Proposta de uma abordagem para a detecção online de mudanças de conceito em fluxos contínuos de dados

Discente: Ruivaldo Neto
Orientador: Ricardo Rios

Universidade Federal da Bahia
Departamento de Ciência da Computação
Programa de Pós-Graduação em Ciência da Computação

Contato: rneto@rneto.dev

14 de Junho de 2019

Roteiro

1. Introdução
2. Revisão Bibliográfica
3. Plano de Pesquisa
4. Experimentos Iniciais
5. Conclusão

Introdução

Contexto e Motivação

- Avanços tecnológicos recentes contribuiram para um aumento exponencial no volume de dados produzidos por sistemas computacionais [39].

Contexto e Motivação

- Avanços tecnológicos recentes contribuiram para um aumento exponencial no volume de dados produzidos por sistemas computacionais [39].
- Parte significativa dos dados é produzida atráves de Fluxos Contínuos de Dados (FCDs): sequências ininterruptas e potencialmente infinitas de eventos [2].

Contexto e Motivação

- Avanços tecnológicos recentes contribuiram para um aumento exponencial no volume de dados produzidos por sistemas computacionais [39].
- Parte significativa dos dados é produzida atráves de Fluxos Contínuos de Dados (FCDs): sequências ininterruptas e potencialmente infinitas de eventos [2].
- FCDs estão presentes em diversos domínios de aplicação:
  - Monitoramento de tráfico;
  - Gestão de redes de telecomunicação;
  - Detecção de intrusos.

Contexto e Motivação

- Técnicas de Aprendizado de Máquina (AM) têm sido aplicadas para extrair informações úteis de grandes conjuntos de dados.

Contexto e Motivação

- Técnicas de Aprendizado de Máquina (AM) têm sido aplicadas para extrair informações úteis de grandes conjuntos de dados.
- Cenários com FCDs limitam a aplicação de técnicas de AM, pois impõem restrições de tempo de resposta, de uso dos recursos computacionais e apresentam comportamento não estacionário.

Contexto e Motivação

- Técnicas de Aprendizado de Máquina (AM) têm sido aplicadas para extrair informações úteis de grandes conjuntos de dados.
- Cenários com FCDs limitam a aplicação de técnicas de AM, pois impõem restrições de tempo de resposta, de uso dos recursos computacionais e apresentam comportamento não estacionário.
- Em cenários não estacionários, o contexto do processo gerador e/ou a distribuição dos dados podem sofrer alterações (mudanças de conceito) ao longo do tempo.

Contexto e Motivação

- Técnicas de Aprendizado de Máquina (AM) têm sido aplicadas para extrair informações úteis de grandes conjuntos de dados.
- Cenários com FCDs limitam a aplicação de técnicas de AM, pois impõem restrições de tempo de resposta, de uso dos recursos computacionais e apresentam comportamento não estacionário.
- Em cenários não estacionários, o contexto do processo gerador e/ou a distribuição dos dados podem sofrer alterações (mudanças de conceito) ao longo do tempo.
- A ocorrência de mudanças de conceito pode impactar a acurácia da técnica aplicada.

Contexto e Motivação

- A atualização periódica de modelos, apesar de computacionalmente ineficiente, foi utilizada como estratégia para mitigar a perda de acurácia causada por tais mudanças.

Contexto e Motivação

- A atualização periódica de modelos, apesar de computacionalmente ineficiente, foi utilizada como estratégia para mitigar a perda de acurácia causada por tais mudanças.
- Visando obter soluções computacionalmente eficientes e com maior precisão, pesquisadores propuseram novos métodos de detecção de mudança de conceito baseados em monitoramento.

Contexto e Motivação

- Entretanto, os métodos disponíveis na literatura ainda apresentam limitações ao serem aplicados em cenários com FCDs [2]:

Contexto e Motivação

- Entretanto, os métodos disponíveis na literatura ainda apresentam limitações ao serem aplicados em cenários com FCDs [2]:
  - Necessidade de rotulação;
  - Eficiência computacional (tempo de resposta e uso de recursos).

Contexto e Motivação

- Entretanto, os métodos disponíveis na literatura ainda apresentam limitações ao serem aplicados em cenários com FCDs [2]:
  - Necessidade de rotulação;
  - Eficiência computacional (tempo de resposta e uso de recursos).
- Visando mitigar estas limitações, este trabalho discute uma abordagem baseada em Redes de Função de Base Radial (redes RBF) para detecção de mudanças de conceito em FCDs.

Hipótese

"A aplicação de Redes de Função de Base Radial em fluxos contínuos de dados permite a detecção de mudanças de conceito em tempo de execução, de forma computacionalmente eficiente e independente de rótulos."

Objetivos

- Validação da hipótese através do desenvolvimento de um novo método baseado em redes RBF.

Objetivos

- Validação da hipótese através do desenvolvimento de um novo método baseado em redes RBF.
- Análise do método proposto através de comparações com o estado da arte.

Objetivos

- Validação da hipótese através do desenvolvimento de um novo método baseado em redes RBF.
- Análise do método proposto através de comparações com o estado da arte.
- Utilizar, no mínimo, dois conjuntos de dados durante os experimentos, sendo um sintético e outro oriundo de uma aplicação da indústria.

Revisão Bibliográfica

Fluxos Contínuos de Dados e Aprendizado de Máquina

- Fluxos Contínuos de Dados (FCDs) são sequências ininterruptas e potencialmente infinitas de eventos [2].

Fluxos Contínuos de Dados e Aprendizado de Máquina

- Fluxos Contínuos de Dados (FCDs) são sequências ininterruptas e potencialmente infinitas de eventos [2].
- Não podem ser armazenados em sua totalidade e, por serem de alta frequência, devem ser analisados em tempo real.

Fluxos Contínuos de Dados e Aprendizado de Máquina

- Fluxos Contínuos de Dados (FCDs) são sequências ininterruptas e potencialmente infinitas de eventos [2].
- Não podem ser armazenados em sua totalidade e, por serem de alta frequência, devem ser analisados em tempo real.
- Algoritmos supervisionados [12, 7, 37, 4, 18] e não-supervisionados [3, 1, 23] da área de AM foram adaptados para atenderem a essas restrições.

Fluxos Contínuos de Dados e Aprendizado de Máquina

- Fluxos Contínuos de Dados (FCDs) são sequências ininterruptas e potencialmente infinitas de eventos [2].
- Não podem ser armazenados em sua totalidade e, por serem de alta frequência, devem ser analisados em tempo real.
- Algoritmos supervisionados [12, 7, 37, 4, 18] e não-supervisionados [3, 1, 23] da área de AM foram adaptados para atenderem a essas restrições.
- Contudo, essas especializações não tratam a ocorrência de mudanças de conceito.

Mudança de Conceito

- A Teoria Bayesiana de Decisão [15] é comumente utilizada para descrever a tarefa de classificação e pode ser utilizada para formalizar a noção de mudança de conceito.

Mudança de Conceito

- A Teoria Bayesiana de Decisão [15] é comumente utilizada para descrever a tarefa de classificação e pode ser utilizada para formalizar a noção de mudança de conceito.
- Considerando que p_{t_0} e p_{t_1} denotam as distribuições de probabilidades conjuntas nos instantes t_0 e t_1, é possível afirmar que há mudança de conceito entre os instantes t_0 e t_1 se:

$$\exists X : p_{t_0}(X, c) \neq p_{t_1}(X, c) \tag{1}$$

Mudança de Conceito

- A Teoria Bayesiana de Decisão [15] é comumente utilizada para descrever a tarefa de classificação e pode ser utilizada para formalizar a noção de mudança de conceito.
- Considerando que p_{t_0} e p_{t_1} denotam as distribuições de probabilidades conjuntas nos instantes t_0 e t_1, é possível afirmar que há mudança de conceito entre os instantes t_0 e t_1 se:

$$\exists X : p_{t_0}(X, c) \neq p_{t_1}(X, c) \tag{1}$$

- Um conjunto de dados possui resultados esperados legítimos em t_0, mas este mesmo conjunto passa a ter resultados esperados diferentes, também legítimos, em t_1 [22].

Mudança de Conceito

- As mudanças de conceito podem ser categorizadas como Virtuais ou Reais [19]:

Mudança de Conceito

- As mudanças de conceito podem ser categorizadas como Virtuais ou Reais [19]:
  - Mudanças Virtuais são causadas por alterações na probabilidade a priori das classes, $P(c)$, e não alteram os conceitos-alvo.

Mudança de Conceito

- As mudanças de conceito podem ser categorizadas como Virtuais ou Reais [19]:
  - Mudanças Virtuais são causadas por alterações na probabilidade a priori das classes, $P(c)$, e não alteram os conceitos-alvo.
  - Mudanças Reais surgem a partir de alterações na probabilidade a posteriori, $p(c|X)$, e modificam os resultados esperados.

Mudança de Conceito

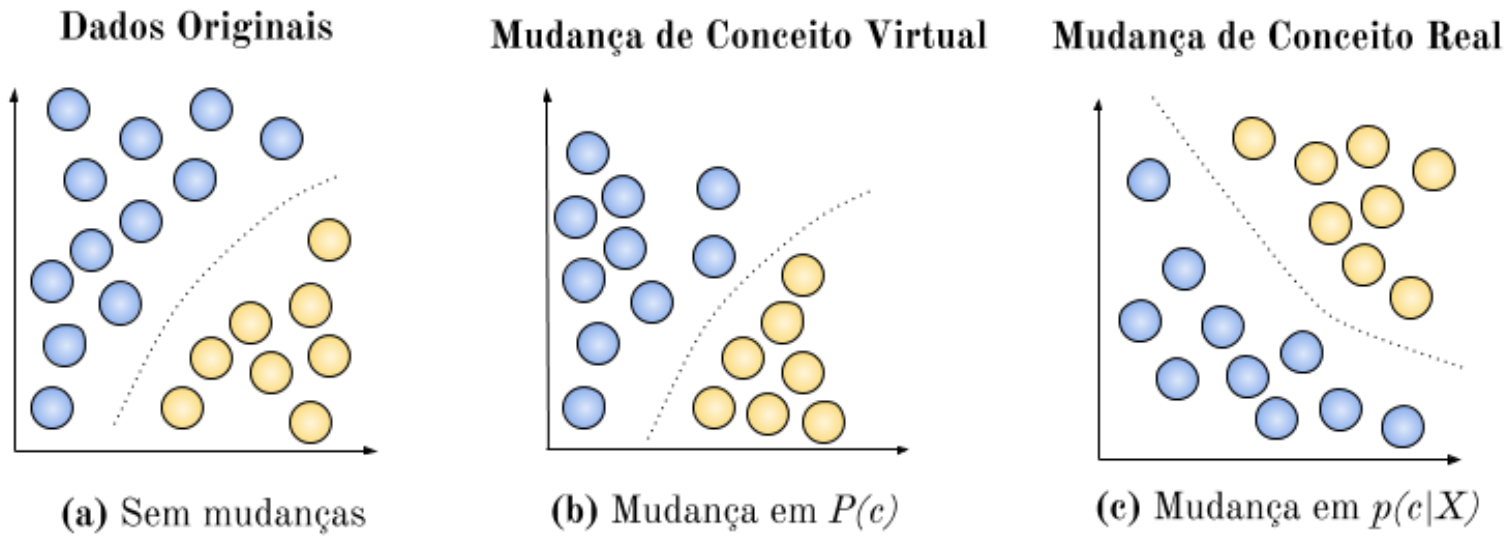


Figura 1: Mudança de Conceito Virtual vs. Mudança de Conceito Real

Mudança de Conceito

- As mudanças de conceito podem ocorrer de forma abrupta, gradual, incremental ou recorrente [38].

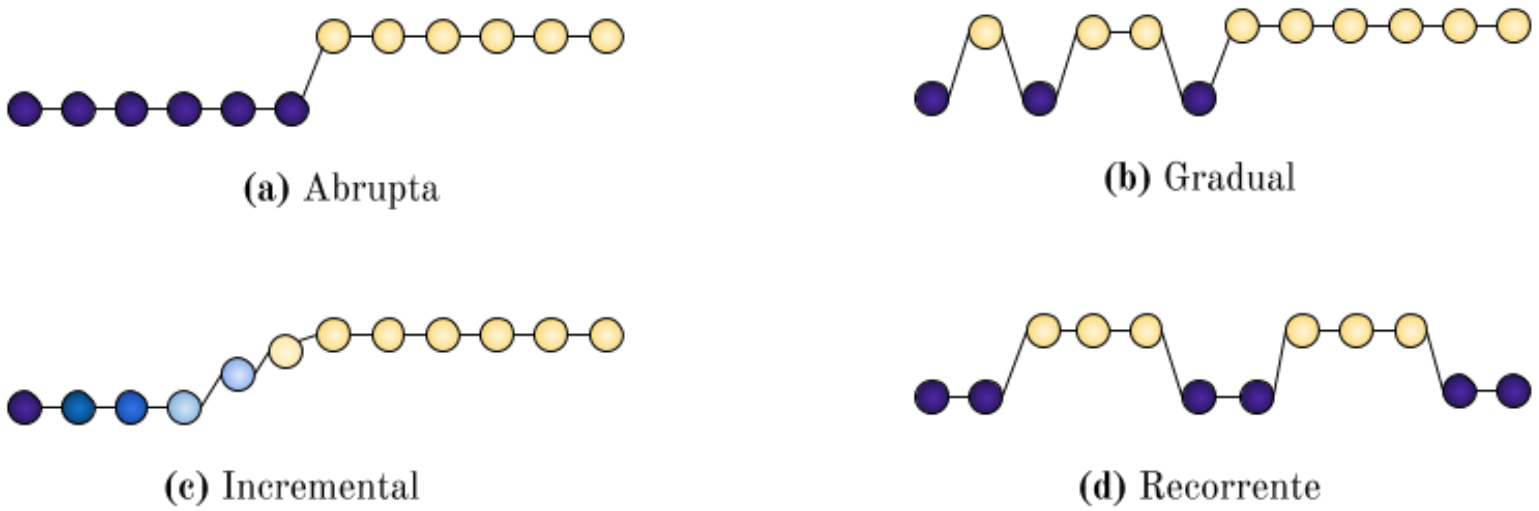


Figura 2: Padrões de ocorrência de Mudanças de Conceito

Mudança de Conceito

- O fenômeno mudança de conceito tem sido estudado em diferentes comunidades de pesquisa, sob diferentes nomeclaturas.

Área	Termos
Mineração de Dados	Mudança de Conceito
Aprendizado de Máquina	Mudança de Conceito, Mudança de Covariável
Computação Evolucionária	Ambiente Evolutivo, Ambiente em Mudança
IA e Robótica	Ambiente Dinâmico
Estatística, Séries Temporais	Não Estacionário
Recuperação de Informação	Evolução Temporal

Tabela 1: Terminologia - Mudança de Conceito [38]

Mudança de Conceito

- Outra fonte comum de equívocos são os termos relacionados aos tipos de técnica de detecção.

Termo	Descrição
Detecção de *Outliers*	Identificam padrões em desacordo com o esperado.
Detecção de Novidades	Identificam padrões ainda não observados.
Detecção de *Change Points*	Identificam variações abruptas de valor em séries temporais unidimensionais estacionárias.
Detecção de Mudança de Conceito	Identificam alterações, na distribuição ou no contexto, que possam afetar a acurácia do modelo em uso.

Tabela 2: Tipos de Técnicas de Detecção

Algoritmos para Detecção de Mudança de Conceito

- Os algoritmos para Detecção de Mudanças de Conceito se dividem em duas categorias, conforme a necessidade de rotulação dos dados [38]:

Algoritmos para Detecção de Mudança de Conceito

- Os algoritmos para Detecção de Mudanças de Conceito se dividem em duas categorias, conforme a necessidade de rotulação dos dados [38]:
  - Explícitos/Supervisionados: Dependem da rotulação dos dados, pois realizam a detecção a partir do monitoramento de medidas de performance como taxa de erro e acurácia.

Algoritmos para Detecção de Mudança de Conceito

- Os algoritmos para Detecção de Mudanças de Conceito se dividem em duas categorias, conforme a necessidade de rotulação dos dados [38]:
  - Explícitos/Supervisionados: Dependem da rotulação dos dados, pois realizam a detecção a partir do monitoramento de medidas de performance como taxa de erro e acurácia.
  - Implícitos/Não Supervisionados: Independem da rotulação dos dados, realizando a detecção através do monitoramento de características dos próprios dados ou de indicadores produzidos pelas técnicas de aprendizado aplicadas.

Algoritmos para Detecção de Mudança de Conceito

- Os algoritmos Explícitos/Supervisionados podem ser segmentados em três subcategorias [19]:

Algoritmos para Detecção de Mudança de Conceito

- Os algoritmos Explícitos/Supervisionados podem ser segmentados em três subcategorias [19]:
  - Métodos Baseados em Análise Sequencial: Monitoram indicadores de performance. Mudança é identificada quando este valor atinge um limiar pré-estabelecido. Exemplos: *Cumulative Sum (CUSUM)*, *PageHinkley (PH)* [28] e *Geometric Moving Average (GMA)* [32].

Algoritmos para Detecção de Mudança de Conceito

- Os algoritmos Explícitos/Supervisionados podem ser segmentados em três subcategorias [19]:
  - Métodos Baseados em Análise Sequencial: Monitoram indicadores de performance. Mudança é identificada quando este valor atinge um limiar pré-estabelecido. Exemplos: *Cumulative Sum (CUSUM)*, *PageHinkley (PH)* [28] e *Geometric Moving Average (GMA)* [32].
  - Abordagens baseadas em Estatística: Identificam mudanças de conceito através da análise de parâmetros estatísticos associados aos resultados das predições. Exemplos: *Drift Detection Method (DDM)* [17], *Early Drift Detection Method (EDDM)* [5], *Exponentially Weighted Moving Average (EWMA)* [33] e *Reactive Drift Detection Method (RDDM)* [10].

Algoritmos para Detecção de Mudança de Conceito

- Os algoritmos Explícitos/Supervisionados podem ser segmentados em três subcategorias [19]:
  - Métodos Baseados em Análise Sequencial: Monitoram indicadores de performance. Mudança é identificada quando este valor atinge um limiar pré-estabelecido. Exemplos: *Cumulative Sum (CUSUM)*, *PageHinkley (PH)* [28] e *Geometric Moving Average (GMA)* [32].
  - Abordagens baseadas em Estatística: Identificam mudanças de conceito através da análise de parâmetros estatísticos associados aos resultados das predições. Exemplos: *Drift Detection Method (DDM)* [17], *Early Drift Detection Method (EDDM)* [5], *Exponentially Weighted Moving Average (EWMA)* [33] e *Reactive Drift Detection Method (RDDM)* [10].
  - Métodos baseados em Janelas: Comparam, continuamente, uma janela fixa com o sumário dos exemplos observados e uma janela deslizante com os dados mais recentes. Exemplos: *Adaptive Windowing (ADWIN)* [6], *SeqDrift* [29], *HDDMA* e *HDDMW* [8].

Algoritmos para Detecção de Mudança de Conceito

- Os algoritmos Implícitos/Não Supervisionados também podem ser segmentados em três subcategorias [20]:

Algoritmos para Detecção de Mudança de Conceito

- Os algoritmos Implícitos/Não Supervisionados também podem ser segmentados em três subcategorias [20]:
  - Detecção de Novidade / Métodos de Agrupamento: Utilizam técnicas derivadas dos métodos de agrupamento e de detecção de *outliers*. Exemplos: *OLINDDA* [36], *MINAS* [16], *Woo* [34], *DETECTNOD* [21], *ECSMiner* [27] e *GC3* [35].

Algoritmos para Detecção de Mudança de Conceito

- Os algoritmos Implícitos/Não Supervisionados também podem ser segmentados em três subcategorias [20]:
  - Detecção de Novidade / Métodos de Agrupamento: Utilizam técnicas derivadas dos métodos de agrupamento e de detecção de *outliers*. Exemplos: *OLINDDA* [36], *MINAS* [16], *Woo* [34], *DETECTNOD* [21], *ECSMiner* [27] e *GC3* [35].
  - Monitoramento de distribuição multivariada: Monitoram diretamente a distribuição dos dados para cada atributo. Exemplos: *CoC* [25], *HDDDM* [11], *PCA-detect* [24].

Algoritmos para Detecção de Mudança de Conceito

- Os algoritmos Implícitos/Não Supervisionados também podem ser segmentados em três subcategorias [20]:
  - Detecção de Novidade / Métodos de Agrupamento: Utilizam técnicas derivadas dos métodos de agrupamento e de detecção de *outliers*. Exemplos: *OLINDDA* [36], *MINAS* [16], *Woo* [34], *DETECTNOD* [21], *ECSMiner* [27] e *GC3* [35].
  - Monitoramento de distribuição multivariada: Monitoram diretamente a distribuição dos dados para cada atributo. Exemplos: *CoC* [25], *HDDDM* [11], *PCA-detect* [24].
  - Monitoramento dependente de modelo: Requerem a aplicação de um algoritmo de classificação probabilístico, pois as mudanças de conceito são detectadas a partir do monitoramento da probabilidade a posteriori. Exemplos: *A-distance* [13], *CDBD* [26] e *Margin* [14].

Ferramenta: MOA

- Principal framework para mineração de dados em fluxos contínuos.
- Permite implementar e validar novos métodos de detecção de mudança de conceito de forma trivial.

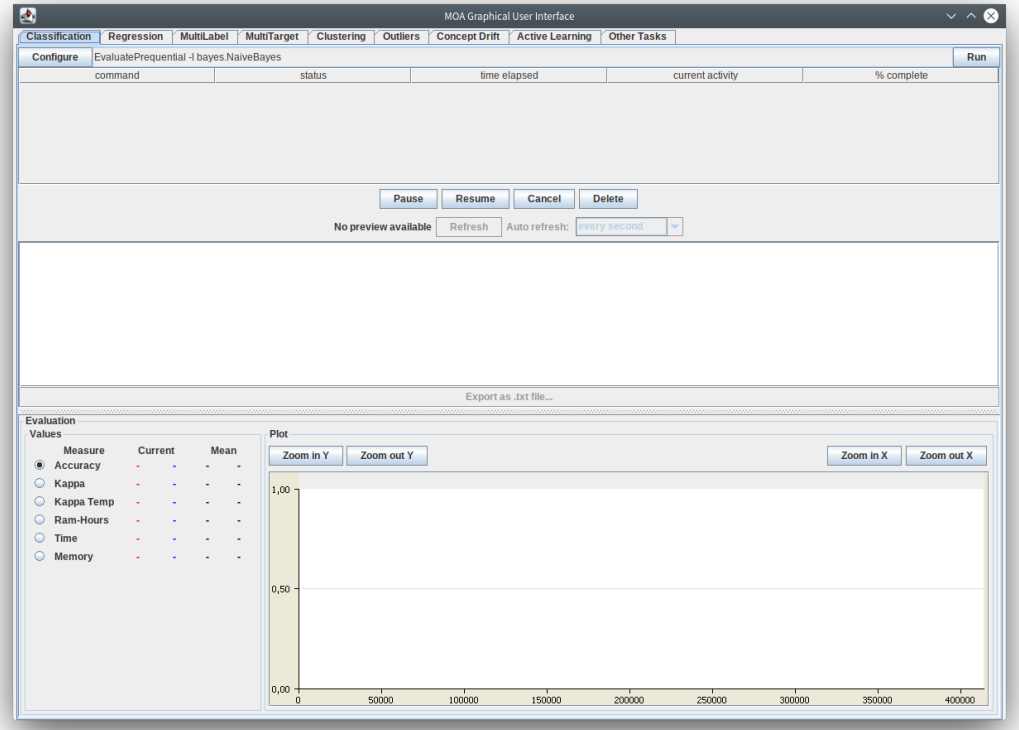


Figura 3: MOA - Tela Inicial

Ferramenta: Tornado

- Framework para avaliação de pares (classificador, detector de mudança de conceito).

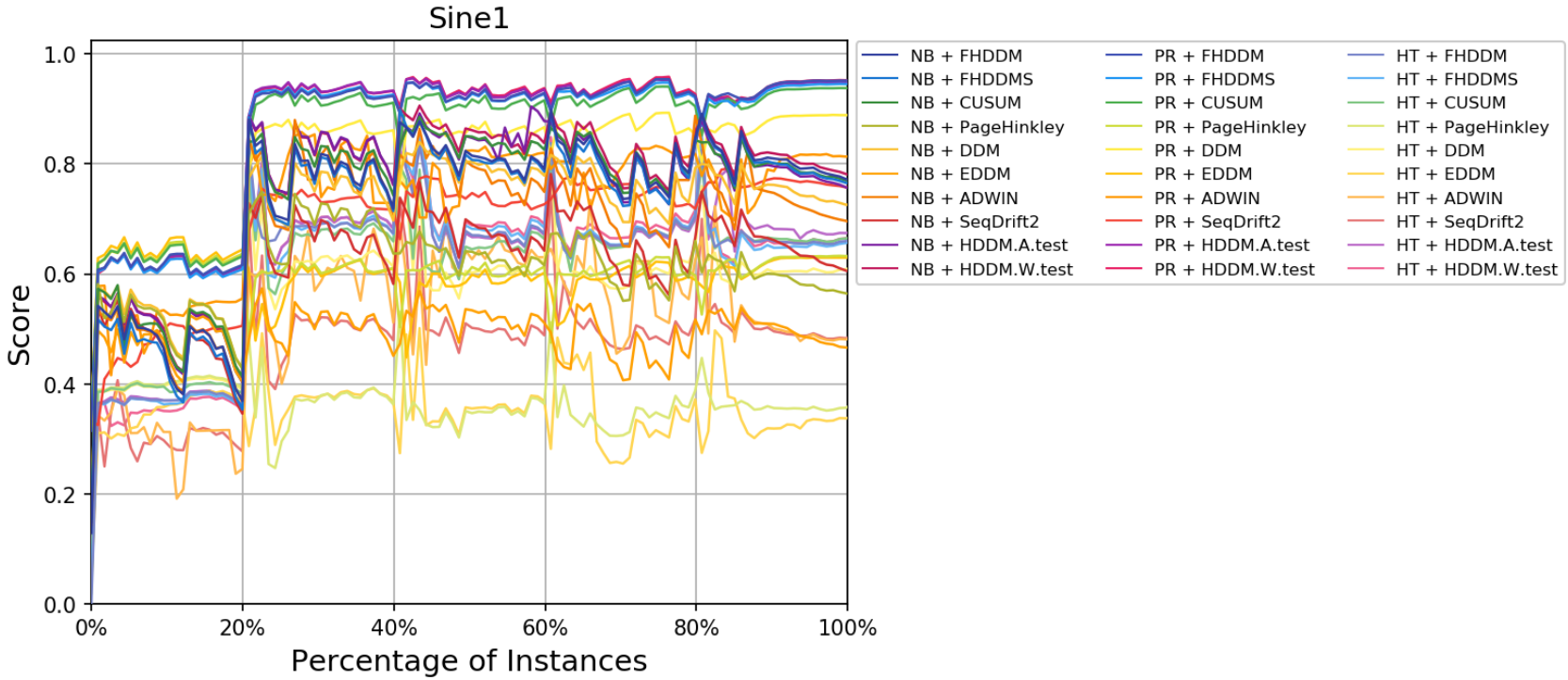


Figura 4: Tornado - Exemplo de resultado [30]

Redes de Função de Base Radial

- Redes de Função de Base Radial são redes neurais cujo principal diferencial é a forma de ativação, realizada através do cálculo da distância entre o dado e um centro definido [9].

Redes de Função de Base Radial

- Redes de Função de Base Radial são redes neurais cujo principal diferencial é a forma de ativação, realizada através do cálculo da distância entre o dado e um centro definido [9].
- A arquitetura de uma rede RBF, em sua forma mais básica, envolve três camadas:

Redes de Função de Base Radial

- Redes de Função de Base Radial são redes neurais cujo principal diferencial é a forma de ativação, realizada através do cálculo da distância entre o dado e um centro definido [9].
- A arquitetura de uma rede RBF, em sua forma mais básica, envolve três camadas:
  - Entrada: Recepciona os dados e encaminha para camada intermediária.

Redes de Função de Base Radial

- Redes de Função de Base Radial são redes neurais cujo principal diferencial é a forma de ativação, realizada através do cálculo da distância entre o dado e um centro definido [9].
- A arquitetura de uma rede RBF, em sua forma mais básica, envolve três camadas:
  - Entrada: Recepciona os dados e encaminha para camada intermediária.
  - Intermediária: Composta por funções de ativação de base radial que atuam como neurônios.

Redes de Função de Base Radial

- Redes de Função de Base Radial são redes neurais cujo principal diferencial é a forma de ativação, realizada através do cálculo da distância entre o dado e um centro definido [9].
- A arquitetura de uma rede RBF, em sua forma mais básica, envolve três camadas:
  - Entrada: Recepciona os dados e encaminha para camada intermediária.
  - Intermediária: Composta por funções de ativação de base radial que atuam como neurônios.
  - Saída: Pondera os resultados da camada intermediária, agregando-os linearmente para compor a resposta final da rede.

Redes de Função de Base Radial

- Redes de Função de Base Radial são redes neurais cujo principal diferencial é a forma de ativação, realizada através do cálculo da distância entre o dado e um centro definido [9].
- A arquitetura de uma rede RBF, em sua forma mais básica, envolve três camadas:
  - Entrada: Recepciona os dados e encaminha para camada intermediária.
  - Intermediária: Composta por funções de ativação de base radial que atuam como neurônios.
  - Saída: Pondera os resultados da camada intermediária, agregando-os linearmente para compor a resposta final da rede.
- Na literatura, as funções Gaussianas são as funções de ativação mais usuais em redes RBF.

Redes de Função de Base Radial

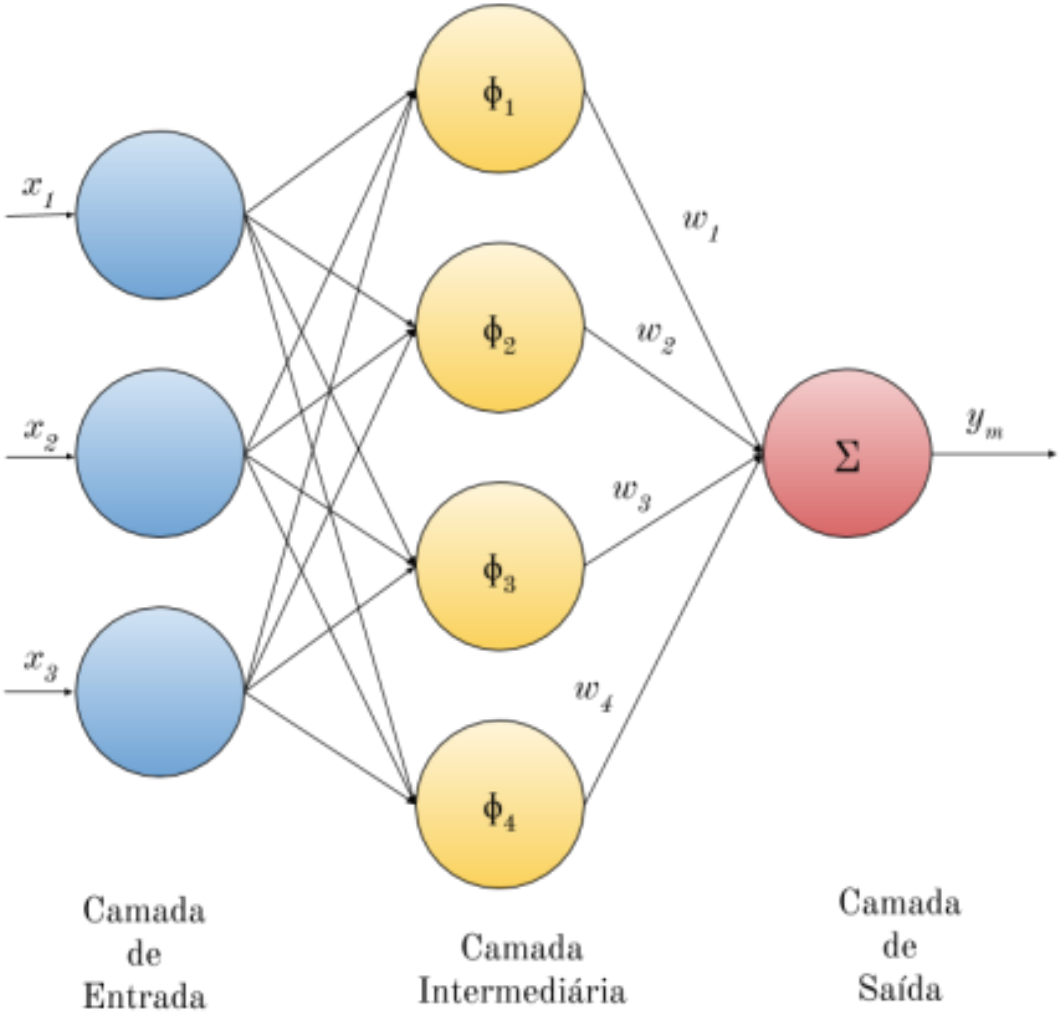


Figura 5: Arquitetura RBF

Trabalhos Relacionados

- Pesquisa na literatura em busca de trabalhos que propõem métodos para identificação de mudanças de conceito em fluxos contínuos de dados, de forma online e independente de rótulos.

Trabalhos Relacionados

- Pesquisa na literatura em busca de trabalhos que propõem métodos para identificação de mudanças de conceito em fluxos contínuos de dados, de forma online e independente de rótulos.
- Também foram estudadas técnicas que pudessem subsidiar o desenvolvimento de novos algoritmos que atendam a esses requisitos.

Trabalhos Relacionados

- Análise dos algoritmos Implícitos/Não Supervisionados da subcategoria Detecção de Novidade / Métodos de Agrupamento.

Trabalhos Relacionados

- Análise dos algoritmos Implícitos/Não Supervisionados da subcategoria Detecção de Novidade / Métodos de Agrupamento.
- Análise dos métodos para detecção de *Change Points* em séries temporais que atuam de forma online:
  - Modelos autoregressivos;
  - Séries com autosimilaridade e periodicidade.

Trabalhos Relacionados

- Análise dos algoritmos Implícitos/Não Supervisionados da subcategoria Detecção de Novidade / Métodos de Agrupamento.
- Análise dos métodos para detecção de *Change Points* em séries temporais que atuam de forma online:
  - Modelos autoregressivos;
  - Séries com autosimilaridade e periodicidade.
- Análise da aplicação de algoritmos de agrupamento estáveis.

Trabalhos Relacionados

- Análise dos algoritmos Implícitos/Não Supervisionados da subcategoria Detecção de Novidade / Métodos de Agrupamento.
- Análise dos métodos para detecção de *Change Points* em séries temporais que atuam de forma online:
  - Modelos autoregressivos;
  - Séries com autosimilaridade e periodicidade.
- Análise da aplicação de algoritmos de agrupamento estáveis.
- Identificação de lacuna de pesquisa.

Plano de Pesquisa

Descrição do Problema

- O método proposto, denominado *RBFDriftDetector*, utiliza as camadas inicial e intermediária da arquitetura básica de uma rede RBF, utilizando a Gaussiana como função de ativação.

Descrição do Problema

- O método proposto, denominado *RBFDriftDetector*, utiliza as camadas inicial e intermediária da arquitetura básica de uma rede RBF, utilizando a Gaussiana como função de ativação.
- Parâmetros: σ, responsável por limitar o raio da radial, e λ, que define um limiar para ativação de um centro.

Descrição do Problema

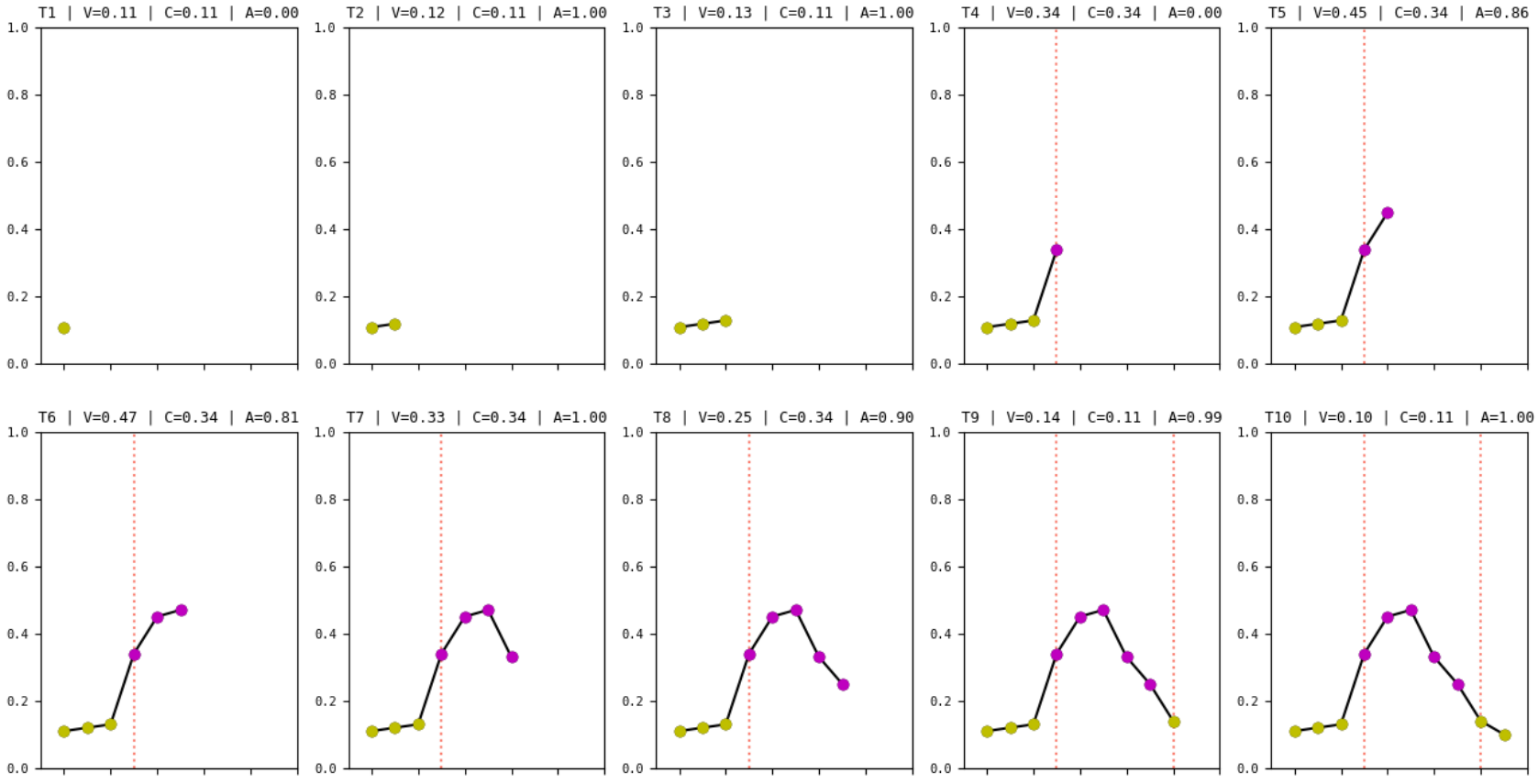


Figura 6: Exemplo de funcionamento do algoritmo

Atividades de Pesquisa

Atividades	Meses																							
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1-Disciplinas	X	X	X	X	X	X	X	X	X	X														
2-Revisão da Literatura							X	X	X	X	X	X												
3-Experimentos									X	X	X	X	•	•	•	•								
4-Análise dos Resultados											X	X				•	•	•						
5-Escrita da qualificação										X	X	X												
6-Estágio docente																•	•	•	•	•	•			
7-Pesquisa Orientada							X	X	X	X	X	X	•	•	•	•	•	•	•	•	•	•	•	•
8-Apresentação da qualificação													•											
9-Escrita de artigos																		•	•					
10-Escrita da dissertação										X	X	X					•	•	•	•	•	•	•	•
11- Defesa da dissertação																								•

Tabela 3: Cronograma de atividades

Experimentos Iniciais

Configuração dos Experimentos

- Conjuntos de dados sintéticos construídos através de especializações das classes geradoras do MOA.

Configuração dos Experimentos

- Conjuntos de dados sintéticos construídos através de especializações das classes geradoras do MOA.
- As classes geradoras originais foram alteradas para permitir a inclusão de ruídos com distribuição uniforme e corrigir limitações.

Configuração dos Experimentos

- Conjuntos de dados sintéticos construídos através de especializações das classes geradoras do MOA.
- As classes geradoras originais foram alteradas para permitir a inclusão de ruídos com distribuição uniforme e corrigir limitações.
- Cada conjunto de dados é composto por 2.500 observações, com valores entre 0 e 1 e conceitos compostos por 400 registros.

Configuração dos Experimentos

Conjunto	Classe utilizada	Qtd. Observações	Tamanho Conceito	Ruído
Mudanças Abruptas	*AbruptChangeGenerator*	2.500	400	$[-0.1, 0.2]$
Mudanças Graduais	*GradualChangeGenerator*	2.500	400	-
Sem Mudanças	*NoChangeGenerator*	2.500	-	$[-0.1, 0.2]$

Tabela 4: Conjuntos de dados produzidos

Configuração dos Experimentos

- Parametrização da classe de avaliação utilizada.

Parâmetro	Valor	Observação
learner	ChangeDetectorLearner	O algoritmo de detecção de mudanças de conceito a ser testado é definido no atributo *driftDetectionMethod* da classe *ChangeDetectorLearner*.
stream	ARFFFileStream	Caminho para um dos arquivos *ARFF* descrito na seção anterior. O atributo *classIndex* deve ser definido como 0, pois não existem rótulos nestes conjuntos de dados.
instanceLimit	-1	Desabilita o limite de instâncias a serem processadas.
timeLimit	-1	Desabilita o limite de tempo de execução.
sampleFrequency	1	Uma linha de resultado do avaliador deve ser gerada para cada instância processada.

Tabela 5: Configuração da classe *BasicConceptDriftPerformanceEvaluator*

Configuração dos Experimentos

- Indicadores de avaliação analisados.

Indicador	Observação
Tempo de Processamento	Tempo médio (seg.) de processamento por instância.
Mudanças Existentes	Quantidade de mudanças existentes.
Mudanças Detectadas	Quantidade de mudanças detectadas corretamente.
Falso-positivos	Quantidade de mudanças detecatadas erroneamente.
Atraso de Detecção	Quantidade média de instâncias até a detecção.

Tabela 6: Indicadores analisados

Configuração dos Experimentos

- Algoritmos comparados e os parâmetros utilizados.

Algoritmo	Parâmetros
RBFDriftDetector	$\sigma = 2;\ \lambda = 0.5$
CUSUM	$\mathit{MinNumInstances} = 30; \delta = 0.005; \lambda = 50$
PageHinkley	$\mathit{MinNumInstances} = 30; \delta = 0.005; \lambda = 50; \alpha = 1$
ADWIN	$\delta = 0.002$

Tabela 7: Parâmetros utilizados para cada algoritmo

Pettitt

- Implementação do teste não paramétrico de Pettitt como método para detecção de mudanças de conceito [31].

Conjunto de Dados	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
Sem mudanças	3.19	0	0	4	–
Mudanças Abruptas	2.09	6	3	2	132
Mudanças Incrementais	1.48	6	4	2	133

Tabela 8: Resultados - Método de Pettitt

Pettitt

- Implementação do teste não paramétrico de Pettitt como método para detecção de mudanças de conceito [31].

Conjunto de Dados	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
Sem mudanças	3.19	0	0	4	–
Mudanças Abruptas	2.09	6	3	2	132
Mudanças Incrementais	1.48	6	4	2	133

Tabela 8: Resultados - Método de Pettitt

- Propenso à produção de falso-positivos e computacionalmente ineficiente.

Experimento 1 - Fluxo sem mudanças de conceito

- Experimento realizado utilizando o conjunto de dados **sem** mudanças de conceito, visando avaliar a tendência de produção de falso-positivos.

Algoritmo	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
RBFDriftDetector	0.22	0	0	0	–
CUSUM	0.31	0	0	0	–
PageHinkley	0.24	0	0	0	–
ADWIN	0.21	0	0	0	–

Tabela 9: Experimento 1 - Fluxo sem mudanças de conceito

Experimento 1 - Fluxo sem mudanças de conceito

- Experimento realizado utilizando o conjunto de dados **sem mudanças de conceito**, visando avaliar a tendência de produção de falso-positivos.

Algoritmo	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
RBFDriftDetector	0.22	0	0	0	–
CUSUM	0.31	0	0	0	–
PageHinkley	0.24	0	0	0	–
ADWIN	0.21	0	0	0	–

Tabela 9: Experimento 1 - Fluxo sem mudanças de conceito

- Todos algoritmos testados demonstraram tolerância a ruídos e baixa propensão a identificar falso-positivos.

Experimento 1 - Fluxo sem mudanças de conceito

- Experimento realizado utilizando o conjunto de dados **sem** mudanças de conceito, visando avaliar a tendência de produção de falso-positivos.

Algoritmo	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
RBFDriftDetector	0.22	0	0	0	–
CUSUM	0.31	0	0	0	–
PageHinkley	0.24	0	0	0	–
ADWIN	0.21	0	0	0	–

Tabela 9: Experimento 1 - Fluxo sem mudanças de conceito

- Todos algoritmos testados demonstraram tolerância a ruídos e baixa propensão a identificar falso-positivos.
- O algoritmo proposto obteve a segunda melhor performance, superado pelo ADWIN por uma pequena margem.

Experimento 1 - Fluxo sem mudanças de conceito

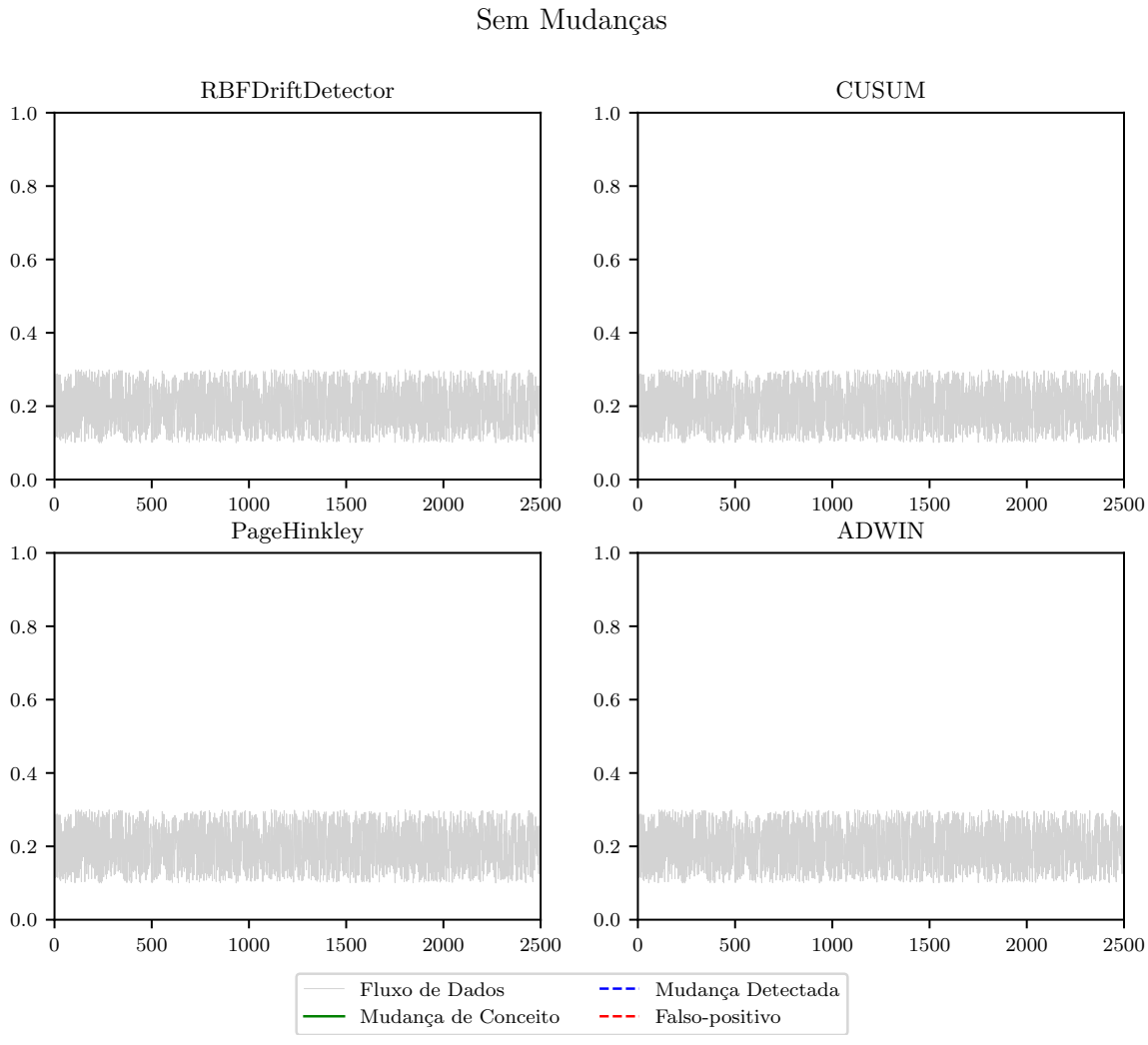


Figura 7: Representação Gráfica - Sem mudanças de conceito

Experimento 2 - Fluxo com mudanças de conceito abruptas

- Experimento realizado utilizando o conjunto de dados com mudanças de conceito **abruptas**.

Algoritmo	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
RBFDriftDetector	0.23	6	6	0	1
CUSUM	0.29	6	3	0	68
PageHinkley	0.22	6	1	0	17
ADWIN	0.21	6	6	2046	9

Tabela 10: Experimento 2 - Fluxo com mudanças de conceito abruptas

Experimento 2 - Fluxo com mudanças de conceito abruptas

- Experimento realizado utilizando o conjunto de dados com mudanças de conceito **abruptas**.

Algoritmo	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
RBFDriftDetector	0.23	6	6	0	1
CUSUM	0.29	6	3	0	68
PageHinkley	0.22	6	1	0	17
ADWIN	0.21	6	6	2046	9

Tabela 10: Experimento 2 - Fluxo com mudanças de conceito abruptas

- Método proposto identificou todas mudanças, sem falso-positivos e com menor atraso de detecção.

Experimento 2 - Fluxo com mudanças de conceito abruptas

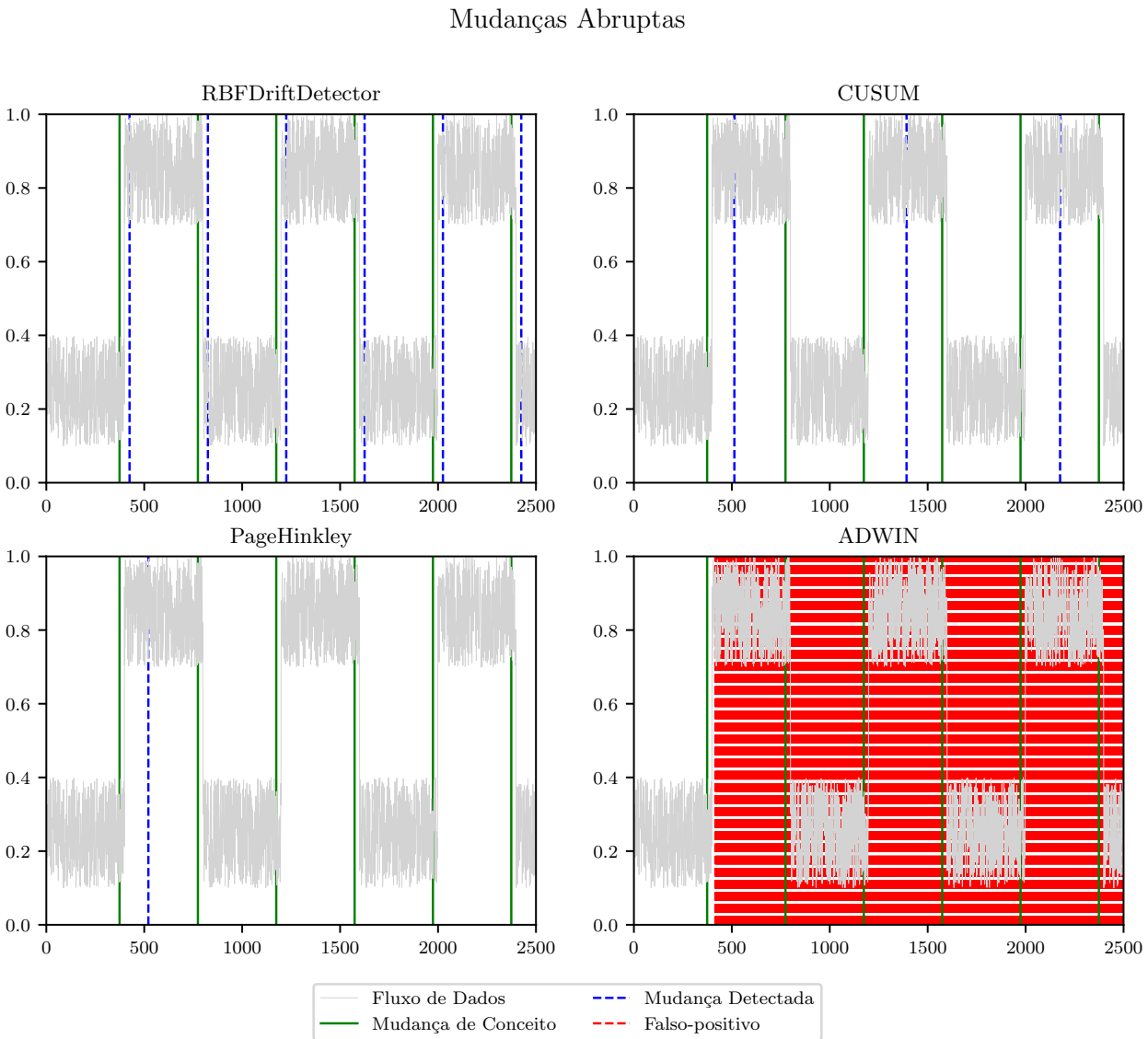


Figura 8: Representação Gráfica - Mudanças Abruptas

Experimento 3 - Fluxo com mudanças de conceito graduais

- Experimento realizado utilizando o conjunto de dados com mudanças de conceito **graduais**.

Algoritmo	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
RBFDriftDetector	0.24	6	5	1	171
CUSUM	0.65	6	3	0	32
PageHinkley	0.26	6	1	0	4
ADWIN	0.27	6	6	2238	1

Tabela 11: Experimento 3 - Fluxo com mudanças de conceito graduais

Experimento 3 - Fluxo com mudanças de conceito graduais

- Experimento realizado utilizando o conjunto de dados com mudanças de conceito **graduais**.

Algoritmo	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
RBFDriftDetector	0.24	6	5	1	171
CUSUM	0.65	6	3	0	32
PageHinkley	0.26	6	1	0	4
ADWIN	0.27	6	6	2238	1

Tabela 11: Experimento 3 - Fluxo com mudanças de conceito graduais

- Método proposto identificou 5 das 6 mudanças existentes e sinalizou um falso-positivo.

Experimento 3 - Fluxo com mudanças de conceito graduais

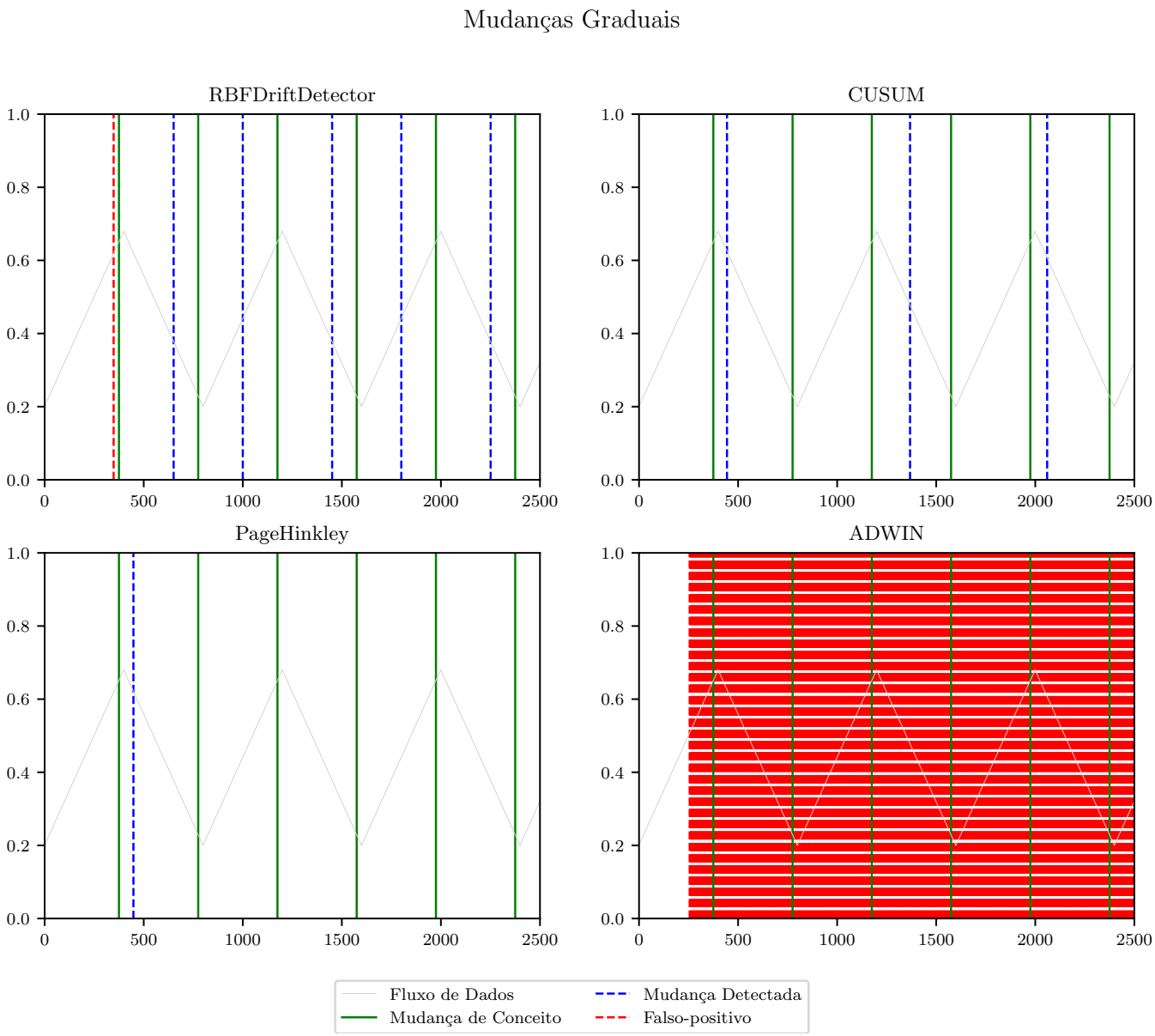
- Experimento realizado utilizando o conjunto de dados com mudanças de conceito **graduais**.

Algoritmo	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
RBFDriftDetector	0.24	6	5	1	171
CUSUM	0.65	6	3	0	32
PageHinkley	0.26	6	1	0	4
ADWIN	0.27	6	6	2238	1

Tabela 11: Experimento 3 - Fluxo com mudanças de conceito graduais

- Método proposto identificou 5 das 6 mudanças existentes e sinalizou um falso-positivo.
- Obteve a melhor acurácia, apesar de apresentar a maior taxa de atraso.

Experimento 3 - Fluxo com mudanças de conceito graduais



Conclusão

Conclusão

- Algoritmo apresenta bom desempenho e acurácia.

Conclusão

- Algoritmo apresenta bom desempenho e acurácia.
- Método proposto pode ser aprimorado para detecção de mudanças graduais.

Conclusão

- Algoritmo apresenta bom desempenho e acurácia.
- Método proposto pode ser aprimorado para detecção de mudanças graduais.
- Os resultados obtidos, apesar de preliminares, mostraram-se promissores, indicando que o tema de pesquisa deve continuar a ser investigado.

Trabalhos Futuros

- Realizar experimentos com fluxos recorrentes e incrementais.
- Integrar uma cadeia de Markov ao método proposto, permitindo a análise do comportamento das mudanças e o aprimoramento da acurácia das detecções.
- Implementar os algoritmos OLINDDA, MINAS e DETECNOD para serem comparados.
- Implementar e validar o algoritmo no framework Tornado.
- Alterar o algoritmo para permitir que o centro se desloque dentro do grupo formado.

Referências i


M. R. Ackermann, M. Märtens, C. Raupach, K. Swierkot, C. Lammersen, and C. Sohler.
Streamkm++: A clustering algorithm for data streams.
J. Exp. Algorithmics, 17:2.4:2.1–2.4:2.30, May 2012.


C. C. Aggarwal.
Data Streams: Models and Algorithms (Advances in Database Systems).
Springer-Verlag, Berlin, Heidelberg, 2006.


C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu.
A framework for clustering evolving data streams.
In *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29*, VLDB '03, pages 81–92. VLDB Endowment, 2003.

Referências ii


C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu.
On demand classification of data streams.
In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 503–508, New York, NY, USA, 2004. ACM.


M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldá, and R. Morales-Bueno.
Early drift detection method.
In *In Fourth International Workshop on Knowledge Discovery from Data Streams*, 2006.


A. Bifet and R. Gavaldà.
Learning from time-changing data with adaptive windowing.
In *SDM*, pages 443–448. SIAM, 2007.

Referências iii


A. Bifet, B. Pfahringer, J. Read, and G. Holmes.
Efficient data stream classification via probabilistic adaptive windows.
In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 801–806, New York, NY, USA, 2013. ACM.


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.


A. Braga, A. C. Carvalho, and T. B. Ludermir.
Redes Neurais Artificiais: Teoria e aplicações, volume 2.
LTC Editora, 2007.

Referências iv


R. S. M. de Barros, D. R. de Lima Cabral, P. M. G. Jr., and S. G. T. de Carvalho Santos.
RDDM: reactive drift detection method.
Expert Syst. Appl., 90:344–355, 2017.


G. Ditzler and R. Polikar.
Hellinger distance based drift detection for nonstationary environments.
In *2011 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*, pages 41–48, April 2011.


P. Domingos and G. Hulten.
Mining high-speed data streams.
In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 71–80, New York, NY, USA, 2000. ACM.

Referências v


M. Dredze, T. Oates, and C. Piatko.
We're not in kansas anymore: Detecting domain changes in streams.
pages 585–595, 2010.
cited By 13.


A. Dries and U. Rückert.
Adaptive concept drift detection.
Statistical Analysis and Data Mining: The ASA Data Science Journal, 2(5-6):311–327, 2009.


R. O. Duda, P. E. Hart, and D. G. Stork.
Pattern Classification (2Nd Edition).
Wiley-Interscience, New York, NY, USA, 2000.

Referências vi


E. R. Faria, J. a. Gama, and A. C. P. L. F. Carvalho.
Novelty detection algorithm for data streams multi-class problems.
In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 795–800, New York, NY, USA, 2013. ACM.


J. Gama, P. Medas, G. Castillo, and P. P. Rodrigues.
Learning with drift detection.
In A. L. C. Bazzan and S. Labidi, editors, *SBIA*, volume 3171 of *Lecture Notes in Computer Science*, pages 286–295. Springer, 2004.

Referências vii


J. a. Gama, R. Rocha, and P. Medas.
Accurate decision trees for mining high-speed data streams.
In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 523–528, New York, NY, USA, 2003. ACM.


J. a. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia.
A survey on concept drift adaptation.
ACM Comput. Surv., 46(4):44:1–44:37, Mar. 2014.


P. M. Gonçalves, S. G. de Carvalho Santos, R. S. Barros, and D. C. Vieira.
A comparative study on concept drift detectors.
Expert Systems with Applications, 41(18):8144 – 8156, 2014.

Referências viii


M. Z. Hayat and M. R. Hashemi.
A dct based approach for detecting novelty and concept drift in data streams.
In *2010 International Conference of Soft Computing and Pattern Recognition*, pages 373–378, Dec 2010.


J. Z. Kolter and M. A. Maloof.
Dynamic weighted majority: An ensemble method for drifting concepts.
J. Mach. Learn. Res., 8:2755–2790, Dec. 2007.


P. Kranen, I. Assent, C. Baldauf, and T. Seidl.
The clustree: Indexing micro-clusters for anytime stream mining.
Knowl. Inf. Syst., 29(2):249–272, Nov. 2011.

Referências ix


L. Kuncheva.
Classifier ensembles for detecting concept change in streaming data: Overview and perspectives.
Proc. Eur. Conf. Artif. Intell., pages 5–10, 2008.
cited By 70.


J. Lee and F. Magoulès.
Detection of concept drift for learning from stream data.
In *2012 IEEE 14th International Conference on High Performance Computing and Communication 2012 IEEE 9th International Conference on Embedded Software and Systems*, pages 241–245, June 2012.


P. Lindstrom, B. Mac Namee, and S. J. Delany.
Drift detection using uncertainty distribution divergence.
Evolving Systems, 4(1):13–25, Mar 2013.

Referências x


M. Masud, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham.
Classification and novel class detection in concept-drifting data streams under time constraints.
IEEE Trans. on Knowl. and Data Eng., 23(6):859–874, June 2011.


E. S. Page.
Continuous Inspection Schemes.
Biometrika, 41(1/2):100–115, 1954.


R. Pears, S. Sakthithasan, and Y. S. Koh.
Detecting concept change in dynamic data streams - A sequential approach based on reservoir sampling.
Machine Learning, 97(3):259–293, 2014.


A. Pesaranghader.
A reservoir of adaptive algorithms for online learning from evolving data streams, 2018.

A. Pettitt.
A non-parametric approach to the change-point problem.
Journal of the Royal Statistical Society. Series C. Applied Statistics, 28, 01 1979.

S. W. Roberts.
Control chart tests based on geometric moving averages.
Technometrics, 42(1):97–101, Feb. 2000.

G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand.
Exponentially weighted moving average charts for detecting concept drift.
Pattern Recogn. Lett., 33(2):191–198, Jan. 2012.


J. W. Ryu, M. M. Kantardzic, M.-W. Kim, and A. Ra Khil.
An efficient method of building an ensemble of classifiers in streaming data.
In S. Srinivasa and V. Bhatnagar, editors, *Big Data Analytics*, pages 122–133, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.


T. S. Sethi, M. Kantardzic, and H. Hu.
A grid density based framework for classifying streaming data in the presence of concept drift.
Journal of Intelligent Information Systems, 46(1):179–211, Feb 2016.


E. J. Spinosa, A. P. de Leon F. de Carvalho, and J. a. Gama.
Olindda: A cluster-based approach for detecting novelty and concept drift in data streams.
In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, pages 448–452, New York, NY, USA, 2007. ACM.


H. Wang, W. Fan, P. S. Yu, and J. Han.
Mining concept-drifting data streams using ensemble classifiers.
In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 226–235, New York, NY, USA, 2003. ACM.


I. Zliobaite.
Learning under concept drift: an overview.
CoRR, abs/1010.4784, 2010.


M. Zwolenski and L. Weatherill.
The digital universe rich data and the increasing value of the internet of things.
Australian Journal of Telecommunications and the Digital Economy, 2, 10 2014.