tanque 3, vazão de saída e sinal de controle u_2 para a monitoração. Também foram gerados três níveis diferentes de ruído para as variáveis y_1 , y_2 e y_3 . A inserção de diferentes níveis de ruído se deu por meio de fatores multiplicadores 1, 2 e 3 na obtenção de y_1 , y_2 e y_3 , como descrito na Equação 4.3. A Tabela 4 resume os detalhes de cada conjunto de dados gerados.

Tabela 4 – Detalhamento dos conjuntos de dados simulados a partir do modelo do STI.

Base	Falha	Nível de Ruído	Fator da falha
simulation_fy2_1.0_nl_1.csv	Falha de sensor de vazão de saída	1	1
simulation_fy2_1.0_nl_2.csv	Falha de sensor de vazão de saída	2	1
simulation_fy2_1.0_nl_3.csv	Falha de sensor de vazão de saída	3	1
simulation_kfba2_1.0_nl_1.csv	Desligamento da bomba BA-2	1	1
simulation_kfba2_1.0_nl_2.csv	Desligamento da bomba BA-2	2	1
simulation_kfba2_1.0_nl_3.csv	Desligamento da bomba BA-2	3	1
simulation_kfcv2_1.0_nl_1.csv	Agarramento da válvula CV-2	1	1
simulation_kfcv2_1.0_nl_2.csv	Agarramento da válvula CV-2	2	1
simulation_kfcv2_1.0_nl_3.csv	Agarramento da válvula CV-2	3	1
simulation_kfe12_0.28_nl_1.csv	Entupimento entre TQ-1 e TQ-2	1	0.28
simulation_kfe12_0.28_nl_2.csv	Entupimento entre TQ-1 e TQ-2	2	0.28
simulation_kfe12_0.28_nl_3.csv	Entupimento entre TQ-1 e TQ-2	3	0.28
simulation_kfv23_0.03_nl_1.csv	Vazamento entre TQ-2 e TQ-3	1	0.03
simulation_kfv23_0.03_nl_2.csv	Vazamento entre TQ-2 e TQ-3	2	0.03
simulation_kfv23_0.03_nl_3.csv	Vazamento entre TQ-2 e TQ-3	3	0.03
simulation_kfv23_0.05_nl_1.csv	Vazamento entre TQ-2 e TQ-3	1	0.05
simulation_kfv23_0.05_nl_2.csv	Vazamento entre TQ-2 e TQ-3	2	0.05
simulation_kfv23_0.05_nl_3.csv	Vazamento entre TQ-2 e TQ-3	3	0.05

A Figura 15 apresenta um exemplo para um conjunto de dados gerados simulando um entupimento entre os tanques TQ-1 e TQ-2 a um fator de 0.28.

4.3.2 Bases Reais

4.3.2.1 Conjunto de Dados “Gas Sensor Array”

O conjunto de dados “Gas Sensor Array” [Vergara, 2012] contém 128 variáveis medidas em 13.910 amostras provenientes de 16 sensores químicos, utilizados em simulações para compensação de *drift* em uma tarefa de discriminação entre 6 tipos de gases em diferentes níveis de concentração. O objetivo principal dos autores, ao disponibilizá-lo, foi torná-lo acessível gratuitamente à comunidade de pesquisa em inteligência artificial, visando o desenvolvimento de estratégias para lidar com mudanças de conceito.

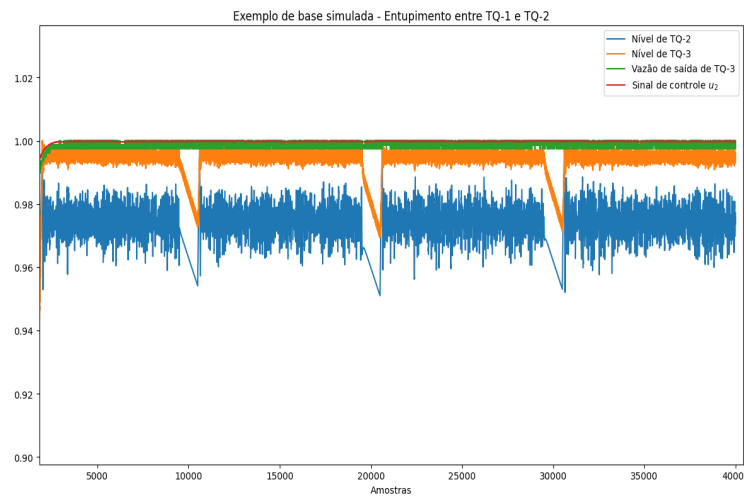


Figura 15 – Exemplo de simulação de entupimento entre TQ-1 e TQ-2 a um fator de 0.28.

Os dados foram coletados entre janeiro de 2007 e fevereiro de 2011 (36 meses), em uma plataforma de entrega de gases localizada no laboratório *ChemoSignals*, do *BioCircuits Institute*, na Universidade da Califórnia em San Diego. Todo o processo foi conduzido por um ambiente totalmente automatizado, controlado pelo software *LabVIEW National Instruments*.

O conjunto de dados compreende registros de seis gases puros distintos: Amônia, Acetaldeído, Acetona, Etileno, Etanol e Tolueno, cada um dosados em uma ampla faixa de concentrações, variando de 5 a 1000 ppmv.

Neste estudo, o conjunto de dados foi organizado em seis grupos, cada um correspondente a uma das substâncias. Como os dados apresentam um *drift* contínuo desde a primeira até a última amostra de cada grupo, foram reproduzidas aleatoriamente 1000 amostras a partir das 300 primeiras amostras originais para representar uma região sem mudanças. Esse procedimento foi aplicado tanto para as 1000 amostras iniciais quanto para o segmento entre as amostras 2000 e 3000 de cada grupo. Como resultado, foram definidas duas regiões distintas de *drift* contínuo por grupo.

Tabela 5 – Distribuição das amostras nos conjuntos de dados de sensores de gás.

Conjunto (classe)	Número de amostras
classe 1	6165
classe 2	6526
classe 3	5241
classe 4	5536
classe 5	6609
classe 6	5433

4.3.2.2 3W

O conjunto de dados 3W [Vargas et al., 2019] é uma base pública disponibilizada pela Petrobras em conjunto com pesquisadores da própria empresa e da Universidade Federal do Espírito Santo para fins de pesquisa e desenvolvimento de técnicas de aprendizagem de máquina. Cada conjunto de dados contém instâncias com oito atributos podendo ser classificadas em sete tipos de eventos indesejados. Esses dados foram obtidos a partir da monitoração de poços de petróleo da empresa. É considerada uma base de difícil análise por possuir os dados tais como obtidos da planta sem qualquer filtragem ou remoção de *outliers*. Também possui muitos valores faltantes, tanto para os rótulos quanto para as características em determinados intervalos.

O 3W é composto por três tipos de instâncias que são determinados pela sua origem: real histórica, simulada, ou desenhada à mão. Instâncias reais históricas, ou apenas reais, são aquelas que ocorreram de fato em poços produtores da Petrobras. A grande maioria destas bases de dados são para condições normais de operação ou *datasets* simulados. Para este estudo, foram escolhidos somente conjuntos de dados **reais** e que apresentaram trechos com condições **normais e anormais** de operação. A Tabela 6 apresenta os detalhes para cada conjunto de dados utilizado.

Tabela 6 – 3W - Conjuntos de dados selecionados.

Nome	Número de amostras	Classe do evento
WELL-00002_20140126161944	24017	Classe 1
WELL-00006_20180617200257	53824	Classe 1
WELL-00009_20170313160804	6738	Classe 2
WELL-00010_20171218190131	9809	Classe 2
WELL-00016_20180404222255	16561	Classe 5
WELL-00016_20180426102531	15514	Classe 5
WELL-00016_20180517184536	25040	Classe 5
WELL-00017_20140319020130	32311	Classe 5
WELL-00002_20140301151700	11478	Classe 6
WELL-00002_20140325170304	12021	Classe 6
WELL-00018_20180611021218	42974	Classe 7
WELL-00018_20190403023307	44814	Classe 7

4.3.3 T85 - Petrobras

O conjunto de dados **T85** consiste em 1805 amostras coletadas ao longo de um período de cinco anos, provenientes de análises laboratoriais da propriedade T85 do Gasóleo leve, um produto de uma unidade de destilação a vácuo no processo de refino de petróleo da refinaria Gabriel Passos - Petrobras. O T85 corresponde à temperatura na qual 85% do óleo foi evaporado, sendo uma especificação crítica na composição de combustíveis da base de diesel. Como a corrente de gasóleo é incorporada ao *pool* de diesel da refinaria, seu valor de **T85** deve ser considerado para garantir que o produto final atenda às especificações do produto. No entanto, nesta planta, não há instrumentos capazes de medir o **T85** em tempo real. Esses valores são obtidos uma ou duas vezes por dia, por meio de análises químicas em laboratório.

Por outro lado, valores de **T85** em tempo real são necessários como entrada para os controladores de processo envolvidos na produção de diesel. Conseqüentemente, modelos de inferência são comumente utilizados para estimar o **T85** com base em outras 14 variáveis de processo que apresentam forte correlação com ele. O valor de **T85** pode ser influenciado pelo tipo de petróleo bruto utilizado, pelas condições de temperatura e pressão do forno a vácuo, ou ainda por decisões operacionais em unidades de processamento anteriores.

Dessa forma, o processo é altamente suscetível a mudanças e, como o **T85** não é medido diretamente, qualquer alteração no comportamento do processo requer o reprocessamento do modelo de inferência responsável por sua estimativa. O conjunto de dados não possui rótulos de *drift*, ou seja, não há uma referência disponível para avaliação direta do método proposto. Ainda assim, como a variável-alvo é inferida, é possível treinar um modelo de regressão e avaliar seu desempenho baseado em retreinamentos.

4.4 Métricas de Avaliação

A avaliação buscou quantificar a capacidade de detecção de mudanças dos métodos propostos. Foram levados em consideração as detecções corretas (TP), detecções incorretas (FP) e não detecções incorretas (FN). A definição de verdadeiro positivo (TP), falso positivo (FP) e falso negativo (FN) ocorreu de acordo com o apresentado por [Gama et al. \[2014\]](#) e [Yu et al. \[2019\]](#).

Foi definido como TP qualquer detecção verificada enquanto a mudança está ocorrendo. Um FP ocorre quando o método sugere um *drift* em qualquer região sem mudança. Um FN ocorre quando o método deixa de detectar uma mudança. Para cada base artificial, a qualidade do detector foi medida por meio da $\text{Precisão} = \text{TP}/(\text{TP}+\text{FP})$, $f1\text{-score} = 2\text{TP}/(2\text{TP}+\text{FP}+\text{FN})$ e atrasos médio na detecção.

Especificamente para bases de dados reais que não possuem *ground truth*, como é o caso da **T85** e **3W** por exemplo, torna-se inviável calcular as métricas de detecção segundo a estratégia mencionada anteriormente. Nesse contexto, foi adotado um classificador ou regressor (dependendo da base) baseado em *Random Forest* como *baseline*, treinado sobre todo o conjunto de dados, permitindo, para o caso do regressor, a avaliação das métricas MSE (*Mean Squared Error*), MAE (*Mean Absolut Error*) e MAPE (*Mean Absolute Percentage Error*) e, para o caso do classificador, Precisão e *F1-Score*. Em seguida, para cada método proposto, foram identificadas possíveis mudanças na base e, a cada ponto de detecção, o classificador/regressor foi retreinado. Por fim, as métricas obtidas após cada retreinamento foram comparadas com as do *baseline*, permitindo avaliar o ganho de desempenho proporcionado por cada abordagem.

4.5 Experimentos

Esta seção apresenta os principais resultados obtidos nos experimentos conduzidos. Os testes foram organizados em duas etapas: (i) uma análise exploratória dos hiperparâmetros para uma base de referência buscando avaliar o efeito de cada configuração e fornecer interpretabilidade; e (ii) uma etapa de avaliação completa, voltada a utilizar essa melhor combinação de parâmetros para todos os cenários daquela categoria.

No método **NR-EIDDM**, os hiperparâmetros avaliados foram o número mínimo de partições (*bins*) e o tamanho da janela w . No método **UDrAE**, avaliou-se o tamanho da janela deslizante. Para bases com *ground truth*, utilizou-se busca em grade sobre os parâmetros; para bases reais sem rótulo, adotou-se parametrização fixa com o ajuste do número de detecções similar visando comparação justa. Os resultados são apresentados a seguir.

4.5.1 Gaussianas (NR-EIDDM)

Para as bases gaussianas sintéticas, foi realizada busca em grade com $w \in [150, 160, \dots, 300]$ e $n_{bins} \in [5, 6, \dots, 15]$ (176 combinações) observando o melhor resultado das métricas para a primeira base de referência. Tomando como referência a precisão, a melhor configuração encontrada foi $w = 180$ e $n_{bins} = 6$. Na sequência essa parametrização foi aplicada a todas as demais bases. A revocação foi 1.0 para todos os conjuntos, indicando que o método detectou todas as mudanças. Por esta razão, todas as comparações foram realizadas somente em termos de precisão e *F1-Score*. Os resultados médios (10 execuções) encontram-se na Tabela 7.

Tabela 7 – Avaliação de desempenho do método NR-EIDDM para os dados gaussianos.

Base	Precisão	<i>F1-score</i>	Atraso Médio
data_0_noise.csv	0.86 ± 0	0.93 ± 0	66.67 ± 0
data_01_l1_s06_noise.csv	0.85 ± 0	0.92 ± 0	118.33 ± 0
data_01_l1_s12_noise.csv	0.85 ± 0	0.92 ± 0	68.0 ± 0
data_01_l15_s06_noise.csv	0.87 ± 0	0.93 ± 0	95.6 ± 0
data_01_l15_s12_noise.csv	0.86 ± 0	0.93 ± 0	66.0 ± 0
data_01_l20_s06_noise.csv	0.86 ± 0	0.93 ± 0	83.33 ± 0
data_01_l20_s12_noise.csv	0.85 ± 0	0.92 ± 0	61.0 ± 0
data_03_l1_s06_noise.csv	0.87 ± 0	0.93 ± 0	89.6 ± 0
data_03_l1_s12_noise.csv	0.86 ± 0	0.92 ± 0	84.0 ± 0
data_03_l15_s06_noise.csv	0.87 ± 0	0.93 ± 0	105.33 ± 0
data_03_l15_s12_noise.csv	0.84 ± 0	0.91 ± 0	71.0 ± 0
data_03_l20_s06_noise.csv	0.86 ± 0	0.93 ± 0	127.0 ± 0
data_03_l20_s12_noise.csv	0.86 ± 0	0.92 ± 0	73.3 ± 0
data_06_l1_s06_noise.csv	0.87 ± 0	0.93 ± 0	82.33 ± 0
data_06_l1_s12_noise.csv	0.86 ± 0	0.93 ± 0	62.0 ± 0
data_06_l15_s06_noise.csv	0.87 ± 0	0.93 ± 0	86.0 ± 0
data_06_l15_s12_noise.csv	0.86 ± 0	0.93 ± 0	55.66 ± 0
data_06_l20_s06_noise.csv	0.87 ± 0	0.93 ± 0	67.33 ± 0
data_06_l20_s12_noise.csv	0.86 ± 0	0.93 ± 0	55.33 ± 0

4.5.2 Gaussianas (UDrAE)

Como no caso anterior, foi avaliado o tamanho da janela deslizante, com busca em grade sobre $w \in [50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600]$ para a primeira base. A melhor configuração foi $w = 50$, utilizada nas demais bases. Os resultados médios (10 execuções) estão na Tabela 8.

Tabela 8 – Avaliação do método UDrAE para o conjunto de bases gaussianas.

Base	Precisão	F1-score	Atraso Médio
data_0_noise.csv	0.79±0.04	0.88±0.02	39.56±4.32
data_01_l1_s06_noise	0.90±0	0.95±0	96.78±6.09
data_01_l1_s12_noise	0.88±0.03	0.90±0.02	51.44±11.44
data_01_l15_s06_noise	0.88±0.06	0.94±0.03	91.56±0.16
data_01_l15_s12_noise	0.93±0.03	0.96±0.02	59.78±5.25
data_01_l20_s06_noise	0.93±0.06	0.96±0.03	99.67±4.24
data_01_l20_s12_noise	0.95±0.04	0.98±0.02	67.78±11.77
data_03_l1_s06_noise	0.95±0.02	0.97±0.01	102.56±11.77
data_03_l1_s12_noise	1.0±0	1.0±0	59.78±3.81
data_03_l15_s06_noise	0.98±0.02	0.99±0.01	106.22±7.93
data_03_l15_s12_noise	0.97±0.02	0.99±0.01	66.78±2.88
data_03_l20_s06_noise	1.0±0	1.0±0	126.44±11.11
data_03_l20_s12_noise	1.0±0	1.0±0	73.44±6.69
data_06_l1_s06_noise	0.98±0.02	0.99±0.01	106.56±2.67
data_06_l1_s12_noise	0.92±0.07	0.95±0.04	64.67±7.25
data_06_l15_s06_noise	0.98±0.03	0.99±0.01	121.11±11.02
data_06_l15_s12_noise	0.95±0.05	0.98±0.02	76.33±15.35
data_06_l20_s06_noise	0.98±0.03	0.99±0.01	141.11±19.65
data_06_l20_s12_noise	0.89±0.03	0.94±0.02	76.78±3.36

4.5.3 STI Simulado (NR-EIDDM)

A busca em grade foi realizada em $w \in [150, 160, \dots, 300]$ e $n_{bins} \in [5, 6, \dots, 15]$. A melhor configuração foi: $w = 300$, $n_{bins} = 13$. Os resultados médios (10 execuções) completos estão apresentados na Tabela 9.

Tabela 9 – Avaliação de desempenho do método NR-EIDDM para os dados do STI.

Base	Precisão	F1-score	Atraso Médio
simulation_fy2_1.0_nl_1.csv	0.96 ± 0	0.98 ± 0	82.33 ± 0
simulation_fy2_1.0_nl_2.csv	0.96 ± 0	0.98 ± 0	64.66 ± 0
simulation_fy2_1.0_nl_3.csv	0.97 ± 0	0.98 ± 0	94.66 ± 0
simulation_kfba2_1.0_nl_1.csv	0.96 ± 0	0.98 ± 0	67.0 ± 0
simulation_kfba2_1.0_nl_2.csv	0.97 ± 0	0.99 ± 0	83.0 ± 0
simulation_kfba2_1.0_nl_3.csv	0.98 ± 0	0.99 ± 0	76.0 ± 0
simulation_kfcv2_1.0_nl_1.csv	0.97 ± 0	0.99 ± 0	71.66 ± 0
simulation_kfcv2_1.0_nl_2.csv	0.97 ± 0	0.99 ± 0	76.66 ± 0
simulation_kfcv2_1.0_nl_3.csv	0.99 ± 0	0.99 ± 0	88.0 ± 0
simulation_kfe12_0.28_nl_1.csv	0.97 ± 0	0.99 ± 0	54.0 ± 0
simulation_kfe12_0.28_nl_2.csv	0.99 ± 0	0.99 ± 0	79.0 ± 0
simulation_kfe12_0.28_nl_3.csv	0.98 ± 0	0.99 ± 0	62.66 ± 0
simulation_kfv23_0.03_nl_1.csv	1.0 ± 0	1.0 ± 0	76.33 ± 0
simulation_kfv23_0.03_nl_2.csv	0.96 ± 0	0.98 ± 0	62.66 ± 0
simulation_kfv23_0.03_nl_3.csv	0.88 ± 0	0.93 ± 0	59.33 ± 0
simulation_kfv23_0.05_nl_1.csv	0.98 ± 0	0.99 ± 0	54.66 ± 0
simulation_kfv23_0.05_nl_2.csv	0.95 ± 0	0.97 ± 0	59.0 ± 0
simulation_kfv23_0.05_nl_3.csv	0.90 ± 0	0.94 ± 0	60.0 ± 0

4.5.4 STI Simulado (UDrAE)

O parâmetro foi avaliado de acordo com: janela deslizante $w \in [50, 100, \dots, 600]$. O melhor resultado foi: $w = 50$. Os resultados médios (10 execuções) estão apresentados na Tabela 10.

Tabela 10 – Avaliação do método UDrAE para as bases STI.

Base	Precisão	F1-Score	Atraso Médio
simulation_fy2_1.0_nl_1	0.92±0.01	0.96±0	4±0
simulation_fy2_1.0_nl_2	0.91±0.04	0.95±0.02	4.22±0.16
simulation_fy2_1.0_nl_3	0.89±0.02	0.94±0.01	4.22±0.16
simulation_kfba2_1.0_nl_1	0.74±0.04	0.85±0.02	4.33±0
simulation_kfba2_1.0_nl_2	0.82±0.01	0.90±0.01	4.56±0.42
simulation_kfba2_1.0_nl_3	0.83±0.03	0.91±0.02	27.89±3.93
simulation_kfcv2_1.0_nl_1	0.79±0.01	0.88±0.01	4.11±0.16
simulation_kfcv2_1.0_nl_2	0.79±0.01	0.88±0.02	4.33±0.27
simulation_kfcv2_1.0_nl_3	0.82±0.06	0.90±0.04	27.33±32.37
simulation_kfe12_0.28_nl_1	0.79±0.05	0.88±0.03	250.89±60.75
simulation_kfe12_0.28_nl_2	0.82±0.03	0.90±0.02	555.33±13.08
simulation_kfe12_0.28_nl_3	0.85±0.04	0.92±0.02	320.67±108.64
simulation_kfv23_0.03_nl_1	0.44±0.04	0.61±0.04	80.67±3.54
simulation_kfv23_0.03_nl_2	0.57±0.12	0.71±0.11	67.78±3.25
simulation_kfv23_0.03_nl_3	0.64±0.07	0.78±0.06	113.44±2.74
simulation_kfv23_0.05_nl_1	0.74±0.03	0.85±0.02	104.33±28.71
simulation_kfv23_0.05_nl_2	0.70±0.06	0.82±0.05	190.22±153.38
simulation_kfv23_0.05_nl_3	0.80±0.03	0.88±0.03	10.22±19.72

4.5.5 Bases de Sensores de Gás (NR-EIDDM)

A busca em grade foi realizada em $w \in [150, 160, \dots, 300]$ e $n_{bins} \in [5, 6, \dots, 15]$. A melhor configuração foi: $w = 200$, $n_{bins} = 6$. Os resultados médios (10 execuções) estão apresentados na Tabela 11.

Tabela 11 – Resultados do método NR-EIDDM para a os conjuntos de dados de Sensores de Gás.

Base	Precisão	F1-Score	Atraso Médio
class_1_expanded	0.85±0	0.92±0	24.0±0
class_2_expanded	0.93±0	0.96±0	125.5±0
class_3_expanded	0.80±0	0.89±0	55.0±0
class_4_expanded	0.84±0	0.91±0	60.0±0
class_5_expanded	0.90±0	0.95±0	62.0±0
class_6_expanded	0.91±0	0.95±0	29.0±0

4.5.6 Bases de Sensores de Gás (UDrAE)

A busca em grade foi realizada em: $w \in [50, 100, \dots, 600]$. A melhor configuração: $w = 200$. Os resultados médios (10 execuções) estão apresentados na Tabela 12.

Tabela 12 – Resultados do método UDrAE para a os conjuntos de dados de Sensores de Gás.

Base	Precisão	F1-Score	Atraso Médio
class_1_expanded	1.0±0	1.0±0	1.0±0
class_2_expanded	1.0±0	1.0±0	24.0±0
class_3_expanded	1.0±0	1.0±0	9.5±0
class_4_expanded	1.0±0	1.0±0	1.5±0
class_5_expanded	1.0±0	1.0±0	1.0±0
class_6_expanded	1.0±0	1.0±0	1.5±0

4.5.7 3W (NR-EIDDM)

Por não haver *ground truth*, utilizou-se parametrização fixa $w = 1500$, $n_{bins} = 3$. Os resultados estão apresentados na Tabela 13.

Tabela 13 – 3W (NR-EIDDM) - Resultados combinados com o classificador.

Nome	Baseline		Adaptado		Detecções
	Precisão	F1-Score	Precisão	F1-Score	
WELL-00002_20140126161944	0.2235	0.3035	0.9213 ± 0	0.9130 ± 0	246
WELL-00006_20180617200257	0.5295	0.6130	0.9699 ± 0	0.9766 ± 0	21
WELL-00009_20170313160804	0.1088	0.1637	0.9372 ± 0	0.9264 ± 0	56
WELL-00010_20171218190131	0.2553	0.3393	0.9076 ± 0	0.8772 ± 0	41
WELL-00016_20180404222255	0.6067	0.6821	0.9415 ± 0	0.9358 ± 0	10
WELL-00016_20180426102531	0.8313	0.8697	1.0000 ± 0	1.0000 ± 0	9
WELL-00016_20180517184536	0.1491	0.2151	1.0000 ± 0	1.0000 ± 0	179
WELL-00017_20140319020130	0.0280	0.0480	0.7588 ± 0	0.3438 ± 0	50
WELL-00002_20140301151700	0.6359	0.7075	0.8978 ± 0	0.9065 ± 0	75
WELL-00002_20140325170304	0.0482	0.0790	0.7803 ± 0	0.4569 ± 0	140
WELL-00018_20180611021218	0.0175	0.0310	0.7187 ± 0	0.3565 ± 0	98
WELL-00018_20190403023307	0.1795	0.2522	0.7651 ± 0	0.4846 ± 0	44

4.5.8 3W (UDrAE)

Objetivando comparações futuras, os parâmetros foram ajustados de modo que o número de detecções fosse o mais similar possível com o obtido pelo método NR-EIDDM. Os resultados estão apresentados na Tabela 14.

Tabela 14 – 3W (UDrAE) - Resultados combinados com o classificador.

Nome	Baseline		Adaptado		Detecções
	Precisão	F1-Score	Precisão	F1-Score	
WELL-00002_20140126161944	0.0027	0.0051	0.9297 ± 0	0.9226 ± 0	245
WELL-00006_20180617200257	0.5395	0.6221	0.9699 ± 0	0.9766 ± 0	16
WELL-00009_20170313160804	0.1088	0.1637	0.9458 ± 0.001	0.9379 ± 0.001	54
WELL-00010_20171218190131	0.2553	0.3393	0.9477 ± 0.002	0.9397 ± 0.003	39
WELL-00016_20180404222255	0.6677	0.7349	0.7610 ± 0	0.8129 ± 0	13
WELL-00016_20180426102531	0.7814	0.8295	0.8627 ± 0	0.8946 ± 0	8
WELL-00016_20180517184536	0.2030	0.2799	1.0000 ± 0	1.0000 ± 0	129
WELL-00017_20140319020130	0.0419	0.0696	0.7741 ± 0.01	0.4467 ± 0.04	58
WELL-00002_20140301151700	0.5870	0.6647	0.8978 ± 0	0.9065 ± 0	79
WELL-00002_20140325170304	0.0117	0.0212	0.7820 ± 0	0.4696 ± 0	121
WELL-00018_20180611021218	0.0261	0.0449	0.5746 ± 0.03	0.2496 ± 0.02	99
WELL-00018_20190403023307	0.1635	0.2329	0.7597 ± 0	0.4671 ± 0	42

4.5.9 T85 (NR-EIDDM)

Sem *ground truth*, adotou-se parametrização fixa visando comparabilidade. Os resultados estão apresentados na Tabela 15 de acordo com a parametrização a seguir. **NR-EIDDM**: $w = 255$, $n_{bins} = 40$.

Tabela 15 – Comparação de métricas para o conjunto de dados T85.

Métrica \ Método	<i>BaseLine</i>	NR-EIDDM
MSE	131.7472	66.0332 \pm 0
MAE	7.9992	5.1739 \pm 0
MAPE	2.18%	1.39 \pm 0%
N. de Detecções	N/A	5

4.5.10 T85 (UDrAE)

Os parâmetros foram ajustados de modo que o número de detecções fosse similar ao obtido com NR-EIDDM. Os resultados estão apresentados na Tabela 16 de acordo com a parametrização a seguir. **UDrAE**: $w = 50$.

Tabela 16 – Comparação de métricas para o conjunto de dados T85.

Métrica \ Método	<i>BaseLine</i>	UDrAE
MSE	131.7472	23.8937 \pm 0
MAE	7.9992	3.2229 \pm 0
MAPE	2.18%	0.88 \pm 0%
N. de Detecções	N/A	5

4.6 Comparações

Os métodos propostos foram comparados com QuantTree (QT) [Boracchi et al., 2018], QT-EWMA [Frittoli et al., 2021], SPLL [Kuncheva, 2011], OCDD [Gözüaçık and Can, 2021], D3 [Gözüaçık et al., 2019] e IKS [dos Reis et al., 2016]. Os três primeiros identificam o *drift* por meio de um único índice amostral, que marca o ponto sugerido da mudança, enquanto os demais mapeiam todas as amostras que compõem a região de transição. As métricas avaliadas foram *F1-score*, precisão e atraso médio, calculadas sobre as bases com regiões de mudança conhecidas. Para as bases sem *ground truth*, a parametrização de cada método foi ajustada de modo que o número de detecções fosse semelhante, permitindo adaptar o treinamento do regressor ou classificador de acordo com os pontos de mudança identificados. Como os três primeiros métodos produzem apenas um ponto de detecção, eles não foram considerados na comparação para as bases sem *ground truth*. Cada experimento foi executado dez vezes por base, e os valores apresentados

correspondem às médias obtidas. Entretanto, nos métodos NR-EIDDM, SPLL, OCDD e IKS, a natureza estatística determinística das abordagens faz com que os resultados sejam idênticos em todas as execuções sobre uma mesma base. Além disso, no caso específico do SPLL, como o método é determinístico e a detecção ocorre em um único ponto, um erro na localização implica métricas nulas, enquanto uma detecção correta resulta em todas as métricas iguais a 1. Os melhores resultados são apresentados em negrito.

4.6.1 Gaussianas

Essa subseção apresenta a comparação realizada para os conjuntos de dados da base de gaussianas. A Tabela 17 apresenta a comparação da precisão para todos os métodos analisados. A Tabela 18 apresenta a comparação do *F1-score* para todos os métodos analisados. Por fim, a Tabela 19 apresenta a comparação do atraso médio de detecção para todos os métodos analisados.

Tabela 17 – Bases de dados Gaussianas - Comparação da precisão entre os métodos QT-EWMA, QT, SPLL, NR-EIDDM, OCDD, D3, IKS e UDrAE.

Base \ Método	QT-EWMA	QT	SPLL	NR-EIDDM	OCDD	D3	IKS	UDrAE
data_0_noise.csv	0.49 ± 0.02	0.56 ± 0.02	0±0	0.86±0	0.93±0	0.28±0	0.69±0	0.79±0.04
data_01_l1_s06_noise.csv	0.45 ± 0.06	0.53 ± 0.04	1.0±0	0.85±0	0.53±0	0.64±0.01	0.64±0	0.90±0
data_01_l1_s12_noise.csv	0.58 ± 0.04	0.58 ± 0.01	1.0±0	0.85±0	0.93±0	0.66±0.01	0.58±0	0.88±0.03
data_01_l15_s06_noise.csv	0.12 ± 0.04	0.35 ± 0.02	0±0	0.87±0	0.85±0	0.76±0.02	0.72±0	0.88±0.06
data_01_l15_s12_noise.csv	0.38 ± 0.01	0.40 ± 0.02	1.0±0	0.86±0	0.86±0	0.79±0.02	0.67±0	0.93±0.03
data_01_l20_s06_noise.csv	0.47 ± 0.01	0.55 ± 0.03	1.0±0	0.86±0	0.72±0	0.76±0.02	0.64±0	0.93±0.06
data_01_l20_s12_noise.csv	0.41 ± 0.00	0.42 ± 0.05	1.0±0	0.85±0	0.94±0	0.79±0.02	0.56±0	0.95±0.04
data_03_l1_s06_noise.csv	0.28 ± 0.05	0.41 ± 0.01	1.0±0	0.87±0	0.54±0	0.76±0.02	0.69±0	0.95±0.02
data_03_l1_s12_noise.csv	0.33 ± 0.04	0.41 ± 0.03	1.0±0	0.86±0	0.88±0	0.74±0.02	0.61±0	1.0±0
data_03_l15_s06_noise.csv	0.43 ± 0.04	0.50 ± 0.04	0±0	0.87±0	0.65±0	0.76±0.02	0.72±0	0.98±0.02
data_03_l15_s12_noise.csv	0.51 ± 0.00	0.46 ± 0.04	1.0±0	0.84±0	0.94±0	0.79±0.02	0.65±0	0.97±0.02
data_03_l20_s06_noise.csv	0.51 ± 0.06	0.50 ± 0.01	1.0±0	0.86±0	0.76±0	0.76±0.02	0.63±0	1.0±0
data_03_l20_s12_noise.csv	0.41 ± 0.05	0.62 ± 0.00	0±0	0.86±0	0.90±0	0.75±0	0.56±0	1.0±0
data_06_l1_s06_noise.csv	0.34 ± 0.01	0.61 ± 0.06	0±0	0.87±0	0.72±0	0.76±0.02	0.60±0	0.98±0.02
data_06_l1_s12_noise.csv	0.36 ± 0.01	0.44 ± 0.06	0±0	0.86±0	0.79±0	0.79±0.02	0.59±0	0.92±0.07
data_06_l15_s06_noise.csv	0.65 ± 0.02	0.63 ± 0.05	1.0±0	0.87±0	0.42±0	0.75±0	0.59±0	0.98±0.03
data_06_l15_s12_noise.csv	0.42 ± 0.03	0.52 ± 0.02	1.0±0	0.86±0	0.83±0	0.75±0	0.68±0	0.95±0.05
data_06_l20_s06_noise.csv	0.36 ± 0.03	0.49 ± 0.01	1.0±0	0.87±0	0.78±0	0.76±0.02	0.70±0	0.98±0.03
data_06_l20_s12_noise.csv	0.49 ± 0.03	0.52 ± 0.01	0±0	0.86±0	0.80±0	0.75±0	0.79±0	0.89±0.03

4.6.2 STI Simulado

Essa subseção apresenta a comparação de desempenho dos métodos aplicados à base do STI Simulado. Os métodos QuantTree [Boracchi et al., 2018], QT-EWMA [Frittoli et al., 2021] e IKS [dos Reis et al., 2016] não foram capazes de detectar nenhuma ocorrência de *drift* nessa base, resultando em todas as métricas iguais a zero. Dessa forma, as análises consideram apenas os demais métodos. A Tabela 20 apresenta a comparação em termos de precisão, a Tabela 21 mostra os resultados de *F1-score*, e a Tabela 22 exhibe a comparação referente ao atraso médio de detecção.

Tabela 18 – Bases de dados Gaussianas - Comparação do *F1-score* entre os métodos QT-EWMA, QT, SPLL, NR-EIDDM, OCDD, D3, IKS e UDrAE.

Base \ Método	QT-EWMA	QT	SPLL	NR-EIDDM	OCDD	D3	IKS	UDrAE
data_0_noise.csv	0.66 ± 0.01	0.72 ± 0.01	0 ± 0	0.93 ± 0	0.96 ± 0	0.44 ± 0	0.82 ± 0	0.88 ± 0.02
data_01_l1_s06_noise	0.62 ± 0.04	0.69 ± 0.02	1.0 ± 0	0.92 ± 0	0.70 ± 0	0.78 ± 0.01	0.78 ± 0	0.95 ± 0
data_01_l1_s12_noise	0.73 ± 0.03	0.73 ± 0.01	1.0 ± 0	0.92 ± 0	0.97 ± 0	0.79 ± 0.01	0.74 ± 0	0.90 ± 0.02
data_01_l15_s06_noise	0.21 ± 0.03	0.51 ± 0.02	0 ± 0	0.93 ± 0	0.92 ± 0	0.86 ± 0.01	0.84 ± 0	0.94 ± 0.03
data_01_l15_s12_noise	0.55 ± 0.01	0.57 ± 0.01	1.0 ± 0	0.93 ± 0	0.93 ± 0	0.88 ± 0.01	0.80 ± 0	0.96 ± 0.02
data_01_l20_s06_noise	0.64 ± 0.01	0.71 ± 0.03	1.0 ± 0	0.93 ± 0	0.84 ± 0	0.86 ± 0.01	0.78 ± 0	0.96 ± 0.03
data_01_l20_s12_noise	0.58 ± 0.00	0.59 ± 0.04	1.0 ± 0	0.92 ± 0	0.97 ± 0	0.88 ± 0.01	0.72 ± 0	0.98 ± 0.02
data_03_l1_s06_noise	0.44 ± 0.04	0.58 ± 0.01	1.0 ± 0	0.93 ± 0	0.70 ± 0	0.86 ± 0.01	0.81 ± 0	0.97 ± 0.01
data_03_l1_s12_noise	0.50 ± 0.03	0.58 ± 0.02	1.0 ± 0	0.92 ± 0	0.94 ± 0	0.85 ± 0.01	0.76 ± 0	1.0 ± 0
data_03_l15_s06_noise	0.60 ± 0.03	0.67 ± 0.03	0 ± 0	0.93 ± 0	0.79 ± 0	0.86 ± 0.01	0.77 ± 0	0.99 ± 0.01
data_03_l15_s12_noise	0.68 ± 0.00	0.63 ± 0.04	1.0 ± 0	0.91 ± 0	0.97 ± 0	0.88 ± 0.01	0.79 ± 0	0.99 ± 0.01
data_03_l20_s06_noise	0.68 ± 0.05	0.67 ± 0.01	1.0 ± 0	0.93 ± 0	0.86 ± 0	0.86 ± 0.01	0.77 ± 0	1.0 ± 0
data_03_l20_s12_noise	0.58 ± 0.04	0.77 ± 0.00	0 ± 0	0.92 ± 0	0.95 ± 0	0.86 ± 0	0.71 ± 0	1.0 ± 0
data_06_l1_s06_noise	0.51 ± 0.01	0.76 ± 0.04	0 ± 0	0.93 ± 0	0.84 ± 0	0.86 ± 0.01	0.75 ± 0	0.99 ± 0.01
data_06_l1_s12_noise	0.53 ± 0.01	0.61 ± 0.05	0 ± 0	0.93 ± 0	0.88 ± 0	0.88 ± 0.01	0.74 ± 0	0.95 ± 0.04
data_06_l15_s06_noise	0.79 ± 0.01	0.70 ± 0.04	1.0 ± 0	0.93 ± 0	0.59 ± 0	0.86 ± 0	0.75 ± 0	0.99 ± 0.01
data_06_l15_s12_noise	0.59 ± 0.02	0.68 ± 0.01	1.0 ± 0	0.93 ± 0	0.91 ± 0	0.75 ± 0	0.81 ± 0	0.98 ± 0.02
data_06_l20_s06_noise	0.53 ± 0.02	0.66 ± 0.01	1.0 ± 0	0.93 ± 0	0.88 ± 0	0.86 ± 0.01	0.82 ± 0	0.99 ± 0.01
data_06_l20_s12_noise	0.66 ± 0.02	0.68 ± 0.01	0 ± 0	0.93 ± 0	0.89 ± 0	0.86 ± 0	0.88 ± 0	0.94 ± 0.02

Tabela 19 – Bases de dados Gaussianas — Comparação do **atraso médio** entre os métodos QT-EWMA, QT, SPLL, NR-EIDDM, OCDD, D3, IKS e UDrAE.

Base \ Método	QT-EWMA	QT	SPLL	NR-EIDDM	OCDD	D3	IKS	UDrAE
data_0_noise.csv	39.35 ± 8.42	55.43 ± 11.26	N/A ± 0	95 ± 0	13.67 ± 0	249.0 ± 0	45.67 ± 0	39.56 ± 4.32
data_01_l1_s06_noise.csv	63.2 ± 14.83	74.72 ± 9.51	120.0 ± 0	128.33 ± 0	44.67 ± 0	249.0 ± 0	74.00 ± 0	96.78 ± 6.09
data_01_l1_s12_noise.csv	41.67 ± 6.22	56.0 ± 10.17	24.0 ± 0	79 ± 0	14.00 ± 0	249.0 ± 0	52.67 ± 0	51.44 ± 11.44
data_01_l15_s06_noise.csv	61.91 ± 18.45	63.31 ± 12.89	N/A ± 0	120 ± 0	44.67 ± 0	249.0 ± 0	71.33 ± 0	91.56 ± 0.16
data_01_l15_s12_noise.csv	40.44 ± 7.35	55.2 ± 9.08	88.0 ± 0	81.33 ± 0	22.67 ± 0	249.0 ± 0	49.67 ± 0	59.78 ± 5.25
data_01_l20_s06_noise.csv	53.70 ± 15.12	62.40 ± 7.67	120.0 ± 0	120.33 ± 0	43.00 ± 0	249.0 ± 0	65.67 ± 0	99.67 ± 4.24
data_01_l20_s12_noise.csv	39.68 ± 9.61	55.24 ± 12.73	88.0 ± 0	114.0 ± 0	14.33 ± 0	249.0 ± 0	27.67 ± 0	67.78 ± 11.77
data_03_l1_s06_noise.csv	54.61 ± 5.84	64.59 ± 16.03	88.0 ± 0	111.33 ± 0	44.33 ± 0	249.0 ± 0	55.00 ± 0	102.56 ± 11.77
data_03_l1_s12_noise.csv	43.18 ± 3.11	55.22 ± 9.82	56.0 ± 0	99.33 ± 0	14.67 ± 0	249.0 ± 0	46.67 ± 0	59.78 ± 3.81
data_03_l15_s06_noise.csv	61.67 ± 11.42	76.48 ± 13.27	N/A ± 0	110.33 ± 0	46.67 ± 0	249.0 ± 0	59.00 ± 0	106.22 ± 7.93
data_03_l15_s12_noise.csv	42.08 ± 9.15	54.61 ± 17.68	88.0 ± 0	84.66 ± 0	22.33 ± 0	249.0 ± 0	47.33 ± 0	66.78 ± 2.88
data_03_l20_s06_noise.csv	53.22 ± 12.67	72.0 ± 21.50	120.0 ± 0	104.0 ± 0	32.00 ± 0	249.0 ± 0	70.00 ± 0	126.44 ± 11.11
data_03_l20_s12_noise.csv	38.44 ± 6.41	55.48 ± 8.27	N/A ± 0	93.0 ± 0	18.67 ± 0	249.0 ± 0	44.33 ± 0	73.44 ± 6.69
data_06_l1_s06_noise.csv	54.12 ± 13.34	63.87 ± 15.45	N/A ± 0	128.33 ± 0	29.67 ± 0	249.0 ± 0	62.67 ± 0	106.56 ± 2.67
data_06_l1_s12_noise.csv	43.39 ± 4.28	56.0 ± 13.61	N/A ± 0	88.0 ± 0	24.00 ± 0	249.0 ± 0	52.33 ± 0	64.67 ± 7.25
data_06_l15_s06_noise.csv	68.35 ± 17.92	80.89 ± 19.13	120.0 ± 0	107.66 ± 0	61.00 ± 0	249.0 ± 0	66.67 ± 0	121.11 ± 11.02
data_06_l15_s12_noise.csv	38.29 ± 5.03	55.38 ± 7.14	56.0 ± 0	77.33 ± 0	17.33 ± 0	249.0 ± 0	63.33 ± 0	76.33 ± 15.35
data_06_l20_s06_noise.csv	63.33 ± 21.89	74.94 ± 26.77	152.0 ± 0	35.0 ± 0	40.67 ± 0	249.0 ± 0	77.33 ± 0	141.11 ± 19.65
data_06_l20_s12_noise.csv	35.39 ± 8.55	52.31 ± 10.91	N/A ± 0	84.33 ± 0	19.67 ± 0	249.0 ± 0	45.00 ± 0	76.78 ± 3.36

4.6.3 Base de Sensores de Gás

Essa subseção apresenta a comparação de desempenho dos métodos aplicados à base de sensores de gás. Os métodos QuantTree [Boracchi et al., 2018], QT-EWMA [Frittoli et al., 2021], SPLL [Kuncheva, 2011] e OCDD [Gözüaçık and Can, 2021] não foram capazes de detectar nenhuma ocorrência de *drift* nessa base, resultando em todas as métricas iguais a zero. Dessa forma, as análises consideram apenas os demais métodos. A Tabela 23 apresenta a comparação em termos de precisão, a Tabela 24 mostra os resultados de *F1-score*, e a Tabela 25 exhibe a comparação referente ao atraso médio de detecção.

Tabela 20 – Bases de dados do STI — Comparação da **precisão** entre os métodos NR-EIDDM, OCDD, D3, UDrAE e SPLL.

Método Base	NR-EIDDM	OCDD	D3	UDrAE	SPLL
simulation_fy2_1.0_nl_1	0.91 ± 0	0.90 ± 0	0.74 ± 0.07	0.92 ± 0.01	1.0 ± 0
simulation_fy2_1.0_nl_2	0.91 ± 0	0.92 ± 0	0.76 ± 0.03	0.91 ± 0.04	0
simulation_fy2_1.0_nl_3	0.92 ± 0	0.87 ± 0	0.72 ± 0.03	0.89 ± 0.02	0
simulation_kfba2_1.0_nl_1	0.85 ± 0	0.87 ± 0	0.64 ± 0.06	0.74 ± 0.04	0
simulation_kfba2_1.0_nl_2	0.90 ± 0	0.87 ± 0	0.65 ± 0.03	0.82 ± 0.01	1.0 ± 0
simulation_kfba2_1.0_nl_3	0.91 ± 0	0.87 ± 0	0.63 ± 0.03	0.83 ± 0.03	1.0 ± 0
simulation_kfcv2_1.0_nl_1	0.82 ± 0	0.77 ± 0	0.69 ± 0.06	0.79 ± 0.01	0
simulation_kfcv2_1.0_nl_2	0.89 ± 0	0.88 ± 0	0.70 ± 0.04	0.79 ± 0.01	0
simulation_kfcv2_1.0_nl_3	0.88 ± 0	0.82 ± 0	0.66 ± 0.03	0.82 ± 0.06	1.0 ± 0
simulation_kfe12_0.28_nl_1	0.74 ± 0	0.84 ± 0	0.45 ± 0.15	0.79 ± 0.05	1.0 ± 0
simulation_kfe12_0.28_nl_2	0.99 ± 0	0.89 ± 0	0.20 ± 0.07	0.82 ± 0.03	0
simulation_kfe12_0.28_nl_3	0.89 ± 0	0.88 ± 0	0.19 ± 0.06	0.85 ± 0.04	1.0 ± 0
simulation_kfv23_0.03_nl_1	0.75 ± 0	0.62 ± 0	0.19 ± 0.06	0.44 ± 0.04	0
simulation_kfv23_0.03_nl_2	0.93 ± 0	0.60 ± 0	0.13 ± 0.03	0.57 ± 0.12	1.0 ± 0
simulation_kfv23_0.03_nl_3	0.80 ± 0	0.62 ± 0	0.07 ± 0.03	0.64 ± 0.07	1.0 ± 0
simulation_kfv23_0.05_nl_1	0.68 ± 0	0.82 ± 0	0.41 ± 0.09	0.74 ± 0.03	1.0 ± 0
simulation_kfv23_0.05_nl_2	0.98 ± 0	0.89 ± 0	0.29 ± 0.03	0.70 ± 0.06	1.0 ± 0
simulation_kfv23_0.05_nl_3	0.88 ± 0	0.71 ± 0	0.25 ± 0.06	0.80 ± 0.03	1.0 ± 0

Tabela 21 – Bases de dados do STI — Comparação do **F1-score** entre os métodos NR-EIDDM, OCDD, D3, UDrAE e SPLL.

Método Base	NR-EIDDM	OCDD	D3	UDrAE	SPLL
simulation_fy2_1.0_nl_1	0.95 ± 0	0.95 ± 0	0.85 ± 0.04	0.96 ± 0	1.0 ± 0
simulation_fy2_1.0_nl_2	0.95 ± 0	0.96 ± 0	0.86 ± 0.02	0.95 ± 0.02	0
simulation_fy2_1.0_nl_3	0.96 ± 0	0.93 ± 0	0.83 ± 0.02	0.94 ± 0.01	0
simulation_kfba2_1.0_nl_1	0.92 ± 0	0.93 ± 0	0.78 ± 0.04	0.85 ± 0.02	0
simulation_kfba2_1.0_nl_2	0.95 ± 0	0.93 ± 0	0.79 ± 0.02	0.90 ± 0.01	1.0 ± 0
simulation_kfba2_1.0_nl_3	0.95 ± 0	0.93 ± 0	0.78 ± 0.02	0.91 ± 0.02	1.0 ± 0
simulation_kfcv2_1.0_nl_1	0.90 ± 0	0.87 ± 0	0.82 ± 0.04	0.88 ± 0.01	0
simulation_kfcv2_1.0_nl_2	0.94 ± 0	0.93 ± 0	0.82 ± 0.02	0.88 ± 0.02	0
simulation_kfcv2_1.0_nl_3	0.93 ± 0	0.90 ± 0	0.80 ± 0.02	0.90 ± 0.04	1.0 ± 0
simulation_kfe12_0.28_nl_1	0.85 ± 0	0.91 ± 0	0.61 ± 0.14	0.88 ± 0.03	1.0 ± 0
simulation_kfe12_0.28_nl_2	0.99 ± 0	0.94 ± 0	0.32 ± 0.09	0.90 ± 0.02	0
simulation_kfe12_0.28_nl_3	0.94 ± 0	0.94 ± 0	0.32 ± 0.08	0.92 ± 0.02	1.0 ± 0
simulation_kfv23_0.03_nl_1	0.85 ± 0	0.76 ± 0	0.30 ± 0.08	0.61 ± 0.04	0
simulation_kfv23_0.03_nl_2	0.96 ± 0	0.75 ± 0	0.23 ± 0.06	0.71 ± 0.11	1.0 ± 0
simulation_kfv23_0.03_nl_3	0.89 ± 0	0.77 ± 0	0.13 ± 0.05	0.78 ± 0.06	1.0 ± 0
simulation_kfv23_0.05_nl_1	0.81 ± 0	0.90 ± 0	0.58 ± 0.09	0.85 ± 0.02	1.0 ± 0
simulation_kfv23_0.05_nl_2	0.99 ± 0	0.94 ± 0	0.45 ± 0.04	0.82 ± 0.05	1.0 ± 0
simulation_kfv23_0.05_nl_3	0.93 ± 0	0.83 ± 0	0.39 ± 0.08	0.88 ± 0.03	1.0 ± 0

4.6.4 3W

Essa subseção apresenta a comparação de desempenho dos métodos aplicados à base 3W. Os métodos QuantTree [Boracchi et al., 2018], QT-EWMA [Frittoli et al., 2021] e SPLL [Kuncheva, 2011] não foram capazes de detectar nenhuma ocorrência de *drift* nesse conjunto de dados, resultando em todas as métricas iguais a zero. Dessa forma, as análises consideram apenas os demais métodos. A Tabela 26 apresenta a comparação em termos de precisão e a Tabela 27 mostra os resultados de *F1-score*.

4.6.5 T85

Essa subseção apresenta a comparação de desempenho dos métodos aplicados à base T85. Os métodos QuantTree [Boracchi et al., 2018], QT-EWMA [Frittoli et al., 2021]

Tabela 22 – Comparação do atraso médio entre os métodos NR-EIDDM, OCDD, D3, UDrAE e SPLL para os conjuntos de dados do STI.

Método Base	NR-EIDDM	OCDD	D3	UDrAE	SPLL
simulation_fy2_1.0_nl_1.csv	138.33 ± 0	58.67 ± 0	147.0 ± 1.41	4.00 ± 0	12.0 ± 0
simulation_fy2_1.0_nl_2.csv	101.67 ± 0	66.67 ± 0	182.0 ± 29.02	4.22 ± 0.16	N/A
simulation_fy2_1.0_nl_3.csv	86.00 ± 0	10.67 ± 0	167.0 ± 47.77	4.22 ± 0.16	N/A
simulation_kfba2_1.0_nl_1.csv	240.00 ± 0	37.33 ± 0	142.0 ± 51.30	4.33 ± 0	N/A
simulation_kfba2_1.0_nl_2.csv	552.66 ± 0	66.0 ± 0	247.0 ± 97.12	4.56 ± 0.42	524 ± 0
simulation_kfba2_1.0_nl_3.csv	382.00 ± 0	54.0 ± 0	207.0 ± 38.37	27.89 ± 3.93	524 ± 0
simulation_kfcv2_1.0_nl_1.csv	108.00 ± 0	36.67 ± 0	142.0 ± 27.6	4.11 ± 0.16	N/A
simulation_kfcv2_1.0_nl_2.csv	89.33 ± 0	54.67 ± 0	202.0 ± 25.34	4.33 ± 0.27	N/A
simulation_kfcv2_1.0_nl_3.csv	108.66 ± 0	35.33 ± 0	247.0 ± 55.79	27.33 ± 32.37	12.0 ± 0
simulation_kfe12_0.28_nl_1.csv	102.66 ± 0	67.33 ± 0	192.0 ± 25.73	250.89 ± 60.75	76.0 ± 0
simulation_kfe12_0.28_nl_2.csv	143.66 ± 0	51.33 ± 0	594.5 ± 303.92	555.33 ± 13.08	N/A
simulation_kfe12_0.28_nl_3.csv	63.33 ± 0	92.67 ± 0	517.0 ± 135.73	320.67 ± 108.64	76.0 ± 0
simulation_kfv23_0.03_nl_1.csv	133.67 ± 0	472.67 ± 0	124.5 ± 48.78	80.67 ± 3.54	N/A
simulation_kfv23_0.03_nl_2.csv	126.67 ± 0	732.67 ± 0	539.5 ± 273.55	67.78 ± 3.25	44.0 ± 0
simulation_kfv23_0.03_nl_3.csv	78.33 ± 0	334.67 ± 0	674.5 ± 657.02	113.44 ± 2.74	44.0 ± 0
simulation_kfv23_0.05_nl_1.csv	100.33 ± 0	52.67 ± 0	132.0 ± 20.54	104.33 ± 28.71	44.0 ± 0
simulation_kfv23_0.05_nl_2.csv	125.66 ± 0	54.0 ± 0	109.5 ± 5.2	190.22 ± 153.38	44.0 ± 0
simulation_kfv23_0.05_nl_3.csv	102.00 ± 0	30.0 ± 0	192.0 ± 73.50	10.22 ± 19.72	44.0 ± 0

Tabela 23 – Comparação da **precisão** entre os métodos D3, IKS, NR-EIDDM e UDrAE para os conjuntos de dados de sensores de gás.

Método Base	D3	IKS	NR-EIDDM	UDrAE
class_1_expanded	0.72±0.06	0.79±0	0.85±0	1.0±0
class_2_expanded	0.77±0.07	0.81±0	0.93±0	1.0±0
class_3_expanded	0.70±0.02	0.70±0	0.80±0	1.0±0
class_4_expanded	0.78±0.12	0.73±0	0.84±0	1.0±0
class_5_expanded	0.81±0.02	0.81±0	0.90±0	1.0±0
class_6_expanded	0.70±0.03	0.70±0	0.91±0	1.0±0

Tabela 24 – Comparação do **F1-score** entre os métodos D3, IKS, NR-EIDDM e UDrAE para os conjuntos de dados de sensores de gás.

Método Base	D3	IKS	NR-EIDDM	UDrAE
class_1_expanded	0.82±0.06	0.88±0	0.92±0	1.0±0
class_2_expanded	0.87±0.05	0.90±0	0.96±0	1.0±0
class_3_expanded	0.82±0.02	0.82±0	0.89±0	1.0±0
class_4_expanded	0.84±0.11	0.84±0	0.91±0	1.0±0
class_5_expanded	0.90±0.01	0.90±0	0.95±0	1.0±0
class_6_expanded	0.82±0.02	0.82±0	0.95±0	1.0±0

e SPLL [Kuncheva, 2011] não foram considerados nas comparações, uma vez que suas implementações não suportam múltiplas detecções em um mesmo *dataset*. Dessa forma, as análises consideram apenas os demais métodos. Por se tratar de um problema de regressão, a Tabela 28 apresenta a comparação em termos de MSE (*Mean Squared Error*), MAE (*Mean Absolute Error*) e MAPE (*Mean Absolute Percentage Error*).

4.7 Considerações

Os resultados apresentados neste capítulo demonstram a efetividade das estratégias propostas para detecção não supervisionada de *drifts* incrementais, tanto em cenários sintéticos quanto em contextos industriais reais. Em geral, os métodos **NR-EIDDM** e

Tabela 25 – Comparação do **atraso médio** entre os métodos D3, IKS, NR-EIDDM e UDrAE para os conjuntos de dados de sensores de gás.

Método Base	D3	IKS	NR-EIDDM	UDrAE
class_1_expanded	259.0±70.98	50.0±0	24.0±0	1.0±0
class_2_expanded	354.5±92.06	74.5±0	125.5±0	24.0±0
class_3_expanded	215.5±77.32	45.0±0	55.0±0	9.5±0
class_4_expanded	292.0±232.29	58.0±0	60.0±0	1.5±0
class_5_expanded	218.5±73.31	46.5±0	62.0±0	1.0±0
class_6_expanded	203.0±47.39	98.5±0	29.0±0	1.5±0

Tabela 26 – 3W — Comparação da **Precisão** entre o *Baseline* e os métodos OCDD, D3, IKS, NR-EIDDM e UDrAE.

Nome	Baseline	Adaptado				
		OCDD	D3	IKS	NR-EIDDM	UDrAE
WELL-00002_20140126161944	0.0027	0.9226± 0	0.9205± 0	0.9205 ± 0	0.9213 ± 0	0.9297± 0
WELL-00006_20180617200257	0.5395	0.9699 ± 0	0.9699 ± 0	0.9699 ± 0	0.9699± 0	0.9699 ± 0
WELL-00009_20170313160804	0.1088	0.9372 ± 0	0.9349 ± 0	0.9349 ± 0	0.9372 ± 0	0.9458± 0.001
WELL-00010_20171218190131	0.2553	0.9451 ± 0	0.9444 ± 0.003	0.9370 ± 0	0.9076 ± 0	0.9477 ± 0.002
WELL-00016_20180404222255	0.6677	0.9063 ± 0	0.7610 ± 0.02	0.9449± 0	0.9415 ± 0	0.7610 ± 0
WELL-00016_20180426102531	0.7814	0.8627 ± 0	0.8627 ± 0	0.8627± 0	1.0000± 0	0.8627 ± 0
WELL-00016_20180517184536	0.2030	1.0000 ± 0	1.0000± 0	1.0000± 0	1.0000± 0	1.0000± 0
WELL-00017_20140319020130	0.0419	0.7501 ± 0	0.7527 ± 0.01	0.7534± 0	0.7588 ± 0	0.7741 ± 0.01
WELL-00002_20140301151700	0.5870	0.8978 ± 0	0.8978± 0	0.8978± 0	0.8978± 0	0.8978± 0
WELL-00002_20140325170304	0.0117	0.7795 ± 0± 0	0.7802	0.7802± 0	0.7803 ± 0	0.7820 ± 0
WELL-00018_20180611021218	0.0261	0.6648 ± 0	0.6447 ± 0	0.6447± 0	0.7187± 0	0.5746 ± 0.03
WELL-00018_20190403023307	0.1635	0.7681 ± 0	0.7611 ± 0	0.7668 ± 0	0.7651± 0	0.7597 ± 0

Tabela 27 – 3W — Comparação do **F1-score** entre o *Baseline* e os métodos em Adaptado (OCDD, D3, IKS, NR-EIDDM e UDrAE).

Nome	Baseline	Adaptado				
		OCDD	D3	IKS	NR-EIDDM	UDrAE
WELL-00002_20140126161944	0.3035	0.9142± 0 ± 0	0.9121 ± 0	0.9121± 0	0.9130± 0	0.9226± 0
WELL-00006_20180617200257	0.6130	0.9766± 0	0.9766± 0	0.9766± 0	0.9766± 0	0.9766± 0
WELL-00009_20170313160804	0.1637	0.9264 ± 0	0.9231 ± 0	0.9231± 0	0.9264± 0	0.9379 ± 0.001
WELL-00010_20171218190131	0.3393	0.9362 ± 0	0.9353 ± 0.003	0.9248 ± 0	0.8772± 0	0.9397 ± 0.003
WELL-00016_20180404222255	0.6821	0.8726 ± 0	0.8129 ± 0.03	0.9405± 0	0.9358± 0	0.8129 ± 0
WELL-00016_20180426102531	0.8697	0.8946 ± 0	0.8946 ± 0	0.8946± 0	1.0000± 0	0.8946 ± 0
WELL-00016_20180517184536	0.2151	1.0000± 0	1.0000± 0	1.0000± 0	1.0000± 0	1.0000± 0
WELL-00017_20140319020130	0.0480	0.2913 ± 0	0.3046 ± 0.01	0.3056 ± 0	0.3438± 0	0.4467 ± 0.04
WELL-00002_20140301151700	0.7075	0.9065 ± 0	0.9065± 0	0.9065 ± 0	0.9065± 0	0.9065± 0
WELL-00002_20140325170304	0.0790	0.4509 ± 0	0.4557 ± 0	0.4557 ± 0	0.4569± 0	0.4696 ± 0
WELL-00018_20180611021218	0.0310	0.2654 ± 0	0.2605 ± 0	0.2605± 0	0.3565± 0	0.2496 ± 0.02
WELL-00018_20190403023307	0.2522	0.5024± 0	0.4668 ± 0	0.4959± 0	0.4846± 0	0.4671 ± 0

UDrAE mostraram-se de certa forma complementares: o primeiro oferece uma caracterização estatística explícita das mudanças, fundamentada na análise de densidades e testes de hipóteses, enquanto o segundo busca a sensibilidade a alterações sutis nos padrões de reconstrução de dados sujeitos a ruído por meio de técnicas de aprendizagem de máquina.

Nas bases sintéticas, os experimentos permitiram avaliar o comportamento controlado dos detectores frente a diferentes magnitudes e velocidades de mudança. Nessas condições, observou-se que ambos os métodos mantêm estabilidade na taxa de falsos positivos, com atrasos médios de detecção similares a grande parte do demais métodos comparados. Em todas as avaliações, especialmente nas simulações do **Sistema de Tanques Interativos (STI)** e nos conjuntos de dados de sensores reais que condizem mais propriamente com cenários industriais, verificou-se que os métodos propostos preservam

Tabela 28 – Comparação de métricas para o conjunto de dados T85.

Métrica \ Método	<i>BaseLine</i>	OCDD	D3	IKS	UDrAE	NR-EIDDM
MSE	131.7472	33.8451 ± 0	26.2081 ± 0	85.0667 ± 0	23.8937 ± 0	66.0332 ± 0
MAE	7.9992	3.6144 ± 0	3.2610 ± 0	5.9845 ± 0	3.2229 ± 0	5.1739 ± 0
MAPE	2.18%	0.98 ± 0%	0.88 ± 0%	1.61 ± 0%	0.88 ± 0%	1.39% ± 0
N. de Detecções	N/A	5	5	5	5	5

desempenho satisfatório, mesmo diante de variações dinâmicas complexas e ruído inerente às medições, com ganhos na maior parte das métricas avaliadas.

As comparações com os detectores da literatura evidenciaram desempenho competitivo dos métodos propostos. Em especial, o **NR-EIDDM** apresentou resultados consistentes em termos de precisão e *F1-score*, destacando-se pela robustez estatística e interpretabilidade dos histogramas. Já nas bases de sensores reais, em especial nas bases **3W** e **T85**, o **UDrAE** obteve resultados mais expressivos. A arquitetura de *Denoising AutoEncoder* mostrou-se mais eficaz em identificar mudanças incrementais sujeitas a ruído, provavelmente por capturar dependências não lineares entre variáveis e reduzir o impacto de flutuações de ruído correlacionado ao longo do tempo. Essa propriedade é particularmente vantajosa em dados industriais de alta variabilidade, nos quais os desvios podem ocorrer de forma contínua e distribuída entre múltiplas dimensões.

Nos cenários em que os métodos propostos apresentaram desempenho inferior aos de referência, as causas mais prováveis estão associadas à natureza conservadora dos limiares adotados. O uso de limites estatísticos mais amplos no **NR-EIDDM**, tende a reduzir a incidência de falsos alarmes, elevando as métricas de precisão e *F1-Score*, porém também pode introduzir atraso na detecção. De modo análogo, o **UDrAE**, ao depender da convergência do erro de reconstrução, requer um número mínimo de amostras após a mudança para acumular evidência suficiente de variação, o que pode retardar a resposta inicial em *drifts* mais abruptos e curtos.

Além disso, vale destacar dois comportamentos específicos observados nos métodos de referência. O método **SPLL**, por ser determinístico e realizar apenas uma detecção por base, apresenta métricas unitárias sempre que a detecção coincide com uma região da mudança incremental. Nesses casos, as métricas de precisão, *F1-score* assumem valor igual a 1, refletindo a natureza do método. O fato de ter somente uma detecção, também leva as métricas a 0 no caso de um falso negativo. Devido a essa natureza, para os métodos que realizam somente uma detecção, tornou-se impossível realizar testes em bases reais sem conhecimento *a priori* das posições de *drift*. Isso dificulta o cenário em que deve-se escolher um número similar de detecções para uma comparação justa. Além disso, o método **OCDD** apresentou atrasos médios menores em algumas bases, resultado que está provavelmente associado à sua maior sensibilidade. Contudo, essa mesma sensibilidade também o torna mais suscetível a flutuações ruidosas, o que explica o fato de suas métricas *F1-score* e

precisão terem sido inferiores às dos métodos propostos para as mesmas bases avaliadas evidenciando uma maior taxa de falso positivos.

Nos experimentos com dados reais sem rótulos provenientes de sistemas industriais (bases **3W** e **T85**), a integração das detecções automáticas ao processo de retreinamento de modelos de regressão/classificação confirmou o potencial prático das abordagens. Os resultados indicaram melhora significativa no desempenho preditivo quando as janelas de reconfiguração foram definidas a partir dos pontos de *drift* identificados pelos métodos propostos, o que comprova seu valor agregado quando aplicado em rotinas de monitoramento de desempenho e ajuste de modelos. Essa estratégia também evidencia as vantagens de se detectar toda a região da mudança.

Em síntese, os experimentos comprovaram que os métodos desenvolvidos atendem aos objetivos propostos neste trabalho: detectar, de forma automática e não supervisionada, mudanças incrementais sob ruído procurando preservar maior interpretabilidade. No próximo capítulo são apresentadas as **conclusões gerais** e as **propostas de continuidade**, discutindo as implicações práticas, limitações e possibilidades de evolução das abordagens introduzidas.

Capítulo 5

Conclusão e Trabalhos Futuros

Este trabalho apresentou duas metodologias não supervisionadas para detecção de *drifts* incrementais em dados de processos industriais: o **NR-EIDDM** (*Noise-Resistant Equal-Intensity Drift Detection Method*) e o **UDrAE** (*Unsupervised Drift detection via Denoising AutoEncoder reconstruction loss*). Ambas as abordagens foram concebidas com o objetivo de oferecer robustez frente a pontos atípicos e ausência de rótulos, condições presentes em fluxos de dados industriais reais.

5.1 Síntese das Contribuições

Ambos os métodos propostos foram validados com bases sintéticas (geradas com funções gaussianas sob diferentes níveis de ruído), simulações do **Sistema de Tanques Interativos (STI)** e dados reais provenientes de sistemas da **PETROBRAS**. Em todas as bases avaliadas, as metodologias detectaram as mudanças presentes e exibiram desempenho competitivo frente a detectores amplamente consolidados, como *QT*, *QT-EWMA*, *SPLL*, *OCDD*, *IKS* e *D3*.

O método **NR-EIDDM** baseia-se em uma abordagem estatística e de densidade, combinando os algoritmos DBSCAN e *K-Means* com restrições para criar partições de densidades iguais. Essas partições permitem a construção de histogramas de densidades uniformes, cuja comparação é realizada por meio do teste qui-quadrado de Pearson. A integração da filtragem inicial realizada pelo DBSCAN associada ao agrupamento baseado em densidade e da análise estatística de similaridade possibilita detectar variações sutis em distribuições, reduzindo falsos positivos mesmo em ambientes com ruído. Além disso, o método permite identificar não apenas um ponto de mudança, mas **regiões de drift**, o que possibilita relaxar a parametrização e reduzir alarmes falsos em detrimento de um possível atraso maior no início da região da mudança.

O método **UDrAE**, por sua vez, apresentou uma proposta simples e eficaz para detecção de *drifts* em fluxos contínuos de dados industriais, com foco especial em cenários

com ruído e mudanças incrementais. Baseado na reconstrução de dados via *Denoising AutoEncoder*, o método combina filtragem de ruído e modelagem de padrões operacionais em um mesmo arcabouço. As detecções são realizadas a partir do monitoramento do erro de reconstrução, tipicamente o erro médio absoluto, que se eleva de forma sistemática quando o sistema passa a operar sob um novo regime. Uma vantagem deste método frente aos demais é o fato de que, além de apresentar toda a região da mudança, também é possível associar uma intensidade à mudança de acordo com a magnitude do erro absoluto médio. Isso traz maior interpretabilidade para a região de *drift*. Os resultados experimentais demonstraram que o **UDrAE** é competitivo frente às principais técnicas da literatura, especialmente sob condições reais. Sua efetividade foi evidenciada nas bases industriais, nas quais a combinação entre reconstrução e remoção de ruído produziu detecções confiáveis.

A disponibilidade de *ground truth* em parte das bases avaliadas permitiu uma análise determinística do desempenho dos métodos. Nos conjuntos **T85** e **3W**, onde apenas uma avaliação indireta foi possível, o **UDrAE** destacou-se por apresentar bom desempenho na maioria das métricas e uma baixa taxa de falsos positivos, configurando-se como um diferencial relevante para aplicações industriais. Em síntese, o método demonstrou um equilíbrio consistente mostrando-se adequado a cenários industriais que exigem confiabilidade, baixo custo de manutenção e robustez a ruído.

5.2 Vantagens e Características Observadas

Entre os principais diferenciais observados, destacam-se:

- **Ausência de supervisão:** Ambos os métodos operam sem necessidade de rótulos, aspecto crucial em cenários industriais onde o *ground truth* raramente é conhecido.
- **Robustez a ruído e outliers:** O **NR-EIDDM** realiza filtragem multidimensional explícita de pontos atípicos por meio do DBSCAN e constrói histogramas de igual densidade; o **UDrAE**, por sua vez, promove essa filtragem de forma implícita durante o treinamento do autoencoder.
- **Interpretação estatística:** O **NR-EIDDM** fornece detecções baseadas em hipóteses formais por meio do teste χ^2 , enquanto o **UDrAE** associa a variação no erro de reconstrução a desvios estruturais do processo.
- **Capacidade de modelar não linearidades:** O **UDrAE** representa relações complexas entre variáveis e identifica mudanças incrementais, contínuas e multidimensionais.

- **Simplicidade e aplicabilidade industrial:** Ambos possuem custo computacional compatível com fácil integração a rotinas de monitoramento, permitindo atualização automática de modelos de regressão/classificação quando *drifts* são detectados.

5.3 Limitações e Hipóteses de Projeto

Os métodos propostos carregam limitações inerentes às suas escolhas de projeto e hipóteses de funcionamento, destacando-se:

- **Esparsidade dos dados:** O **NR-EIDDM** é sensível a dados esparsos, pois tanto o **DBSCAN** quanto o *K-Means* com restrições dependem de proximidade local entre amostras. Entretanto, os métodos propostos são voltados para fluxos contínuos de séries temporais, típicos de processos industriais, nos quais as amostras sequenciais são próximas; quando essa localidade está ausente (por exemplo, em entradas altamente esparsas ou não estruturadas), a estimação de densidade e o particionamento podem se degradar. Nesses casos, pode ser necessário um mapeamento prévio para um espaço latente contínuo (por exemplo, via PCA/PLS ou autoencoders), o que introduz suposições adicionais de modelagem.
- **Estrutura desbalanceada:** Regiões com poucas amostras podem ser rotuladas como ruído pelo **DBSCAN** ou inviabilizar o particionamento de igual intensidade, exigindo ajuste fino dos parâmetros. Neste caso, de forma similar ao apresentado anteriormente, o método espera fluxos contínuos de séries temporais, típicos de processos industriais, nos quais as amostras sequenciais são próximas, o que não condiz com estrutura desbalanceada.
- **Pressupostos do teste χ^2 :** No **NR-EIDDM**, o teste χ^2 opera sobre contagens agrupadas em *bins*. Isso significa que seu poder e comportamento quanto a alarmes falsos dependem do número de *bins* escolhido e do tamanho da janela amostral. As contagens esperadas por *bin* devem ser suficientemente grandes (por exemplo, $E_i \geq 5$), e a janela deslizante deve conter amostras em quantidade suficiente para produzir estatísticas confiáveis. Aumentar n_{bins} eleva os graus de liberdade (e o valor crítico), afetando a sensibilidade; assim, n_{bins} e o tamanho da janela w representam um compromisso entre **responsividade** e **estabilidade**.
- **Domínio de aplicabilidade:** Tanto o **NR-EIDDM** quanto o **UDrAE** foram projetados para dados contínuos e séries temporais; O **UDrAE** pressupõe que as mudanças se manifestem de forma incremental no espaço de reconstrução. Dados categóricos ou não estruturados não são suportados e, caso necessário, requerem representação numérica consistente.

- **Atraso em drifts abruptos:** O UDrAE, devido à necessidade de estabilização da função de perda, apresenta atraso adicional na detecção de *drifts* abruptos, embora compense esse efeito com maior robustez em regimes ruidosos.
- **Não Linearidade:** Embora essa característica represente uma vantagem para o método UDrAE, por possibilitar o uso de representações não lineares, ela também introduz complexidade adicional que pode reduzir a eficiência do PCA e tornar a rede mais extensa e computacionalmente mais custosa.

5.4 Trabalhos Futuros

Em continuidade a este estudo, são sugeridas as seguintes direções de pesquisa:

- **Aprimoramento adaptativo de parâmetros:** Investigar estratégias automáticas de ajuste de limiares e janelas deslizantes, baseadas em *meta-learning* ou técnicas de otimização, de modo a reduzir a dependência de parâmetros fixos e aumentar a adaptabilidade frente a diferentes regimes operacionais [Vilalta and Drissi, 2002, Finn et al., 2017, Webb et al., 2016].
- **Integração híbrida:** Combinar o NR-EIDDM e o UDrAE em um arcabouço unificado, permitindo que o primeiro atue como um filtro estatístico inicial e o segundo refine as detecções no espaço latente, seguindo abordagens híbridas que conciliam métodos estatísticos e modelos baseados em aprendizado profundo [Gama et al., 2014, Lu et al., 2019].
- **Escalabilidade e operação em tempo real:** Avaliar o desempenho das metodologias em fluxos de dados de alta frequência, desenvolvendo versões paralelas ou distribuídas para integração em plataformas industriais, como sistemas de historiadores de dados e arquiteturas baseadas em OPC-UA, visando garantir baixa latência e viabilidade operacional [Bifet et al., 2018].
- **Interpretação dos *drifts*:** Criar mecanismos para identificar as variáveis mais relevantes associadas a cada mudança detectada, promovendo maior interpretabilidade e suporte à tomada de decisão, conforme discutido em abordagens de explicabilidade aplicadas a modelos de aprendizado de máquina e detecção de mudanças [Lundberg and Lee, 2017, Webb et al., 2016].
- **Tratamento em tempo real de ruído:** Investigar novas estratégias para lidar com ruídos não estacionários durante a operação em fluxo contínuo, incluindo métodos robustos e adaptativos de filtragem e aprendizado, de forma a otimizar a resposta dos detectores em ambientes industriais ruidosos [Qin, 2012, Bishop, 2006].

5.5 Considerações Finais

Em síntese, os métodos propostos, **NR-EIDDM** e **UDrAE**, demonstraram robustez e aplicabilidade em cenários industriais reais. O primeiro destacou-se pela fundamentação estatística e resistência a ruído, enquanto o segundo evidenciou sua capacidade de adaptação a dados espúrios. Ambos contribuem para o avanço de soluções não supervisionadas de monitoramento e reconfiguração automática de modelos, reforçando o potencial de aplicação prática em sistemas industriais dinâmicos e sujeitos à mudanças incrementais.

Referências

- S. Agrahari and A. K. Singh. Concept drift detection in data stream mining: A literature review. *Journal of King Saud University-Computer and Information Sciences*, 34(10): 9523–9540, 2022.
- C. Alippi and M. Roveri. Just-in-time adaptive classifiers - part I: detecting nonstationary changes. *IEEE Trans. Neural Networks*, 19(7):1145–1153, 2008. doi: 10.1109/TNN.2008.2000082. URL <https://doi.org/10.1109/TNN.2008.2000082>.
- D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.
- R. Arya and G. Sikka. An optimized approach for density based spatial clustering application with noise. In *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol I: Hosted by CSI Vishakapatnam Chapter*, pages 695–702. Springer, 2014.
- A. Ashfahani, M. Pratama, E. Lughofer, and Y.-S. Ong. Devdan: Deep evolving denoising autoencoder. *Neurocomputing*, 390:297–314, 2020.
- M. Baena-Garcia, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno. Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams*, volume 6, pages 77–86, 2006.
- R. S. M. Barros and S. G. T. C. Santos. A large-scale comparison of concept drift detectors. *Information Sciences*, 451:348–370, 2018.
- F. Bayram, B. S. Ahmed, and A. Kassler. From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*, page 108632, 2022.
- A. Bhattacharya, R. Jaiswal, and A. Kumar. Faster algorithms for the constrained k-means problem. *Theory of computing systems*, 62:93–115, 2018. doi: 10.1007/s00224-017-9820-7.
- A. Bifet, R. Gavaldà, G. Holmes, and B. Pfahringer. *Data Stream Mining: A Practical Approach*. MIT Press, 2018.

- C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- I. I. F. Blanco, J. del Campo-Ávila, G. Ramos-Jiménez, R. M. Bueno, A. A. O. Díaz, and Y. C. Mota. Online and non-parametric drift detection methods based on hoeffding's bounds. *IEEE Trans. Knowl. Data Eng.*, 27(3):810–823, 2015. doi: 10.1109/TKDE.2014.2345382. URL <https://doi.org/10.1109/TKDE.2014.2345382>.
- G. Boracchi, C. Cervellera, and D. Macciò. Uniform histograms for change detection in multivariate data. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1732–1739. IEEE, 2017.
- G. Boracchi, D. Carrera, C. Cervellera, and D. Maccio. Quanttree: Histograms for change detection in multivariate data streams. In *International Conference on Machine Learning*, pages 639–648. PMLR, 2018.
- P. S. Bradley, K. P. Bennett, and A. Demiriz. Constrained k-means clustering. *Microsoft Research, Redmond*, 20(0):0, 2000.
- A. R. Braga. Implementação de estratégias de controle multimalha e multivariável. Master's thesis, Universidade Federal de Minas Gerais, 1994.
- L. Bu, C. Alippi, and D. Zhao. A pdf-free change detection test based on density difference estimation. *IEEE Trans. Neural Networks Learn. Syst.*, 29(2):324–334, 2018. doi: 10.1109/TNNLS.2016.2619909. URL <https://doi.org/10.1109/TNNLS.2016.2619909>.
- W. M. Caminhas et al. *Estratégias de detecção e diagnóstico de falhas em sistemas dinâmicos*. PhD thesis, Universidade Estadual de Campinas, 1997.
- G. Casella and R. L. Berger. *Statistical Inference*. Duxbury, 2nd edition, 2002.
- R. A. Coelho and C. L. de Castro. Abordagem espacial via quadtree para detecção de mudança de conceito em fluxos de dados contínuos. In *Congresso Brasileiro de Automática-CBA*, volume 2, 2020.
- R. A. Coelho, L. C. B. Torres, and C. L. de Castro. Concept drift detection with quadtree-based spatial mapping of streaming data. *Information Sciences*, 2023.
- M. F. D'Angelo, R. M. Palhares, R. H. Takahashi, and R. H. Loschi. Fuzzy/bayesian change point detection approach to incipient fault detection. *IET control theory & applications*, 5(4):539–551, 2011.
- T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi. An information-theoretic approach to detecting changes in multidimensional data streams. *Interfaces*, 01 2006.

- J. de Andrade Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. P. de Leon Ferreira de Carvalho, and J. Gama. Data stream clustering: A survey. *ACM Comput. Surv.*, 46(1):13:1–13:31, 2013. doi: 10.1145/2522968.2522981. URL <https://doi.org/10.1145/2522968.2522981>.
- R. F. de Mello, Y. Vaz, C. H. Grossi, and A. Bifet. On learning guarantees to unsupervised concept drift detection on data streams. *Expert Systems with Applications*, 117:90–102, 2019.
- D. M. dos Reis, P. Flach, S. Matwin, and G. Batista. Fast unsupervised online drift detection using incremental kolmogorov-smirnov test. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1545–1554, 2016.
- M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.
- C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *Proceedings of the 34th International Conference on Machine Learning*, 2017.
- P. Fränti and J. Kivijärvi. Randomised local search algorithm for the clustering problem. *Pattern Analysis & Applications*, 3:358–369, 2000.
- L. Frittoli, D. Carrera, and G. Boracchi. Change detection in multivariate datastreams controlling false alarms. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I 21*, pages 421–436. Springer, 2021.
- J. Gama. A survey on learning from data streams: current and future trends. *Prog. Artif. Intell.*, 1(1):45–55, 2012. doi: 10.1007/s13748-011-0002-6. URL <https://doi.org/10.1007/s13748-011-0002-6>.
- J. Gama and G. Castillo. Learning with local drift detection. In *Advanced Data Mining and Applications: Second International Conference, ADMA 2006, Xi'an, China, August 14–16, 2006 Proceedings 2*, pages 42–55. Springer, 2006.
- J. Gama, P. Medas, G. Castillo, and P. P. Rodrigues. Learning with drift detection. In A. L. C. Bazzan and S. Labidi, editors, *Advances in Artificial Intelligence - SBIA 2004, 17th Brazilian Symposium on Artificial Intelligence, São Luis, Maranhão, Brazil, September 29 - October 1, 2004, Proceedings*, volume 3171 of *Lecture Notes in Computer Science*, pages 286–295. Springer, 2004. doi: 10.1007/978-3-540-28645-5_29. URL https://doi.org/10.1007/978-3-540-28645-5_29.

- J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.
- R. N. Gemaque, A. F. J. Costa, R. Giusti, and E. M. Dos Santos. An overview of unsupervised drift detection methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(6):e1381, 2020.
- H. Ghomeshi, M. M. Gaber, and Y. Kovalchuk. Eacd: evolutionary adaptation to concept drifts in data streams. *Data Mining and Knowledge Discovery*, 33(3):663–694, 2019.
- Ö. Gözüaçık and F. Can. Concept learning using one-class classifiers for implicit drift detection in evolving data streams. *Artificial Intelligence Review*, 54(5):3725–3747, 2021.
- Ö. Gözüaçık, A. Büyükçakır, H. Bonab, and F. Can. Unsupervised concept drift detection with a discriminative classifier. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2365–2368, 2019.
- E. B. Gulcan and F. Can. Unsupervised concept drift detection for multi-label data streams. *Artificial Intelligence Review*, 56(3):2401–2434, 2023.
- H. Guo, H. Li, Q. Ren, and W. Wang. Concept drift type identification based on multi-sliding windows. *Information Sciences*, 585:1–23, 2022.
- F. Hinder, V. Vaquet, and B. Hammer. One or two things we know about concept drift—a survey on monitoring in evolving environments. part a: detecting concept drift. *Frontiers in Artificial Intelligence*, 7:1330257, 2024.
- A. S. Iwashita and J. P. Papa. An overview on concept drift learning. *Ieee Access*, 7:1532–1547, 2018.
- S. Jaramillo-Valbuena, J. M. Londoño-Peláez, and S. A. Cardona. Performance evaluation of concept drift detection techniques in the presence of noise. *Performance evaluation*, 38(39), 2017.
- F. Jota, A. Braga, and R. Pena. Performance assessment of advanced process control algorithms using an interacting tank system. In *IAS'95. Conference Record of the 1995 IEEE Industry Applications Conference Thirtieth IAS Annual Meeting*, volume 2, pages 1565–1571. IEEE, 1995.
- P. Kadlec, B. Gabrys, and S. Strandt. Data-driven soft sensors in the process industry. *Computers & Chemical Engineering*, 33(4):795–814, 2009. ISSN 0098-1354. doi: <https://doi.org/10.1016/j.compchemeng.2008.12.012>. URL <https://www.sciencedirect.com/science/article/pii/S0098135409000076>.

- M. M. R. Khan, M. A. B. Siddique, R. B. Arif, and M. R. Oishe. Adbscan: Adaptive density-based spatial clustering of applications with noise for identifying clusters with varying densities. In *2018 4th international conference on electrical engineering and information & communication technology (iCEEiCT)*, pages 107–111. IEEE, 2018.
- D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *VLDB*, volume 4, pages 180–191. Toronto, Canada, 2004.
- B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Wozniak. Ensemble learning for data stream analysis: A survey. *Inf. Fusion*, 37:132–156, 2017. doi: 10.1016/j.inffus.2017.02.004. URL <https://doi.org/10.1016/j.inffus.2017.02.004>.
- L. I. Kuncheva. Change detection in streaming multivariate data using likelihood detectors. *IEEE transactions on knowledge and data engineering*, 25(5):1175–1180, 2011.
- D. S. D. Lara, R. A. Coelho, C. L. Castro, and W. M. Caminhas. An unsupervised noise-resistant method for detection of incremental drifts. *Knowledge and Information Systems*, 2025a. doi: 10.1007/s10115-025-02592-2. (Publicado).
- D. S. D. Lara, R. A. Coelho, C. L. Castro, and W. M. Caminhas. Unsupervised drift detection via denoising autoencoder reconstruction loss. *IEEE Access*, 2025b. (Em Revisão).
- H. Li et al. Measurement and characterization of electromagnetic noise in industrial environments. *Sensors*, 2019. URL <https://www.mdpi.com/1424-8220/19/14/3104>. Acesso em: 2026-01-06.
- A. Likas, N. Vlassis, and J. J. Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.
- A. Liu, Y. Song, G. Zhang, and J. Lu. Regional concept drift detection and density synchronized drift adaptation. In C. Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 2280–2286. ijcai.org, 2017a. doi: 10.24963/ijcai.2017/317. URL <https://doi.org/10.24963/ijcai.2017/317>.
- A. Liu, G. Zhang, and J. Lu. Fuzzy time windowing for gradual concept drift adaptation. In *2017 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2017, Naples, Italy, July 9-12, 2017*, pages 1–6. IEEE, 2017b. doi: 10.1109/FUZZ-IEEE.2017.8015596. URL <https://doi.org/10.1109/FUZZ-IEEE.2017.8015596>.
- A. Liu, J. Lu, F. Liu, and G. Zhang. Accumulating regional density dissimilarity for concept drift detection in data streams. *Pattern Recognition*, 76:256–272, 2018.

- A. Liu, J. Lu, and G. Zhang. Concept drift detection via equal intensity k-means space partitioning. *IEEE transactions on cybernetics*, 51(6):3198–3211, 2020a.
- A. Liu, J. Lu, and G. Zhang. Concept drift detection: Dealing with missing values via fuzzy distance estimations. *IEEE Transactions on Fuzzy Systems*, 29(11):3219–3233, 2020b.
- B. Lu and L. Chiang. Semi-supervised online soft sensor maintenance experiences in the chemical industry. *Journal of Process Control*, 67:23–34, 2018.
- J. Lu, A. Liu, F. Dong, and F. Gu. Concept drift detection for industrial processes. *IEEE Transactions on Industrial Informatics*, 2019. doi: 10.1109/TII.2019.2894472.
- J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under concept drift: A review. *CoRR*, abs/2004.05785, 2020. URL <https://arxiv.org/abs/2004.05785>.
- N. Lu, G. Zhang, and J. Lu. Concept drift detection via competence models. *Artif. Intell.*, 209:11–28, 2014. doi: 10.1016/j.artint.2014.01.001. URL <https://doi.org/10.1016/j.artint.2014.01.001>.
- N. Lu, J. Lu, G. Zhang, and R. L. de Mántaras. A concept drift-tolerant case-base editing technique. *Artif. Intell.*, 230:108–133, 2016a. doi: 10.1016/j.artint.2015.09.009. URL <https://doi.org/10.1016/j.artint.2015.09.009>.
- N. Lu, J. Lu, G. Zhang, and R. L. De Mántaras. A concept drift-tolerant case-base editing technique. *Artificial Intelligence*, 230:108–133, 2016b.
- D. Lukats, O. Zielinski, A. Hahn, and F. Stahl. A benchmark and survey of fully unsupervised concept drift detectors on real-world data streams. *International Journal of Data Science and Analytics*, 19(1):1–31, 2025.
- S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017.
- J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Math., Stat., and Prob*, page 281, 1965.
- M. I. Malinen and P. Fränti. Balanced k-means for clustering. In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, S+ SSPR 2014, Joensuu, Finland, August 20-22, 2014. Proceedings*, pages 32–41. Springer, 2014.
- H. K. Mittermaier, J. D. le Roux, and I. K. Craig. Model-plant mismatch diagnosis using plant model ratios for a grinding mill circuit under model predictive control. *Minerals Engineering*, 227:109278, 2025.

- G. Montavon, G. Orr, and K.-R. Müller. *Neural networks: tricks of the trade*, volume 7700. springer, 2012.
- A. M. Mustafa, G. Ayoade, K. Al-Naami, L. Khan, K. W. Hamlen, B. Thuraisingham, and F. Araujo. Unsupervised deep embedding for novel class detection over data stream. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 1830–1839. IEEE, 2017.
- A. Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.
- R. Parameswaran. Book review: Statistics for experimenters: An introduction to design, data analysis, and model building, 1979.
- K. Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.
- K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11): 559–572, 1901.
- D. Pelleg, A. W. Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734, 2000.
- M. Pereira. *Desenvolvimento e Aplicação de um Método de Detecção e Identificação de Falhas Baseado em Observadores Dedicados*. PhD thesis, Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica . . . , 1996.
- A. Pesaranghader, H. L. Viktor, and E. Paquet. Mcdiarmid drift detection methods for evolving data streams. In *2018 International joint conference on neural networks (IJCNN)*, pages 1–9. IEEE, 2018.
- R.-E. Precup, P. Angelov, B. S. J. Costa, and M. Sayed-Mouchaweh. An overview on fault diagnosis and nature-inspired optimal control of industrial process applications. *Computers in Industry*, 74:75–94, 2015.
- S. J. Qin. Survey on data-driven industrial process monitoring and diagnosis. *Control Engineering Practice*, 21(9):1069–1083, 2012. doi: 10.1016/j.conengprac.2012.02.001.
- S. Ramírez-Gallego, B. Krawczyk, S. García, M. Wozniak, and F. Herrera. A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing*, 239:39–57, 2017. doi: 10.1016/j.neucom.2017.01.078. URL <https://doi.org/10.1016/j.neucom.2017.01.078>.

- J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data mining and knowledge discovery*, 2: 169–194, 1998.
- E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017. doi: 10.1145/3068335.
- P. Sobhani and H. Beigy. New drift detection method for data streams. In *Adaptive and Intelligent Systems: Second International Conference, ICAIS 2011, Klagenfurt, Austria, September 6-8, 2011. Proceedings*, pages 88–97. Springer, 2011.
- V. M. Souza, F. A. Chowdhury, and A. Mueen. Unsupervised drift detection on high-speed data streams. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 102–111. IEEE, 2020.
- Z. Sun, J. Tang, J. Qiao, and C. Cui. Review of concept drift detection method for industrial process modeling. In *2020 39th Chinese Control Conference (CCC)*, pages 5754–5759. IEEE, 2020.
- E. Teixeira, A. Braga, and H. Yehia. Cop-kmeans e clustering semi-supervisionado através de restrições. In G. d. A. Barreto and J. A. F. Costa, editors, *Anais do 10 Congresso Brasileiro de Inteligência Computacional*, pages 1–8, Fortaleza, CE, 2011. SBIC.
- R. E. V. Vargas, C. J. Munaro, P. M. Ciarelli, A. G. Medeiros, B. G. do Amaral, D. C. Barrionuevo, J. C. D. de Araújo, J. L. Ribeiro, and L. P. Magalhães. A realistic and public dataset with rare undesirable real events in oil wells. *Journal of Petroleum Science and Engineering*, 181:106223, 2019. ISSN 0920-4105. doi: <https://doi.org/10.1016/j.petrol.2019.106223>. URL <http://www.sciencedirect.com/science/article/pii/S0920410519306357>.
- A. Vergara. Gas Sensor Array Drift Dataset. UCI Machine Learning Repository, 2012. DOI: <https://doi.org/10.24432/C5RP6W>.
- R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95, 2002. doi: 10.1023/A:1019956318069.
- G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30:964–994, 2016. doi: 10.1007/s10618-015-0448-4.
- Y. Wu, K. Ianakiev, and V. Govindaraju. Improved k-nearest neighbor classification. *Pattern recognition*, 35(10):2311–2318, 2002. doi: 10.1016/s0031-3203(01)00132-7.

-
- S. Xu and J. Wang. Dynamic extreme learning machine for data stream classification. *Neurocomputing*, 238:433–449, 2017.
- K. Yonekawa, K. Saito, and M. Kurokawa. Riden: Neural-based uniform density histogram for distribution shift detection. In *Proceedings of the Second International Conference on AI-ML Systems*, pages 1–9, 2022.
- S. Yu, Z. Abraham, H. Wang, M. Shah, Y. Wei, and J. C. Príncipe. Concept drift detection and adaptation with hierarchical hypothesis testing. *Journal of the Franklin Institute*, 356(5):3187–3215, 2019.
- L. Yuan, H. Li, B. Xia, C. Gao, M. Liu, W. Yuan, and X. You. Recent advances in concept drift adaptation methods for deep learning. In *IJCAI*, pages 5654–5661, 2022.
- Y. Zhang, G. Chu, P. Li, X. Hu, and X. Wu. Three-layer concept drifting detection in text data streams. *Neurocomputing*, 260:393–403, 2017.