



Universidade Federal da Bahia
Instituto de Matemática

Programa de Pós-Graduação em Ciência da Computação

**APLICANDO REDES DE FUNÇÃO DE BASE
RADIAL PARA DETECÇÃO DE MUDANÇAS
DE CONCEITO EM FLUXOS CONTÍNUOS
DE DADOS**

Ruivaldo Azevedo Lobão Neto

QUALIFICAÇÃO DE MESTRADO

Salvador
03 de Abril de 2019

RUIVALDO AZEVEDO LOBÃO NETO

**APLICANDO REDES DE FUNÇÃO DE BASE RADIAL PARA
DETECÇÃO DE MUDANÇAS DE CONCEITO EM FLUXOS
CONTÍNUOS DE DADOS**

Esta Qualificação de Mestrado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Ricardo Araújo Rios

Salvador
03 de Abril de 2019

TERMO DE APROVAÇÃO

RUIVALDO AZEVEDO LOBÃO NETO

APLICANDO REDES DE FUNÇÃO DE BASE RADIAL PARA DETECÇÃO DE MUDANÇAS DE CONCEITO EM FLUXOS CONTÍNUOS DE DADOS

Esta Qualificação de Mestrado foi julgada adequada à obtenção do título de Mestre em Ciência da Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia.

Salvador, 03 de Abril de 2019

Prof. Dr. Ricardo Araújo Rios
UFBA

RESUMO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Palavras-chave: Aprendizado de Máquina, Fluxos Contínuos de Dados, Mudanças de Conceito, Redes de Função de Base Radial, Não supervisionado

ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Keywords: Machine Learning, Data Streams, Concept Drift, Radial Basis Function Networks, RBF Network, Unlabeled

SUMÁRIO

Capítulo 1—Introdução	1
1.1 Contexto e Motivação	1
1.2 Hipótese e Objetivo	2
Capítulo 2—Revisão Bibliográfica	5
2.1 Considerações Iniciais	5
2.2 Fluxos Contínuos de Dados e Aprendizado de Máquina	5
2.3 Mudança de Conceito	7
2.3.1 Terminologia	9
2.3.2 Algoritmos para Detecção de Mudança de Conceito	10
2.3.3 Ferramentas	12
2.3.4 MOA	12
2.3.5 Tornado	14
2.4 Redes de Função de Base Radial	16
2.5 Trabalhos Relacionados	18
2.6 Considerações Finais	19
Capítulo 3—Plano de Pesquisa	21
3.1 Considerações Iniciais	21
3.2 Descrição do Problema	21
3.3 Atividades de Pesquisa	24
3.4 Considerações Finais	25
Capítulo 4—Experimentos Iniciais	27
4.1 Considerações Iniciais	27
4.2 Configuração dos Experimentos	27
4.3 Critérios de avaliação	29
4.4 RBFDriftDetector	31
4.5 Considerações Finais	31

LISTA DE FIGURAS

2.1	Mudança de Conceito Virtual vs. Mudança de Conceito Real	8
2.2	Padrões de ocorrência de Mudanças de Conceito	8
2.3	MOA - Tela Inicial	13
2.4	MOA - Configuração detector	14
2.5	Framework Tornado (PESARANGHADER, 2018).	15
2.6	Tornado - Exemplo de resultado (PESARANGHADER, 2018)	16
2.7	Arquitetura RBF	17
3.1	Exemplo de funcionamento do algoritmo	23
4.1	Parametrização da classe <i>AbruptChangeGenerator</i>	28
4.2	Parametrização da classe <i>GradualChangeGenerator</i>	29

LISTA DE TABELAS

2.1	Terminologia - Mudança de Conceito (ZLIOBAITE, 2010)	9
2.2	Resumo - Algoritmos de detecção (SETHI; KANTARDZIC, 2017)	12
3.1	Cronograma de atividades	24
4.1	Configuração da tarefa <i>BasicConceptDriftPerformanceEvaluator</i>	30
4.2	Indicadores analisados	30

LISTA DE ALGORITMOS

1	RBFDRIFTDETECTOR	22
---	----------------------------	----

INTRODUÇÃO

1.1 CONTEXTO E MOTIVAÇÃO

Nos últimos anos, o volume de dados produzidos por sistemas computacionais tem crescido de forma acentuada. Esse crescimento foi favorecido por avanços tecnológicos recentes, como a pervasividade dos dispositivos móveis, a popularização das redes sociais e a expansão da internet das coisas (COHEN et al., 2009). A dimensão desse aumento é verificada em (ZWOLENSKI; WEATHERILL, 2014), no qual se estima que, entre os anos de 2014 e 2020, a quantidade de informações produzidas anualmente irá aumentar de 4,4 zettabytes (trilhões de gigabytes) para 44 zettabytes.

Parte significativa dessas informações é produzida na forma de sequências ininterruptas e potencialmente infinitas (AGGARWAL, 2006). Na literatura, sequências com essas características são denominadas Fluxos Contínuos de Dados (FCDs) e estão presentes em diversos domínios de aplicação, por exemplo: monitoramento do mercado financeiro (ZHOU et al., 2015), acompanhamento de tráfego rodoviário (WANG et al., 2015), gerenciamento de redes de telecomunicação (DELATTRE; IMBERT, 2015), análise de sentimento em tempo real (KRANJC et al., 2015) e sistemas de prevenção e identificação de intrusos (KENKRE; PAI; COLACO, 2015).

Para extrair informações úteis dessa grande quantidade de dados, pesquisadores têm aplicado técnicas da área de Aprendizado de Máquina (AM), a qual estuda algoritmos que melhoram seu desempenho conforme ganham experiência (MITCHELL, 1997). Entretanto, as estratégias tradicionais de aprendizado de máquina têm aplicação limitada para contextos com fluxos contínuos de dados, pois nesses cenários os algoritmos devem atender a severas restrições de tempo de execução e de uso dos recursos computacionais (BIFET; KIRKBY, 2009).

Além dessas limitações, as técnicas de aprendizado de máquina, quando aplicadas em contextos com fluxos contínuos, também devem lidar com variações na distribuição dos dados ou no contexto do processo gerador. Essas alterações são denominadas Mudanças de Conceito (GAMA, 2010) e a sua ocorrência pode impactar a acurácia do algoritmo.

Inicialmente, a atualização periódica do modelo foi utilizada como estratégia para evitar a perda de acurácia causada por tais mudanças. Contudo, esta solução é pouco sofisticada e computacionalmente custosa. Diante disso, pesquisadores propuseram técnicas de detecção de mudanças de conceito baseadas em monitoramento (GAMA et al., 2014). Estes métodos identificam o momento exato da mudança, permitindo que o modelo de decisão seja atualizado somente quando necessário. Exemplos de algoritmos baseados nesta abordagem, incluem: DDM (GAMA et al., 2004), EDDM (BAENA-GARCÍA et al., 2006), ADWIN (BIFET; GAVALDÀ, 2007), ECDD (ROSS et al., 2012), PL (Bach; Maloof, 2008), FCWM (SEBASTIÃO et al., 2010) e STEPDM (NISHIDA; YAMAUCHI, 2007).

Entretanto, as técnicas baseadas em monitoramento necessitam que o rótulo correto de cada exemplo esteja disponível. Em muitos cenários, o tempo ou o custo para obter esses rótulos é proibitivo (AGGARWAL, 2006). Consequentemente, foram desenvolvidos novos algoritmos independentes de rótulos. Nestes métodos, a detecção se baseia na identificação de exemplos que não se enquadram na estrutura dos dados (SPINOSA; CARVALHO; GAMA, 2007). Essa análise é implementada com base em técnicas de agrupamento, detecção de *outliers* e medidas de dissimilaridade (RYU et al., 2012). Os seguintes algoritmos são exemplos desta metodologia: OLINDDA (SPINOSA; CARVALHO; GAMA, 2007), MINAS (FARIA; GAMA; CARVALHO, 2013), ECSMiner (MASUD et al., 2011) e GC3 (SETHI; KANTARDZIC; HU, 2016).

Todavia, segundo Aggarwal (2006), as técnicas de detecção de mudanças de conceito propostas apresentam limitações ao serem aplicadas em cenários com fluxos contínuos de dados. Os algoritmos dependentes de rótulo se tornam inviáveis, por causa do custo e do tempo necessário para obter os rótulos corretos. Enquanto as técnicas independentes têm dificuldade em atender as severas restrições de tempo de execução e de uso dos recursos computacionais desses cenários.

Visando resolver essas limitações, este projeto de mestrado discute uma abordagem baseada em redes de função de base radial para detecção de mudanças de conceito em fluxos contínuos de dados. A metodologia proposta se diferencia por detectar as mudanças em tempo de execução, de forma computacionalmente eficiente e independente de rótulos.

1.2 HIPÓTESE E OBJETIVO

Com base nas observações citadas anteriormente, a seguinte hipótese foi formulada:

“ A aplicação de redes de função de base radial a fluxos contínuos de dados permite a detecção de mudanças de conceito em tempo de execução, de forma computacionalmente eficiente e independente de rótulos. ”

Assim, o objetivo deste trabalho de mestrado será a validação desta hipótese. Para atingir este objetivo, será desenvolvido um método para detecção de mudanças de conceito baseado em redes de função de base radial. A técnica proposta será validada através de comparações com o estado da arte. Os dados utilizados durante a validação serão divididos em dois conjuntos. Um conjunto formado por dados sintéticos, que permitirão uma análise detalhada da abordagem, uma vez que as características e os comportamentos

dos fluxos serão conhecidos. O outro conjunto será composto por dados obtidos a partir de sistemas computacionais utilizados na indústria, visando apresentar uma aplicação prática para a solução proposta.

O restante deste projeto está organizado conforme a seguinte estrutura: O **Capítulo 2** apresenta uma revisão bibliográfica dos principais conceitos utilizados neste trabalho como, por exemplo, fluxos contínuos de dados e aprendizado de máquina, mudança de conceito e redes de função de base radial; No **Capítulo 3** o plano de pesquisa é detalhado, identificando a metodologia que será aplicada na pesquisa e o cronograma de atividades. Por fim, o **Capítulo 4** apresenta um conjunto de experimentos preliminares e a análise dos resultados obtidos.

REVISÃO BIBLIOGRÁFICA

2.1 CONSIDERAÇÕES INICIAIS

Este capítulo apresenta uma discussão geral sobre os principais conceitos utilizados neste projeto. Inicialmente, será abordada a relação entre fluxos contínuos de dados e técnicas de aprendizado de máquina. Em seguida, o fenômeno mudança de conceito e seus métodos de detecção são discutidos. Posteriormente, as redes de função de base radial são detalhadas. Por fim, são apresentados os trabalhos relacionados encontrados na literatura.

2.2 FLUXOS CONTÍNUOS DE DADOS E APRENDIZADO DE MÁQUINA

Fluxos Contínuos de Dados (FCDs) podem ser definidos como sequências ininterruptas e potencialmente infinitas de eventos (AGGARWAL, 2006). Nestes fluxos, os eventos ocorrem em alta frequência, sendo necessário processá-los em tempo real. Além disso, por serem de tamanho potencialmente ilimitado, não é possível armazená-los de forma permanente em memória.

As características dos fluxos contínuos de dados implicam nas seguintes restrições aos algoritmos que os processam (BIFET; KIRKBY, 2009):

1. É impossível armazenar todos os dados do fluxo. Somente uma pequena parcela pode ser processada e armazenada, enquanto o restante é descartado;
2. A velocidade de chegada dos eventos no fluxo exige que os elementos sejam processados em tempo real;
3. A distribuição dos dados pode mudar com o tempo. Assim, os dados do passado podem se tornar irrelevantes ou mesmo prejudiciais para a descrição dos conceitos atuais.

A área de Aprendizado de Máquina (AM) estuda algoritmos que melhoram o seu desempenho conforme ganham experiência (MITCHELL, 1997). Esses algoritmos se dividem em duas categorias principais: não supervisionados (agrupamento ou *clustering*)

e supervisionados (classificação ou regressão). Algoritmos de ambas as categorias foram adaptados para que pudessem ser aplicados em cenários com fluxos contínuos de dados. As principais características de cada categoria e as especializações propostas serão discutidas a seguir.

As técnicas não supervisionadas realizam o agrupamento automático de dados segundo o seu grau de semelhança. Essas técnicas têm como objetivo a formação de grupos com alta similaridade intragrupo e baixa similaridade intergrupo (JAIN; DUBES, 1988). Os seguintes algoritmos são exemplos de técnicas não supervisionadas para cenários em lote: K-Means (LLOYD, 2006), DBSCAN (ESTER et al., 1996), PAM (KAUFMAN; ROUSSEEUW, 1990) e OPTICS (ANKERST et al., 1999).

De acordo com Gama (2010), a principal dificuldade ao aplicar técnicas não supervisionadas em cenários com fluxos contínuos é a manutenção da qualidade e consistência dos grupos formados conforme novos dados são observados. Portanto, é necessário que os algoritmos atuem de forma incremental, evoluindo os grupos formados ao longo do tempo (BARBARÁ, 2002). Sendo assim, foram desenvolvidos métodos não supervisionados especializados para fluxos contínuos de dados. Os seguintes trabalhos são exemplos dessas especializações: CluStream (AGGARWAL et al., 2003), StreamKM++ (ACKERMANN et al., 2012), DenStream (CAO et al., 2006), D-Stream (LING; LING-JUN; LI, 2009) e ClusTree (KRANEN et al., 2011).

Os algoritmos supervisionados realizam previsões para novos exemplos utilizando um modelo criado a partir de uma base de treinamento (KOTSIANTIS, 2007). Se a previsão é categórica, entende-se como um problema de classificação. Se a previsão resulta em um valor numérico, trata-se de uma tarefa de regressão. Exemplos de algoritmos supervisionados para cenários em lote, incluem: árvores de decisão (BREIMAN et al., 1984), métodos baseados em regras, redes neurais e máquinas de vetores suporte (SVM) (VAPNIK, 1998).

Segundo Gama e Gaber (2010), as técnicas supervisionadas tradicionais não podem ser aplicadas a contextos com fluxos contínuos de dados, pois estes métodos não contemplam as severas restrições de uso de memória e de tempo de execução desses cenários. Dessa forma, novos algoritmos supervisionados foram propostos para esses contextos (DOMINGOS; HULTEN, 2000; BIFET et al., 2013; WANG et al., 2003; AGGARWAL et al., 2004; GAMA; ROCHA; MEDAS, 2003).

As especializações mencionadas buscam atender às restrições de uso de memória e de tempo de execução dos contextos com fluxos contínuos de dados. Contudo, não consideram que na maioria desses cenários as informações são geradas por uma distribuição não estacionária e por processos que evoluem ao longo do tempo. Ou seja, a distribuição dos dados ou o contexto do processo gerador podem sofrer variações, alterando os resultados esperados. Na literatura, essas alterações são denominadas mudanças de conceito e a sua ocorrência pode impactar a acurácia da técnica aplicada (GAMA, 2010).

Neste projeto de mestrado, considera-se que os dados são obtidos a partir de fluxos contínuos de dados com ocorrência de mudanças de conceito. Na próxima seção, o fenômeno mudança de conceito será discutido em detalhes.

2.3 MUDANÇA DE CONCEITO

Técnicas de aprendizado de máquina aplicadas a cenários com fluxos contínuos de dados devem ser capazes de lidar com alterações na distribuição dos dados ou no contexto do processo gerador. Essas alterações são denominadas mudanças de conceito (*concept drift*) e podem alterar os resultados esperados (conceitos-alvo) dos algoritmos, prejudicando sua acurácia (WIDMER; KUBAT, 1996).

Na literatura, é comum utilizar a Teoria Bayesiana de Decisão (DUDA; HART; STORK, 2000) para descrever a tarefa de classificação. Esta descrição será utilizada como base para formalização do fenômeno de mudança de conceito.

Sendo $X \in \mathbb{R}^p$ uma instância em um espaço p -dimensional de atributos e $X \in c_i$ onde c_1, c_2, \dots, c_k é o conjunto de classes, o classificador ótimo para classificar $x \rightarrow c_i$ é determinado a partir das probabilidades a priori das classes $P(c_i)$ e pela função de densidade de probabilidade condicionada às classes $p(X|c_i), i = 1, \dots, k$. Logo, é possível definir um conceito como um conjunto de probabilidades a priori e condicionais das classes, como mostra a Equação 2.1:

$$S = \{(P(c_1), P(X|c_1)), (P(c_2), P(X|c_2)), \dots, (P(c_k), P(X|c_k))\} \quad (2.1)$$

Ainda segundo a Teoria Bayesiana, a classificação de uma instância X baseada na máxima probabilidade a posteriori pode ser obtida através da Equação 2.2:

$$p(c_i|X) = \frac{p(c_i) * p(X|c_i)}{p(X)} \quad (2.2)$$

Assim, é possível afirmar que há mudança de conceito entre os instantes t_0 e t_1 se:

$$\exists X : p_{t_0}(X, c) \neq p_{t_1}(X, c) \quad (2.3)$$

onde p_{t_0} e p_{t_1} denotam as distribuições de probabilidades conjuntas nos instantes t_0 e t_1 , respectivamente, para X e c (GAMA et al., 2014). Em outras palavras, 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 (KOLTER; MALOOF, 2007).

De acordo com Gama et al. (2014), as mudanças de conceito podem ser categorizadas como virtuais ou reais. As mudanças virtuais são causadas por alterações na probabilidade a priori das classes, $P(c)$, e não alteram os conceitos-alvo. Enquanto as mudanças de conceito reais surgem a partir de alterações na probabilidade a posteriori, $p(c|X)$, e modificam os resultados esperados. Os dois tipos de mudança de conceito estão representados na Figura 2.1.

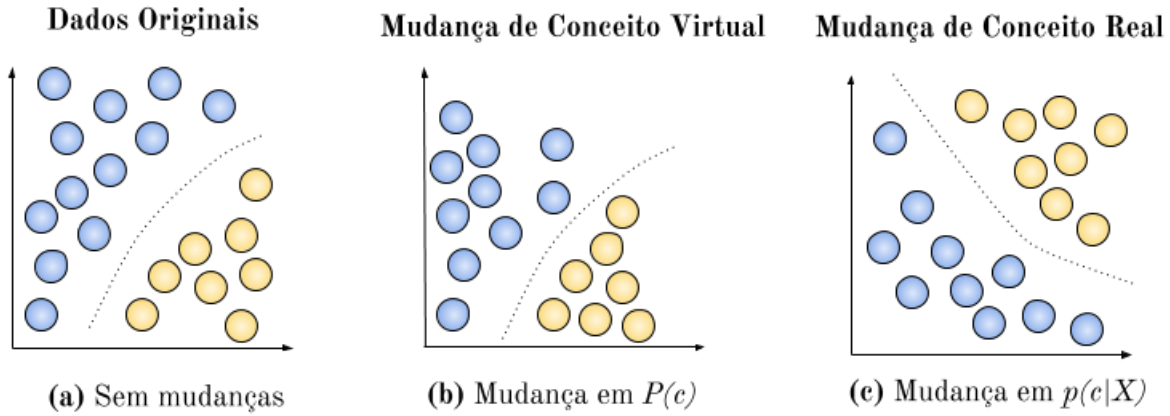


Figura 2.1 Mudança de Conceito Virtual vs. Mudança de Conceito Real

Conforme Zliobaite (2010), as mudanças de conceito podem ocorrer de forma abrupta, gradual, incremental ou recorrente. A Figura 2.2 ilustra estes padrões, utilizando círculos na cor azul para representar o conceito A e círculos na cor bege para o conceito B :



Figura 2.2 Padrões de ocorrência de Mudanças de Conceito

Na mudança abrupta, o conceito A é repentinamente substituído pelo conceito B (Figura 2.2 (a)).

Na mudança gradual, ocorre uma transição mais suave entre os conceitos A e B . Inicialmente, eventos pertencentes a ambos os conceitos coexistem. Com o passar do tempo, os eventos pertencentes ao conceito A diminuem de frequência, até pararem de ocorrer. Por fim, os eventos pertencentes a B se tornam predominantes (Figura 2.2 (b)).

A mudança incremental descreve a evolução de um único conceito ao longo do tempo. Essa evolução pode ser discretizada como uma sequência de conceitos consecutivos. Nesta

sequência, cada conceito intermediário difere pouco dos seus conceitos antecessor e sucessor. Portanto, as mudanças são notáveis apenas à longo prazo (Figura 2.2 (c)).

A mudança recorrente acontece quando um conceito anteriormente ativo reaparece após um determinado período de tempo. Contudo, não se trata de uma sazonalidade periódica, pois não é evidente o momento no qual o conceito voltará a ser ativo (Figura 2.2 (d)).

Este trabalho de mestrado propõe um método baseado em redes de função de base radial para detecção de mudanças de conceito reais em fluxos contínuos de dados, independente do padrão de ocorrência. Na próxima subseção, será apresentada a terminologia do fenômeno mudança de conceito.

2.3.1 Terminologia

O fenômeno mudança de conceito tem sido estudado em diferentes comunidades de pesquisa, incluindo mineração de dados, aprendizado de máquina, estatística e recuperação de informação (ZLIOBAITE, 2010). Contudo, o mesmo conceito pode ter diferentes nomenclaturas em cada comunidade. Na Tabela 2.1 são listados os termos correspondentes a mudança de conceito em cada área de pesquisa.

Tabela 2.1 Terminologia - Mudança de Conceito (ZLIOBAITE, 2010)

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
Estatísticas, Séries Temporais	Não Estacionário
Recuperação de Informação	Evolução Temporal

Outra fonte comum de equívocos são os termos detecção de *outliers*, detecção de novidade, detecção de *change points* e detecção de mudança de conceito. Estes termos são muitas vezes utilizados de forma indistinta, mas, para o contexto deste trabalho, é importante distingui-los.

As técnicas para detecção de *outliers* têm como objetivo identificar padrões de dados em desacordo com o comportamento esperado. Estes padrões são geralmente classificados como anomalias ou ruídos (CHANDOLA; BANERJEE; KUMAR, 2009).

Os métodos para detecção de novidade identificam padrões ainda não observados, mas que se enquadram no comportamento esperado. Estes métodos se diferenciam das técnicas para detecção de *outliers* pois os novos padrões são incorporados ao modelo (CHANDOLA; BANERJEE; KUMAR, 2009).

As estratégias para detecção de *change points* identificam variações abruptas de valor, que podem representar transições entre estados, em séries temporais unidimensionais estacionárias (AMINIKHANGHAHI; COOK, 2017).

Por fim, os métodos para detecção de mudanças de conceito monitoram a distribuição dos dados ou indicadores (por exemplo: taxa de erro) das técnicas de aprendizado aplicadas, a fim de identificar a ocorrência de mudanças de conceito (GAMA et al., 2014).

Na próxima subseção, os principais algoritmos para detecção de mudança de conceito serão descritos.

2.3.2 Algoritmos para Detecção de Mudança de Conceito

Os algoritmos para detecção de mudança de conceito caracterizam e quantificam as mudanças de conceito através da delimitação dos instantes ou intervalos de tempo em que as mudanças ocorrem (BASSEVILLE; NIKIFOROV, 1993).

Esses algoritmos se dividem em duas categorias, conforme a necessidade de rotulação dos dados (ZLIOBAITE, 2010):

Algoritmos Explícitos/Supervisionados Dependem da rotulação dos dados por um especialista. Estes rótulos são utilizados no cálculo de medidas de performance como taxa de erro e acurácia, que são monitoradas ao longo do tempo. Mudanças de conceito são sinalizadas quando essas medidas atingem um limite previamente definido.

Algoritmos Implícitos/Não Supervisionados Independem da rotulação por especialistas, baseando-se em características dos próprios dados ou indicadores das técnicas de aprendizado aplicadas. São mais propensos a alarmes falsos, mas essa independência os torna interessantes para contextos onde a obtenção de rótulos é dispendiosa, demorada ou inviável.

Segundo Gama et al. (2014), os algoritmos *explícitos* / *supervisionados* podem ser segmentados em três subcategorias:

Métodos Baseados em Análise Sequencial Avaliam continuamente os indicadores de performance (por exemplo: taxa de erro) do classificador aplicado. A mudança de conceito é detectada quando esses indicadores atingem um limite pré-definido. Os algoritmos *Cumulative Sum (CUSUM)*, *PageHinkley (PH)* (PAGE, 1954) e *Geometric Moving Average (GMA)* (ROBERTS, 2000) são representantes desta subcategoria.

Abordagens baseadas em Estatística Identificam mudanças de conceito através da análise de parâmetros estatísticos como média e desvio padrão associados aos resultados das predições. Os métodos *Drift Detection Method (DDM)* (GAMA et al., 2004), *Early Drift Detection Method (EDDM)* (BAENA-GARCÍA et al., 2006), *Exponentially Weighted Moving Average (EWMA)* (ROSS et al., 2012) e *Reactive Drift Detection Method (RDDM)* (BARROS et al., 2017) são exemplos desta subcategoria.

Métodos baseados em Janelas Utilizam uma janela de tamanho fixo para sumarizar informações passadas e uma janela deslizante para sumarizar os dados recentes. Uma diferença significativa entre as distribuições dessas janelas implica na ocorrência de mudança de conceito. Esta diferença é verificada a partir de testes estatísticos ou desigualdades matemáticas, considerando como hipótese nula a igualdade das distribuições. Os algoritmos *Adaptive Windowing (ADWIN)* (BIFET; GAVALDÀ, 2007), *SeqDrift* (PEARS; SAKTHITHASAN; KOH, 2014), *HDDMA* e *HDDMW* (BLANCO et al., 2015) pertencem a esta subcategoria.

De forma similar, os algoritmos *implícitos / não supervisionados* também foram divididos em três subcategorias (GONÇALVES et al., 2014):

Deteção de Novidade / Métodos de Agrupamento Utilizam técnicas derivadas dos métodos de agrupamento e de deteção de *outliers* para identificar padrões ainda não observados. A partir dessa identificação, são realizados cálculos de distância e/ou densidade para confirmar a ocorrência de mudança de conceito (RYU et al., 2012). Os métodos *OLINDDA* (SPINOSA; CARVALHO; GAMA, 2007), *MINAS* (FARIA; GAMA; CARVALHO, 2013), *Woo* (RYU et al., 2012), *DETECTNOD* (Hayat; Hashemi, 2010), *ECSMiner* (MASUD et al., 2011) e *GC3* (SETHI; KANTARDZIC; HU, 2016) fazem parte desta subcategoria.

Monitoramento de distribuição multivariada Monitoram diretamente a distribuição dos dados para cada atributo. A distribuição de um conjunto de treinamento é sumarizada e utilizada como referência. Esta referência é, então, comparada à distribuição dos dados do conjunto atual. Diferenças significativas entre esses conjuntos indicam a ocorrência de mudança de conceito. Os algoritmos *CoC* (Lee; Magoulès, 2012), *HDDDM* (Ditzler; Polikar, 2011), *PCA-detect* (KUNCHEVA, 2008) são representantes desta subcategoria.

Monitoramento dependente de modelo Dependem da aplicação de um algoritmo de classificação probabilístico, pois as mudanças de conceito são detectadas a partir do monitoramento da probabilidade a posteriori calculada (ZLIOBAITE, 2010). Estes algoritmos conseguem reduzir a ocorrência de falsos positivos e tornar o processo computacionalmente eficiente, pois apenas um único fluxo univariado de valores é observado. Os métodos *A-distance* (DREDZE; OATES; PIATKO, 2010), *CDBD* (LINDSTROM; NAMEE; DELANY, 2013) e *Margin* (DRIES; RÜCKERT, 2009) integram esta subcategoria.

Por fim, a Tabela 2.2 resume as categorias, as subcategorias e as respectivas técnicas abordadas nesta seção.

O método de deteção proposto neste trabalho se enquadra na categoria de algoritmos *implícitos / não supervisionados*, mais especificamente na subcategoria *deteção de novidades / métodos de agrupamento*. Na próxima seção, as ferramentas utilizadas para implementação e validação deste método serão apresentadas.

Tabela 2.2 Resumo - Algoritmos de detecção (SETHI; KANTARDZIC, 2017)

Algoritmos Explícitos/Supervisionados	Métodos Baseados em Análise Sequencial	Cumulative Sum (CUSUM) PageHinkley (PH) (PAGE, 1954) Geometric Moving Average (GMA) (ROBERTS, 2000)
	Abordagens baseadas em Estatística	Drift Detection Method (DDM) (GAMA et al., 2004) Early Drift Detection Method (EDDM) (BAENA-GARCÍA et al., 2006) Exponentially Weighted Moving Average (EWMA) (ROSS et al., 2012) Reactive Drift Detection Method (RDDM) (BARROS et al., 2017)
	Métodos baseados em Janelas	Adaptive Windowing (ADWIN) (BIFET; GAVALDÀ, 2007) SeqDrift (PEARS; SAKTHITHASAN; KOH, 2014) HDDMA/HDDMW (BLANCO et al., 2015)
Algoritmos Implícitos/Não Supervisionados	Detecção de Novidade / Métodos de Agrupamento	OLINDDA (SPINOSA; CARVALHO; GAMA, 2007) MINAS (FARIA; GAMA; CARVALHO, 2013) Woo (RYU et al., 2012) DETECTNOD (Hayat; Hashemi, 2010) ECSMiner (MASUD et al., 2011) GC3 (SETHI; KANTARDZIC; HU, 2016)
	Monitoramento de distribuição multivariada	CoC (Lee; Magoulès, 2012) HDDM (Ditzler; Polikar, 2011) PCA-detect (KUNCHEVA, 2008)
	Monitoramento dependente de modelo	A-distance (DREDZE; OATES; PIATKO, 2010) CDBD (LINDSTROM; NAMEE; DELANY, 2013) Margin (DRIES; RÜCKERT, 2009)

2.3.3 Ferramentas

Nesta seção, os frameworks *Massive Online Analysis* (MOA) e *Tornado* serão apresentados. Estas ferramentas permitem a implementação e a validação de novas técnicas de detecção de mudanças de conceito. Além de possibilitarem a comparação com o estado da arte, pois dispõem de um vasto conjunto de algoritmos já implementados. Ambas as ferramentas foram utilizadas durante o desenvolvimento deste projeto de trabalho de mestrado.

2.3.4 MOA

Atualmente, o *MOA – Massive Online Analysis*¹ é o principal framework para mineração de dados em fluxos contínuos (BIFET et al., 2010). O projeto é de código-aberto² e apresenta uma comunidade bastante ativa e crescente. A aplicação é composta por uma ampla coleção de algoritmos da área de aprendizado de máquina, contemplando técnicas de classificação, regressão, agrupamento, busca por padrões, detecção de *outliers*, detecção de mudanças de conceito e sistemas de recomendação. Além das implementações, também estão disponíveis rotinas para avaliação dessas técnicas. A aplicação é desenvolvida em Java, o que permite a sua execução nos principais sistemas operacionais e a integração com o projeto WEKA (HALL et al., 2009).

¹<https://moa.cms.waikato.ac.nz/>

²<https://github.com/Waikato/moa>

O MOA divide as suas funcionalidades em tarefas (*tasks*). Estas tarefas podem ser executadas a partir da interface gráfica (GUI) ou por linha de comando. A interface gráfica permite executar múltiplas tarefas de forma concorrente, controlar suas execuções e visualizar os resultados parciais. A tela principal da aplicação é demonstrada na Figura 2.3.

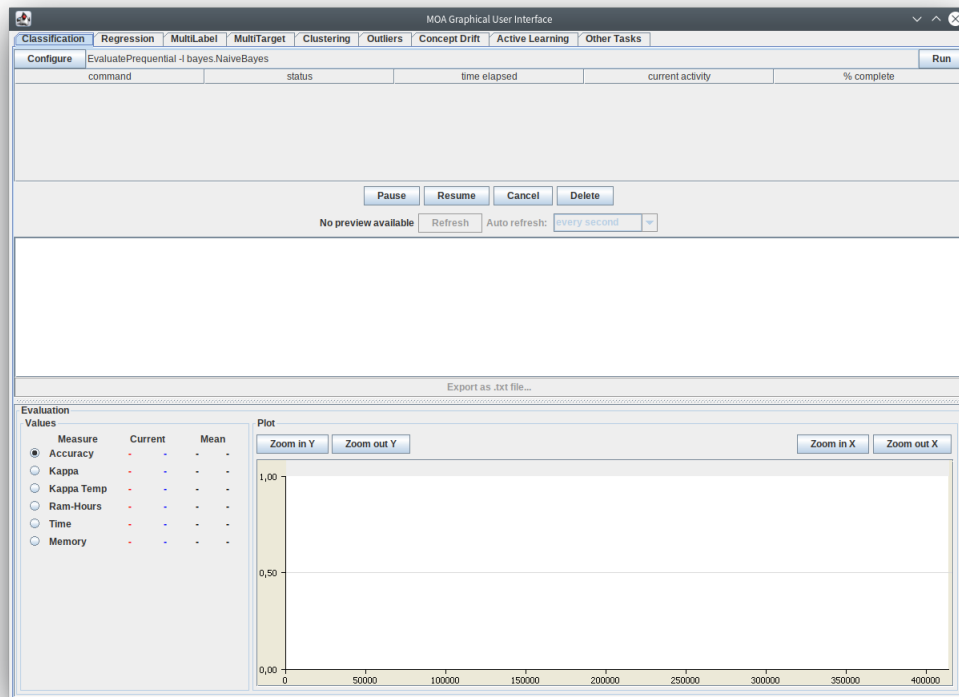


Figura 2.3 MOA - Tela Inicial

A aplicação é capaz de ler arquivos em formato *ARFF*, popularizados pelo projeto WEKA. A ferramenta também permite a produção de fluxos de dados dinamicamente, através de geradores. Alguns dos geradores de fluxo disponíveis no MOA são: *Random Trees* (DOMINGOS; HULTEN, 2000) *SEA* (STREET; KIM, 2001), *STAGGER* (SCHLIMMER; GRANGER, 1986), *Rotating Hyperplane* (WANG et al., 2003), *Random RBF*, *LED* (GAMA; ROCHA; MEDAS, 2003), *Waveform* (GAMA; ROCHA; MEDAS, 2003), e *Function* (JIN; AGRAWAL, 2003).

Outra funcionalidade importante do framework é a possibilidade de adicionar mudanças de conceito a fluxos estacionários existentes. Esse processo é realizado através de uma função sigmóide, que modela o evento de mudança de conceito como uma combinação balanceada de duas distribuições homogêneas, que caracterizam os conceitos-alvo antes e depois da mudança. Além destes conceitos, o usuário também pode definir o momento da mudança e a sua duração (BIFET et al., 2010).

Os principais métodos para detecção de mudança de conceito propostos na literatura estão disponíveis no MOA. Além disso, a arquitetura do framework é modular, permitindo a implementação de novos detectores de forma trivial. Por exemplo, para criar um novo detector, basta estender a classe abstrata *AbstractChangeDetector* e implementar o algoritmo desejado. A janela de configuração deste detector, similar a Figura 2.4, será criada dinamicamente, a partir dos atributos definidos na classe.

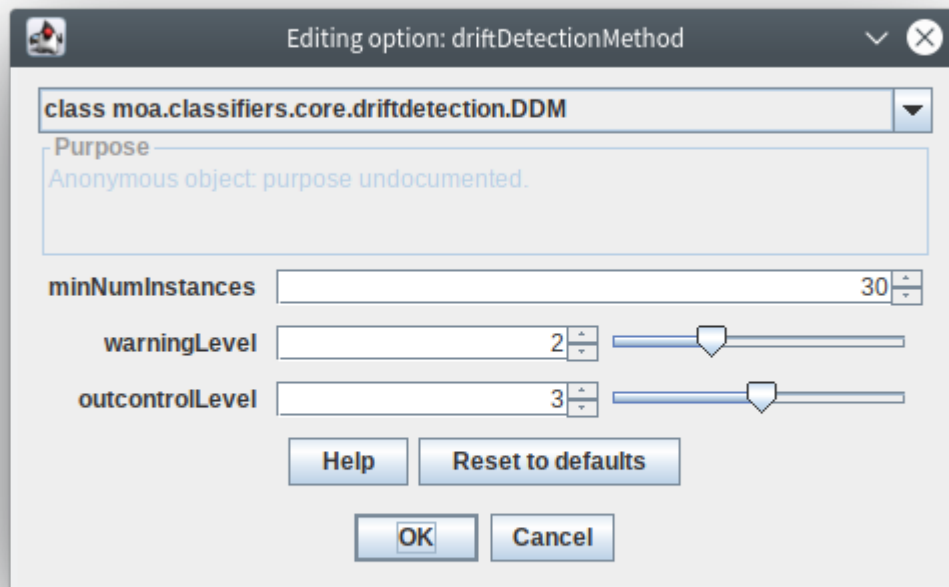


Figura 2.4 MOA - Configuração detector

O MOA dispõe de diversas classes para avaliação de técnicas de aprendizado de máquina. Para este trabalho, a classe *BasicConceptDriftPerformanceEvaluator* é mais relevante, por realizar a análise de algoritmos para detecção de mudança de conceito.

Na próxima subseção, o framework *Tornado* será discutido.

2.3.5 Tornado

O *Tornado* é um framework para avaliação de algoritmos de detecção de mudança de conceito (PESARANGHADER, 2018). O projeto é desenvolvido na linguagem Python

e o seu código está disponível³. O framework se diferencia do *MOA* por apresentar um cenário de avaliação específico: analisar a execução, em paralelo, de pares (classificador, detector de mudança de conceito), para identificar o par ótimo ao longo do tempo, em relação ao fluxo de dados.

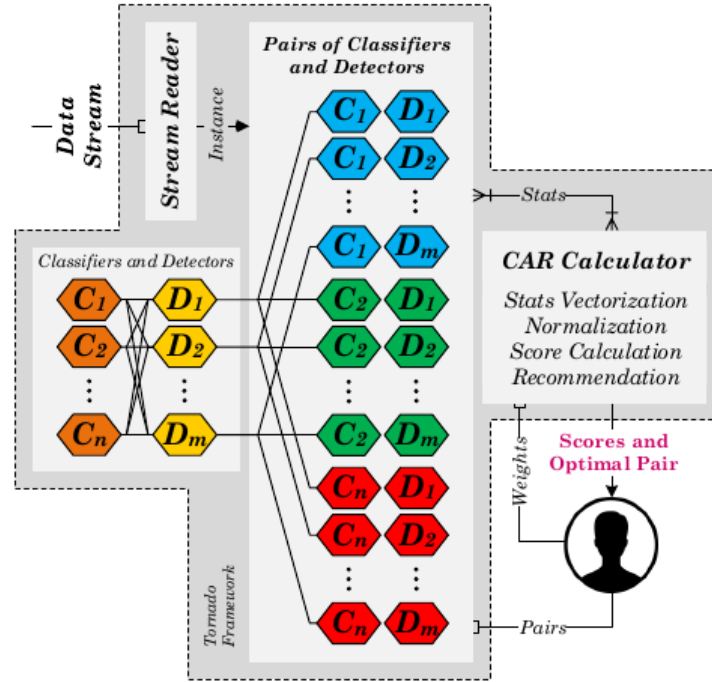


Figura 2.5 Framework Tornado (PESARANGHADER, 2018).

Conforme apresentado na Figura 2.5, os principais componentes do framework são: *Stream Reader*, *Classifiers*, *Detectors*, *Classifier-Detector Pairs* e *CAR Calculator*. A entrada de dados é composta por um fluxo (*Stream*), uma lista de pares (classificador, detector) e um vetor com pesos.

O componente *Stream Reader* recebe as instâncias e as encaminha para construção do modelo de forma incremental. Por seguir a abordagem *prequential*, cada instância é primeiramente utilizada para testes e depois como treinamento. Simultaneamente, os classificadores enviam suas estatísticas aos detectores, para que a mudança de conceito possa ser sinalizada. Por fim, o componente *CAR Calculator* calcula uma pontuação para cada par (classificador e detector), considerando taxa de erro, atraso para detecção da mudança de conceito, falsos positivos, falsos negativos, quantidade de memória utilizada e tempo de execução (PESARANGHADER, 2018).

O framework apresenta ao usuário o par ótimo para cada instante da execução. A abordagem de avaliação adotada pelo framework é relevante, pois este par pode mudar ao longo do tempo, devido ao aprendizado incremental ou às mudanças de conceito. A Figura 2.6 apresenta um exemplo de resultado produzido pela ferramenta.

³<https://github.com/alipsgh/tornado>

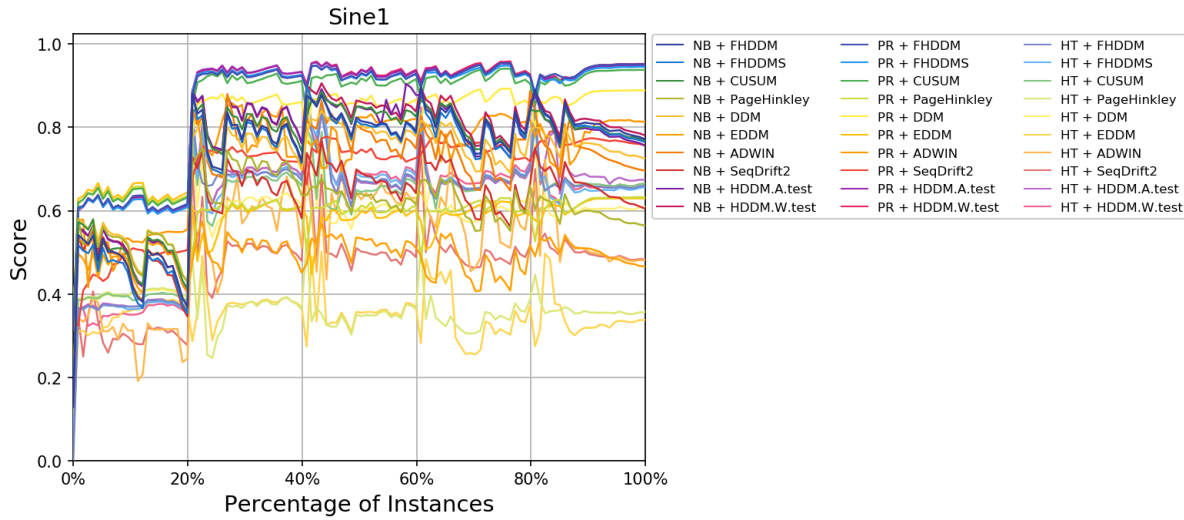


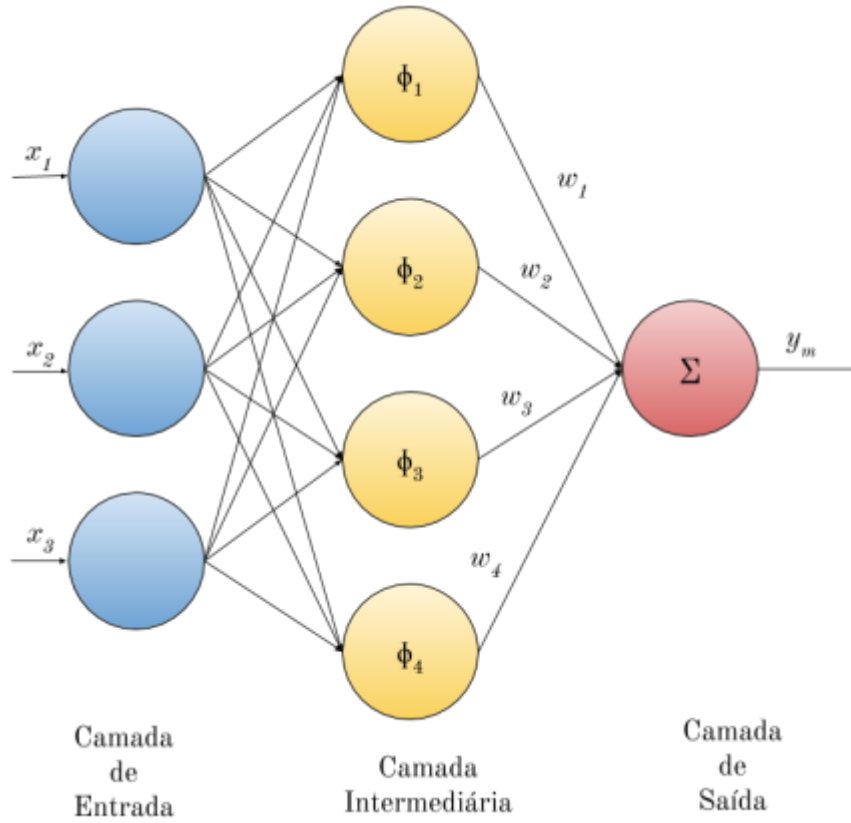
Figura 2.6 Tornado - Exemplo de resultado (PESARANGHADER, 2018)

Neste trabalho, o algoritmo proposto será implementado e testado nas duas ferramentas apresentadas. Os detalhes de implementação e os resultados desses testes serão discutidos no Capítulo 4. A seguir, as redes de função de base radial são detalhadas.

2.4 REDES DE FUNÇÃO DE BASE RADIAL

Uma Rede de Função de Base Radial (*radial basis function*, *RBF*, em inglês) pode ser definida como um modelo de múltiplas camadas alimentadas adiante (*feedforward*), capaz de analisar padrões complexos e resolver problemas não-linearmente separáveis, utilizando uma abordagem de aproximação de funções. Estas redes têm como principal diferencial a sua forma de ativação, realizada através do cálculo da distância entre o dado e um centro definido (BRAGA; CARVALHO; LUDERMIR, 2007).

A arquitetura de uma rede de função de base radial, em sua forma mais básica, envolve três camadas. A camada de entrada contém nós de fonte (unidades sensoriais) que conectam a rede ao seu ambiente. A camada intermediária, única camada oculta da rede, utiliza funções de base radial para realizar uma transformação não-linear dos dados de entrada para um espaço de alta dimensionalidade. Por fim, a camada de saída, através de uma combinação linear, fornece a resposta da rede ao padrão (sinal) de ativação aplicado à camada de entrada (ROJAS, 1996). A Figura 2.7 demonstra essa arquitetura.

**Figura 2.7** Arquitetura RBF

A transformação não-linear dos dados de entrada para um espaço de alta dimensionalidade é justificada matematicamente pelo teorema de Cover (1965), segundo o qual, “um problema complexo de classificação de padrões disposto não linearmente em um espaço de alta dimensionalidade tem maior probabilidade de ser linearmente separável do que em um espaço de baixa dimensionalidade”.

Essa transformação é realizada por funções de base radial presentes na camada intermediária. Na literatura, uma das funções mais utilizadas para esta tarefa é a função gaussiana, representada na Equação 2.4:

$$\varphi(v_i) = e^{-(\sigma r)^2} \quad (2.4)$$

onde, r é a distância euclidiana ($\|\mathbf{v}_i - \mathbf{c}_i\|$), v é o valor de entrada, c_i representa o centro e σ é o parâmetro limitador do raio. Assim, a classificação realizada por uma rede de função de base radial consiste na resolução das funções 2.5 e 2.6:

$$f(x) = \sum_{i=1}^N w_{ij} \varphi(\|\mathbf{x} - \mathbf{t}_i\|) \quad (2.5)$$

$$y_i = \sum_{j=1}^N w_{ij} \phi(\|\mathbf{x} - \mathbf{t}_i\|) + w_{j_0} \quad (2.6)$$

A resolução dessas equações dá origem ao sistema responsável por produzir o resultado final da rede (Equação 2.7):

$$\begin{bmatrix} \varphi(\|x_1 - t_1\|) & \varphi(\|x_1 - t_2\|) & \dots & \varphi(\|x_1 - t_N\|) \\ \varphi(\|x_2 - t_1\|) & \varphi(\|x_2 - t_2\|) & \dots & \varphi(\|x_2 - t_N\|) \\ \dots & \dots & \dots & \dots \\ \varphi(\|x_N - t_1\|) & \varphi(\|x_N - t_2\|) & \dots & \varphi(\|x_N - t_N\|) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad (2.7)$$

onde w_{ij} são os pesos de cada conexão, ϕ é a matriz de interpolação originada do conjunto de N funções de base radial aplicadas nas entradas x e dos seus respectivos centros t_i , w_{j_0} representa o bias, $\varphi(\|x - t_i\|)$ é o conjunto de N funções de base radial, $\|\dots\|$ é a norma euclidiana e y são as saídas geradas pela rede.

O algoritmo proposto neste trabalho utiliza as camadas inicial e intermediária das redes de função de base radial para compôr um novo método de detecção de mudanças de conceito. Isto é possível, pois a camada intermediária cria, de forma implícita, agrupamentos no espaço oculto de alta dimensionalidade. Dessa forma, a criação de novos centros e a mudança do centro ativo podem sinalizar a ocorrência de mudanças de conceito. A técnica desenvolvida será discutida em detalhes no Capítulo 3.

Na próxima seção, os trabalhos relacionados encontrados na literatura são apresentados.

2.5 TRABALHOS RELACIONADOS

Além das referências básicas apresentadas neste capítulo, foi realizada uma pesquisa na literatura visando identificar trabalhos que utilizam redes de função de base radial para identificação de mudanças de conceito ou para outras tarefas correlatas.

Roberts, Penny e Pillot (1996) propuseram um método para detecção de novidades baseado em um comitê de redes de função de base radial. Neste método, novos padrões são detectados a partir do monitoramento das taxas de erro e confiança do comitê. A técnica foi utilizada para classificação de pacientes com problemas de tremor muscular.

Jianping e Venkateswarlu (2002) também utilizaram as redes de função de base radial em tarefas de detecção de novidades. A técnica proposta atua sobre cenários estacionários e foi aplicada ao problema de identificação da fala. Seu principal diferencial é a utilização do algoritmo *k-means* para definir os centros e as matrizes de covariância da rede.

Por fim, Bazargani e Namee (2018) utilizaram redes RBF para detecção de anomalias em fluxos de dados. O método proposto modifica as funções de perda, transformando as redes de função de base radial em classificadores de classe única, permitindo a identificação de exemplos divergentes dos padrões conhecidos.

2.6 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os principais conceitos utilizados neste projeto de trabalho de mestrado. Foram discutidos conceitos de fluxos contínuos de dados, técnicas de aprendizado de máquina, mudanças de conceito, técnicas de detecção de mudanças de conceito e redes de função de base radial. Por fim, foram apresentados os trabalhos relacionados encontrados na literatura. No próximo capítulo, o plano de pesquisa será detalhado.

PLANO DE PESQUISA

3.1 CONSIDERAÇÕES INICIAIS

Este capítulo descreve como a pesquisa proposta neste mestrado será desenvolvida para permitir que redes de função de base radial sejam aplicadas para detectar mudanças de conceito em fluxos contínuos de dados. Espera-se que com a utilização de redes RBF, seja possível detectar mudanças de conceito em fluxos contínuos em tempo de execução, de forma computacionalmente eficiente e independente de rótulos. A seguir, são apresentados detalhes sobre cada etapa do desenvolvimento do projeto.

3.2 DESCRIÇÃO DO PROBLEMA

Nos últimos anos, a quantidade de dados produzidos por sistemas computacionais tem crescido de forma exponencial (ZWOLENSKI; WEATHERILL, 2014). Parte significativa dessas informações é produzida na forma de fluxos contínuos de dados, que são sequências potencialmente infinitas e de alta frequência (AGGARWAL, 2006).

Devido a esse crescimento, pesquisadores passaram a utilizar técnicas de aprendizado de máquina para extrair informações úteis de grandes volumes de dados. Essas técnicas precisaram ser adaptadas para contextos com fluxos contínuos, pois estes cenários apresentam severas restrições de tempo de execução e de uso dos recursos computacionais.

Contudo, as adaptações propostas não tratam variações na distribuição dos dados ou no contexto do processo gerador do fluxo. Estas alterações são denominadas mudanças de conceito e podem afetar negativamente a acurácia do algoritmo (GAMA et al., 2014).

Para mitigar este problema, técnicas de detecção de mudanças de conceito foram desenvolvidas. Estes métodos identificam com precisão o momento da mudança, permitindo que o modelo de decisão seja atualizado de forma eficiente e tempestiva.

Entretanto, as técnicas de detecção encontradas na literatura apresentam limitações ao serem aplicadas em cenários com fluxos contínuos de dados. Os métodos de detecção supervisionados/explicitos necessitam que o rótulo correto de cada exemplo processado seja informado, o que os torna inviáveis, por causa do custo e do tempo de rotulação.

Enquanto as técnicas não supervisionadas/implícitas têm dificuldade para antever às restrições de tempo de execução e de uso dos recursos computacionais desses cenários.

Redes de função de base radial são modelos de redes neurais multicamadas, alimentadas adiante, capazes de analisar padrões complexos e resolver problemas não-linearmente separáveis (BRAGA; CARVALHO; LUDERMIR, 2007). A arquitetura básica dessas redes é composta por três camadas. A camada de entrada recebe os dados. A camada intermediária, através de funções de base radial, realiza uma transformação não-linear dos dados para um espaço com alta dimensionalidade, buscando tornar o problema linearmente separável (COVER, 1965). Por fim, a camada de saída produz o resultado final da rede através de uma combinação linear dos resultados da camada intermediária (ROJAS, 1996).

A camada intermediária das redes de função de base radial forma, implicitamente, grupos (*clusters*) no espaço de alta dimensionalidade criado. O centro ativo deste agrupamento pode se alterar, conforme o valor processado. Assim, este trabalho de mestrado implementa as camadas inicial e intermediária, e utiliza as alterações de centro no espaço de alta dimensionalidade como indicadores da ocorrência de mudanças de conceito.

O método proposto, denominado *RBFDriftDetector*, utiliza a Gaussiana (Equação 2.4) como função de base radial e requer a definição de dois parâmetros: σ , responsável por limitar o raio da função gaussiana, e λ , que define o valor mínimo para ativação de um centro. A técnica é descrita na forma de pseudocódigo em Algoritmo 1, tendo sido também implementada na linguagem Java¹, para realização dos experimentos na plataforma MOA.

Entrada: *valor*, σ , λ

Saída: booleano indicando a ocorrência ou não de mudança de conceito

início

```

centros  $\leftarrow$  (); centroAtual  $\leftarrow$  null; centroAtivo  $\leftarrow$  null
mudanca  $\leftarrow$  falso
para todo centro faça
    | ativacao  $\leftarrow$  gaussiana(valor,  $\sigma$ )
    | se ativacao  $\geq \lambda$  então
    | | centroAtivo  $\leftarrow$  centro
    | |  $\lambda \leftarrow$  ativacao
se centroAtivo == null então
    | centros  $\leftarrow$  valor
    | centroAtivo  $\leftarrow$  valor
se centroAtual == null então
    | centroAtual  $\leftarrow$  valor
se centroAtual  $\neq$  centroAtivo então
    | centroAtual  $\leftarrow$  centroAtivo
    | mudanca  $\leftarrow$  verdadeiro
retorna mudanca

```

Algoritmo 1: RBFDRIFTDETECTOR

¹<https://git.io/fjGuv>

Para exemplificar a execução do algoritmo proposto neste projeto, considere o conjunto $S = \{0.11, 0.12, 0.13, 0.34, 0.45, 0.47, 0.33, 0.25, 0.14, 0.10\}$ como fonte de dados. Para este exemplo, os parâmetros foram definidos de forma empírica. O parâmetro σ foi definido com o valor 0.2 e o parâmetro λ foi fixado em 0.6. A seguir, será descrito o comportamento do algoritmo para cada dado recebido a partir de S .

No instante $T1$, o valor 0.11 é recepcionado. Como não existem centros estabelecidos, o valor é definido como centro e ativado. Em $T2$ e $T3$ são recebidos, respectivamente, os valores 0.12 e 0.13 que são imediatamente vinculados ao centro atualmente ativo (0.11), pois seus valores de ativação foram maiores que 0.6 (λ).

No instante $T4$, o valor 0.34 não atinge o valor mínimo de ativação (λ) para ser vinculado ao centro ativo (0.11). Dessa forma, o valor é definido como um novo centro e ativado. Devido a alteração do centro ativo, o algoritmo sinaliza a ocorrência de mudança de conceito.

Os valores dos instantes $T5$, $T6$, $T7$ e $T8$ são vinculados ao último centro ativo (0.34). Contudo, em $T9$, o valor 0.14 é recebido e apresenta maior valor de ativação para o centro 0.11, que é reativado. Assim, uma nova mudança de conceito é sinalizada. Finalmente, o valor do instante $T10$, 0.10, é vinculado ao centro ativo (0.11).

A Figura 3.1 apresenta esse comportamento graficamente. Os círculos na cor amarela representam valores vinculados ao primeiro centro estabelecido (0.11 no momento $T1$), enquanto os de cor lilás foram ativados pelo segundo centro definido (0.34 em $T4$). As linhas verticais tracejadas, de cor vermelha, indicam a ocorrência de mudança de conceito.

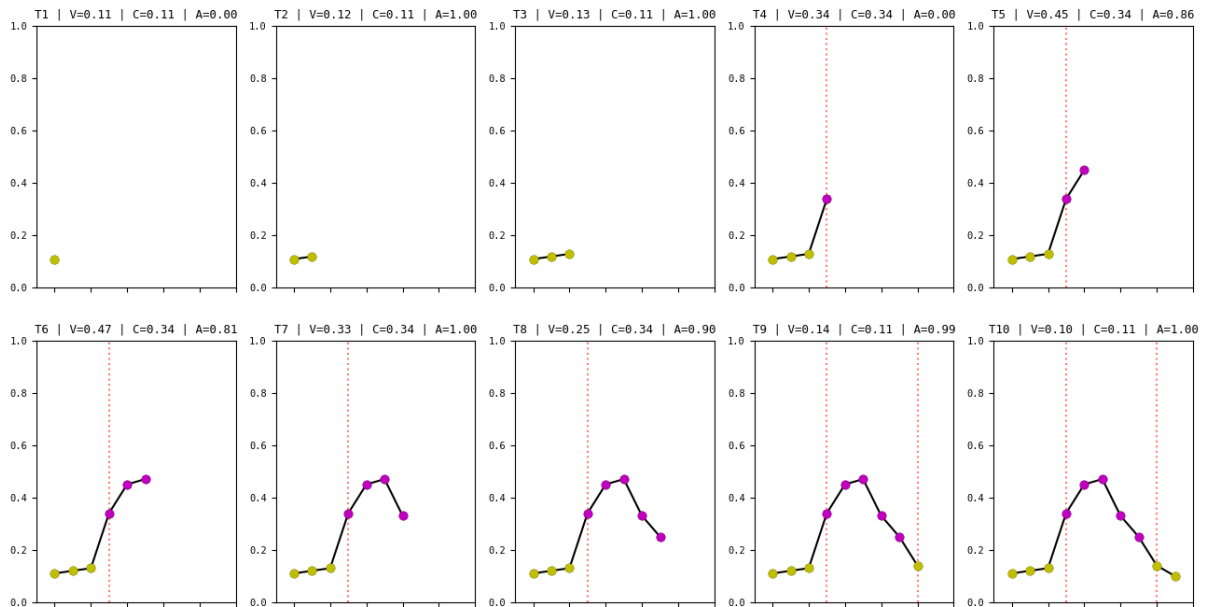


Figura 3.1 Exemplo de funcionamento do algoritmo

3.3 ATIVIDADES DE PESQUISA

A Tabela 3.1 apresenta o cronograma das atividades planejadas para a realização da pesquisa. Atividades concluídas são representadas pelo símbolo *X* e as futuras por *•*.

Tabela 3.1 Cronograma de atividades

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																								•

Para a conclusão da atividade 1, o Programa de Pós-Graduação em Ciência da Computação (PGCOMP) da Universidade Federal da Bahia (UFBA) exige que um mestrando obtenha um total de 18 créditos em disciplinas. Nos semestres 2018.1 e 2018.2, foram obtidos os créditos exigidos ao cursar as disciplinas: MATD74 – Algoritmos e Grafos, MATE32 – Tópicos em Inteligência Computacional II, MATE64 – Seminários Científicos, MATE65 – Fundamentos de Pesquisa em Ciência da Computação I, MATE10 – Tópicos em Inteligência Computacional I e MATE84 – Tópicos em Fundamentos da Computação IV.

A segunda atividade planejada neste cronograma foi realizada a partir da disciplina MATE65 – Fundamentos de Pesquisa em Ciência da Computação I e nos meses subsequentes. Além disso, durante a execução desta tarefa foi realizada a prova de proficiência em inglês.

As atividades 3 e 4 do cronograma consistem na realização dos experimentos e análise dos resultados. Essas atividades foram divididas em duas partes. A primeira contém apenas experimentos preliminares que foram realizados para validar esta proposta de trabalho. Nesta parte, fluxos contínuos de dados sintéticos com mudança de conceito foram analisados conforme apresentado no Capítulo 4. A segunda parte dos experimentos e suas análises serão realizadas após a qualificação.

As atividades 5 e 8 estão relacionadas com o componente curricular MATD75 – Exame de qualificação. A atividade 5 refere-se à escrita deste texto e as atividades 6 e 7 representam os componentes curriculares MATA32 – Estágio Docente e MATA31 – Pesquisa Orientada, respectivamente. A atividade 8 está relacionada à apresentação desta qualificação de mestrado.

A escrita de artigos, listada no item 9 do cronograma, será realizada com base nos resultados gerados com os experimentos (Atividades 3 e 4) e nas contribuições obtidas

com a apresentação da qualificação. Por fim, como requisito para a defesa de dissertação, fica pendente a atividade MATE93 — Defesa de Proposta de Mestrado, a qual se refere aos itens 10 e 11 da Tabela 3.1.

3.4 CONSIDERAÇÕES FINAIS

Neste capítulo, apresentou-se de maneira detalhada o projeto de pesquisa, o plano de atividades e o cronograma planejado para a conclusão do mestrado. No próximo capítulo, serão discutidos os resultados preliminares realizados com o objetivo de analisar a viabilidade da proposta de mestrado.

EXPERIMENTOS INICIAIS

4.1 CONSIDERAÇÕES INICIAIS

Este capítulo apresenta um conjunto de experimentos preliminares realizados com dados sintéticos, cujo objetivo foi demonstrar a viabilidade da proposta de mestrado. Os resultados obtidos se mostraram promissores, justificando a continuidade da pesquisa. A próxima seção apresenta como os dados sintéticos foram produzidos para condução dos experimentos.

4.2 CONFIGURAÇÃO DOS EXPERIMENTOS

Os fluxos de dados sintéticos foram produzidos através de classes geradoras do framework *MOA* e salvos em arquivos no formato *ARFF*, para que pudessem ser reutilizados. Cada fluxo gerado representa até dois conceitos e é composto por 20.000 observações, com valores entre 0 e 1. O restante desta seção apresenta as classes geradoras utilizadas, as modificações realizadas e os parâmetros aplicados.

A classe geradora *AbruptChangeGenerator* produz fluxos de dados sintéticos com mudanças de conceito abruptas. Os fluxos gerados simulam as mudanças através da intercalação de sequências com o valor 0.2 e sequências com o valor 0.8, conforme o tamanho definido para os conceitos. Com o objetivo de tornar os dados produzidos mais próximos da realidade, a classe foi alterada¹ para permitir a adição de um ruído randômico, com amplitude configurável, para cada exemplo produzido. A versão com suporte a ruído foi utilizada na produção do fluxo com mudanças abruptas utilizado nos experimentos. O gerador foi parametrizado para produzir exemplos não binários, com conceitos formados por 2.500 instâncias e ruído limitado ao intervalo $[-0.1, 0.2]$. Esta configuração é apresentada na Figura 4.1.

¹<https://git.io/fjGEj>

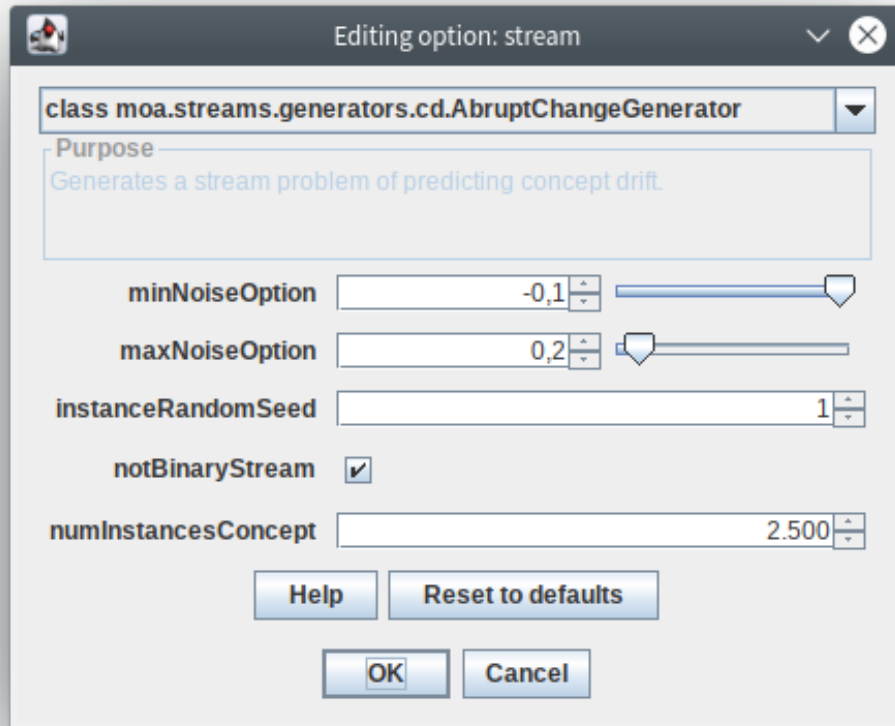


Figura 4.1 Parametrização da classe *AbruptChangeGenerator*

A classe *GradualChangeGenerator*, responsável por produzir fluxos sintéticos com mudanças de conceito graduais, também não permite a adição de ruído aos resultados. Além disso, a classe produz apenas uma mudança de conceito gradual, ignorando o tamanho parametrizado para os conceitos. Para mitigar estas limitações, a classe foi modificada² para permitir a adição de um ruído randômico e parametrizável aos resultados e para respeitar o tamanho dos conceitos, produzindo uma quantidade de mudanças adequada. Esta classe foi utilizada para gerar o fluxo sintético com mudanças graduais utilizado nos experimentos, sendo parametrizada para produzir exemplos não binários, com conceitos formados por 2.500 instâncias e ruído limitado ao intervalo $[-0.1, 0.2]$. A Figura 4.2 apresenta a tela de configuração desta classe geradora.

²<https://git.io/fjGue>

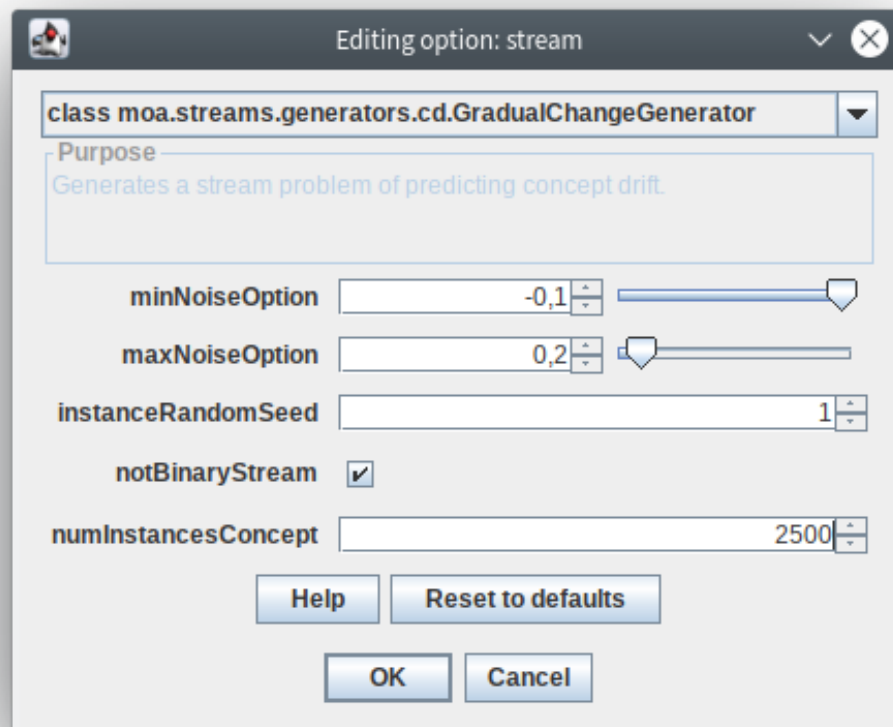


Figura 4.2 Parametrização da classe *GradualChangeGenerator*

Por fim, a classe *NoChangeGenerator*, que produz fluxos sem ocorrência de mudanças de conceito, também foi modificada³ para permitir a incidência de ruído nos resultados gerados. Esta versão foi utilizada para produzir um fluxo sintético sem mudanças, com ruído entre $[-0.1, 0.1]$, para ser utilizado como *baseline* nos experimentos.

4.3 CRITÉRIOS DE AVALIAÇÃO

A classe *BasicConceptDriftPerformanceEvaluator* do MOA será utilizada para avaliar o método proposto neste trabalho e compará-lo ao estado da arte. Esta classe permite avaliar algoritmos para detecção de mudanças de conceito, analisando dimensões de acurácia e eficiência computacional. Para utilizá-la, é necessário construir uma tarefa do tipo

³<https://git.io/fjGu3>

EvaluateConceptDrift através da aba *Concept Drift*, presente na tela inicial da aplicação.

Para execução dos experimentos deste projeto de mestrado, os parâmetros da tarefa *BasicConceptDriftPerformanceEvaluator* foram configurados conforme a Tabela 4.1.

Tabela 4.1 Configuração da tarefa *BasicConceptDriftPerformanceEvaluator*

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

A classe *BasicConceptDriftPerformanceEvaluator* produz diversos indicadores sobre a performance e a acurácia da técnica de detecção de mudanças analisada. Os indicadores analisados neste trabalho estão listados na Tabela 4.2.

Tabela 4.2 Indicadores analisados

Indicador	Observação
<i>Tempo de Processamento</i>	Tempo médio (segundos) de processamento para cada exemplo.
<i>Mudanças Existentes</i>	Quantidade de mudanças existentes de fato.
<i>Mudanças Detectadas</i>	Quantidade de mudanças detectadas.
<i>Mudanças Reais Detectadas</i>	Quantidade de mudanças detectadas corretamente.
<i>Atraso de Detecção</i>	Quantidade média de instâncias necessárias para detectar uma mudança.

4.4 RBFDriftDetector

Esta seção apresenta um conjunto de experimentos realizados com o objetivo de validar o método de detecção de mudanças de conceito proposto neste trabalho, denominado *RBFDriftDetector*.

Embora os experimentos ainda não sejam suficientes para comprovar a hipótese proposta, a análise empírica aqui apresentada evidencia a importância de analisar a aplicação de redes de função de base radial para detecção de mudanças de conceito em fluxos contínuos de dados.

Para execução dos experimentos, foram utilizados os três conjuntos de dados sintéticos gerados, conforme a seção 4.2. A seguir *ADWIN*,

4.5 CONSIDERAÇÕES FINAIS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

REFERÊNCIAS BIBLIOGRÁFICAS

- ACKERMANN, M. R. et al. Streamkm++: A clustering algorithm for data streams. *J. Exp. Algorithmics*, ACM, New York, NY, USA, v. 17, p. 2.4:2.1–2.4:2.30, maio 2012. ISSN 1084-6654. Disponível em: <http://doi.acm.org/10.1145/2133803.2184450>.
- AGGARWAL, C. C. *Data Streams: Models and Algorithms (Advances in Database Systems)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387287590.
- AGGARWAL, C. C. et al. A framework for clustering evolving data streams. In: *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29*. VLDB Endowment, 2003. (VLDB '03), p. 81–92. ISBN 0-12-722442-4. Disponível em: <http://dl.acm.org/citation.cfm?id=1315451.1315460>.
- AGGARWAL, C. C. et al. On demand classification of data streams. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2004. (KDD '04), p. 503–508. ISBN 1-58113-888-1. Disponível em: <http://doi.acm.org/10.1145/1014052.1014110>.
- AMINIKHANGHAHI, S.; COOK, D. J. A survey of methods for time series change point detection. *Knowl. Inf. Syst.*, Springer-Verlag New York, Inc., New York, NY, USA, v. 51, n. 2, p. 339–367, maio 2017. ISSN 0219-1377. Disponível em: <https://doi.org/10.1007/s10115-016-0987-z>.
- ANKERST, M. et al. Optics: Ordering points to identify the clustering structure. *SIGMOD Rec.*, ACM, New York, NY, USA, v. 28, n. 2, p. 49–60, jun. 1999. ISSN 0163-5808. Disponível em: <http://doi.acm.org/10.1145/304181.304187>.
- Bach, S. H.; Maloof, M. A. Paired learners for concept drift. In: *2008 Eighth IEEE International Conference on Data Mining*. [S.l.: s.n.], 2008. p. 23–32. ISSN 1550-4786.
- BAENA-GARCÍA, M. et al. Early drift detection method. In: *In Fourth International Workshop on Knowledge Discovery from Data Streams*. [S.l.: s.n.], 2006.
- BARBARÁ, D. Requirements for clustering data streams. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 3, n. 2, p. 23–27, jan. 2002. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/507515.507519>.
- BARROS, R. S. M. de et al. RDDM: reactive drift detection method. *Expert Syst. Appl.*, v. 90, p. 344–355, 2017.
- BASSEVILLE, M.; NIKIFOROV, I. V. *Detection of Abrupt Changes: Theory and Application*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993. ISBN 0-13-126780-9.

BAZARGANI, M. H.; NAMEE, B. M. Radial basis function data descriptor (rbfdd) network : An anomaly detection approach. In: . [S.l.: s.n.], 2018.

BIFET, A.; GAVALDÀ, R. Learning from time-changing data with adaptive windowing. In: *SDM*. SIAM, 2007. p. 443–448. ISBN 978-1-61197-277-1. Disponível em: <http://dblp.uni-trier.de/db/conf/sdm/sdm2007.html#BifetG07>.

BIFET, A. et al. Moa: Massive online analysis. *J. Mach. Learn. Res.*, JMLR.org, v. 11, p. 1601–1604, ago. 2010. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=1756006.1859903>.

BIFET, A.; KIRKBY, R. Data stream mining a practical approach. Citeseer, 2009.

BIFET, A. et al. Efficient data stream classification via probabilistic adaptive windows. In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2013. (SAC '13), p. 801–806. ISBN 978-1-4503-1656-9. Disponível em: <http://doi.acm.org/10.1145/2480362.2480516>.

BLANCO, I. I. F. et al. Online and non-parametric drift detection methods based on hoeffding's bounds. *IEEE Trans. Knowl. Data Eng.*, v. 27, n. 3, p. 810–823, 2015.

BRAGA, A.; CARVALHO, A. C.; LUDERMIR, T. B. *Redes Neurais Artificiais: Teoria e aplicações*. LTC Editora, 2007. ISBN 9788521615644. Disponível em: <http://www.worldcat.org/isbn/9788521615644>.

BREIMAN, L. et al. *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

CAO, F. et al. Density-based clustering over an evolving data stream with noise. In: GHOSH, J. et al. (Ed.). *SDM*. SIAM, 2006. p. 328–339. ISBN 978-1-61197-276-4. Disponível em: <http://dblp.uni-trier.de/db/conf/sdm/sdm2006.html#CaoEQZ06>.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 41, n. 3, p. 15:1–15:58, jul. 2009. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/1541880.1541882>.

COHEN, J. et al. Mad skills: New analysis practices for big data. *Proc. VLDB Endow.*, VLDB Endowment, v. 2, n. 2, p. 1481–1492, ago. 2009. ISSN 2150-8097. Disponível em: <https://doi.org/10.14778/1687553.1687576>.

COVER, T. M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *Electronic Computers, IEEE Transactions on*, EC-14, n. 3, p. 326–334, 1965. Disponível em: <http://hebb.mit.edu/courses/9.641/2002/readings/Cover65.pdf>.

DELATTRE, M.; IMBERT, B. *Method for management of data stream exchanges in an autonomic telecommunications network*. [S.l.]: Google Patents, 2015. US Patent 8,949,412.

Ditzler, G.; Polikar, R. Hellinger distance based drift detection for nonstationary environments. In: *2011 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*. [S.l.: s.n.], 2011. p. 41–48.

DOMINGOS, P.; HULTEN, G. Mining high-speed data streams. In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2000. (KDD '00), p. 71–80. ISBN 1-58113-233-6. Disponível em: <http://doi.acm.org/10.1145/347090.347107>.

DREDZE, M.; OATES, T.; PIATKO, C. We're not in kansas anymore: Detecting domain changes in streams. In: . [s.n.], 2010. p. 585–595. Cited By 13. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-80053285630&partnerID=40&md5=2eea89f635e2cbc0920069028e9f7746>.

DRIES, A.; RÜCKERT, U. Adaptive concept drift detection. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, v. 2, n. 5-6, p. 311–327, 2009. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.10054>.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification (2Nd Edition)*. New York, NY, USA: Wiley-Interscience, 2000. ISBN 0471056693.

ESTER, M. et al. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996. (KDD'96), p. 226–231. Disponível em: <http://dl.acm.org/citation.cfm?id=3001460.3001507>.

FARIA, E. R.; GAMA, J. a.; CARVALHO, A. C. P. L. F. Novelty detection algorithm for data streams multi-class problems. In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2013. (SAC '13), p. 795–800. ISBN 978-1-4503-1656-9. Disponível em: <http://doi.acm.org/10.1145/2480362.2480515>.

GAMA, J. *Knowledge Discovery from Data Streams*. 1st. ed. [S.l.]: Chapman & Hall/-CRC, 2010. ISBN 1439826110, 9781439826119.

GAMA, J.; GABER, M. M. *Learning from Data Streams: Processing Techniques in Sensor Networks*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 3642092853, 9783642092855.

GAMA, J. et al. Learning with drift detection. In: BAZZAN, A. L. C.; LABIDI, S. (Ed.). *SBIA*. Springer, 2004. (Lecture Notes in Computer Science, v. 3171), p. 286–295. ISBN 3-540-23237-0. Disponível em: <http://dblp.uni-trier.de/db/conf/sbia/sbia2004.html#GamaMCR04>.

GAMA, J. a.; ROCHA, R.; MEDAS, P. Accurate decision trees for mining high-speed data streams. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2003. (KDD '03),

p. 523–528. ISBN 1-58113-737-0. Disponível em: <http://doi.acm.org/10.1145/956750.956813>.

GAMA, J. a. et al. A survey on concept drift adaptation. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 46, n. 4, p. 44:1–44:37, mar. 2014. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/2523813>.

GONÇALVES, P. M. et al. A comparative study on concept drift detectors. *Expert Systems with Applications*, v. 41, n. 18, p. 8144 – 8156, 2014. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417414004175>.

HALL, M. et al. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/1656274.1656278>.

Hayat, M. Z.; Hashemi, M. R. A dct based approach for detecting novelty and concept drift in data streams. In: *2010 International Conference of Soft Computing and Pattern Recognition*. [S.l.: s.n.], 2010. p. 373–378.

JAIN, A. K.; DUBES, R. C. *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. ISBN 0-13-022278-X.

JIANPING, D.; VENKATESWARLU, R. Speaker recognition using radial basis function neural networks. In: ABRAHAM, A.; KÖPPEN, M. (Ed.). *Hybrid Information Systems*. Heidelberg: Physica-Verlag HD, 2002. p. 57–64. ISBN 978-3-7908-1782-9.

JIN, R.; AGRAWAL, G. Efficient decision tree construction on streaming data. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2003. (KDD '03), p. 571–576. ISBN 1-58113-737-0. Disponível em: <http://doi.acm.org/10.1145/956750.956821>.

KAUFMAN, L.; ROUSSEEUW, P. *Finding Groups in Data: an introduction to cluster analysis*. [S.l.]: Wiley, 1990.

KENKRE, P. S.; PAI, A.; COLACO, L. Real time intrusion detection and prevention system. In: SATAPATHY, S. C. et al. (Ed.). *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*. Cham: Springer International Publishing, 2015. p. 405–411. ISBN 978-3-319-11933-5.

KOLTER, J. Z.; MALOOF, M. A. Dynamic weighted majority: An ensemble method for drifting concepts. *J. Mach. Learn. Res.*, JMLR.org, v. 8, p. 2755–2790, dez. 2007. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=1314498.1390333>.

KOTSIANTIS, S. B. Supervised machine learning: A review of classification techniques. In: *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*. Amsterdam, The Netherlands, The

Netherlands: IOS Press, 2007. p. 3–24. ISBN 978-1-58603-780-2. Disponível em: <http://dl.acm.org/citation.cfm?id=1566770.1566773>.

KRANEN, P. et al. The clustree: Indexing micro-clusters for anytime stream mining. *Knowl. Inf. Syst.*, Springer-Verlag New York, Inc., New York, NY, USA, v. 29, n. 2, p. 249–272, nov. 2011. ISSN 0219-1377. Disponível em: <http://dx.doi.org/10.1007/s10115-010-0342-8>.

KRANJC, J. et al. Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the clowdflows platform. *Information Processing & Management*, v. 51, n. 2, p. 187 – 203, 2015. ISSN 0306-4573. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0306457314000296>.

KUNCHEVA, L. Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. *Proc. Eur. Conf. Artif. Intell.*, p. 5–10, 2008. Cited By 70.

Lee, J.; Magoulès, F. 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*. [S.l.: s.n.], 2012. p. 241–245.

LINDSTROM, P.; NAMEE, B. M.; DELANY, S. J. Drift detection using uncertainty distribution divergence. *Evolving Systems*, v. 4, n. 1, p. 13–25, Mar 2013. ISSN 1868-6486. Disponível em: <https://doi.org/10.1007/s12530-012-9061-6>.

LING, C.; LING-JUN, Z.; LI, T. Stream data classification using improved fisher discriminate analysis. *Journal of Computers*, 01 2009.

LLOYD, S. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, IEEE Press, Piscataway, NJ, USA, v. 28, n. 2, p. 129–137, set. 2006. ISSN 0018-9448. Disponível em: <http://dx.doi.org/10.1109/TIT.1982.1056489>.

MASUD, M. et al. Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Trans. on Knowl. and Data Eng.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 23, n. 6, p. 859–874, jun. 2011. ISSN 1041-4347. Disponível em: <http://dx.doi.org/10.1109/TKDE.2010.61>.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.

NISHIDA, K.; YAMAUCHI, K. Detecting concept drift using statistical testing. In: CORRUBLE, V.; TAKEDA, M.; SUZUKI, E. (Ed.). *Discovery Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 264–269. ISBN 978-3-540-75488-6.

PAGE, E. S. Continuous Inspection Schemes. *Biometrika*, Biometrika Trust, v. 41, n. 1/2, p. 100–115, 1954. ISSN 00063444. Disponível em: <http://dx.doi.org/10.2307/2333009>.

PEARS, R.; SAKTHITHASAN, S.; KOH, Y. S. Detecting concept change in dynamic data streams - A sequential approach based on reservoir sampling. *Machine Learning*, v. 97, n. 3, p. 259–293, 2014.

PESARANGHADER, A. *A Reservoir of Adaptive Algorithms for Online Learning from Evolving Data Streams*. Université d'Ottawa / University of Ottawa, 2018. Disponível em: <http://ruor.uottawa.ca/handle/10393/38190>.

Roberts, S. J.; Penny, W.; Pillot, D. Novelty, confidence and errors in connectionist systems. In: *IEE Colloquium on Intelligent Sensors (Digest No: 1996/261)*. [S.l.: s.n.], 1996. p. 10/1–10/6.

ROBERTS, S. W. Control chart tests based on geometric moving averages. *Technometrics*, American Society for Quality Control and American Statistical Association, Alexandria, Va, USA, v. 42, n. 1, p. 97–101, fev. 2000. ISSN 0040-1706. Disponível em: <http://dx.doi.org/10.2307/1271439>.

ROJAS, R. *Neural Networks: A Systematic Introduction*. Berlin, Heidelberg: Springer-Verlag, 1996. ISBN 3-540-60505-3.

ROSS, G. J. et al. Exponentially weighted moving average charts for detecting concept drift. *Pattern Recogn. Lett.*, Elsevier Science Inc., New York, NY, USA, v. 33, n. 2, p. 191–198, jan. 2012. ISSN 0167-8655. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2011.08.019>.

RYU, J. W. et al. An efficient method of building an ensemble of classifiers in streaming data. In: SRINIVASA, S.; BHATNAGAR, V. (Ed.). *Big Data Analytics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 122–133. ISBN 978-3-642-35542-4.

SCHLIMMER, J. C.; GRANGER, R. H. Incremental learning from noisy data. *Machine Learning*, v. 1, n. 3, p. 317–354, Sep 1986. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00116895>.

SEBASTIÃO, R. et al. Monitoring incremental histogram distribution for change detection in data streams. In: GABER, M. M. et al. (Ed.). *Knowledge Discovery from Sensor Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 25–42. ISBN 978-3-642-12519-5.

SETHI, T. S.; KANTARDZIC, M. On the reliable detection of concept drift from streaming unlabeled data. *Expert Syst. Appl.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 82, n. C, p. 77–99, out. 2017. ISSN 0957-4174. Disponível em: <https://doi.org/10.1016/j.eswa.2017.04.008>.

SETHI, T. S.; KANTARDZIC, M.; HU, H. A grid density based framework for classifying streaming data in the presence of concept drift. *Journal of Intelligent Information Systems*, v. 46, n. 1, p. 179–211, Feb 2016.

SPINOSA, E. J.; CARVALHO, A. P. de Leon F. de; GAMA, J. a. Olinda: A cluster-based approach for detecting novelty and concept drift in data streams. In: *Proceedings of the 2007 ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2007. (SAC '07), p. 448–452. ISBN 1-59593-480-4. Disponível em: <http://doi.acm.org/10.1145/1244002.1244107>.

STREET, W. N.; KIM, Y. A streaming ensemble algorithm (sea) for large-scale classification. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2001. (KDD '01), p. 377–382. ISBN 1-58113-391-X. Disponível em: <http://doi.acm.org/10.1145/502512.502568>.

VAPNIK, V. N. *Statistical Learning Theory*. [S.l.]: Wiley-Interscience, 1998.

WANG, F. et al. Estimating online vacancies in real-time road traffic monitoring with traffic sensor data stream. *Ad Hoc Netw.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 35, n. C, p. 3–13, dez. 2015. ISSN 1570-8705. Disponível em: <https://doi.org/10.1016/j.adhoc.2015.07.003>.

WANG, H. et al. Mining concept-drifting data streams using ensemble classifiers. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2003. (KDD '03), p. 226–235. ISBN 1-58113-737-0. Disponível em: <http://doi.acm.org/10.1145/956750.956778>.

WIDMER, G.; KUBAT, M. Learning in the presence of concept drift and hidden contexts. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 23, n. 1, p. 69–101, abr. 1996. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A:1018046501280>.

ZHOU, L. et al. Fpga based low-latency market data feed handler. In: XU, W. et al. (Ed.). *Computer Engineering and Technology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. p. 69–77. ISBN 978-3-662-45815-0.

ZLIOBAITE, I. Learning under concept drift: an overview. *CoRR*, abs/1010.4784, 2010. Disponível em: <http://arxiv.org/abs/1010.4784>.

ZWOLENSKI, M.; WEATHERILL, L. The digital universe rich data and the increasing value of the internet of things. *Australian Journal of Telecommunications and the Digital Economy*, v. 2, 10 2014.