



Pós-Graduação em Ciência da Computação

Gustavo Henrique Ferreira de Miranda Oliveira

**PREVISÃO DE SÉRIES TEMPORAIS NA PRESENÇA DE MUDANÇA  
DE CONCEITO: UMA ABORDAGEM BASEADA EM PSO.**



Universidade Federal de Pernambuco  
posgraduacao@cin.ufpe.br  
<www.cin.ufpe.br/~posgraduacao>

RECIFE  
2018

Gustavo Henrique Ferreira de Miranda Oliveira

**PREVISÃO DE SÉRIES TEMPORAIS NA PRESENÇA DE MUDANÇA  
DE CONCEITO: UMA ABORDAGEM BASEADA EM PSO.**

*Trabalho apresentado ao Programa de Pós-graduação em  
Ciência da Computação do Centro de Informática da Univer-  
sidade Federal de Pernambuco como requisito parcial para  
obtenção do grau de Mestre em Ciência da Computação.*

Orientador: *Adriano Lorena Inácio de Oliveira*

Co-Orientador: *Rodolfo Carneiro Cavalcante*

RECIFE

2018

Catálogo na fonte  
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

O48p Oliveira, Gustavo Henrique Ferreira de Miranda  
Previsão de séries temporais na presença de mudança de conceito: uma abordagem baseada em PSO / Gustavo Henrique Ferreira de Miranda Oliveira. – 2018.  
76 f.: il., fig., tab.

Orientador: Adriano Lorena Inácio de Oliveira.  
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2018.  
Inclui referências.

1. Inteligência artificial. 2. Previsão de séries temporais. 3. Redes neurais artificiais. I. Oliveira, Adriano Lorena Inácio de (orientador). II. Título.

006.3                      CDD (23. ed.)                      UFPE- MEI 2018-057

Dissertação de Mestrado apresentada por **Gustavo Henrique Ferreira de Miranda Oliveira** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “**PREVISÃO DE SÉRIES TEMPORAIS NA PRESENÇA DE MUDANÇA DE CONCEITO: UMA ABORDAGEM BASEADA EM PSO**”, orientada pelo **Prof. Adriano Lorena Inácio de Oliveira** e aprovada pela Banca Examinadora formada pelos professores:

---

Prof. Paulo Salgado Gomes de Mattos Neto  
Centro de Informática / UFPE

---

Prof. João Fausto Lorenzato de Oliveira  
Departamento de Licenciatura em Computação / UPE

---

Prof. Adriano Lorena Inácio de Oliveira  
Centro de Informática / UFPE

Visto e permitida a impressão.  
Recife, 19 de fevereiro de 2018.

---

**Profa. Edna Natividade da Silva Barros**  
Coordenadora da Pós-Graduação em Ciência da Computação do  
Centro de Informática da Universidade Federal de Pernambuco.

*Eu dedico esse trabalho aos meus pais, que sempre me orientaram, incentivaram, encorajaram e nunca mediram esforços para me ajudar independentemente da situação.*

## **Agradecimentos**

Acima de tudo agradeço a Deus, por me conceder muita força de vontade, determinação e inspiração para a construção desse trabalho.

Agradeço a minha família, meu pai Gilberto, minha mãe Maria Luci, minha irmã Marília, minha tia-irmã Luciana e minha namorada Hellen Carolina, por todo o apoio durante esses anos e em especial ao meu irmão Augusto, pela companhia e parceria durante os anos dessa pesquisa.

Agradeço ao meu orientador Adriano Lorena por toda confiança, por todas as orientações, reuniões e por me apresentar pessoas importantíssimas para a construção desse trabalho. Agradeço ao meu co-orientador Rodolfo Carneiro, por sempre tirar minhas dúvidas, pela parceria e revisão de todos os meus trabalhos.

Agradeço aos meus amigos de pesquisa: Alison Pereira, Ellen Polliana, George Gomes, Leandro Minku, Fausto Lorenzato e Diego Santos pela parceria e colaboração nos trabalhos ao longo desses anos, cada reunião foi uma aula para mim.

Agradeço aos meus tios Jailton e Rogério, por abdicar de parte do seu tempo para me trazer em Recife e me levar para Bom Conselho, sempre que foi necessário. Agradeço aos meus tios Joseval, Flávia, Josemir e minha prima Mirella, pelas hospedagens e presenças ao longo desses anos em Recife.

Agradeço aos meus amigos do laboratório de pesquisa: Hugo Souza, Leandro Marques, Reinaldo Antônio, Francisco de Assis, Vanderberger e Osmar, por tornarem o meu dia a dia mais divertido e pelas caronas no horário de almoço.

Agradeço também, aos meus professores do Centro de Informática, Francisco, Renata, Adriano, Cleber e Fábio pelas aulas e pelo crescimento intelectual que me forneceram.

Agradeço aos amigos que fiz em Recife Eraylson Galdino, Emanuel Cordeiro, Germano Esteves, Luiz Vicentin e João Lucas, pela convivência durante esses anos.

Por fim, agradeço a todos os meus amigos e familiares que estavam torcendo por mim durante essa jornada.

*O cavalo prepara-se para o dia da batalha, mas do Senhor vem a vitória.*

—PROVÉRBIOS 21:31

## Resumo

Previsão de séries temporais é um problema com muitas aplicações. No entanto, em muitos domínios, tal como mercado de ações, o processo gerador das observações das séries temporais pode mudar, tornando os modelos de previsão obsoletos. Esse problema é conhecido como mudança de conceito (Concept Drift). Abordagens para previsão de séries temporais devem ser capazes de detectar e reagir a uma mudança de conceito em tempo hábil, de modo que o modelo de previsão possa ser atualizado o mais rápido possível. Apesar do fato de que o problema da mudança de conceito ser bem investigado na literatura, principalmente em problemas de classificação de dados, pouco esforço tem sido feito para resolver esse problema para previsão de séries temporais até agora. A fim de preencher essa lacuna, esse trabalho propõe três métodos para detecção de mudança de conceito e duas arquiteturas que usam inteligência de enxames da detecção de mudança de conceito até a adaptação depois da mudança em previsão de séries temporais. Os sistemas monitoram o erro através de Testes de Detecção de Mudanças (CDT's) por meio de vários modelos de previsão gerados por um método de inteligência de enxames, chamado de Algoritmo de Enxame de Partículas (PSO). Para adaptação, técnicas de Otimização Dinâmica Evolucionária (EDO), como o uso de memória e seleção das melhores partículas, foram usadas para melhorar a adaptabilidade dos modelos. Experimentos mostraram que em relação a detecção de mudança de conceito os métodos propostos alcançaram os melhores resultados em séries que possuem mudanças abruptas. Além disso, em relação a acurácia de previsão, os métodos com adaptação a mudança de conceito obtiveram os melhores resultados segundo o ranking de Friedman, apresentando diferença estatística dos demais métodos experimentados.

**Palavras-chave:** Previsão de Séries Temporais. Mudança de Conceito. Inteligência de Enxames. Redes Neurais Artificiais.

## **Abstract**

Time series forecasting is a problem with many applications. However, in many domains, such as stock market, the underlying generating process of the time series observations may change, making forecasting models obsolete. This problem is known as Concept Drift. Approaches for time series forecasting should be able to detect and react to concept drift in a timely manner, so that the forecasting model can be updated as soon as possible. Despite the fact that the concept drift problem is well investigated in the literature, mainly in problems of data classification, little effort has been made to solve this problem for time series forecasting so far. Therefore, to overcome the aforementioned problems and challenges, we propose three methods of concept drift detection and two architectures that use swarm intelligence from the detection of concept change to adaptation after the change in time series prediction. The systems monitor the error, through Change Detection Tests (CDT's), through several prediction models generated by a swarm intelligence method, called the Particle Swarm Optimization (PSO). For adaptation, techniques of Evolutionary Dynamic Optimization (EDO), as the use of memory and selection of best particles, are used to improve the adaptability of the models. Experiments show that in relation of detection of concept drift the proposed methods reach the best results in time series which that have abrupt changes. Besides that, in relation to forecasting accuracy, the methods with adaptation to concept drift got the best results according to the Friedman rank, showing statistic difference against the other experimented methods.

**Keywords:** Time Series Forecasting. Concept Drift. Swarm Intelligence. Artificial Neural Network.

## Lista de Figuras

2.1	Representação dos componentes de tendência e sazonalidade. . . . .	22
2.2	Representação dos componentes de estacionariedade e volatilidade. . . . .	23
2.3	Estrutura de uma MLP para previsão de séries temporais. . . . .	25
2.4	Codificação de uma partícula para a otimização dos pesos de uma rede neural MLP. .	27
2.5	Mudança de conceito real na série financeira Down Jones Industrial Average de 03 de julho de 1972 até 30 de junho de 1975 . . . . .	29
2.6	A esquerda um exemplo de mudança de conceito real. A direita um exemplo de mudança de conceito virtual . . . . .	30
2.7	Representação da mudança de conceito na ótica da otimização. . . . .	31
4.1	Fluxograma do procedimento geral dos sistemas com mecanismos de adaptação. . .	45
4.2	Distribuição de erros de partículas. . . . .	46
5.1	Exemplos de Séries Artificiais. . . . .	54
5.2	Séries reais com mudanças conhecidas. . . . .	57
5.3	Séries temporais financeiras. . . . .	58
5.4	Média das métricas FPR e DD para todas as séries artificiais. . . . .	59
5.5	Curva FPR x DD para as séries artificiais. . . . .	60
5.6	Teste Nemenyi para os modelos treinados e avaliados nas séries artificiais com mudanças de conceitos. . . . .	61
5.7	Curva FPR x DD para as séries reais com conceitos conhecidos. . . . .	63
5.8	Teste Nemenyi para os modelos treinados e avaliados nas séries reais financeiras. . .	65
5.9	Execução do método P-IDPSO-ELM-DDSV para as séries reais financeiras. . . . .	66

## Lista de Tabelas

5.1 Parâmetros usados nos modelos das séries temporais . . . . .	55
5.2 Erro médio absoluto para as séries artificiais. . . . .	67
5.3 Erro médio absoluto para as séries reais financeiras . . . . .	67
5.4 Erro médio absoluto para as séries reais com mudanças conhecidas. . . . .	67

## Lista de Acrônimos

<b>ARIMA</b>	Modelo Autorregressivo Integrado de Médias Móveis (Autoregressive Integrated Moving Average)
<b>CDT</b>	Teste de Controle de Detecção (Change Detection Method)
<b>DD</b>	Atrasos na Detecção (Drift Delay)
<b>DDM</b>	Método de Detecção de Mudanças (Drift Detection Method)
<b>DOP</b>	Problemas de Otimização Dinâmicos (Dynamic Optimization Problems)
<b>DyFor GP</b>	Programação Genética para Previsão Dinâmica (Dynamic Forecasting Genetic Program)
<b>ECDD</b>	Média Móvel Exponencial Ponderada Para Detectar Mudanças de Conceito (Exponentially Weighted Moving Average Charts for Detecting Concept Drift)
<b>EDO</b>	Otimização Dinâmica Evolucionária (Evolutionary Dynamic Optimization)
<b>ELM</b>	Máquina de Aprendizagem Extrema (Extreme Learning Machine)
<b>EWMA</b>	Média Móvel Exponencial Ponderada (Exponentially Weighted Moving Average)
<b>FEDD</b>	Extração de Características Para Detecção Explícita de Mudança de Conceito (Feature Extraction for Explicit Concept Drift Detection)
<b>FPR</b>	Detecções Falso Positivas (False Positive Rate)
<b>GARCH</b>	Modelo de Heterocedasticidade Condicional Autorregressivo Generalizado (Generalized Autoregressive Conditional Heteroskedasticity)
<b>IDPSO</b>	Algoritmo de Enxames de Partículas Auto-Adaptativo (Improved Self-adaptive Particle Swarm Optimization)
<b>IDPSO-ELM-B</b>	Detecção de Mudança de Conceito Baseado no Comportamento do Enxame (Concept Drift Detection Based on Swarm Behavior)
<b>IDPSO-ELM-S</b>	Detecção de Mudança de Conceito Baseada em Sensores (Concept Drift Detection Based on Sensors)
<b>IDPSO-ELM-DDSV</b>	Detecção de Mudança de Conceito Baseada no Voto Majoritário de Sensores (Concept Drift Detection Based on Majority Vote of Sensors)
<b>MAE</b>	Erro Médio Absoluto (Mean Absolute Error)
<b>MPB</b>	Função de Referência de Picos Móveis (Moving Peaks Benchmark)
<b>M-IDPSO-ELM-DDSV</b>	Método de Adaptação à Mudança de Conceito Baseado em Memória (Memory-based Concept Drift Adaptation Method)
<b>OS-ELM</b>	Máquina de Aprendizagem Extrema On-line (On-line Extreme Learning Machine)

<b>P-IDPSO-ELM-DDSV</b>	Método de Adaptação à Mudança de Conceito Baseado em Partículas (Particle-based Concept Drift Adaptation Method)
<b>PSO</b>	Algoritmo de Enxames de Partículas (Particle Swarm Optimization)
<b>RNA</b>	Redes Neurais Artificiais (Artificial Neural Network)
<b>RPSO</b>	Reiniciar o Algoritmo de Enxame de Partículas (Reinitialize Particle Swarm Optimization)
<b>SARIMA</b>	Modelo Autorregressivo Integrado de Médias Móveis Sazonal (Seasonal Autoregressive Integrated Moving Average)
<b>SVR</b>	Regressão por Vetores de Suporte (Support Vector Regression)
<b>IW</b>	Marca de Água Inteligente (Intelligent Watermarking)

## Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	Motivação . . . . .	15
1.2	Formulação do Problema . . . . .	16
1.3	Objetivos, Perguntas de Pesquisa e Hipóteses . . . . .	17
1.4	Estrutura do Documento . . . . .	19
<b>2</b>	<b>FUNDAMENTOS</b>	<b>21</b>
2.1	Análise de Séries Temporais . . . . .	21
2.1.1	Descrição, Classificação e Definição de Séries Temporais . . . . .	22
2.1.2	Modelagem e Previsão de Séries Temporais. . . . .	23
2.1.2.1	<i>Previsão de Séries Temporais com Redes Neurais Artificiais</i> . . . . .	24
2.1.2.2	<i>Otimização Aplicada a Redes Neurais Artificiais</i> . . . . .	26
2.2	Mudança de Conceito . . . . .	28
2.2.1	Mudança de Conceito em Previsão de Séries Temporais . . . . .	29
2.2.2	Mudança de Conceito em Problemas de Otimização . . . . .	31
2.3	Adaptação a Mudança de Conceito . . . . .	31
2.3.1	Métodos Passivos de Adaptação . . . . .	32
2.3.2	Métodos Ativos de Adaptação . . . . .	32
2.4	Síntese do Capítulo de Fundamentos . . . . .	33
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>34</b>
3.1	Detecção de Mudança de Conceito . . . . .	34
3.1.1	Detecção de Mudança de Conceito em Séries Temporais . . . . .	34
3.1.2	Detecção de Mudança de Conceito em Problemas de Otimização . . . . .	36
3.2	Estratégias de Adaptação após a Mudança de Conceito . . . . .	37
3.2.1	Estratégias de Adaptação para Previsão de Séries Temporais . . . . .	37
3.2.2	Estratégias de Adaptação para Problemas de Otimização . . . . .	38
3.3	Síntese e Discussões do Capítulo de Trabalhos Relacionados . . . . .	38
<b>4</b>	<b>MÉTODO PROPOSTO</b>	<b>40</b>
4.1	Treinamento da Rede Neural Artificial . . . . .	40
4.2	Detecção de Mudança de Conceito Baseado em Inteligência de Enxames . . . . .	41
4.2.1	Visão Geral dos Métodos de Detecção . . . . .	41
4.2.2	Detecção de Mudança Baseado no Comportamento do Enxame . . . . .	42
4.2.3	Detecção de Mudança Baseado em Sensores . . . . .	43
4.2.4	Detecção de Mudança Baseado no Voto Majoritário dos Sensores . . . . .	44

4.3	<b>Métodos de Adaptação Após a Detecção de Mudança de Conceito</b>	44
4.3.1	Visão Geral dos Métodos de Adaptação	44
4.3.2	Adaptação Por Partículas	45
4.3.3	Adaptação Por Memória	47
4.4	<b>Síntese do Método Proposto</b>	49
<b>5</b>	<b>EXPERIMENTOS</b>	<b>50</b>
5.1	<b>Configuração Experimental</b>	50
5.1.1	Habilidade de Detecção de Mudança de Conceito	50
5.1.2	Habilidade de Previsão de Séries Temporais	52
5.1.3	Descrição dos Dados	53
5.1.3.1	<i>Séries Artificiais</i>	53
5.1.3.2	<i>Séries Reais Com Mudanças de Conceitos Conhecidas</i>	56
5.1.3.3	<i>Séries Reais Financeiras</i>	57
5.2	<b>Resultados</b>	58
5.2.1	Avaliação para as Séries Artificiais	58
5.2.1.1	<i>Habilidade de Detecção</i>	58
5.2.1.2	<i>Habilidade de Previsão</i>	61
5.2.2	Avaliação para as Séries Reais Com Mudanças de Conceitos Conhecidas	62
5.2.2.1	<i>Habilidade de Detecção</i>	62
5.2.2.2	<i>Habilidade de Previsão</i>	64
5.2.3	Avaliação para as Séries Reais Financeiras	64
5.2.3.1	<i>Habilidade de Previsão</i>	64
5.3	<b>Síntese do Capítulo de Experimentos</b>	68
<b>6</b>	<b>CONCLUSÃO</b>	<b>69</b>
6.1	<b>Limitações do Método Proposto</b>	70
6.2	<b>Trabalhos Futuros</b>	70
	<b>Referências</b>	<b>72</b>

# 1

## INTRODUÇÃO

Uma série temporal é uma coleção de observações medidas sequencialmente ao longo do tempo. Nesse tipo de base de dados, as observações tipicamente apresentam uma correlação serial, ou seja, as observações possuem dependência de uma para outra. Vários processos do mundo real medem dados ao longo do tempo que podem ser modelados como séries temporais, tal como folha de pagamento, movimento do preço de ações, taxas de câmbio, temperaturas de uma cidade, entre outros (CAVALCANTE; OLIVEIRA, 2015)

Na literatura, a maioria das abordagens propostas para análise de séries temporais modelam os dados da série de forma estática. Essas abordagens tipicamente processam a série temporal de modo *off-line*, através do processamento dos dados históricos em várias etapas. No entanto, na maioria das aplicações de séries temporais do mundo real, os dados chegam sequencialmente em um fluxo de dados. Nesse cenário de transmissão, os dados podem fluir em alta velocidade e evoluir ao longo do tempo (DITZLER et al., 2015).

Os fluxos de dados dinâmicos apresentam vários desafios para modelos preditivos (GAMA et al., 2014), uma vez que o processo de geração pode mudar ao longo do tempo, fazendo com que as observações históricas não sejam úteis para definir comportamentos futuros. Mudanças no processo de geração dos dados são conhecidas como mudanças de conceito (MINKU; YAO, 2012). Esse fenômeno é muito comum em fluxos de dados de séries temporais reais. Por exemplo, séries podem mudar devido a mudanças políticas, fatores econômicos ou mudanças na psicologia dos investidores (CAVALCANTE; MINKU; OLIVEIRA, 2016).

Portanto, devido a importância da mudança de conceito em séries temporais e os sérios problemas que surgem para as abordagens de previsão tradicionais, este trabalho investiga o desenvolvimento de métodos de aprendizagem mais sofisticados, capazes tanto de detectar como de adaptar-se precisamente às mudanças em tempo real.

### 1.1 Motivação

O problema da mudança de conceito tem sido amplamente estudado na literatura, principalmente em problemas de classificação de dados (CAVALCANTE et al., 2016). Esse problema tem sido tratado por duas principais classes de métodos, que são os métodos ativos e passivos de

controle de mudança de conceito (GAMA et al., 2004).

Os métodos passivos de adaptação, adaptam o modelo aprendido a cada instância de chegada sem qualquer detecção explícita de mudanças (GAMA et al., 2014). Esses métodos podem não ser eficientes, pois continuam adaptando o modelo aprendido mesmo em casos onde não há mudanças no processo de geração dos dados. Por outro lado, os métodos ativos de adaptação usam um mecanismo para detectar explicitamente as mudanças. Estes podem ser mais eficientes, pois implementam mecanismos de detecção e reação (OLIVEIRA et al., 2017).

Apesar dos métodos ativos de adaptação serem computacionalmente mais eficientes, o seu uso implica em enfrentar uma série de desafios. O primeiro deles é melhorar a sua acurácia de detecção de mudança de conceito, de modo a fornecer pontos de mudança para o usuário e o auxiliar na tomada de decisões de investimentos. O segundo é utilizar uma estratégia eficiente de adaptação após a mudança de conceito, de modo a minimizar a degradação do modelo e continuar fornecendo previsões confiáveis (OLIVEIRA et al., 2017).

Portanto, a principal motivação desse trabalho é o desenvolvimento de sistemas capazes de lidar com a mudança de conceito, bem como os problemas encontrados pelos métodos ativos de adaptação da literatura. Para isso, buscamos investigar o procedimento de técnicas da literatura que operam bem em determinadas situações e que podem apoiar essa investigação, como por exemplo: o treinamento do modelo preditivo através de inteligência de enxames, pois na literatura essas técnicas geralmente superam os processos de treinamento convencionais em Redes Neurais Artificiais (Artificial Neural Network) (RNA) (RAKITIANSKAIA; ENGELBRECHT, 2009); A detecção de mudança de conceito baseada em vários métodos, já que esse tipo de abordagem permite uma informação mais confiável de quando houve a mudança de conceito, uma vez que minimiza-se as chances de problemas causados pelo mal treinamento de apenas um modelo de previsão (OLIVEIRA et al., 2017); Finalmente, técnicas de adaptação a mudança de conceito, que podem tornar o modelo mais robusto as mudanças de conceito, uma vez que o modelo de previsão pode ser atualizado constantemente, podendo diminuir a sua degradação ao longo do tempo (NGUYEN; YANG; BRANKE, 2012).

## 1.2 Formulação do Problema

Séries temporais são um tipo especial de fluxo de dados que geralmente apresentam mudança de conceito (CAVALCANTE; MINKU; OLIVEIRA, 2016). A mudança de conceito pode ser definida como uma mudança na relação entre os atributos de entrada e os seus respectivos alvos (DITZLER et al., 2015).

Para entendermos esse problema, tomemos como exemplo um modelo de treinamento baseado em um conjunto de entradas  $X$  e um conjunto de alvos  $y$ , tal que  $X_b, y_b = (b = 1, 2, 3, \dots, n)$ , cuja a distribuição de probabilidade conjunta é  $P(X, y) = P(y|X)P(X)$ , onde  $P(X)$  é a distribuição dos componentes de entrada e  $P(y|X)$  representa um modelo de aprendizagem de máquina que relaciona o conjunto de entradas  $X$  com o conjunto de saídas  $y$ . Sabendo disso, mudanças podem

ocorrer tanto em  $P(X)$ , quanto em  $P(y|X)$  (YAMAUCHI, 2010). Quando ocorrem em  $P(X)$  isso indica que os dados mudaram de distribuição, mas a previsão do modelo não é afetada. Já se ocorrem em  $P(y|X)$ , os modelos de previsão tornam-se obsoletos e ineficientes em prever os comportamentos futuros da série (GAMA et al., 2014).

### 1.3 Objetivos, Perguntas de Pesquisa e Hipóteses

O principal objetivo dessa pesquisa é investigar e propor sistemas de aprendizagem adaptativos para modelar e realizar previsões em séries temporais que eventualmente são afetadas por mudanças de conceito. Motivados a superar os principais problemas de aprendizagem adaptativa descritos na Seção 1.1, os algoritmos propostos devem ser:

1. capazes de lidar com mudanças que ocorrem em  $P(y|X)$ , uma vez os módulos monitoram a relação de entrada e saída dos atributos através do erro do modelo preditivo.
2. precisos em detectar mudanças de conceito, uma vez que se baseiam no monitoramento de vários modelos gerados por inteligência de enxames;
3. transparentes para o usuário, uma vez que implementam mecanismos explícitos de detecção da mudança;
4. capazes de lidar com mudanças de conceitos recorrentes, uma vez que utilizam técnicas de otimização dinâmica;
5. precisos na previsão de séries temporais com padrões em evolução, uma vez que o modelo é atualizado sempre que o seu erro se degrada;
6. eficientes, uma vez que atualizam o modelo aprendido somente após detectar uma mudança na série.

Para atingir o objetivo de construir os sistemas de aprendizagem adaptativos, algumas questões de pesquisa surgem e são respondidas nesse trabalho:

1. Métodos ativos que lidam com mudanças de conceito monitoram algum aspecto do fluxo de dados para detectar explicitamente mudanças. Uma das estratégias mais utilizadas é monitorar o erro do classificador ou regressor em uma abordagem online. A principal suposição desses métodos é que, quando uma mudança acontece, o modelo aprendido não é mais eficaz na descrição do comportamento dos dados e seu erro de previsão deve aumentar. Nesses trabalhos apenas um modelo individual é monitorado, mas em Problemas de Otimização Dinâmicos (Dynamic Optimization Problems) (DOP), monitora-se por meio do Algoritmo de Enxames de Partículas (Particle Swarm Optimization) (PSO) (KENNEDY, 2011) várias características de

um enxame. Um enxame é definido por um conjunto de modelos treinados em um mesmo conceito, chamadas de partículas. Ao monitorar diversas características (por exemplo: média do fitness do enxame (comportamento) e o fitness de cada partícula (sensores)) de um enxame é possível monitorar o erro do modelo de várias perspectivas. **Portanto, monitorar mais de um modelo melhora a capacidade de detecção? Monitorar mais características de um enxame beneficia a detecção? Até que ponto o monitoramento dessas soluções beneficiam a detecção?**

2. Quando o fenômeno da mudança de conceito ocorre, os modelos de previsão podem se tornar obsoletos. Então, modelos que lidam com esse problema devem ser capazes de se adaptar de forma rápida e precisa às possíveis mudanças (MINKU; YAO, 2012). Para lidar com a previsão de séries temporais, sistemas de aprendizagem híbridos baseados em lote usam o PSO como algoritmo de otimização para construir modelos e melhorar a sua acurácia de previsão (HAN et al., 2016). Esses algoritmos utilizam várias partículas que representam variações da solução (HAN et al., 2016), (DING et al., 2015), (YEH, 2013) para o atual conceito. Então, cada partícula tem uma diferente capacidade de solução para o problema treinado, o que pode implicar em uma diferente capacidade de se adaptar ao novo conceito. **Portanto, o PSO, pode ser usado para obter uma melhoria na capacidade de adaptação ao novo conceito em previsão de séries temporais? Quais tipos de mudança de conceito se beneficiam? Até que ponto elas podem se beneficiar?**
3. O processo de aprendizagem em fluxos de dados é contínuo e depois de algum tempo os conceitos mudam e modelos aprendidos são esquecidos. Estas mudanças, segundo Hosseini, Ahmadi e Beigy (2013), podem ser categorizadas nos seguintes grupos: (i) mudanças abruptas; (ii) graduais; (iii) incrementais e (iv) recorrentes. Em conceitos recorrentes, um conceito antigo pode reaparecer e um modelo recente, por ser treinado em um conceito antigo, pode realizar uma má previsão devido sua solução desatualizada. Então, se for possível usar conhecimento aprendido anteriormente, este problema pode ser resolvido de forma adequada (HOSSEINI; AHMADI; BEIGY, 2013). Além disso, reutilizar informações passadas pode fazer sentido em outros tipos de mudanças. Segundo Minku e Yao (2012) com estas informações é possível ajudar a adaptação a um conceito que não seja recorrente, usando estratégias de conjunto de modelos com diversidade. Em algoritmos de otimização dinâmica soluções de enxames podem ser armazenadas para serem usadas futuramente em memórias de longo prazo (MAVROVOUNIOTIS; LI; YANG, 2017). Estas soluções podem ajudar as partículas a se adaptarem mais rápido a um conceito novo, como também evitar que elas fiquem presas em ótimos locais. **Portanto, o uso de soluções de enxames passados pode levar a uma aceleração na adaptação de conceitos recorrentes em previsão de séries temporais? Em que medida isso**

**pode ajudar na adaptação? Outros tipos de mudanças de conceito podem se beneficiar disso?**

As principais hipóteses formuladas nesse trabalho estão organizadas para responder as perguntas de pesquisa listadas acima, respectivamente:

1. Acredita-se que ao monitorar diversas variações do erro, como um grupo de soluções individualmente ou a média desse grupo por meio de enxames, pode-se dar mais consistência em afirmar que houve uma mudança de conceito, diminuindo a probabilidade de detecções falso positivas.
2. Acredita-se que explorar os modelos criados pelo PSO para a adaptação pode melhorar a acurácia de previsão dos modelos, pois há as chances de pelo menos um modelo ser suficiente para o novo conceito.
3. Acredita-se que caso existam semelhanças entre conceitos antigos e conceitos novos, o uso de soluções passadas pode contribuir para melhorar a acurácia de previsão após a ocorrência da mudança de conceito.

#### **1.4 Estrutura do Documento**

Esse trabalho está organizado da seguinte maneira: Nesse capítulo, apresentamos as motivações desse trabalho, a formulação do problema, os objetivos e as principais questões e hipóteses de pesquisa.

No capítulo 2, contextualizamos algumas abordagens para análise de séries temporais, como os elementos que compõem uma série e como construir um modelo de previsão. Discutimos também o fenômeno da mudança de conceito, bem como ele ocorre e como se comporta nas áreas de previsão de séries temporais e problemas de otimização. Ademais, discutimos sobre as principais abordagens que são utilizadas para lidar com a mudança de conceito.

O capítulo 3, apresenta os principais estudos da literatura que se relacionam com este trabalho. Nesse capítulo, discutimos as vantagens e desvantagens desse métodos tanto para a área de detecção de mudança de conceito, bem como estratégias de adaptação após a mudança.

O capítulo 4, apresenta os sistemas propostos, para isso apresentamos uma visão geral dos sistemas e em seguida detalhamos cada um dos seus componentes, como por exemplo: o processo de treinamento do modelo; abordagens de detecção de mudança e estratégias de adaptação ao novo ambiente, dado uma mudança.

No capítulo 5, descrevemos a forma como avaliamos os algoritmos propostos e também os conjuntos de dados utilizados. Além disso, discutimos como os sistemas propostos responderam as questões de pesquisas formuladas no capítulo 1.

E por fim, o capítulo 6 resume os principais resultados e contribuições obtidos nesse trabalho, como também indica algumas direções para pesquisas futuras.

Parte do material usado nesse trabalho foi publicado no seguinte artigo:

1. **OLIVEIRA, G. H. F. M.; CAVALCANTE, R. C.; CABRAL, G. G.; MINKU, L. L.; OLIVEIRA, ADRIANO L. I.** Time Series Forecasting in the Presence of Concept Drift: A PSO-based Approach. In: 29th IEEE International Conference on Tools with Artificial Intelligence (ICTAI2017)

# 2

## FUNDAMENTOS

Neste capítulo são contextualizadas as seguintes áreas de pesquisa: 2.1: análise de séries temporais; 2.2: mudança de conceito; e 2.3: adaptação a mudança de conceito. Na análise de séries temporais, são discutidos conceitos que ajudam a entender como os dados se comportam e são apresentadas técnicas que ajudam a criar modelos para previsão de séries temporais. Na mudança de conceito, serão contextualizados como esse fenômeno ocorre e também afeta as áreas de aplicação de previsão de séries temporais e problemas de otimização. Na adaptação à mudança de conceito, serão apresentadas as principais abordagens que são utilizadas para lidar com o problema da mudança de conceito.

### 2.1 Análise de Séries Temporais

Uma série temporal é uma coleção de observações medidas sequencialmente ao longo do tempo. Nesse tipo de base de dados, as observações são ordenadas no tempo e tipicamente apresentam uma correlação serial (OLIVEIRA et al., 2017). Geralmente, as séries temporais podem ser representadas como  $y_t = [y_1, y_2, y_3, \dots, y_n]$ , onde  $n$  é o seu comprimento e  $y_t$  representa a observação no tempo  $t$  (ADHIKARI, 2015).

Vários processos do mundo real medem suas variáveis ao longo do tempo em formato de série temporal, como: folha de pagamento de uma empresa (OLIVEIRA; MEIRA, 2006), movimentos dos preços das ações (CAVALCANTE; OLIVEIRA, 2014), taxas de câmbio (BRITO; OLIVEIRA, 2014), temperaturas de uma cidade (LEE et al., 2006), eletroencefalograma (ANDRZEJAK et al., 2001), entre outros.

Modelar um processo real como séries temporais possibilita entender como os dados se comportam em determinados períodos de tempos (HAMILTON, 1994). Com isso, é possível identificar os fenômenos que ocorrem na área em questão, uma vez que consegue-se observar em quais instantes de tempos os dados variam. Além disso, através do estudo das observações históricas é possível modelar o processo gerador dessas observações (OLIVEIRA; LUDERMIR, 2016), tornando possível realizar previsões (FIRMINO; NETO; FERREIRA, 2015) e ajudar pessoas e instituições na tomada de decisões. De acordo com Palit e Popovic (2006) os principais estudos da análise de séries temporais são: (i) Descrição, Classificação, Definição; (ii)

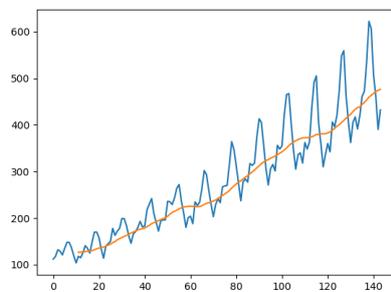
## Modelagem e Previsão de Séries Temporais.

## 2.1.1 Descrição, Classificação e Definição de Séries Temporais

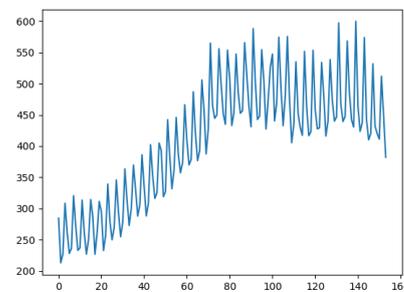
O estudo da descrição, classificação e definição de séries temporais tem como objetivo identificar e entender alguns componentes contidos na série, bem como os seus principais comportamentos. Segundo Morettin e Toloí (2006), séries temporais possuem comportamentos distintos e eles podem ser integrados em quatro elementos: (i) tendência, (ii) sazonalidade, (iii) estacionariedade e (iv) volatilidade.

Tendência é caracterizado por um aumento ou decrescimento do nível dos valores das observações ao longo do tempo (COWPERTWAIT; METCALFE, 2009). Esse comportamento pode ser encontrado em séries de ações de empresas que obtiveram um crescimento ou decrescimento contínuo dos seus lucros, por exemplo séries de vendas. Quando observado pela ótica da análise, esse componente pode indicar o sentido que provavelmente os dados futuros tenderão a ir. Um exemplo desse componente pode ser visto na figura 2.1(a).

Sazonalidade consiste em um movimento ondulatório de curta duração das observações da série temporal (MORETTIN; TOLOI, 2006). Esse comportamento geralmente pode ser observado em séries climáticas e de vendas que recebem alterações de acordo com as temporadas climáticas, como pode ser visto na figura 2.1(b). Ao se analisar esse componente pode-se entender que determinados padrões podem ressurgir em conformidade com a época do ano.



(a) Componente de tendência.



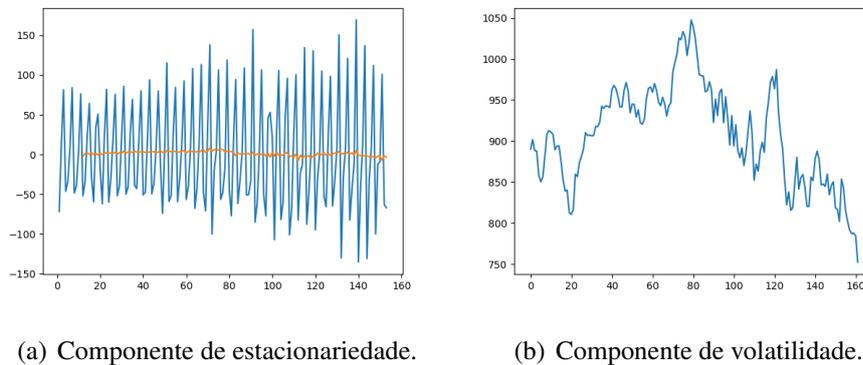
(b) Componente de sazonalidade mais tendência.

**Figura 2.1:** Representação dos componentes de tendência e sazonalidade.

Estacionariedade é um componente que diz respeito quando as observações da série temporal se desenvolvem no tempo aleatoriamente ao redor de uma média constante, refletindo alguma forma de equilíbrio estável (MORETTIN, 2008) como ilustrado na figura 2.2(a). Esse tipo de comportamento é frequente em aplicações de reconhecimento de áudio (MUDA; BEGAM; ELAMVAZUTHI, 2010). Apesar de muitas séries não serem estacionárias, esse comportamento pode ser obtido através de operações matemáticas, como a diferença entre observações de um tempo a frente. Através dessas operações, pode-se remover, ou subtrair tendências

(COWPERTWAIT; METCALFE, 2009), o que pode facilitar na modelagem do processo gerador das observações.

Por fim, o elemento chamado de volatilidade é caracterizado pela distribuição irregular dos dados ou variâncias no processo gerador das observações. Esse comportamento pode ser encontrado em séries financeiras e é um dos parâmetros de maior relevância no apreçamento de ações (MORETTIN, 2008). Esse fenômeno vem acompanhado de aleatoriedade o que dificulta o processo de previsão através de modelos matemáticos. Um exemplo desse componente pode ser visto na figura 2.2(b).



**Figura 2.2:** Representação dos componentes de estacionariedade e volatilidade.

### 2.1.2 Modelagem e Previsão de Séries Temporais.

A análise de séries temporais consiste na criação de um modelo do comportamento da série. Através desse modelo é possível identificar os componentes contidos na série temporal (OLIVEIRA; LUDERMIR, 2016) e com isso é possível entender o passado e tentar prever o comportamento futuro da série.

Na literatura existem diversos modelos matemáticos que podem ser usados para a modelagem e previsão de séries temporais. Esses modelos são classificados em lineares e não-lineares (BABU; REDDY, 2014). Como modelos lineares podemos citar: o Modelo de Heterocedasticidade Condicional Autorregressivo Generalizado (Generalized Autoregressive Conditional Heteroskedasticity) (GARCH), o Modelo Autorregressivo Integrado de Médias Móveis (Autoregressive Integrated Moving Average) (ARIMA) e o Modelo Autorregressivo Integrado de Médias Móveis Sazonal (Seasonal Autoregressive Integrated Moving Average) (SARIMA), enquanto os não-lineares: Redes Neurais Artificiais (Artificial Neural Network) (RNA) e Regressão por Vetores de Suporte (Support Vector Regression) (SVR).

A escolha do método é uma das etapas mais importantes no processo de modelagem de séries temporais. A razão para isso está relacionada com os diferentes padrões que as séries temporais podem possuir. Segundo Zhang (2003) nenhum método de previsão é melhor em todas as situações, portanto é necessário fazer um estudo do modelo mais adequado para o problema.

Como exemplo de modelos específicos, podemos citar o método SARIMA (BOX et al., 2015) que foi desenvolvido para lidar com componentes sazonais; o método GARCH (LAMOUREUX; LASTRAPES, 1990) para lidar com dados que possuem volatilidade e; RNA para dados com descontinuidades e não-linearidades (BABU; REDDY, 2014).

Após a escolha do modelo mais apropriado, o segundo passo consiste em encontrar os melhores parâmetros do modelo para descrever os dados observados (CAVALCANTE, 2017). Os melhores parâmetros são aqueles que apresentam a maior qualidade de ajuste do modelo, por exemplo a quantidade de neurônios de entrada e da camada escondida de uma RNA. A qualidade do ajuste do modelo é medida através de métricas de previsão. Essas métricas calculam a similaridade da previsão do modelo em relação aos dados reais, na qual quanto mais próximo dos dados originais, melhor qualidade o modelo possui. Como exemplo de métrica de previsão, pode-se citar o Erro Médio Absoluto (Mean Absolute Error) (MAE), que é uma métrica de simples interpretação que calcula a diferença absoluta média de duas sequências. Esta métrica está descrita na equação 2.1.

$$MAE = \frac{1}{N} \sum_{n=1}^N |O_n - P_n| \quad (2.1)$$

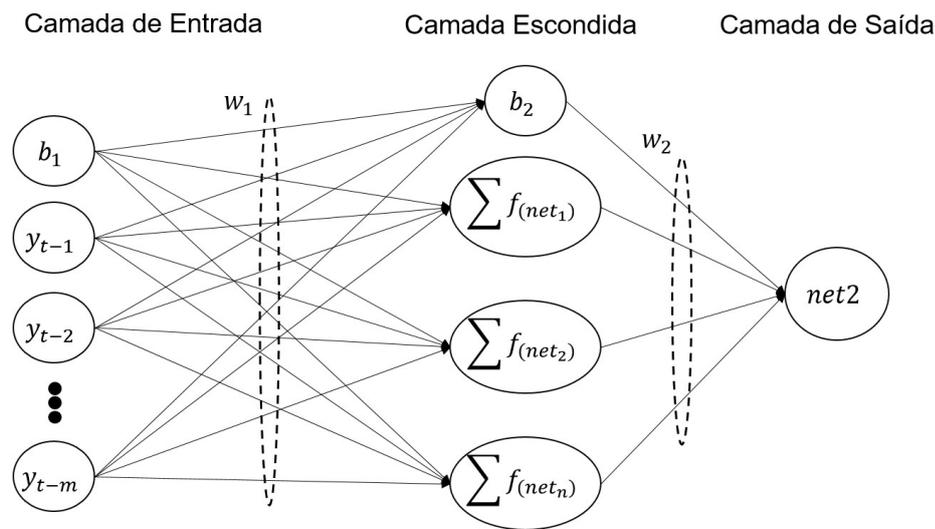
Em que  $N$  é a quantidade de exemplos usados para modelagem,  $O_n$  é a observação real no tempo  $n$  e  $P_n$  é a previsão gerada pelo modelo para o tempo  $n$ .

### 2.1.2.1 Previsão de Séries Temporais com Redes Neurais Artificiais

Técnicas de aprendizagem de máquina, especialmente RNA's, têm sido amplamente usadas no domínio financeiro (BEIRANVAND; BAKAR; OTHMAN, 2012). Essas técnicas foram inspiradas por estudos dos neurônios biológicos do cérebro humano e são capazes de aprender com a experiência e fazer generalizações baseada no seu conhecimento previamente acumulado (WAGNER et al., 2007).

Na literatura, a maioria dos trabalhos que propõem o uso de RNA's para problemas de previsão financeiros tem usado a rede neural de multicamadas (MLP) com grande sucesso (CAVALCANTE et al., 2016). A principal razão para esse método ter se tornado tão popular é devido a sua capacidade de lidar com dados categorizados por não-linearidades, descontinuidades e componentes polinomiais de alta frequência. RNA's são métodos capazes de capturar comportamentos não-lineares de séries temporais sem qualquer suposição estatística sobre os dados (CAVALCANTE et al., 2016).

As redes MLP foram originalmente concebidas para executar tarefas de natureza estática, não foram, portanto, idealizadas para tratar problemas temporais (BARROS, 2005). Devido a isto o método de janela de tempo foi desenvolvido para criar uma adaptação da rede MLP para o processamento dinâmico. A técnica da janela de tempo consiste em definir como entrada da RNA um conjunto de observações passadas ordenadas no tempo e como saída o valor que precede a



**Figura 2.3:** Estrutura de uma MLP para previsão de séries temporais.

esta sequência. Esse processamento na série original é dado através da seguinte equação 2.2:

$$y'_t = f(y_{t-1}, y_{t-2}, y_{t-3}, \dots, y_{t-m}) \quad (2.2)$$

Em que  $y_t$  é o ponto da série correspondente ao tempo  $t$ ,  $m$  é o tamanho da janela de tempo e  $y'_t$  é a nova série gerada. Então se a RNA possui como entrada  $m = 3$ , ela receberá de entrada sequências de três números, como por exemplo:  $y_1, y_2, y_3$  para tentar prever  $y_4$ . Essas novas séries são usadas iterativamente até a última observação para o treinamento da rede. A rede neural por sua vez encontra a relação entre os padrões de entrada com as saídas esperadas.

Além da camada entrada, existem mais duas, sendo elas a camada escondida e camada de saída. Com exceção da última camada, que deve ser sempre uma unidade, em cada camada o número de unidades (neurônios) é empregado de acordo com a acurácia do modelo para a determinada tarefa (ZHANG, 2001). Outro componente importante nessa estrutura são os pesos que interconectam as unidades, descritos pelos elementos  $w_1, b_1, w_2$  e  $b_2$ . Esses elementos são os responsáveis pelo poder de aprendizado e generalização da RNA.

Portanto, os valores de saída da rede neural são obtidos através de três etapas. A primeira pelo cálculo das unidades de entrada pelos pesos da camada escondida, conforme descrito na Equação 2.3:

$$\text{net}_1 = \left[ \sum_{j=1}^m w_{1j} y_{t-j} + b_1 \right] \quad (2.3)$$

A segunda etapa que consiste em atribuir a não-linearidade da rede neural. Esse atividade é dada através do emprego de uma função de ativação não-linear nas unidades de entrada. Para

esse processo, geralmente é usado a função sigmoide logística, como descrito na equação 2.4.

$$net_1 = f_1(net_1) = \frac{1}{1 + \exp^{-net_1}} \quad (2.4)$$

E por fim o terceiro passo consiste na computação da ativação dos neurônios da camada escondida pelos pesos da camada de saída. A camada de saída é caracterizada por uma função linear. Para esse processo tome  $n$  como a quantidade de unidades para a camada escondida da rede conforme a equação 2.5:

$$f(net_2) = net_2 = \left[ \sum_{j=1}^n w_{2,j}net_j + b_2 \right] \quad (2.5)$$

### 2.1.2.2 Otimização Aplicada a Redes Neurais Artificiais

A otimização dos pesos de uma RNA é um dos processos mais importantes antes do seu uso. Essa otimização consiste em selecionar os pesos que apresentam a melhor acurácia. Esta atividade é chamada de processo de treinamento ou aprendizagem. Na literatura existem diversos métodos que podem otimizar os pesos de uma RNA, entre esses métodos podemos citar algoritmos baseados em gradiente e descendente (HECHT-NIELSEN et al., 1988), geometria analítica (HUANG; ZHU; SIEW, 2006), algoritmos evolucionários (MONTANA; DAVIS, 1989) e inteligência de enxames (HAN; YAO; LING, 2013).

Dentre as técnicas acima listadas destaca-se o uso de inteligência de enxames, devido a sua capacidade de busca global. Alguns trabalhos têm mostrado que esse tipo de otimização é muito efetivo para treinar RNA's (RAKITIANSKAIA; ENGELBRECHT, 2009), (HAN et al., 2016), (OLIVEIRA et al., 2017). A principal motivação para esses trabalhos é que o treinamento por inteligência de enxames geralmente apresenta melhores resultados que algoritmos convencionais, pois esses algoritmos conseguem lidar de forma mais eficiente com mínimos locais, devido a sua estratégia de criar vários modelos de previsão (RAKITIANSKAIA; ENGELBRECHT, 2009).

Uma das técnicas da área que tem recebido bastante atenção para esse treinamento é o algoritmo de enxame de partículas (PSO) (RAKITIANSKAIA; ENGELBRECHT, 2012). O PSO (KENNEDY, 2011) é um algoritmo de busca inteligente inspirado no comportamento social dos indivíduos. Esse algoritmo faz sua busca através de partículas que representam soluções distintas para o problema em questão. O processo de busca é realizado pelo movimento das partículas no espaço de características através do cálculo da velocidade que atualiza cada posição das partículas a cada iteração.

As partículas são compostas por um vetor numérico e quando aplicadas a otimização dos pesos de uma RNA, esses valores são constituídos pelos pesos da rede (OLIVEIRA et al., 2017). Um exemplo dessa codificação pode ser visto na figura 2.4. Onde  $w_{mn}$  são os pesos da camada de entrada, na qual  $m$  representa a quantidade de unidades de entrada e  $n$  a quantidade

de unidades escondidas.  $w_{2n}$  representa os pesos da camada escondida e  $b_1$  e  $b_2$  os bias da rede.

$w_{11}$	...	$w_{1n}$	$w_{2n}$	$w_{3n}$	...	$w_{mn}$	$w_{21}$	...	$w_{2n}$	$b_1$	...	$b_n$
----------	-----	----------	----------	----------	-----	----------	----------	-----	----------	-------	-----	-------

**Figura 2.4:** Codificação de uma partícula para a otimização dos pesos de uma rede neural MLP.

Cada partícula representa uma solução distinta para o atual problema. Ao se movimentar no espaço de busca, cada partícula atualiza a sua solução, durante esse processo, elas guardam a sua melhor solução encontrada em um vetor numérico chamado  $pBest$ . Já a melhor solução global encontrada entre todas as partículas é salva em um vetor numérico chamado de  $gBest$ . Ambos  $gBest$  e  $pBest$  são tomados em consideração para o movimento das partículas, como mostrado na equação 2.6.

Para entender o funcionamento do cálculo da velocidade, tome  $v_i$  como um vetor numérico que representa a velocidade da  $i$ -ésima partícula. Tome  $pBest_i$ , como a melhor solução pessoal encontrada e  $g_i$  como a solução atual da  $i$ -ésima partícula. Portanto a velocidade é calculada como segue:

$$v_i(t+1) = w * v_i(t) + c_1 \cdot rnd() \cdot (pBest_i - g_i(t)) + c_2 \cdot rnd() \cdot (gBest - g_i(t)) \quad (2.6)$$

Em que  $w$ ,  $c_1$  e  $c_2$  representam a inércia e os fatores pessoais e sociais. Com isso a nova solução é calculada da seguinte forma:

$$g_i(t+1) = g_i(t) + v_i(t+1) \quad (2.7)$$

Outras pesquisas tem melhorado o processo de busca do PSO tradicional, oferecendo um melhor balanço entre uma busca global e local. Nesse sentido podemos citar o trabalho de Zhang, Xiong e Zhang (2013) que desenvolveram um Algoritmo de Enxames de Partículas Auto-Adaptativo (Improved Self-adaptive Particle Swarm Optimization) (IDPSO). Nesse algoritmo foi desenvolvido uma função não-linear  $\varphi(t)$  que modifica os valores dos parâmetros  $w$ ,  $c_1$  e  $c_2$  ao longo do tempo para cada partícula, possibilitando que cada uma tenha uma racionalidade própria. Com isso o algoritmo consegue realizar uma busca mais eficiente, uma vez que cada partícula pode decidir como realizar sua busca conforme a sua experiência.

No algoritmo 1 apresentamos com detalhes o procedimento do IDPSO, que necessita do cálculo do valor de  $\varphi(t)$  e as modificações de  $w$ ,  $c_1$  e  $c_2$ . A seguir são apresentadas suas respectivas Equações 2.8 2.9, 2.10 e 2.11:

$$\varphi(t) = \frac{\|gBest - g_i(t-1)\|}{\|pBest_i - g_i(t-1)\|} \quad (2.8)$$

$$w(t) = \frac{w_{inicial} - w_{final}}{1 + e^{\varphi(t) \cdot (t - ((1 + \ln(\varphi(t))) \cdot K_{max}) / \mu)}} + w_{final} \quad (2.9)$$

$$c_1(t) = c_1 \cdot \varphi^{-1}(t) \quad (2.10)$$

$$c_2(t) = c_2 \cdot \varphi(t) \quad (2.11)$$

No Algoritmo 1, pode-se ver através de uma sequência de atividades como funciona o procedimento do IDPSO aplicado a otimização dos pesos de uma RNA. Vale salientar que nesse processo utilizamos como métrica de avaliação das partículas o MAE, por ser uma métrica que usa o erro absoluto e possibilita uma análise mais simples da distância das soluções.

---

**Algoritmo 1:** Otimização dos pesos de uma RNA com IDPSO

---

**Entrada:** tamanho\_enxame, iterações,  $c_1$ ,  $c_2$ ,  $\omega_{inicial}$  e  $\omega_{final}$

- 1 Inicializa as partículas com velocidade e posições aleatórias entre [0, 1];
- 2 **enquanto** *máximo de iterações não é atendido* **faça**
- 3     **para cada partícula faça**
- 4         Computa previsões para os dados de treinamento conforme a Eq.2.5;
- 5         Computa o fitness das partículas conforme a Eq.2.1;
- 6         **se** *valor de fitness é melhor que pBest* **então**
- 7             Atualiza o valor atual para o novo pBest;
- 8         **fim**
- 9     **fim**
- 10    Escolhe a partícula com o melhor fitness e salva em gBest;
- 11    **para cada partícula faça**
- 12         Calcula a velocidade das partículas conforme a Eq.2.6;
- 13         Atualiza os parâmetros das partículas conforme as Eqs. 2.9, 2.10 e 2.11;
- 14         Atualiza a posição das partículas conforme a Eq.2.7;
- 15    **fim**
- 16 **fim**

---

## 2.2 Mudança de Conceito

A prevalência de telefones celulares e tecnologias ligadas a internet têm lidado com um enorme crescimento na quantidade dos dados. Esses dados agora estão disponíveis em forma de fluxos de dados (DITZLER et al., 2015). Um fluxo de dados é uma coleção de dados de entrada que chegam sequencialmente e continuamente ao longo do tempo. Esses dados tendem a evoluir ao longo do tempo, devido a dinâmica das atividades do mundo real.

Nos trabalhos da literatura, frequentemente assume-se, implicitamente ou explicitamente, que o processo de geração de um fluxo de dados é estacionário, que os dados foram gerados através de uma distribuição de probabilidade fixa e imutável (DITZLER et al., 2015).

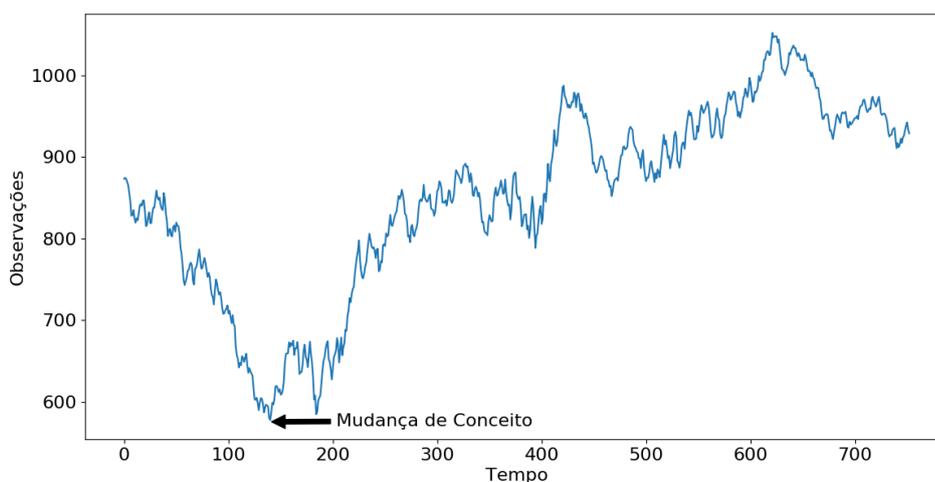
No entanto, em muitos cenários do mundo real essa suposição não é verdadeira, o pro-

cesso de geração dos fluxos de dados é intrinsecamente não-estacionário, ou seja, a distribuição dos dados evolui ao longo do tempo. Esse fenômeno é chamado de mudança de conceito (GAMA et al., 2014).

A mudança de conceito tem se tornado uma área que atrai atenção de muitas pesquisas nas últimas décadas, devido a sua ocorrência em muitas aplicações do mundo real (ZLIOBAITE et al., 2014). Como exemplos aplicações, podemos citar: (i) mudança de conceito em previsão de séries temporais e (ii) mudança de conceito em problemas de otimização.

### 2.2.1 Mudança de Conceito em Previsão de Séries Temporais

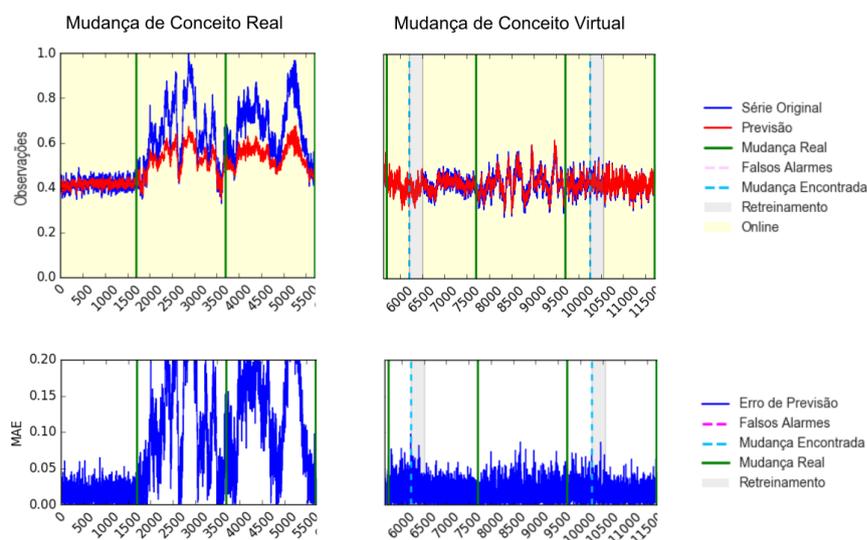
Séries temporais financeiras são um tipo especial de fluxos de dados, e devido ao dinamismo das aplicações do mundo real os seus padrões evoluem ao longo do tempo (CAVALCANTE; OLIVEIRA, 2015). A mudança de conceito pode acontecer devido a diversos motivos, como por exemplo: eventos e notícias que impactam o mundo financeiro, mudanças políticas, fatores econômicos ou mudança na psicologia dos investidores (CAVALCANTE; MINKU; OLIVEIRA, 2016). Podemos ver uma dessas mudanças na imagem 2.5 que apresenta uma mudança de conceito causada pela condenação dos assessores do ex-presidente Nixon em 30 de Janeiro de 1973 (ADAMS; MACKAY, 2007).



**Figura 2.5:** Mudança de conceito real na série financeira Down Jones Industrial Average de 03 de julho de 1972 até 30 de junho de 1975

Esse fenômeno pode ser classificado como mudança de conceito real e mudança de conceito virtual. Para entender a diferença dessas nomenclaturas, tomemos como exemplo um modelo de treinamento baseado em um conjunto de entradas  $X$  e um conjunto de alvos  $y$ , tal que  $X_b, y_b = (b = 1, 2, 3, \dots, n)$ , cuja a distribuição de probabilidade conjunta é  $P(X, y) = P(y|X)P(X)$ , onde  $P(X)$  é a distribuição dos componentes de entrada e  $P(y|X)$  é a função que relaciona o conjunto de entradas  $X$  com o conjunto de saídas  $y$ . Esse último por sua vez pode ser representado por um modelo de aprendizagem de máquina (YAMAUCHI, 2010). Com isso podemos exemplificar os tipos de mudanças. A mudança de conceito real é caracterizada por

mudanças em  $P(y|X)$ , ou seja, uma mudança que afeta a acurácia do modelo de previsão que relaciona os dados de entrada e saída, também conhecida como deslocação de mudança ou mudança condicional (GAMA et al., 2014). Já a mudança de conceito virtual acontece se  $P(X)$  muda, ou seja, a distribuição dos exemplos de chegada muda sem afetar a acurácia do modelo de previsão ( $P(y|X)$ ). Essas duas mudanças estão ilustradas na Figura 2.6.



**Figura 2.6:** A esquerda um exemplo de mudança de conceito real. A direita um exemplo de mudança de conceito virtual

Na Figura 2.6, observa-se que no bloco da mudança de conceito real, o erro do modelo preditivo aumenta quando ocorre uma mudança, isso pode ser visto com mais detalhes ao se analisar o bloco do erro, onde em um conceito treinado o erro se comportava de forma uniforme, mas ao ocorrer a mudança, ele aumenta drasticamente. Já no bloco da mudança de conceito virtual, percebe-se que a distribuição dos dados muda ao longo do tempo, mas o modelo de previsão permanece acurado, uma vez que o seu erro se mantém sempre de forma uniforme.

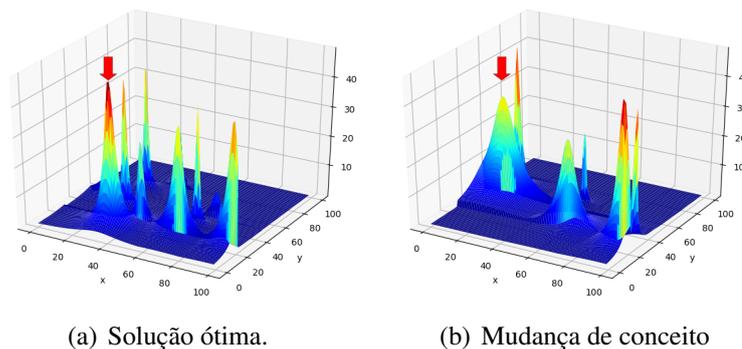
A mudança de conceito real apresenta diversos desafios para os modelos de previsão tradicionais. Um desses desafios é que o processamento dos dados de forma *off-line* pode tornar os modelos de previsão ineficazes e inapropriados, fazendo com que não sejam úteis para definir comportamentos futuros. (GAMA et al., 2014).

Apesar desse problema ser muito importante para a área de pesquisa de previsão de séries temporais, na literatura, há, uma escassez de pesquisas para lidar com esse problema (CAVALCANTE et al., 2016). Um dos principais obstáculos para o desenvolvimento dessas pesquisas é que ao se combinar técnicas que lidam com mudança de conceito, com modelos de regressão, o esquema final torna-se bastante complexo (SONG et al., 2017).

### 2.2.2 Mudança de Conceito em Problemas de Otimização

Na literatura vários estudos têm investigado como usar técnicas de otimização para melhorar a acurácia de modelos de aprendizagem de máquina. O objetivo desses métodos é encontrar o ótimo global ou local do espaço de busca, que tipicamente apresenta as melhores configurações para o modelo de previsão. No entanto, em muitas aplicações do mundo real os problemas podem ser dinâmicos e não estacionários. A função a ser otimizada pode mudar ao longo do tempo, o que torna as soluções antigas ineficientes (OLIVEIRA et al., 2017).

Um exemplo dessa situação está ilustrado na figura 2.7. A imagem 2.7(a) representa o espaço de características de um problema de maximização, ou seja, a busca do ponto máximo da função. Em abordagens convencionais o objetivo seria encontrar a solução ótima, representado pela seta vermelha. No entanto, se a função de otimização muda ao longo do tempo a melhor solução deixa de ser ótima (Essa situação pode ser vista na figura 2.7(b)). A seta vermelha que antes representava a melhor solução do espaço de busca, após a mudança da função, deixou de ser ótima.



**Figura 2.7:** Representação da mudança de conceito na ótica da otimização.

Na literatura esses tipos de problema são conhecidos como Problemas de Otimização Dinâmicos (Dynamic Optimization Problems) (DOP) (NGUYEN; YANG; BRANKE, 2012). Nesse campo de estudo, os algoritmos buscam encontrar o ótimo, detectar as mudanças, rastrear o ótimo e localizar novos melhores ótimos assim que eles aparecem (NGUYEN; YANG; BRANKE, 2012).

## 2.3 Adaptação a Mudança de Conceito

Adaptação a mudança de conceito consiste no estudo e desenvolvimento de técnicas capazes de minimizar ou lidar por completo com esse problema. De acordo com Gama et al. (2014), duas principais abordagens lidam com esse problema: (i) os métodos passivos e (ii) ativos de controle de mudança de conceito.

### 2.3.1 Métodos Passivos de Adaptação

Os métodos passivos de adaptação, também conhecidos como métodos cegos (GAMA et al., 2014), lidam com a mudança sem nenhuma informação explícita sobre ela (KUNCHEVA; GUNN, 2016). Esses métodos geralmente usam técnicas de janelas de tamanho fixo e deslizantes para armazenar os dados (KUNCHEVA; GUNN, 2016). A cada instante de tempo essa janela é atualizada e os dados mais antigos são apagados. Esses métodos periodicamente atualizam o modelo preditivo utilizando os dados mais recentes armazenados em sua memória.

Um caso especial dos métodos passivos são os modelos de aprendizagem *on-line*, onde os modelos evoluem em conformidade com os dados (GAMA et al., 2014). Esses métodos sem qualquer estratégia de detecção explícita de mudança de conceito se adaptam aos dados mais recentes (GAMA et al., 2014).

Como vantagem, essas abordagens são capazes de se adaptar mais rapidamente a mudança de conceito, uma vez que se atualizam sempre que novos dados chegam (KUNCHEVA; GUNN, 2016). No entanto, como desvantagens, esses métodos não são transparentes e eficientes (OLIVEIRA et al., 2017). Transparentes no sentido que não informam quando ocorre a mudança de conceito e eficientes no sentido que não devem ativar uma adaptação quando não é necessário. Além disso, podem ter previsões enviesadas, uma vez que os modelos são baseados em dados de um conceito antigo, devido o esquecimento lento das informações (GAMA et al., 2014).

### 2.3.2 Métodos Ativos de Adaptação

A outra classe métodos de adaptação que lida com a mudança de conceito na literatura são os métodos ativos ou informados (GAMA et al., 2014). Esses métodos possuem a estratégia de detecção-reação. Eles possuem um mecanismo de gatilho, que informa quando ocorre a mudança de conceito. Sempre que informados, um mecanismo de adaptação reage a mudança atualizando ou construindo um novo modelo (DITZLER et al., 2015)

O gatilho pode ser estabelecido em um Teste de Controle de Detecção (Change Detection Method) (CDT) aplicado a mudança de conceito. Esses testes são especificamente projetados para analisar sequencialmente o comportamento estatístico de algum indicador de desempenho do classificador ou monitorar a distribuição dos dados em diferentes porções do conjunto de dados (OLIVEIRA et al., 2017). Esses métodos são normalmente caracterizados pela baixa complexidade computacional, uma vez que eles tem que monitorar continuamente o fluxo de dados (DITZLER et al., 2015).

A reação consiste na reciclagem do modelo de previsão. Existem dois tipos de abordagens que são capazes de atualizar o modelo, são abordagens de substituição global ou local (GAMA et al., 2014). A substituição global é a mais radical, pois consiste na reconstrução completa do modelo. A substituição local por sua vez, consiste na atualização de parte do modelo de previsão. Essa atualização se baseia no pressuposto que apenas algumas regiões do espaço de busca mudaram (GAMA et al., 2014), dessa forma, mantendo algumas informações passadas.

Essas abordagens possuem como desvantagens a dependência do gatilho para ativar a reação. O gatilho não consegue garantir uma detecção perfeita, a detecção pode vir com um incontrolável atraso, não ser confiável e enviesada (DITZLER et al., 2015). Por outro lado, essas abordagens são mais eficientes que as abordagens passivas, pois não reagem indevidamente quando não ocorre a mudança de conceito. Outra vantagem é a sua transparência, pois conseguem informar ao usuário quando ocorre a mudança de conceito (OLIVEIRA et al., 2017). Essa última pode ser útil na tomada de decisão.

## **2.4 Síntese do Capítulo de Fundamentos**

Nesse capítulo, foram apresentados conceitos básicos para o entendimento sobre a análise de séries temporais, o fenômeno da mudança de conceito e as estratégias usadas pela literatura para lidar com esse problema.

Em relação a análise de séries temporais, foram apresentados os principais elementos que compõem as séries temporais e como esses elementos influenciam no comportamento dos dados. Ainda foi apresentado o processo de construção de modelos para previsão de séries temporais, dando destaque para o uso de RNA e o seu processo de treinamento através de técnicas de otimização.

Na sequência, foi contextualizado o fenômeno da mudança de conceito. Foi apresentado um breve estudo de como esse problema se dá e quais as suas implicações nas áreas de pesquisa de previsão de séries temporais e problemas de otimização.

Por fim, na seção de adaptação a mudança de conceito foi listado uma breve revisão da literatura das principais áreas de pesquisas que desenvolvem estratégias para conseguir lidar com esse problema.

# 3

## TRABALHOS RELACIONADOS

As abordagens ativas de controle de mudança de conceito são conhecidas como abordagens "detectar e reagir": uma vez que a mudança é detectada, o modelo descarta o conhecimento obsoleto, e adapta-se ao novo ambiente (DITZLER et al., 2015). Existem diversas abordagens que podem ser usadas nesses dois módulos, por essa razão esse capítulo discutirá importantes trabalhos da literatura, para saber como eles lidam a respeito: da detecção de mudança de conceito (Seção 3.1); e estratégias de adaptação após a mudança de conceito (Seção 3.2).

### 3.1 Detecção de Mudança de Conceito

As abordagens ativas são métodos que lidam explicitamente com a mudança de conceito, ou seja, primeiramente detectam a mudança na distribuição dos dados para poder reagir. Essas abordagens têm se mostrado eficientes para lidar com fluxos de dados (DITZLER et al., 2015). Uma das principais razões para o desenvolvimento de pesquisas para essa área é que essas abordagens trabalham de forma transparente, informando aos usuários quando ocorre a mudança de conceito, que por sua vez, pode ser usada para a tomada de decisões (CAVALCANTE; MINKU; OLIVEIRA, 2016).

Na literatura várias áreas de pesquisa utilizam métodos de detecção para lidar com a mudança de conceito. Cada área de pesquisa possui diferentes métodos e estratégias para realizar as detecções. Nesse sentido, destacaremos trabalhos nos seguintes campos de pesquisa: (i) detecção de mudança de conceito em séries temporais e; (ii) detecção de mudança de conceito em problemas de otimização.

#### 3.1.1 Detecção de Mudança de Conceito em Séries Temporais

Em séries temporais alguns trabalhos investigaram como detectar pontos de mudanças nas séries, que são variações abruptas nos parâmetros geradores dos dados, porém com uma abordagem retrospectiva e *off-line*. Esses métodos necessitam ter todos os dados disponíveis para poderem encontrar essas variações. Esse campo de estudo é conhecido como detecção de pontos de mudança (ADAMS; MACKAY, 2007).

Neste campo, Boracchi e Roveri (2014) propuseram uma abordagem que mede a auto-similaridade entre os segmentos das séries temporais e usa os valores desta característica como variável do detector de mudanças. O método proposto é chamado de interseção de intervalos de confiança (ICI). Ele é usado como CDT para monitorar mudanças no processo de geração das observações da série temporal. No entanto, essa abordagem monitora apenas um aspecto do comportamento das séries temporais e sua aplicabilidade é restrita a séries que apresentam auto-similaridade.

Lavielle e Teyssiere (2006), desenvolveram um método para detectar pontos de mudanças em séries multivariadas, cujas observações são independentes e identicamente distribuídas (IID). O método assume que a estrutura de covariância da série muda abruptamente em alguns instantes de tempo desconhecidos. O método proposto conseguiu encontrar pontos de mudanças tanto em séries reais de índices de ações diários, quanto séries geradas artificialmente.

Já outros trabalhos na literatura buscam detectar mudanças de conceito de forma *on-line*. Neste sentido, destaca-se (CAVALCANTE; MINKU; OLIVEIRA, 2016) que aplicou um CDT para monitorar características de séries temporais. Eles usaram um conjunto de características estatísticas de séries temporais para caracterizar conceitos e monitorar as diferenças entre vetores subsequentes extraídos da série de forma *on-line*. Os experimentos foram realizados em séries artificiais gerados por modelos lineares e não-lineares, de modo a fornecer tanto mudanças graduais como abruptas. O método proposto se mostrou mais eficiente, em termos de detecção, do que alguns métodos baseados no erro do modelo preditivo.

Em outro trabalho de Cavalcante e Oliveira (2015) foi investigado a combinação de CDT's com redes neurais artificiais do tipo Máquina de Aprendizagem Extrema (Extreme Learning Machine) (ELM) (HUANG; ZHU; SIEW, 2006). Foram investigados CDT's que se baseiam no erro do modelo preditivo como o ECDD (ROSS et al., 2012) e DDM (GAMA et al., 2004). O sistema proposto trabalha de forma *on-line*, assim que erro do modelo se degrada os CDT's informam que houve uma mudança de conceito e o modelo é atualizado. Neste estudo não foi avaliado a precisão das detecções, apenas a acurácia de previsão dos sistemas. Observou-se que utilizar um módulo de detecção melhora o desempenho de previsão em algoritmos de treinamento baseados em lote.

Neste mesmo sentido Guo e Liew (2016), propuseram uma abordagem baseada em filtro de Kalman como CDT para detectar mudanças de conceito e atualizar redes neurais. O filtro de Kalman usa uma abordagem recursiva para monitorar o erro do modelo preditivo através de dois passos. No primeiro passo quando o modelo realiza a previsão do conjunto de treinamento o filtro produz estimativas sobre o erro e no segundo passo quando o erro é calculado de forma *on-line* as estimativas são atualizadas usando uma média ponderada. Neste estudo, somente a acurácia de previsão foi considerada, e observou-se que utilizar um módulo de detecção, diminuiu o tempo de execução do método proposto, sem perder a acurácia de previsão.

Sobre os trabalhos acima discutidos, podemos destacar alguns problemas: poucos estudos detectam a mudança de conceito de forma *on-line*, e desses poucos, somente um trabalho consi-

derou a acurácia de detecção dos modelos. Avaliar a acurácia de detecção torna-se importante, uma vez que pode-se apresentar ao usuário a confiabilidade do método em informar quando ocorre a mudança de conceito.

### 3.1.2 Detecção de Mudança de Conceito em Problemas de Otimização

Na literatura de otimização, há um campo chamado de Problemas de Otimização Dinâmicos (Dynamic Optimization Problems) (NGUYEN; YANG; BRANKE, 2012). Neste campo de estudo, geralmente os algoritmos tratam a mudança de conceito com abordagens ativas de detecção de mudanças. Os algoritmos propostos rastreiam o ótimo (melhor solução para o problema) ao longo do tempo e para isso dependem da detecção da mudança de conceito (HENDTLASS; MOSER; RANDALL, 2009).

Os algoritmos de otimização dinâmica, geralmente se baseiam na comparação de um ponto fixo do espaço de busca. Para isto os algoritmos utilizam estatísticas descritivas como média e desvio padrão. A razão para isto é que essas estatísticas são sensíveis e reconhecem pequenas variações da solução ótima. As detecções propostas podem ser integradas nos seguintes grupos: (i) técnicas baseadas em sensores, (ii) comportamento do enxame e em (iii) testes de detecção (NGUYEN; YANG; BRANKE, 2012).

A nomenclatura de sensores é dada quando a detecção se baseia em muitos pontos do espaço de busca. Esses pontos são representados por partículas do enxame que representam diferentes soluções para o problema. Se a maioria ou todas as soluções anunciam que houve uma mudança de conceito, então o algoritmo reage (NGUYEN; YANG; BRANKE, 2012). Neste sentido podemos citar os trabalhos de (HU; EBERHART, 2002), (FOULADGAR; LOTFI, 2016) e (RICHTER, 2009) que propuseram sensores para detectar mudanças na Função de Referência de Picos Móveis (Moving Peaks Benchmark) (MPB). Essa função simula mudanças de conceitos ao mudar a função principal ao longo de tempo.

Já a nomenclatura de comportamento é dada aos estudos que se baseiam no monitoramento da queda do valor da média das melhores soluções encontradas ao longo de várias gerações (NGUYEN; YANG; BRANKE, 2012). Como exemplos desses estudos podemos citar os trabalhos de (NASIRI; MEYBODI; EBADZADEH, 2016) e (JANSON; MIDDENDORF, 2006) que também detectam mudanças na função de otimização MPB.

Por fim, existem os testes de detecção que utilizam abordagens estatísticas mais complexas para comparar pontos do espaço de busca em diferentes instantes de tempo. Vellasques, Sabourin e Granger (2011) e Vellasques, Sabourin e Granger (2013) investigaram essa abordagem em problemas de Marca de Água Inteligente (Intelligent Watermarking) (IW). As abordagens propostas comparam soluções que são armazenadas em um memória, se a distância entre a solução da memória para a sua referente no atual ambiente for muito alta, então uma mudança de conceito é declarada.

### 3.2 Estratégias de Adaptação após a Mudança de Conceito

Um dos principais desafios encontrados ao se lidar com fluxos de dados é manter o modelo preciso ao longo do tempo (MINKU; YAO, 2012). A mudança de conceito por mudar a distribuição dos dados faz com que os modelos treinados nesses dados tornem-se obsoletos e ineficazes em prever os comportamentos futuros (MINKU; YAO, 2012).

Os algoritmos capazes de manter sua acurácia ao longo do tempo são chamados de algoritmos adaptativos (GAMA et al., 2014). Nesta seção serão apresentados alguns sistemas adaptativos propostos na literatura para as áreas de (i) previsão de séries temporais e (ii) problemas de otimização.

#### 3.2.1 Estratégias de Adaptação para Previsão de Séries Temporais

Apesar da mudança de conceito não ser uma nova área de pesquisa, ela ainda não é amplamente investigada em previsão de séries temporais (CAVALCANTE; MINKU; OLIVEIRA, 2016), existem poucos trabalhos que desenvolveram sistemas adaptativos.

Dentre esses trabalhos, podemos citar o trabalho de Wagner et al. (2007). Nesse trabalho foi desenvolvido um sistema passivo para lidar com a mudança de conceito. O modelo proposto é chamado de Programação Genética para Previsão Dinâmica (Dynamic Forecasting Genetic Program) (DyFor GP). O modelo utiliza várias janelas de tamanhos diferentes que incorporam diferentes conjuntos de dados ao longo do tempo. Quando a mudança de conceito ocorre, o sistema utiliza modelos treinados em diferentes conjuntos de dados para se adaptar. O método proposto foi testado tanto em séries simuladas, quanto reais. Os resultados mostraram que o DyFor GP obteve o melhor desempenho em todos os experimentos, sugerindo assim sua aplicação para a previsão no mundo real.

Em Cavalcante e Oliveira (2015) dois tipos de sistemas adaptativos foram desenvolvidos. O primeiro utiliza uma abordagem de treinamento em lotes e o segundo utiliza uma abordagem de treinamento *on-line*. No primeiro sistema é utilizado uma ELM que possui um módulo de detecção de mudança de conceito. Quando o sistema identifica que houve uma mudança, ele inicia uma coleta de dados novos referentes ao novo conceito. Quando o modelo possui dados suficientes para um retreinamento ele é atualizado. O segundo sistema utiliza o algoritmo Máquina de Aprendizagem Extrema On-line (On-line Extreme Learning Machine) (OS-ELM) seguindo essa mesma lógica. A diferença é que esse algoritmo consegue atualizar o modelo a cada instante de tempo.

Guo e Liew (2016) propuseram uma estrutura que combina ELM e OS-ELM em paralelo para lidar com a mudança de conceito. A estrutura detecta a mudança de conceito através de um filtro de Kalman, quando isso acontece o OS-ELM é aplicado para atualizar o modelo, enquanto isso uma ELM é aplicada para realizar as previsões.

Por fim em Song et al. (2017) foi desenvolvido um algoritmo auto-adaptativo baseado em uma abordagem de agrupamento com fuzzy kernel c-means e um algoritmo de aprendizado

preguiçoso, chamado de FKLL. Usando o FKLL, o sistema atualiza o conjunto de dados usando o aprendizado preguiçoso, então o agrupamento de fuzzy kernel c-means é usado para determinar o conjunto de aprendizagem mais relevante para realizar as previsões.

### 3.2.2 Estratégias de Adaptação para Problemas de Otimização

Otimização dinâmica evolucionária (EDO) é um campo de estudo que tem atraído esforços durante os últimos 20 anos (NGUYEN; YANG; BRANKE, 2012). Nesse campo são desenvolvidos algoritmos de otimização que são capazes de lidar com problemas que mudam ao longo do tempo. Esses algoritmos implementam técnicas baseadas em: (i) diversidade, (ii) multi-enxames e (iii) memórias.

Sobre a diversidade, o objetivo dos algoritmos é evitar a convergência completa dos seus indivíduos. Isso ocorre porque se o ambiente de busca muda e não tem nenhum indivíduo do algoritmo na área, então ele não poderá reagir efetivamente e, portanto, pode deixar de rastrear o ótimo global (NGUYEN; YANG; BRANKE, 2012). Nesse sentido Rakitianskaia e Engelbrecht (2009) e Rakitianskaia e Engelbrecht (2012) investigaram um conjunto de variantes de PSO com mecanismos de diversidade para ajudar no treinamento de uma RNA em um problema de classificação dinâmica.

Sobre o uso de multi-enxames, essa técnica diz respeito ao uso de várias populações de indivíduos trabalhando paralelamente. Cada sub-população pode lidar com uma área separada do espaço de busca, como também podem se dividir em buscar o ótimo e encontrar mudanças (NGUYEN; YANG; BRANKE, 2012). Os trabalhos de Fouladgar e Lotfi (2016) e Nasiri, Meybodi e Ebadzadeh (2016) são exemplos desse tipo de abordagem, em ambos os casos duas populações de indivíduos são geradas com o objetivo de evitar uma total convergência e conseguir refinar a melhor solução encontrada até o momento.

Por fim, existem as abordagens baseadas em memórias que utilizam soluções encontradas anteriormente para economizar tempo computacional e impulsionar o processo de busca (NGUYEN; YANG; BRANKE, 2012). Nasiri, Meybodi e Ebadzadeh (2016), Vellasques, Sabourin e Granger (2011) e Vellasques, Sabourin e Granger (2013) utilizam dois tipos de memórias no seus trabalhos, sendo uma memória de curto prazo para impulsionar a busca de uma mudança de ambiente para o outro, e uma memória de longo prazo que fornece soluções passadas quando um ambiente similar aparece.

## 3.3 Síntese e Discussões do Capítulo de Trabalhos Relacionados

Nesse capítulo, foi apresentado uma breve revisão da literatura acerca dos métodos de detecção e das adaptações usadas nas áreas de previsão de séries temporais e problemas de otimização.

Em relação a detecção de mudança de conceito, pode-se observar que na área de séries temporais dois campos estudam esse fenômeno. O primeiro chamado de detecção de pontos

---

de mudança, em que as abordagens são retrospectivas e *off-line* e o segundo campo em que as abordagens utilizam a detecção de forma *on-line* através de CDT's. Sobre os problemas de otimização foi possível observar que as abordagens são sempre *on-line* e dependem do monitoramento de alguma característica do algoritmo de otimização.

Em relação a adaptação a mudança de conceito, pode-se observar que ainda existem poucos sistemas adaptativos para a área de previsão de séries temporais. Os trabalhos existentes se dividem entre abordagens ativas e passivas. Por outro lado em problemas de otimização, foi possível verificar que os sistemas desfrutam de uma grande quantidade de técnicas desenvolvidas para lidar com as mudanças nos dados.

Por meio dessa revisão, conclui-se que nenhum dos trabalhos que detectam mudanças em séries temporais estudou como aplicar técnicas de otimização nesse processo. Além disso, nenhum dos sistemas propostos utilizou técnicas de algoritmos dinâmicos de otimização como estratégia para lidar com a mudança de conceito. Utilizar uma combinação de otimização com o modelo preditivo, pode fornecer um modelo com melhor treinamento, e conseqüentemente um modelo com melhor acurácia de previsão. Além disso, ao utilizar técnicas de otimização dinâmica é possível construir um sistema mais robusto aos problemas causados pela mudança de conceito.

# 4

## MÉTODO PROPOSTO

Esse capítulo apresenta dois tipos de sistemas, que possuem procedimentos diferentes, sendo eles: os métodos de detecção baseados em enxame; e os métodos de adaptação baseados em técnicas de otimização dinâmica. A principal diferença entre esses sistemas está relacionado com a forma em que eles coletam os dados para um retreinamento e como atualizam o seu modelo preditivo. Além disso, essa estrutura foi montada de modo a conseguir responder as questões de pesquisa desse estudo e investigar a diferença entre usar e não usar mecanismos de adaptação.

Portanto a estrutura desse capítulo está organizada da seguinte forma: Na subseção 4.1 apresentamos o processo de treinamento da rede neural, que é universal para os dois tipos de sistemas; em seguida na subseção 4.2 introduzimos os algoritmos de detecção de mudança de conceito; e na subseção 4.3 os sistemas com mecanismos de adaptação ao novo ambiente (dado uma mudança de conceito). Para ambos os sistemas, primeiramente apresentamos uma visão geral das abordagens e em seguida as suas particularidades.

### 4.1 Treinamento da Rede Neural Artificial

Para este trabalho, a rede neural ELM foi aplicada como método de regressão devido ao seu procedimento de treinamento rápido e boa capacidade de generalização. Na rede ELM, as conexões entre a camadas escondida e de saída têm seus pesos calculados analiticamente enquanto que os pesos entre a entrada e a camada escondida são atribuídos aleatoriamente. Os criadores das redes ELM defendem que a atribuição aleatória dos pesos na primeira camada não afetam o desempenho do método (HUANG; ZHU; SIEW, 2006). No entanto, alguns trabalhos já apresentaram abordagens diferentes para melhorar o desempenho da ELM ao encontrar valores mais adequados para os pesos da primeira camada (HAN; YAO; LING, 2013). Este trabalho usa o algoritmo IDPSO para encontrar um conjunto adequado de pesos para as conexões da primeira camada. O IDPSO foi escolhido porque, em comparação com outros algoritmos de pesquisa global, como por exemplo ABC, PSO, o IDPSO apresentou um bom desempenho (SOUZA et al., 2016).

Durante a fase de treinamento, cada partícula do IDPSO representa um modelo ELM que

se movimenta no espaço dos parâmetros, ou seja, um espaço  $i$ -dimensional, em que  $i$  representa o número de conexões da primeira camada, dado por:

$$i = l * h \tag{4.1}$$

Em que  $l$  representa a quantidade de neurônios de entrada e  $h$  representa a quantidade de neurônios na camada escondida da RNA. Ao se movimentar no espaço de busca, cada modelo (partícula) busca encontrar o menor MAE, conforme a equação 2.1 para o conjunto de validação. O MAE é utilizado por ser uma métrica de simples entendimento e utiliza apenas o erro absoluto no seu cálculo, o que facilita a detecção de mudança de conceito. Já o conjunto de validação é usado para avaliar o poder de generalização dos modelos. O procedimento adotado para o treinamento da rede pode ser visto com mais detalhes no algoritmo 1.

## 4.2 Detecção de Mudança de Conceito Baseado em Inteligência de Enxames

Para responder as perguntas de pesquisa do bloco 1, que diz que monitorar vários modelos de previsão melhora a acurácia de detecção de mudança de conceito. Esse trabalho investiga o uso de três abordagens baseadas em características do enxame, esses algoritmos serão detalhados na seguinte sequência: (i) visão geral dos métodos de detecção; (ii) IDPSO-ELM-B - detecção de mudança baseado no comportamento do enxame; (iii) IDPSO-ELM-S - detecção de mudança baseado em sensores e; (iv) IDPSO-ELM-DDSV - detecção de mudança baseado no voto majoritário dos sensores.

### 4.2.1 Visão Geral dos Métodos de Detecção

O Algoritmo 2 apresenta o procedimento geral para os algoritmos com apenas a detecção de mudança de conceito. Esse procedimento geral recebe como entrada do fluxo de dados  $X$ , o tamanho do conjunto de treinamento  $n$ , o tamanho do enxame e os parâmetros específicos dos métodos que vão ser detalhados a seguir nas subseções 4.2.2, 4.2.3 e 4.2.4. Inicialmente, o conjunto de treinamento ( $tr$ ) é construído e um enxame de modelos ELM é usado para procurar o melhor modelo de previsão. Em seguida, as estatísticas dos erros dos modelos para o conjunto de treinamento são computadas para apoiar a detecção de mudança de conceito no futuro. Neste ponto, a melhor partícula é usada como modelo de previsão e o sistema monitora o fluxo de dados para detectar mudanças. Uma vez que ocorre uma mudança de conceito, um novo enxame

de ELM é treinado.

---

**Algoritmo 2:** Algoritmo dos métodos de detecção

---

**Entrada:**  $X$ ,  $n$ , tamanhoEnxame, paramEsp

```

1 tr := X(t : t + n);          // tr é o atual conjunto de treinamento
2 S = {p1, p2 .. ptamanhoEnxame};      // Treina o enxame S com IDPSO
3 para cada nova observação faça
4      $\hat{x}_t := gBest(x_{(t)});$           // previsão gBest
5     mudanca := DetecMudanca(paramEsp,  $\hat{x}_t$ );
6     se (mudanca) então
7         Espera  $n$  exemplos de train e define tr := X(t : t + n);
8         S = {p1, p2 .. ptamanhoEnxame};      // Treina um novo enxame S
9         Computa o erro do enxame em tr e define gBest;
10    fim
11 fim

```

---

#### 4.2.2 Detecção de Mudança Baseado no Comportamento do Enxame

Dada a convergência do exame para uma solução específica, é possível assumir que o enxame aprendeu o atual conceito. Sendo  $S$  um enxame composto de partículas  $p_i$  e cada uma dessas partículas computa seu próprio vetor de erro de previsão para as  $n$  observações contidas no conjunto de treinamento ( $tr$ ), a média  $\mu_p$  e desvio padrão  $\sigma_p$  dos erros de cada partícula  $p$  são computados.

Esse método é baseado na suposição de que a média do erro muda quando um novo conceito é introduzido (GAMA et al., 2004). Portanto, o erro de treinamento médio fornecido pelas partículas forma uma distribuição que muda estatisticamente conforme um novo conceito acontece.

Para monitorar o comportamento do erro do enxame, a média e o desvio padrão do erro médio das partículas é empregado:

$$\mu_s = 1/tamanhoEnxame \sum_p \mu_p \quad (4.2)$$

$$\sigma_s = \sqrt{1/tamanhoEnxame \sum_p (\mu_p - \mu_s)^2} \quad (4.3)$$

Para detectar a mudança de conceito, o módulo de detecção adapta equações introduzidas pelo ECDD (ROSS et al., 2012) que aplica a Média Móvel Exponencial Ponderada (Exponentially Weighted Moving Average) (EWMA), que consiste em uma estimação da média de uma sequência de variáveis tal que as variáveis mais recentes são consideradas mais importantes.

Portanto, tome  $Z_0 = \mu_s$  e a condição de detecção a equação 4.4:

$$Z_t > \mu_s + c * \sigma_{Z_t} \quad (4.4)$$

Em que:

$$Z_t = (1 - \lambda) * Z_{(t-1)} + \lambda * e_t^s, \quad s.t. \quad t > 0 \quad (4.5)$$

$$\sigma_{Z_t} = \sqrt{\frac{\lambda}{2 - \lambda} (1 - (1 - \lambda)^{2t})} \sigma_s \quad (4.6)$$

Nessa equação, a constante pré-definida  $c$  aumenta ou diminui o nível de tolerância para o erro de previsão,  $\lambda$  indica a importância do nível dado aos erros de previsão mais recentes e  $e_t^s$  significa o erro médio do exame para um único exemplo novo, chegando ao tempo  $t$ .

Finalmente, espera-se que esse método supere os métodos da literatura, uma vez que ele opera monitorando uma distribuição de erro estatístico, diferente dos outros métodos, que monitoram apenas um único resultado de um modelo.

#### 4.2.3 Detecção de Mudança Baseado em Sensores

Para esse método, a ideia é escolher um conjunto de partículas (sensores -  $S$ ) e monitorar essas partículas em uma maneira isolada. Esse método tem duas diferenças em relação ao método de comportamento: (i) apenas as melhores partículas são consideradas e (ii) a mudança de conceito é detectada apenas quando todos os sensores concordam. Nesse caso a mudança de conceito acontece quando:

$$Z_t > \mu_p + c * \sigma_{Z_t}, \quad \forall \quad p \in S \quad (4.7)$$

Em que:

$$Z_t = (1 - \lambda) * Z_{(t-1)} + \lambda * e_t^p \quad (4.8)$$

$$\sigma_{Z_t} = \sqrt{\frac{\lambda}{2 - \lambda} (1 - (1 - \lambda)^{2t})} \sigma_p \quad (4.9)$$

é satisfeito para todos os sensores.

Diferente do método de comportamento, individualmente cada partícula usa a média dos erros ( $\mu_p$ ) e o desvio padrão ( $\sigma_p$ ) do conjunto de treinamento e o atual erro das partículas ( $e^p$ ). Como todas as melhores partículas devem concordar que a mudança de conceito foi detectada, esperamos que esse método reduza o número de detecções de mudanças falso positivas em relação ao método de comportamento.

#### 4.2.4 Detecção de Mudança Baseado no Voto Majoritário dos Sensores

Esse método é uma variação do método de sensores e investiga a mudança de conceito analisando uma porção deles. A razão para isso é que depender do acordo de todos os sensores torna a detecção muito conservativa. Conservativa no sentido que o algoritmo depende de muitos fatores para conseguir acionar uma mudança de conceito, neste caso muitos sensores. O grande problema de ser conservativo é que quanto mais tempo o algoritmo demora para alertar sobre uma mudança, maiores serão os atrasos na detecção e maior será a degradação do modelo preditivo.

Então, dado um vetor de sensores  $S = (s_1, s_2, \dots, s_n)$  onde  $n$  é a quantidade, cada sensor no vetor pode detectar separadamente e  $\text{moda}(S)$  indica uma mudança (por exemplo., a mudança de conceito vai acontecer se a maioria dos sensores concorda).

Espera-se com essa abordagem obter uma quantidade semelhante de detecções falso positivas do método de sensores e obter menos atrasos sobre as detecções.

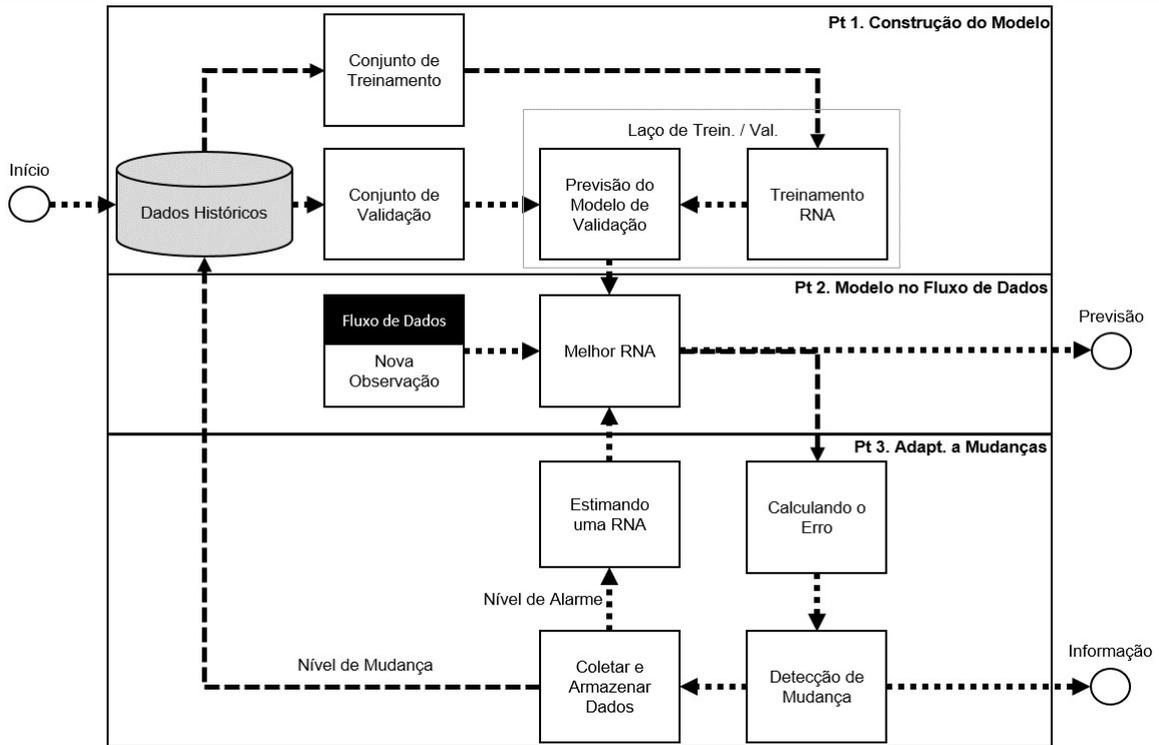
### 4.3 Métodos de Adaptação Após a Detecção de Mudança de Conceito

Com o objetivo de responder as questões de pesquisa do bloco 2 e 3, que dizem que utilizar as partículas e soluções de enxames passados na adaptação melhoram o desempenho do sistema, apresentamos dois novos métodos que usam mecanismos de otimização dinâmica, esses métodos serão descritos na seguinte ordem: (i) visão geral dos métodos de adaptação; (ii) P-IDPSO-ELM-DDSV – adaptação por partículas e (iii) M-IDPSO-ELM-DDSV – adaptação por memória. Ambos usam o detector de mudança de conceito baseado no voto majoritário dos sensores (DDSV) 4.2.4.

#### 4.3.1 Visão Geral dos Métodos de Adaptação

A Figura 4.1 é dividida em três partes e representa a estrutura dos sistemas com mecanismos de adaptação. A parte 1 refere-se a construção do modelo preditivo. Na parte 2, a fase de previsão é apresentada e na parte 3 as estratégias de mudança de conceito são consideradas.

Para construir o primeiro modelo de previsão, conforme descrito na parte 1, um conjunto de  $n$  observações do fluxo de dados é coletado e dividido em conjunto de treinamento e validação. O conjunto de validação desempenha um papel importante para evitar que o modelo se ajuste demais. Uma vez que o modelo adequado está disponível, ele se torna o preditor atual do fluxo de dados.



**Figura 4.1:** Fluxograma do procedimento geral dos sistemas com mecanismos de adaptação.

Conforme mostrado na parte 2, o preditor atual é aplicado ao fluxo de dados para prever a observação ( $n + 1$ ) com base nas últimas observações. Neste momento, a cada iteração o erro de regressão é computado e armazenado.

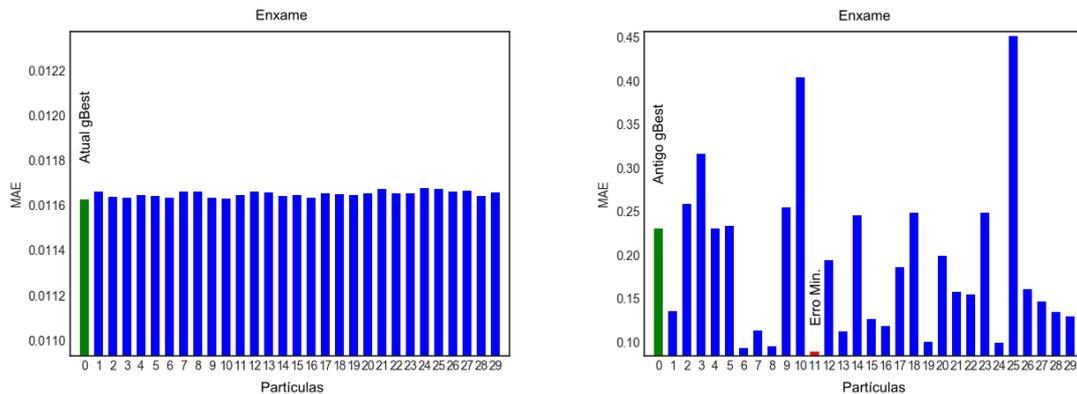
Por fim, a parte 3 usa a distribuição de erros para detectar mudanças no processo gerador dos dados. O nível de erro atual é então categorizado como normal (ou seja, o nível de erro permanece baixo), o alarme (isto é, o nível de erro sugere uma degradação) e o nível de mudança (onde o detector de mudança confirma e informa ao usuário uma mudança de conceito). Cada nível de erro produz uma resposta diferente: (i) para o nível de erro normal, o modelo permanece intacto; (ii) para o nível de erro de alarme, o sistema começa a coletar as observações atuais para retreinar o modelo usando essas novas observações no caso de uma mudança de conceito (para este nível de erro, a cada nova observação o modelo é atualizado); e (iii) dado que ocorreu um nível de mudança, são coletadas mais observações e adicionadas às observações coletadas durante o nível de alarme, quando a quantidade de observações chega a  $n$  um novo modelo é retreinado.

#### 4.3.2 Adaptação Por Partículas

Como neste trabalho adota-se uma abordagem de enxame e as partículas desse enxame representam diferentes modelos com capacidades distintas para se adaptarem a um novo conceito, é importante reconhecer qual modelo (partícula) se adapta melhor ao novo conceito. A Figura 4.2 mostra os desempenhos de um enxame com 30 partículas em dois momentos diferentes,

quando o conceito permanece com as mesmas características na fase de treinamento (Figura 4.2(a)) e quando ocorre um novo conceito (Figura 4.2(b)).

Na Figura 4.2, pode-se notar que as partículas funcionam de forma semelhante e o erro e desvio padrão são baixos. Por outro lado, quando um novo conceito ocorre, os desempenhos das partículas são arbitrariamente organizados e o desvio padrão do erro aumenta altamente.



(a) Conceito Recentemente Treinado

(b) Novo Conceito.

**Figura 4.2:** Distribuição de erros de partículas.

Quando os modelos são treinados, a partícula verde (Figura 4.2) é empregada como modelo de regressão, no entanto, esse modelo torna-se um dos piores modelos quando ocorre um novo conceito. O Algoritmo 3 visa resolver esse problema.

No Algoritmo 3, a variável *paramEsp* representa os parâmetros específicos do método de detecção de mudança de conceito. Quando o módulo de detecção indica um nível de alarme para o erro, os dados são coletados. Dado que o sistema está no nível de alarme e que a mudança de conceito dura mais de  $n/2$  (onde  $n$  é o número de observações necessárias para treinar os modelos) unidades de tempo para ocorrer, o sistema esquece as observações armazenadas porque elas podem ser obsoletas. Pelo contrário, se uma mudança de conceito for detectada, o sistema coleta  $n$  observações para construir os novos modelos.

As linhas 7 a 12 apresentam o processo de coleta e esquecimento de dados no período de alarme, conforme descrito anteriormente. As linhas 14 a 19 mostram como os dados são

coletados após a detecção de um novo conceito, também descrito anteriormente.

---

**Algoritmo 3:** Adaptação por partículas

---

**Entrada:**  $X$ ,  $n$ , tamanhoEnxame, paramEsp

```

1 bufferTr =  $\emptyset$ ; // Obs coletadas para retreinamento
2 tr :=  $X(t : t + n)$ ; // tr é o atual conjunto de treinamento
3  $S = \{p_1, p_2 \dots p_{\text{tamanhoEnxame}}\}$ ; // Treina o enxame S com IDPSO
4 para cada nova observação faça
5      $\hat{x}_t := gBest(x_t)$ ; // previsão gBest
6     nivelErro := DetecMudanca(paramEsp,  $\hat{x}_t$ );
7     se (nivelErro == alarme) então
8         bufferTr = bufferTr +  $(x_t, y_t)$ ;
9         se (comprimento(bufferTr) ==  $(n/2)$ ) então
10            | bufferTr =  $\emptyset$ ;
11        fim
12    fim
13    se (nivelErro == mudanca) então
14        | se (comprimento(bufferTr) ==  $n$ ) então
15            |  $S = \{p_1, p_2 \dots p_{\text{tamanhoEnxame}}\}$ ; // Treina um novo enxame S
16        | fim
17        | bufferTr = bufferTr +  $(x_t, y_t)$ ;
18        | Avalia todas as partículas em bufferTr e atualiza o gBest;
19    fim
20 fim

```

---

### 4.3.3 Adaptação Por Memória

Esta abordagem baseia-se no pressuposto de que boas soluções passadas possam se tornar válidas novamente no futuro. A expectativa é que os conceitos antigos possam retornar e depois serem tratados por uma solução pré-treinada. O Algoritmo 4 apresenta uma versão do Algoritmo 3 capaz de usar uma memória com modelos criados para conceitos passados. Em contraste com Algoritmo 3, quando ocorre uma alteração no Algoritmo 4, o novo  $gBest$  é escolhido entre todas

as soluções contidas em uma memória.

---

**Algoritmo 4:** Adaptação por memória

---

**Entrada:**  $X$ ,  $n$ , tamanhoEnxame, paramEsp

```

1 bufferTr =  $\emptyset$ ; // Obs coletadas para retreinamento
2 tr :=  $X(t : t + n)$ ; // tr é o atual conjunto de treinamento
3  $S = \{p_1, p_2 \dots p_{\text{tamanhoEnxame}}\}$ ; // Treina o enxame S com IDPSO
4 Guardar Solução( $gBest$ );
5 para cada nova observação faça
6    $\hat{x}_t := gBest(x_t)$ ; //  $gBest$  previsão
7   nivelErro := DetecMudanca(paramEsp,  $\hat{x}_t$ );
8   se (nivelErro == alarme) então
9     bufferTr = bufferTr +  $(x_t, y_t)$ ;
10    se (length(bufferTr) ==  $(n/2)$ ) então
11      bufferTr =  $\emptyset$ ;
12    fim
13  fim
14  se (nivelErro == mudanca) então
15    se (length(bufferTr) ==  $n$ ) então
16       $S = \{p_1, p_2 \dots p_{\text{tamanhoEnxame}}\}$ ; // Treina um novo enxame S
17      Guardar Solução( $gBest$ );
18    senão
19      bufferTr = bufferTr +  $(x_t, y_t)$ ;
20       $gBest :=$  Lembrar Solução( $gBest$ , bufferTr);
21    fim
22  fim
23 fim
24 fim

```

---

A memória contendo soluções passadas tem um tamanho máximo representado por  $tam\_max\_mem$ . Se o sistema tentar inserir uma solução na memória e não houver mais espaço disponível, a distância euclidiana entre o  $gBest$  e o vizinho mais próximo na memória é calculada e se  $d$  for menor do que  $limiar\_exc$ , o vizinho mais próximo é substituído por  $gBest$ . O algoritmo

5 apresenta este processo.

---

**Algoritmo 5:** Guardar Solução
 

---

**Entrada:** gBest

```

1 se comprimento(memoria) < tam_max_mem então
2   memoria := memoria + gBest;
3   senão
4     d = minDist(memoria, gBest); // retorna menor distancia
5     se d < limiar_exc então
6       memoria := memoria - vizinho;
7       memoria := memoria + gBest;
8     fim
9   fim
10 fim
```

---

Finalmente, o algoritmo 6 retorna a solução com o melhor desempenho para as observações armazenadas recentemente no fluxo de dados (*bufferTr*). Esta solução pode ser o atual *gBest* ou uma solução contida na memória.

---

**Algoritmo 6:** Lembrar Solução
 

---

**Entrada:** gBest, memoria, bufferTr

```

1 minError = error(memoria, bufferTr); // retorna a melhor solução
2  $\hat{x}_t := gBest(bufferTr)$ ;
3 se minError <  $\hat{x}_t$  então
4   gBest := melhor solução na memória;
5   retorna gBest;
6 fim
```

---

#### 4.4 Síntese do Método Proposto

Neste capítulo propomos 5 algoritmos para a previsão de séries temporais. Três deles introduzem técnicas de detecção baseadas em inteligência de enxames e dois introduzem o uso de técnicas de otimização dinâmica.

Os métodos de detecção monitoram o erro do modelo preditivo através de CDT's por meio de vários modelos de previsão gerados por um método de inteligência de enxames.

Já os sistemas adaptativos utilizam técnicas de Otimização Dinâmica Evolucionária (Evolutionary Dynamic Optimization (EDO) para melhorar a adaptabilidade dos modelos.

As principais contribuições dessa dissertação são: (i) os métodos de detecção de mudança de conceito baseados em características de enxames, que tentam responder as perguntas de pesquisa do primeiro (1) bloco desse trabalho; e (ii) as técnicas de otimização dinâmica dos sistemas adaptativos, que tentam responder as perguntas do segundo (2) e terceiro (3) bloco desse trabalho.

# 5

## EXPERIMENTOS

Nesse capítulo descreveremos os experimentos realizados para avaliar os sistemas propostos tanto em séries artificiais quanto em séries reais. Discutiremos os objetivos dos experimentos, como os algoritmos foram avaliados, a configuração dos parâmetros utilizados e os conjuntos de dados utilizados para tornar essa investigação reproduzível. Então, apresentaremos e analisaremos os principais resultados obtidos nesses experimentos e como eles induzem a validação das hipóteses formuladas no Capítulo 1.

### 5.1 Configuração Experimental

A configuração dos experimentos está organizada da seguinte forma: 5.1.1: configurações para avaliar a habilidade de detecção dos algoritmos; 5.1.2: configurações usadas para avaliar a habilidade de previsão dos algoritmos e; 5.1.3: os conjuntos de dados utilizados.

#### 5.1.1 Habilidade de Detecção de Mudança de Conceito

Avaliar a acurácia de detecção de mudança de conceito é o primeiro tipo de análise estudada nesse trabalho. O objetivo desses experimentos é confirmar a hipótese referente as perguntas do bloco 1 desse trabalho, que diz que monitorar estatísticas do erro do exame é efetivo para diminuir o número de detecções falsas positivas. Além disso, o experimento avalia o grau de confiabilidade dos métodos em termos de atrasos na detecção para uma operação transparente para o usuário.

Como a mudança de conceito não é amplamente investigada em séries temporais, ainda há uma lacuna de séries com mudanças de conceitos conhecidas. Então, para ser possível o estudo e a análise de detecção, 180 séries artificiais contendo diferentes mudanças de conceitos foram geradas. Além destas, duas séries reais com mudanças de conceitos conhecidas também foram experimentadas. Mais detalhes sobre essas séries podem ser vistas na seção 5.1.3. Para cada série temporal, cada método de detecção foi executado 10 vezes e a média dos resultados das execuções foi usada para a comparação. Em relação as séries reais, para ser possível uma análise mais eficiente os algoritmos foram executados 30 vezes.

Para permitir a análise discutida acima, 7 métodos baseados em CDT foram utilizados:

Método de Detecção de Mudanças (Drift Detection Method)) (DDM) (GAMA et al., 2004); Média Móvel Exponencial Ponderada Para Detectar Mudanças de Conceito (Exponentially Weighted Moving Average Charts for Detecting Concept Drift) (ECDD) (ROSS et al., 2012); e Extração de Características Para Detecção Explícita de Mudança de Conceito (Feature Extraction for Explicit Concept Drift Detection) (FEDD); (CAVALCANTE; MINKU; OLIVEIRA, 2016) e os métodos de detecção baseados em inteligência de enxames: Detecção de Mudança de Conceito Baseado no Comportamento do Enxame (Concept Drift Detection Based on Swarm Behavior) (IDPSO-ELM-B) (OLIVEIRA et al., 2017); Detecção de Mudança de Conceito Baseada em Sensores (Concept Drift Detection Based on Sensors) (IDPSO-ELM-S) (OLIVEIRA et al., 2017); Detecção de Mudança de Conceito Baseada no Voto Majoritário de Sensores (Concept Drift Detection Based on Majority Vote of Sensors) (IDPSO-ELM-DDSV); e Reiniciar o Algoritmo de Enxame de Partículas (Reinitialize Particle Swarm Optimization) (RPSO) (RAKITIANSKAIA; ENGELBRECHT, 2012).

Os algoritmos foram comparados de acordo com a curva de Detecções Falso Positivas (False Positive Rate) (FPR) vs Atrasos na Detecção (Drift Delay) (DD) propostas por (BORACCHI; ROVERI, 2014). Essas curvas mostram o comportamento dos métodos comparados para todas as configurações de parâmetros investigadas. Portanto, não há necessidade de fixar ou escolher os melhores parâmetros para cada método. Cada ponto da curva corresponde a um parâmetro diferente. Os melhores métodos são aqueles com curvas próximas aos eixos do gráfico, que são aqueles com pequenos atrasos e poucas detecções falso positivas. Para que a comparação seja justa, todos os métodos de detecção experimentados utilizaram o mesmo tipo de regressor, uma rede neural artificial do tipo ELM.

Os parâmetros usados neste experimento são os seguintes: RPSO-ELM:  $\omega = \{1, 2, 3, 4\}$ ; ELM-DDM:  $\omega = \{2, 4, 6, 8\}$ ; ELM-FEDD:  $c = \{0, 0.25, 0.5, 0.75\}$ ; ELM-ECDD, IDPSO-ELM-B, IDPSO-ELM-S e IDPSO-ELM-DDSV:  $c = \{0.25, 0.5, 0.75, 1\}$ . A constante  $c$  controla a tolerância para o erro de previsão. Altos valores de  $c$  aumentam os atrasos na detecção. Se baixos valores são utilizados, menos tolerância ao erro é permitida, então aumenta o número de detecções falso positivas. A constante  $\omega$  representa o número de desvios padrões e comporta-se de forma semelhante a  $c$ . Essa constante aumenta a tolerância do erro em relação a média do erro do conjunto de treinamento.

Com o objetivo de atribuir um valor adequado para o número de observações para treinamento dos modelos de regressão, os valores 100, 200, 300, 400 e 500 foram experimentados. Para as séries artificiais o valor 300 produziu os melhores resultados. Para as séries reais com mudanças de conceitos conhecidos atribuímos o valor igual a 100, uma vez que as séries são curtas e valores maiores que 100 cobrem algumas mudanças de conceito.

Para os métodos que empregam o IDPSO e PSO, foram realizados experimentos preliminares para encontrar os melhores parâmetros em relação à precisão de previsão. Os melhores parâmetros foram os seguintes: 50 iterações; 30 partículas (tamanho do enxame); 3 iterações sem melhora como critério de parada. Para o IDPSO:  $\omega_{inicial} = 0.8$ ;  $\omega_{final} = 0.4$ ;  $c_1 = c_2 =$

2;  $posicao\_max = velocidade\_max = 1$ ;  $posicao\_min = velocidade\_min = -1$  e para o PSO:  $\omega = 0.8$ .

### 5.1.2 Habilidade de Previsão de Séries Temporais

O segundo conjunto de experimentos foi executado para avaliar a precisão dos métodos propostos em termos de erro de previsão de séries temporais. Este experimento visa validar as hipóteses dos blocos 2 e 3 desse trabalho, que diz que usar uma abordagem de adaptação melhora o desempenho de previsão do sistema.

Para este caso, o MAE (Eq. 2.1) foi considerado por ser uma métrica amplamente usada nos trabalhos de séries temporais e por medir de forma mais simples a diferença entre a previsão e o dado real. Além disso, para atestar de forma estatística os resultados, o teste não paramétrico de Friedman, com nível de significância  $\alpha = 0.05$  foi usado, conforme proposto por (DEMSAR, 2006). Se a hipótese nula de que não há diferença estatística entre os métodos é rejeitada, o teste de Nemenyi é usado com nível de significância  $\alpha = 0.05$ . Os testes foram escolhidos pelos seguintes motivos: O teste de Friedman Ranqueia os algoritmos de acordo com o seu desempenho para o conjunto de dados e o teste de Nemenyi compara todos os algoritmos entre si, para identificar quais são diferentes estatisticamente. Além disso, esses testes são não-paramétricos e são mais adequados para poucos dados e dados sem normalidade (FRIEDMAN, 1940).

Testes preliminares foram realizados para obter os melhores parâmetros da rede ELM considerando a precisão da previsão. A melhor configuração encontrada usou 10 neurônios escondidos, 5 neurônios de entrada e a função sigmoide como função de ativação. Além disso, o restante dos parâmetros foram atribuídos de acordo com a seção 5.1.1.

Além das séries da seção 5.1.1, foram utilizadas três séries reais contendo índices do mercado de ações, descritas na seção 5.1.3. Para cada série temporal, foram feitas 30 execuções. Essa quantidade de execuções permite uma análise mais confiável dos resultados. Uma comparação com uma rede ELM sem detecção de mudança de conceito também foi realizada para destacar os benefícios da detecção de mudança de conceito.

Além disso, para encontrarmos os parâmetros que levavam ao menor erro de predição, avaliamos a média das acurácias de previsão dos algoritmos para todas as séries. Os melhores parâmetros foram: RPSO-ELM-ELM:  $\omega = 1$ ; ELM-DDM:  $\omega = 8$ ; ELM-FEDD:  $c = 0.25$ ; ECDD  $c = 0.25$ , IDPSO-ELM-B, IDPSO-ELM-S, IDPSO-ELM-DDSV. Para os métodos com mecanismos de adaptação: Método de Adaptação à Mudança de Conceito Baseado em Partículas (Particle-based Concept Drift Adaptation Method) (P-IDPSO-ELM-DDSV) e Método de Adaptação à Mudança de Conceito Baseado em Memória (Memory-based Concept Drift Adaptation Method) (M-IDPSO-ELM-DDSV):  $c = 0.25$ . Como nível de alarme, o parâmetro que obteve o menor erro foi:  $w = 0.1$ . Por fim, para o método M-IDPSO-ELM-DDSV, utilizamos 30 espaços para alocação de soluções e limiar de exclusão de soluções igual a 3. Ambos parâmetros foram

definidos por apresentarem melhores resultados em experimentos preliminares.

### 5.1.3 Descrição dos Dados

Esta subseção descreve as séries usadas nos experimentos deste trabalho, sendo elas: séries artificiais; séries reais com mudanças de conceitos conhecidas; e séries reais financeiras.

#### 5.1.3.1 Séries Artificiais

Foram criados 6 tipos de séries temporais, sendo elas: (i) séries temporais lineares com mudanças graduais; (ii) séries temporais lineares com mudanças abruptas; (iii) séries temporais não-lineares com mudanças graduais; (iv) séries temporais não-lineares com mudanças abruptas; (v) séries lineares sazonais e (vi) séries com conceitos misturados, chamadas de híbridas. As mudanças abruptas foram simuladas por uma grande alteração nos parâmetros do modelo, já as graduais por uma pequena alteração. Estas séries estão ilustradas na Figura 5.1.

Cada um dos seis tipos de séries temporais é composto por 30 séries diferentes, totalizando 180 séries. Cada série contém 20000 observações e são compostas por 10 conceitos, cada conceito com 2000 observações. As séries lineares foram simuladas usando um processo autorregressivo (AR), processo definido por:

$$x_t = a_1x_{(t-1)} + a_2x_{(t-2)} + \dots + a_px_{(t-p)} + w_t \quad (5.1)$$

Em que  $a_1, \dots, a_p$  são parâmetros do modelo AR e  $w_t : t = 1, \dots, n$  é um ruído branco Gaussiano, em que as variáveis  $w_t$  pertencem a uma distribuição normal e são independentes e identicamente distribuídas (IID).

As séries temporais não-lineares usadas nos experimentos foram geradas por dois modelos de suavização, dados por Cavalcante, Minku e Oliveira (2016):

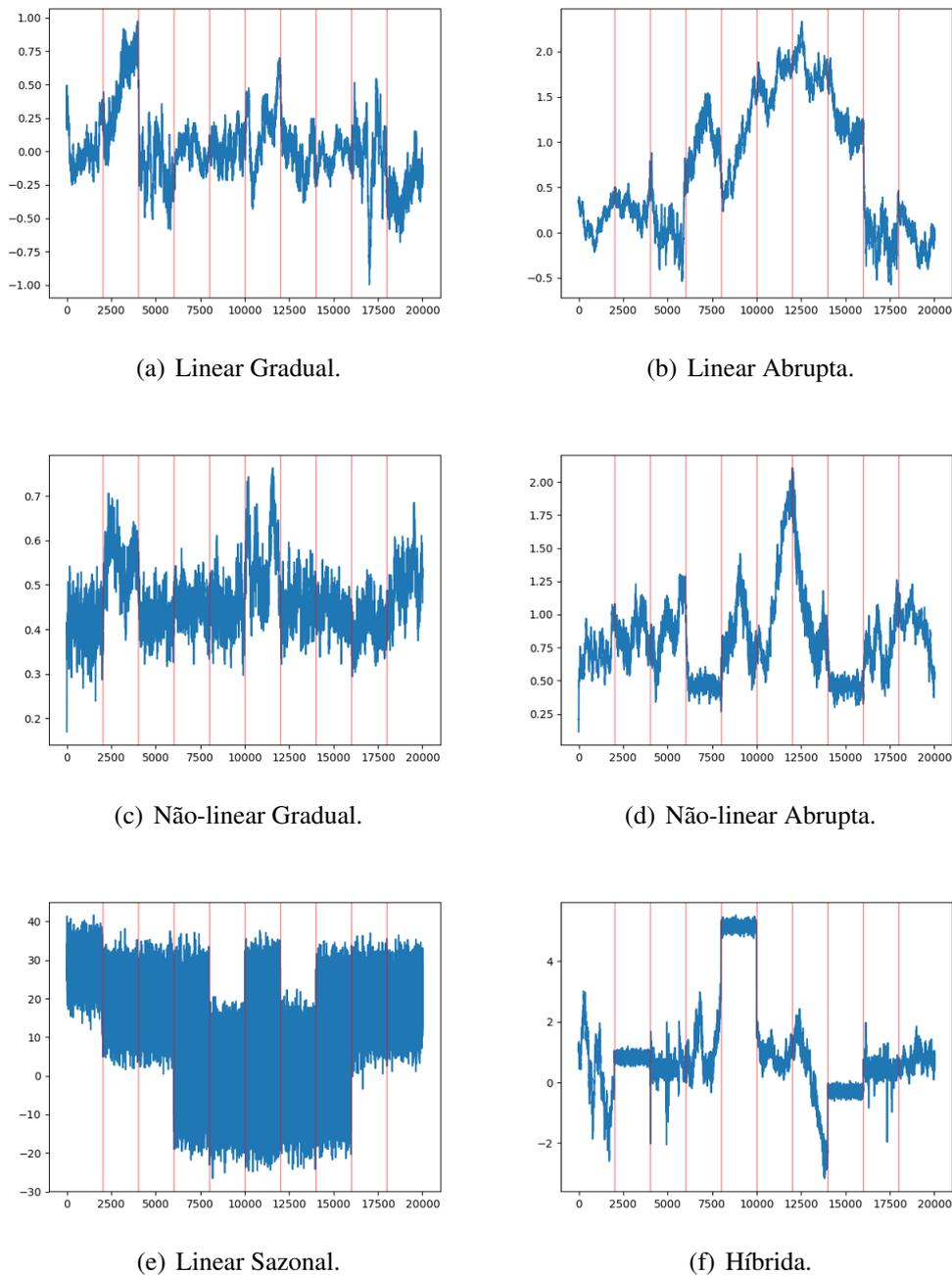
$$x_t = [a_1x_{(t-1)} + a_2x_{(t-2)} + a_3x_{(t-3)} + a_4x_{(t-4)}] * [1 - \exp(-10x_{t-1})]^{-1} + w_t \quad (5.2)$$

$$x_t = a_1x_{(t-1)} + a_2x_{(t-2)} + [a_3x_{(t-1)} + a_4x_{(t-2)}] * [1 - \exp(-10x_{t-1})]^{-1} + w_t \quad (5.3)$$

As séries temporais lineares sazonais foram simuladas por um modelo linear usando variáveis ( $s$ ) indicando o número de períodos das séries. Esse modelo contém  $s$  temporadas definidos por:

$$x_t = m_t + \beta_{(1+\text{mod}(t-1,s))} + w_t \quad (5.4)$$

onde  $m_t$  representa a tendência,  $\beta_{(1+\text{mod}(t-1,s))}$  o fator sazonal e  $w_t$  o ruído gaussiano.



**Figura 5.1:** Exemplos de Séries Artificiais.

Os parâmetros dos modelos foram variados de modo a representar as respectivas mudanças de conceitos. A Tabela 5.1 contém os parâmetros investigados, bem como as respectivas equações. Por fim, para tentar respondermos as perguntas de pesquisa do bloco 3, que descrevem o uso de soluções passadas, simulamos conceitos recorrentes. Esses conceitos recorrem a partir do sexto conceito de forma retrospectiva.

Tabela 5.1: Parâmetros usados nos modelos das séries temporais

		$\alpha$	<b>p</b>
<b>Linear Gradual</b>	1	{0.007,-0.253,0.855,0.391}	4
	2	{-0.443,0.447,1.352,-0.356}	4
	3	{0.003,-0.328,0.146,1.172}	4
	4	{0.333,-0.113,0.054,0.715}	4
	5	{-0.634,0.335,1.36,-0.074}	4
	6	{-0.441,0.074,1.257,0.108}	4
		$\alpha$	<b>p</b>
<b>Linear Abrupto</b>	1	{0.149, 0.051, 0.433, 0.367}	4
	2	{-0.318, 0.413, 1.148, -0.245}	4
	3	{0.003, -0.328, 0.146, 1.172}	4
	4	{-0.443, 0.447, 1.352, -0.356}	4
	5	{-0.027, 0.22, -0.038, 0.845}	4
	6	{-0.479, 0.856, 0.025, 0.598}	4
		$\alpha$	<b>Modelo</b>
<b>Não-linear Gradual</b>	1	{0.02, 0.149, 0.122, 0.691}	NL 1
	2	{0.214, 0.175, 0.256, 0.349}	NL 1
	3	{0.675, 0.04, 0.129, 0.141}	NL 1
	4	{0.259, 0.187, 0.251, 0.291}	NL 1
	5	{0.333, -0.113, 0.054, 0.715}	NL 1
	6	{0.178, -0.091, 0.363, 0.545}	NL 1
		$\alpha$	<b>Modelo</b>
<b>Não-linear Abrupto</b>	1	{-0.067, 0.234, 0.155, 0.677}	NL 1
	2	{-0.507, 0.259, 1.397, -0.15}	NL 1
	3	{-0.439, 0.375, 1.333, -0.269}	NL 1
	4	{0.07, -0.052, 0.635, 0.334}	NL 1
	5	{-0.443, 0.447, 1.352, -0.356}	NL 1
	6	{-0.276, 0.334, 0.41, 0.532}	NL 1
		$\beta$	<b>s</b>
<b>Linear Sazonal</b>	1	{34, 32, 30, 28, 26, 24, 22, 24, 26, 28, 30, 32}	12
	2	{34, 26, 18, 10, 18, 26, 10}	7
	3	{34, 26, 18, 10, 18, 26}	6
	4	{34, 26, 18, 10, 2, -6, -14, -6, 2, 10, 18, 26}	12
	5	{34, 10, -14, 10}	4
	6	{38, 28, 18, 8, 0, -8,-18,-8, 0, 8, 18, 28}	12
		$\alpha, \beta$	<b>Modelo</b>
<b>Híbrida</b>	1	{0.003, -0.005, 1.0}	AR
	2	{Últimas 3 observações}	Sazonal
	3	{0.059, 0.086, 0.62, 0.21}	NL 2
	4	{0.018, 0.95, 0.032}	AR
	5	{Últimas 3 observações}	Sazonal
	6	{0.55, 0.024, 0.41, 0.009}	NL 1

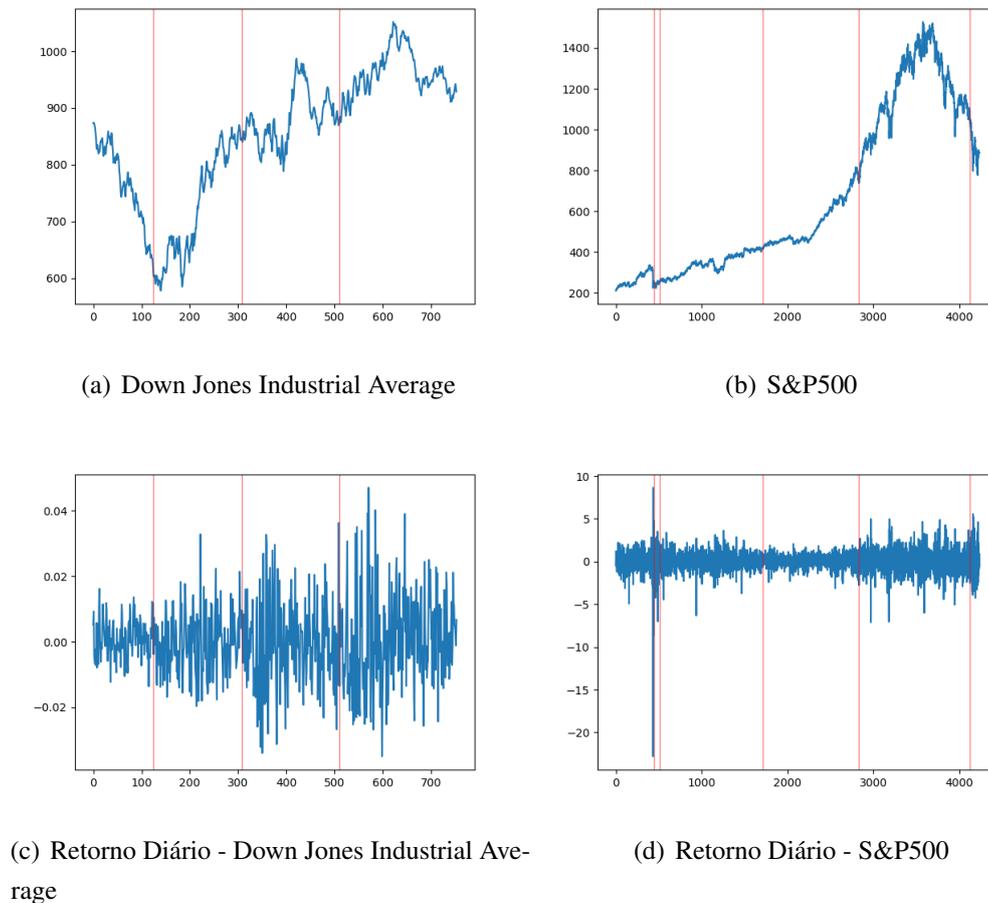
### 5.1.3.2 *Séries Reais Com Mudanças de Conceitos Conhecidas*

As séries reais descritas nesta Subseção, primeiramente foram estudadas em artigos de detecção de pontos de mudanças que têm como objetivo encontrar instantes de tempo onde ocorreram mudanças de conceito (CAVALCANTE; MINKU; OLIVEIRA, 2016). A primeira das séries trata-se da série financeira diária Dow Jones Industrial Average, estudada no artigo de Adams e MacKay (2007) e ilustrada Figura 5.2(a). Esta série possui 735 pontos e segundo os resultados de Adams e MacKay (2007), as mudanças ocorrem nos seguintes instantes de tempo: 124, 307 e 510. Em uma análise visual pode ser observado que os conceitos dessa série são abruptos, uma vez que dividem a série em comportamentos bem distintos. Além da análise estatística da detecção de pontos de mudanças, os autores cruzaram as detecções encontradas com momentos históricos que possivelmente trouxeram mudanças na economia das ações da série, sendo elas nos seguintes instantes de tempo:

1. **124:** Condenação dos assessores do ex-presidente Nixon, G. Gordon Liddy e James W. McCord, Jr. Em 30 de Janeiro de 1973.
2. **307:** Início do embargo OPEC contra os Estados Unidos em 19 de outubro de 1973.
3. **510:** A renúncia do ex-presidente Nixon em 9 de agosto de 1974.

A segunda série trata-se da série financeira S&P500, uma série diária com 4229 pontos, estudada no artigo de Lavielle e Teyssiere (2006) e ilustrada na Figura 5.2(b). De acordo com os resultados as detecções acontecem nos seguintes instantes de tempo: 448, 508, 1715, 2826 e 4119. Pode-se observar que as mudanças de conceito nessa série são graduais, uma vez que os dados vão evoluindo lentamente ao longo do tempo. Os autores também cruzaram as detecções encontradas pelo método com momentos históricos, essas mudanças foram causados pelos seguintes motivos:

1. **448 e 508:** Aumento e diminuição da volatilidade após o acidente de ações de outubro de 1987.
2. **2826:** Crise asiática do verão de 1997.
3. **4119:** Maio de 2002, um aumento na volatilidade devido ao colapso da bolha da internet e às falências fraudulentas da Enron e da WorldCom, que lançaram dúvidas sobre a posição contábil atual de várias empresas e aumentaram o nível de incerteza.



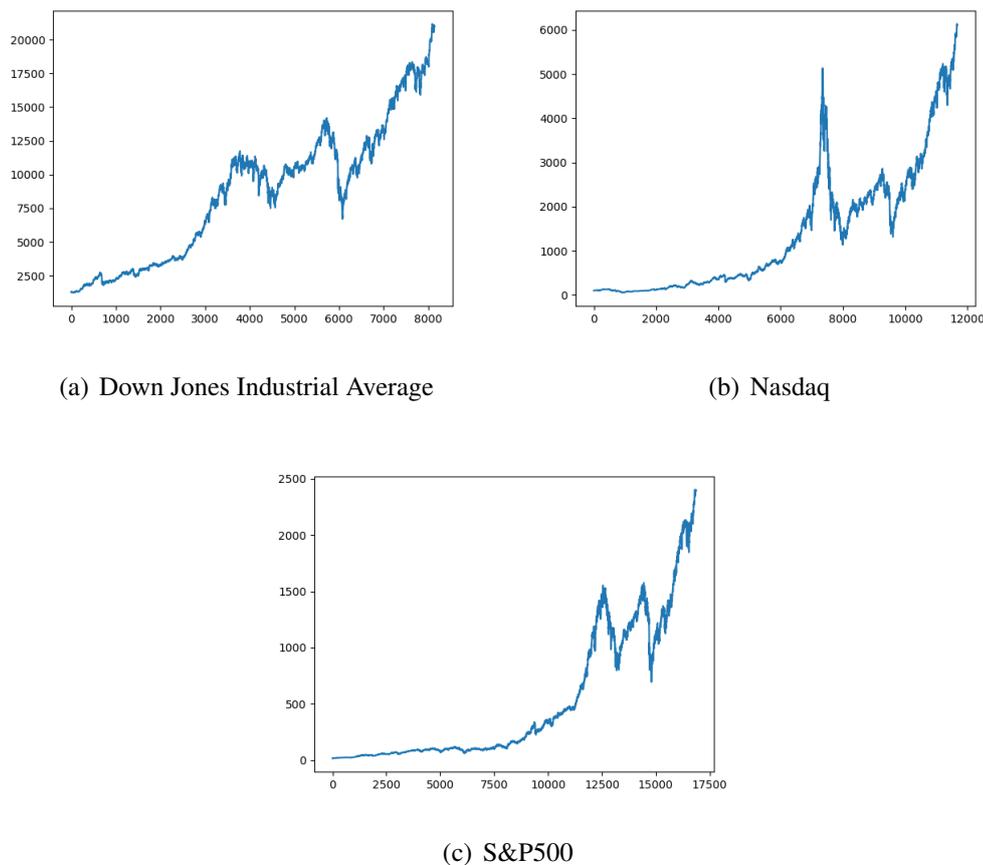
**Figura 5.2:** Séries reais com mudanças conhecidas.

Essas séries em seus respectivos artigos, foram estudadas em sua forma estacionária pelo cálculo do retorno diário, como ilustrado nas Figuras 5.2(c) e 5.2(d), mas para este trabalho, as séries foram usadas em sua forma original. A razão para isto é tentar legitimar uma situação real onde os dados chegam dessa maneira.

### 5.1.3.3 Séries Reais Financeiras

Por fim, com o propósito de analisarmos como os métodos propostos se comportam em termos de desempenho de previsão em séries reais, utilizamos três séries reais financeiras, sendo elas: (i) Dow Jones Industrial Average, uma série diária, contendo 8139 pontos, com os dados coletados desde 29 de janeiro de 1985 até 12 de maio de 2017; (ii) Nasdaq, também uma série diária, contendo 11667 pontos, com os dados coletados desde 02 de maio de 1985 até 12 de maio de 2017 e; (iii) S&P 500, uma série diária, contendo 16857 pontos, com os dados coletados desde 15 de maio de 1950 até 12 de maio de 2017. Todas estas séries estão disponíveis no Yahoo finanças <sup>1</sup> e estão ilustradas na figura 5.3.

<sup>1</sup><https://finance.yahoo.com/>



**Figura 5.3:** Séries temporais financeiras.

## 5.2 Resultados

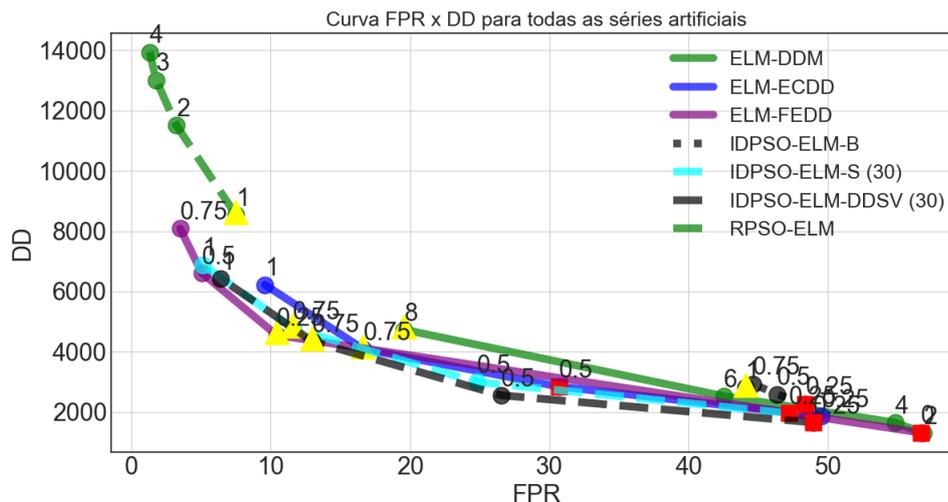
Nessa Seção são apresentados os resultados e discussões sobre os métodos descritos na Seção 5.1. Na Subseção 5.2.1 buscamos responder as questões de pesquisa desse trabalho utilizando as séries artificiais, em seguida na subseção 5.2.2 buscamos validar essas discussões usando séries do mundo real com mudanças de conceitos conhecidas. Por fim, na subseção 5.2.3 avaliamos a capacidade dos métodos propostos em séries reais financeiras.

### 5.2.1 Avaliação para as Séries Artificiais

Para esta Subseção, duas análises foram feitas. A primeira pertinente a capacidade de detecção de mudança de conceito e a segunda referente a habilidade de predição dos modelos.

#### 5.2.1.1 Habilidade de Detecção

Na Figura 5.4, apresenta-se as curvas formadas pelos métodos de detecção para todas as séries artificiais. Os resultados dos pontos dessas curvas consideram as médias das taxas FPR e DD de todas as detecções para todas as séries.



**Figura 5.4:** Média das métricas FPR e DD para todas as séries artificiais.

Na Figura 5.4, percebe-se claramente que dois métodos de detecções possuíram a pior acurácia, uma vez que suas curvas estão muito distantes do ponto (0, 0) (detecção ótima), sendo eles os métodos ELM-DDM e RPSO-ELM. Ambos métodos utilizam estatísticas descritivas para declarar uma mudança de conceito. A diferença entre eles é que o método DDM utiliza uma abordagem que monitora individualmente o erro de cada instância de chegada, verificando se o mesmo está dentro da distribuição do conjunto de treinamento. Já o RPSO-ELM, monitora a média do erro para um conjunto de dados. A dependência de muitos dados para o monitoramento é o que torna essa abordagem inferior, em termos de detecção, dos outros métodos. Se a detecção depende de muitos dados, a média do conjunto só é afetada se, o erro de previsão for muito alto. Além disso, percebe-se com as curvas que todos os métodos que utilizaram CDT's obtiveram melhores detecções que o método RPSO-ELM, o que indica que o mesmo não possui boa transparência quando comparado com os demais.

Em relação a pergunta de pesquisa do bloco 1, que diz respeito que ao monitorar mais características de um enxame a detecção se beneficia, ela pode ser respondida ao comparar os métodos IDPSO-ELM-S (30) e IDPSO-ELM-DDSV (30) com o método ELM-ECDD. Esses métodos utilizam 30 modelos para monitorar a mudança de conceito e também utilizam como CDT o método ECDD, a única diferença deles em relação ao ELM-ECDD é que esses métodos monitoram mais modelos. O mesmo não pode ser dito para o método IDPSO-ELM-B, pois sua curva não apresentou boa acurácia, o que pode estar relacionado com o problema de pontos aberrantes já que esse modelo baseia sua detecção na média do fitness das partículas. Como pode ser visto na Figura 5.5, as curvas dos métodos de detecção baseados em enxames estão mais próximas do eixo ótimo, o que podemos concluir que monitorar mais características de um enxame melhora a detecção.

Outra análise importante a ser discutida é a comparação do método IDPSO-ELM-DDSV (30) com o método IDPSO-ELM-S (30). Percebe-se que o método de enxames que usa a condição

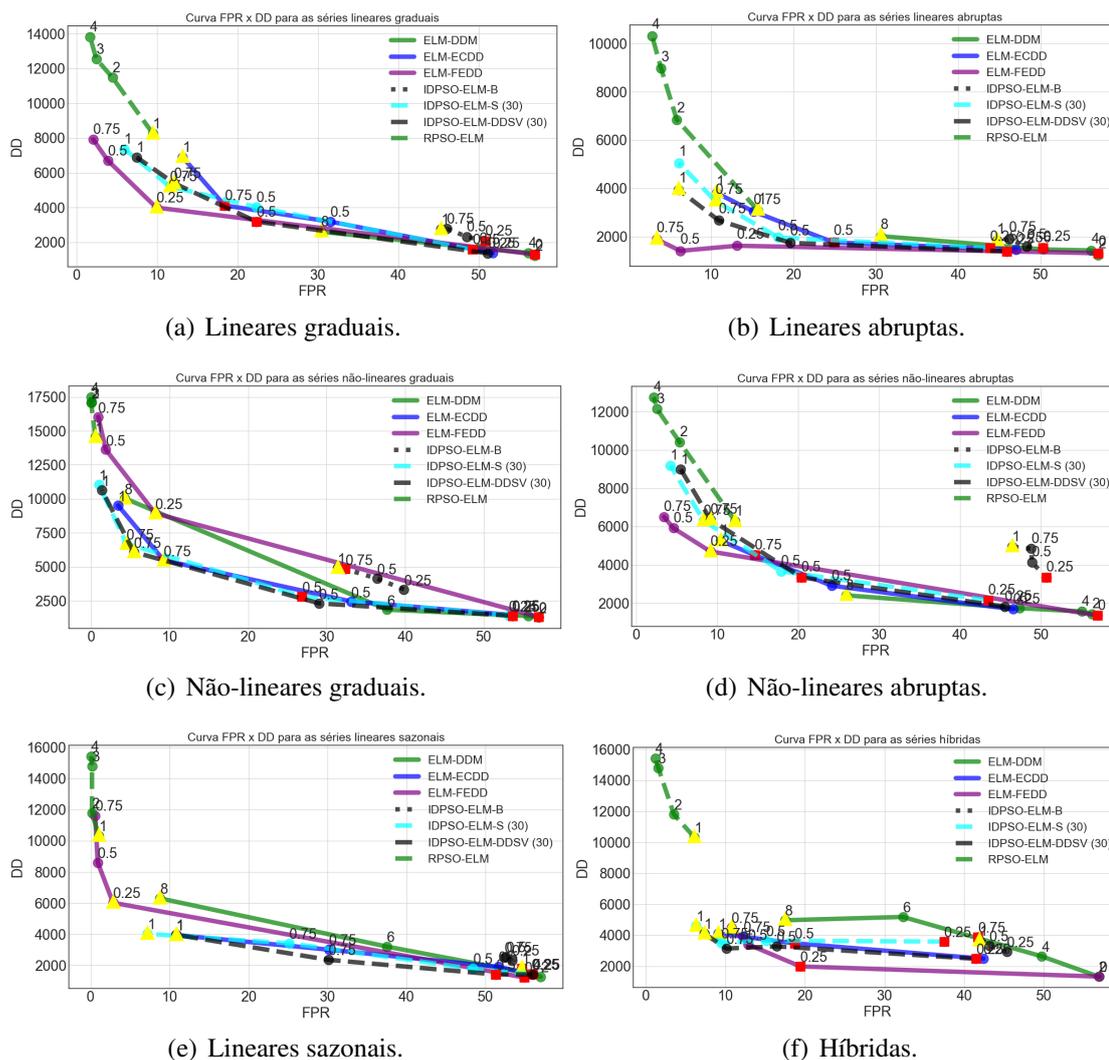


Figura 5.5: Curva FPR x DD para as séries artificiais.

do voto majoritário (IDPSO-ELM-DDSV (30)) alcançou a melhor acurácia de detecção quando comparado com os outros métodos baseados no erro, conforme visto nas suas curvas na figura 5.5. Esse método alcançou melhores resultados em relação ao seu variante (IDPSO-ELM-S (30)) pelo seguinte motivo: conceber a mudança pela maioria dos sensores e não por todos fornece uma grande melhoria no atraso da detecção, enquanto que, a taxa de falsos alarmes não é tão afetada já que uma grande amostra de sensores ainda é considerada.

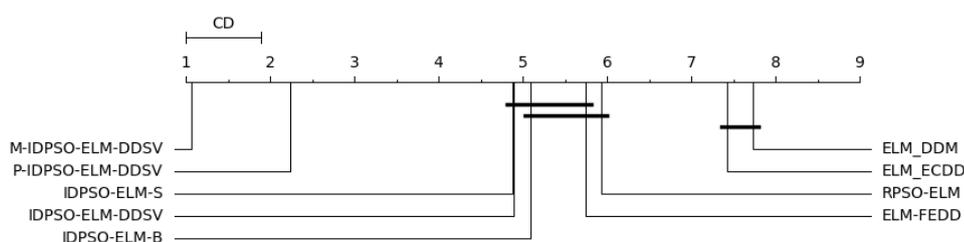
Uma tarefa importante da análise desses resultados é identificar qual o algoritmo que alcançou a melhor acurácia de detecção no geral. As curvas mais próximas do eixo ótimo são dos algoritmos ELM-FEDD e IDPSO-ELM-DDSV (30). Ao observar as imagens da figura 5.5, percebe-se claramente que o algoritmo ELM-FEDD alcançou as melhores acurácias de detecção para as séries lineares graduais (figura 5.5(a)), lineares abruptas (figura 5.5(b)) e não-lineares abruptas (figura 5.5(d)), enquanto que os métodos baseados no erro de previsão do modelo, alcançaram melhores colocações nas séries não-lineares graduais (figura 5.5(c)), séries

sazonais (figura 5.5(e)) e séries híbridas (figura 5.5(f)). A correlação que encontramos para esses resultados foi a forma com que os métodos detectam as mudanças de conceito, por exemplo: nas séries em que os dados são menos voláteis, ou seja, em que a variação dos dados é menor, o algoritmo ELM-FEDD alcançou as melhores acurácias. O FEDD por analisar características estatísticas das séries consegue identificar com maior facilidade quando as mudanças mais simples ocorrem. Já nas séries em que as mudanças de conceitos são mais abruptas e voláteis os métodos baseados no erro do modelo preditivo foram melhores. A explicação para estes resultados é que os métodos baseados no erro dependem que o modelo se degrade para que eles identifiquem uma mudança, então quanto mais abrupto e volátil a mudança, mais rapidamente o modelo se desgasta, favorecendo desse modo os seus testes de detecção.

### 5.2.1.2 Habilidade de Previsão

Essa análise tem como objetivo responder as questões de pesquisa do bloco 2 e 3 desse trabalho. A Tabela 5.2 apresenta os resultados para os modelos com melhor acurácia de previsão. Esses modelos são representados pelos quadrados vermelhos na figura 5.5. Os melhores valores, estatisticamente testados (figura 5.6), são apresentados em negrito.

A Figura 5.6 apresenta o teste de Nemenyi para os melhores modelos. Essa visualização apresenta o Ranking de Friedman, sendo iniciado da esquerda para a direita. O tamanho da distância crítica que mede o quanto os algoritmos são diferentes entre si é demonstrado no início do Ranking pela sigla CD. Se os algoritmos estão interligados por uma linha horizontal, isso significa que eles são equivalentes estatisticamente.



**Figura 5.6:** Teste Nemenyi para os modelos treinados e avaliados nas séries artificiais com mudanças de conceitos.

De acordo com o teste estatístico (figura 5.6) os métodos com treinamento em IDPSO, alcançaram melhores colocações no Ranking em relação aos métodos com o treinamento padrão do ELM. O que confirma a nossa hipótese de que o treinamento por enxames possibilita melhorar a acurácia de predição em redes neurais do tipo ELM. Além disso, estes métodos também alcançaram melhores resultados em relação ao método RPSO-ELM. Estes resultados podem ser explicados por dois motivos. O primeiro devido ao uso de algoritmo do treinamento IDPSO em comparação ao PSO tradicional. O artigo que apresenta o IDPSO (ZHANG; XIONG; ZHANG, 2013) demonstra em experimentos que o seu algoritmo atinge melhor desempenho que o PSO tradicional em questão de busca global. O segundo, e mais importante, está relacionado com

os CDT's. Como dito anteriormente, o método RPSO-ELM baseia sua detecção em estatísticas descritivas, como média e desvio padrão. Essas estatísticas são sensíveis a pontos aberrantes, então se ocorrer algum problema no treinamento, como por exemplo um erro muito alto para alguma das instâncias, a condição de detecção será afetada, dificultando a definição de quando uma mudança de conceito ocorre já que elas receberam uma maior tolerância. Se o módulo de detecção possui dificuldades para identificar quando ocorre uma mudança de conceito, então ele não poderá informar ao regressor quando ele deve ser atualizado, conseqüentemente o deixando se degradar.

Em relação aos métodos que utilizam mecanismos de adaptação, podemos observar na Figura 5.6 que eles alcançaram as melhores colocações do Ranking, atestando diferença estatística dos demais métodos. Com essa primeira constatação, podemos responder as questões de pesquisa referentes ao uso de partículas (bloco 2 de perguntas) e soluções passadas (bloco 3 de perguntas). Concluimos com base no teste (figura 5.6) que utilizar qualquer uma dessas estratégias ajuda o sistema a se adaptar as mudanças melhorando o desempenho de previsão.

Além dessa constatação, também pudemos confirmar que o uso de soluções passadas são eficientes quando conceitos recorrem em séries temporais. Podemos afirmar isto, pois as séries artificiais experimentadas possuem conceitos recorrentes a partir do sexto conceito, como descrito em detalhes na seção 5.1.3.1. Este formato das séries possibilitou o método M-IDPSO-ELM-DDSV alcançar a melhor colocação do Ranking de Friedman (figura 5.6) e atestar diferença estatística dos demais métodos. Além disso, também pudemos confirmar nossa hipótese (hipótese 3) em relação ao método M-IDPSO-ELM-DDSV que se estratégias de diversidade são aplicadas as soluções armazenadas, elas podem ser usadas para diferentes tipos de mudanças de conceitos.

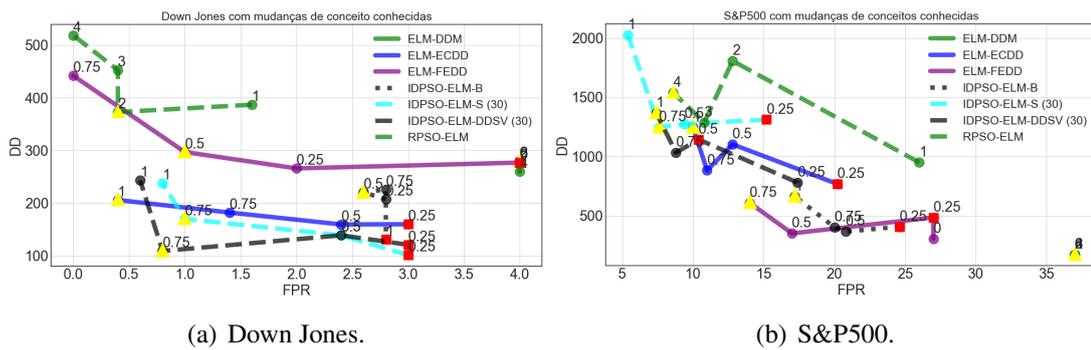
Outro ponto importante a se destacar é o desempenho do método P-IDPSO-ELM-DDSV. Mesmo não utilizando mecanismos de soluções passadas, ele conseguiu obter uma das melhores colocações do Ranking. O que implica dizer que independente do tipo de mudança de conceito, o uso de partículas para adaptação melhora significativamente o desempenho de predição em relação a não usar.

## 5.2.2 Avaliação para as Séries Reais Com Mudanças de Conceitos Conhecidas

Nesta subseção também serão discutidas as análises de detecção de mudança de conceito e previsão de séries temporais.

### 5.2.2.1 Habilidade de Detecção

As figuras 5.7(a) e 5.7(b), semelhante ao processo discutido na subseção 5.2.1, mostram as curvas formadas pelos métodos de detecção para as bases Down Jones e S&P500 com mudanças conhecidas.



**Figura 5.7:** Curva FPR x DD para as séries reais com conceitos conhecidos.

De acordo com a figura 5.7(a), pode-se observar que os métodos baseados no erro do modelo preditivo alcançaram as melhores taxas de detecção, pois suas curvas mais se aproximaram da solução ótima. Esses resultados refletem o que foi discutido na seção 5.2.1, que diz que, quando a série apresenta mudanças de conceitos muito abruptas os métodos baseados no erro apresentam melhor acurácia de detecção, pois o modelo preditivo se degrada mais rapidamente.

Pode-se observar também que, para essa série o método IDPSO-ELM-DDSV obteve a melhor curva de detecção, pois quando comparada com as curvas dos métodos IDPSO-ELM-S e ELM-ECDD a mesma apresenta uma distância relevante. A razão para esses resultados também foi discutido na seção 5.2.1, que diz que, ao monitorar mais de um modelo preditivo a detecção torna-se mais robusta a taxas de FPR sem afetar significativamente a taxa de DD.

Em relação a figura 5.7(b), pode-se observar que o FEDD, o método de detecção baseado nas características da série alcançou as menores taxas de DD, mas também as maiores taxas de FPR. Enquanto que os métodos baseados no erro, obtiveram desempenho contrário. Outra observação feita, foi que o método IDPSO-ELM-S (30), obteve as maiores taxas de DD, resultado este ligado com sua forma conservativa de declarar uma mudança de conceito. Enquanto que o método IDPSO-ELM-DDSV, conseguiu manter uma boa taxa de DD quando comparado ao ELM-ECDD.

Os resultados para essa série explica-se da mesma maneira que as constatações feitas para as séries artificiais na subseção 5.2.1. O FEDD obtém melhores taxas de detecções quando experimentado em séries em que as mudanças de conceitos são mais graduais, como essa série. Percebe-se com a figura 5.7(b) que o FEDD obteve as menores taxas de DD o que implica dizer que ele conseguiu de forma eficiente, detectar todas as mudanças de conceitos quando comparado com os demais métodos. Já os métodos baseados no erro do modelo preditivo, obtiveram as maiores taxas de DD o que implica dizer que devido as mudanças dessa série serem graduais a previsão dos modelos não se degradou, impossibilitando os testes de detecção de encontrar as mudanças de conceito.

### 5.2.2.2 *Habilidade de Previsão*

Além dos métodos de detecção experimentados na subseção passada, os métodos que utilizam mecanismos de adaptação, M-IDPSO-ELM-DDSV e P-IDPSO-ELM-DDSV, também foram experimentados nessas séries reais com mudanças conhecidas. A razão disto é também validar seus mecanismos de adaptação. O desempenho de previsão dos melhores modelos estão apresentados na tabela 5.4. Esses desempenhos foram atestados pelo teste de Nemenyi ilustrado na figura 5.8.

De acordo com a tabela 5.4 todos os métodos de detecção baseados no erro do modelo preditivo alcançaram menor erro em relação ao FEDD, com exceção apenas do método ELM-ECDD na série S&P500. O que confirma a análise feita na subseção anterior que afirma que os métodos de detecção baseados no erro não conseguiram detectar bem as mudanças de conceito porque suas previsões não se degradaram. Para atestarmos essa suposição, podemos observar com a figura 5.8 que todos os métodos baseados no erro alcançaram melhores colocações no Ranking quando comparados com o ELM-FEDD. Alguns deles como os métodos IDPSO-ELM-S, IDPSO-ELM-DDSV e IDPSO-ELM-B conseguiram atestar diferença.

Pudemos também validar o desempenho de predição dos métodos com adaptação. De acordo com a tabela 5.4 o método P-IDPSO-ELM-DDSV obteve a melhor média para a série S&P500, enquanto que o método M-IDPSO-ELM-DDSV obteve a melhor média para a série Dow Jones. Esse último resultado implica dizer que essa série possui alguns segmentos que podem ser correlacionados, ou melhor, recorrentes. Para atestar a veracidade desses desempenhos, pode ser visto no teste estatístico (figura 5.8) que os métodos com adaptação alcançaram as melhores posições do Ranking atestando diferença estatística de todos os outros métodos, com execução do M-IDPSO-ELM-DDSV para método RPSO-ELM.

### 5.2.3 Avaliação para as Séries Reais Financeiras

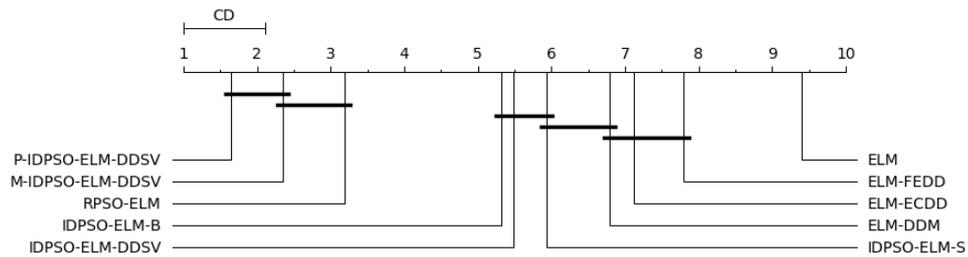
Nesta subseção, avaliamos a acurácia de previsão de todos os algoritmos nas três séries reais discutidas na seção 5.1.3.3.

#### 5.2.3.1 *Habilidade de Previsão*

Semelhante as outras análises de acurácia de previsão, na figura 5.8 apresenta-se o teste estatístico de Nemenyi para atestar o desempenho dos algoritmos e na tabela 5.3, apresentamos estatísticas descritivas (média e desvio padrão) de cada algoritmo para cada base.

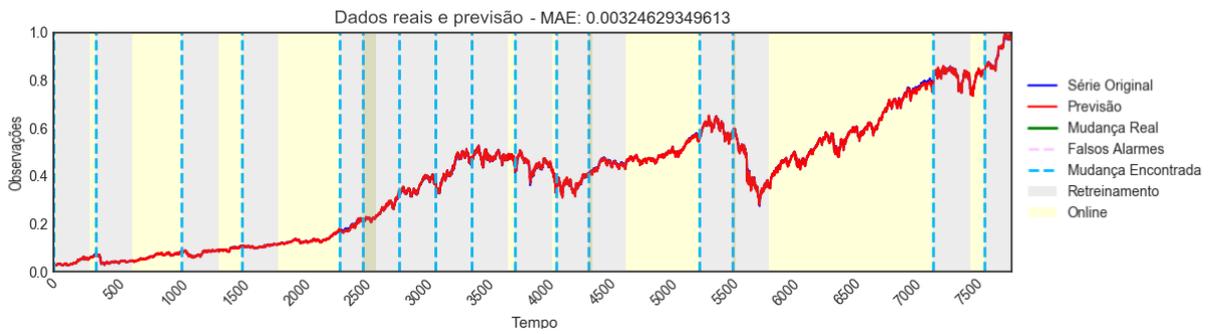
Com os métodos de adaptação, pudemos confirmar nossas hipóteses que independente dos tipos de mudanças a adaptação com técnicas de EDO melhoram o desempenho de previsão. De acordo com o teste estatístico (figura 5.8), os métodos adaptativos P-IDPSO-ELM-DDSV e M-IDPSO-ELM-DDSV, alcançaram as melhores colocações no Ranking, onde o método P-IDPSO-ELM-DDSV obteve diferença estatística dos demais, com exceção do método M-IDPSO-ELM-DDSV. A principal razão para essas colocações é que nessas séries não se sabe se

existem conceitos recorrentes, portanto o método P-IDPSO-ELM-DDSV recebe uma vantagem em relação ao método M-IDPSO-ELM-DDSV.

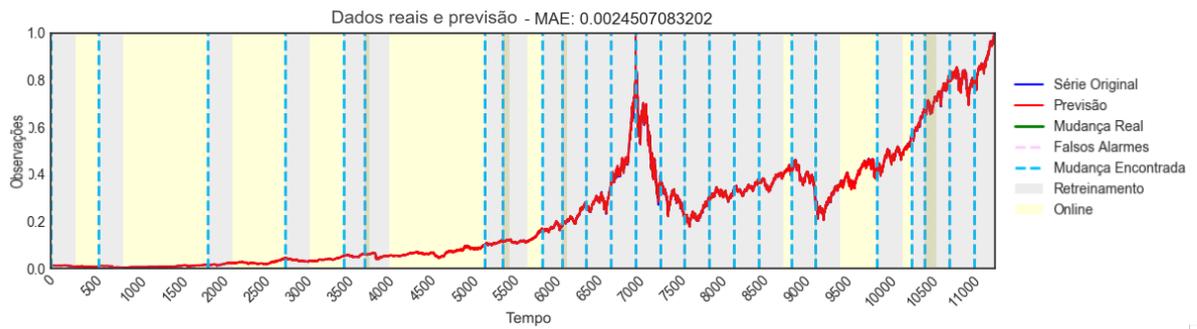


**Figura 5.8:** Teste Nemenyi para os modelos treinados e avaliados nas séries reais financeiras.

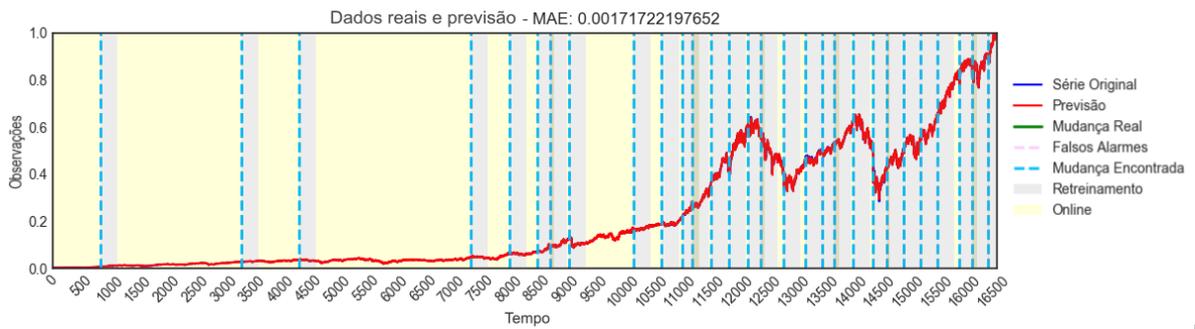
As execuções do método P-IDPSO-ELM-DDSV para essas séries podem ser vistas nas Figuras 5.9(a), 5.9(b) e 5.9(c). Os quadros em amarelo representam o período onde o modelo preditivo realizou a previsão de forma *on-line*, já os quadros em cinza representam o período onde o sistema estava armazenando informações para atualizar o melhor modelo. As linhas pontilhadas azuis representam os instantes de tempo onde o módulo de detecção informou a ocorrência de mudanças de conceito.



(a) DowJones



(b) Nasdaq.



(c) S&P500.

**Figura 5.9:** Execução do método P-IDPSO-ELM-DDSV para as séries reais financeiras.

Tabela 5.2: Erro médio absoluto para as séries artificiais.

Bases	ELM-DDM	ELM-ECDD	ELM-FEDD	IDPSO-ELM-B	IDPSO-ELM-S	IDPSO-ELM-DDSV	P-IDPSO-ELM-DDSV	M-IDPSO-ELM-DDSV	RPSO-ELM
Linear Graduais	0.0257 (0.009)	0.0306 (0.018)	0.0186 (0.004)	0.0147 (0.002)	0.0147 (0.002)	0.0147 (0.002)	0.0129 (0.002)	<b>0.0116 (0.001)</b>	0.0167 (0.004)
Linear Abruptas	0.0179 (0.005)	0.0176 (0.003)	0.0135 (0.003)	0.0134 (0.003)	0.0132 (0.002)	0.0128 (0.003)	0.0105 (0.002)	<b>0.0091 (0.001)</b>	0.0124 (0.004)
Não-linear Graduais	0.0351 (0.005)	0.0353 (0.006)	0.0332 (0.004)	0.0329 (0.004)	0.033 (0.004)	0.0327 (0.004)	0.0315 (0.004)	<b>0.0313 (0.002)</b>	0.036 (0.004)
Não-linear Abruptas	0.0483 (0.035)	0.0438 (0.031)	0.0467 (0.04)	0.0179 (0.004)	0.0173 (0.003)	0.0187 (0.005)	0.0131 (0.003)	<b>0.0118 (0.002)</b>	0.0181 (0.004)
Linear Sazonais	0.0782 (0.007)	0.0726 (0.005)	0.0659 (0.002)	0.0663 (0.002)	0.0666 (0.002)	0.0668 (0.002)	0.0647 (0.002)	<b>0.0596 (0.002)</b>	0.0834 (0.004)
Híbridas	0.0511 (0.034)	0.0644 (0.049)	0.0834 (0.069)	0.0413 (0.02)	0.0328 (0.013)	0.0317 (0.011)	0.0163 (0.003)	<b>0.0129 (0.002)</b>	0.0246 (0.004)

Tabela 5.3: Erro médio absoluto para as séries reais financeiras

Bases	ELM	ELM-DDM	ELM-ECDD	ELM-FEDD	IDPSO-ELM-B	IDPSO-ELM-S	IDPSO-ELM-DDSV	P-IDPSO-ELM-DDSV	M-IDPSO-ELM-DDSV	RPSO-ELM
Dow Jones	8.2735 (9.0484)	0.0097 (0.0021)	0.0098 (0.0023)	0.0612 (0.0437)	0.0076 (0.0017)	0.0088 (0.0025)	0.0087 (0.0022)	<b>0.0033 (0.0002)</b>	0.0043 (0.0005)	0.0047 (0.0008)
Nasdaq	10.2544 (7.4285)	0.009 (0.0027)	0.0097 (0.0046)	0.0075 (0.0022)	0.0052 (0.0013)	0.0066 (0.0015)	0.0061 (0.0011)	<b>0.0026 (0.0003)</b>	0.0032 (0.0005)	0.0037 (0.0006)
S&P500	23.3326 (23.3314)	0.0063 (0.0026)	0.0053 (0.0011)	0.0074 (0.0024)	0.0055 (0.002)	0.0044 (0.001)	0.005 (0.0014)	<b>0.0018 (0.0001)</b>	0.0022 (0.0002)	0.0023 (0.0004)

Tabela 5.4: Erro médio absoluto para as séries reais com mudanças conhecidas.

Bases	ELM-DDM	ELM-ECDD	ELM-FEDD	IDPSO-ELM-B	IDPSO-ELM-S	IDPSO-ELM-DDSV	P-IDPSO-ELM-DDSV	M-IDPSO-ELM-DDSV	RPSO-ELM
Dow Jones-drift	0.1905 (0.1572)	0.167 (0.1196)	0.2039 (0.1375)	0.0593 (0.0248)	0.0501 (0.0153)	0.0428 (0.02)	0.0301 (0.0027)	<b>0.0226 (0.0019)</b>	0.0356 (0.008)
S&P500-drift	0.031 (0.0319)	0.0326 (0.0321)	0.0343 (0.0295)	0.0298 (0.0325)	0.0316 (0.0328)	0.0305 (0.0328)	<b>0.0063 (0.0002)</b>	0.0075 (0.0006)	0.0113 (0.0005)

### 5.3 Síntese do Capítulo de Experimentos

Os experimentos reportados nesse capítulo foram realizados de modo a responder as questões de pesquisas descritas no capítulo 1. Os principais resultados mostraram que os métodos propostos são competitivos tanto em tarefas de previsão de séries temporais quanto de detecção de mudança de conceito.

Sobre a previsão, os métodos com mecanismos com adaptação à mudança de conceito obtiveram as melhores acurácias em todas as séries experimentadas. Sobre a detecção, os métodos de detecção propostos obtiveram os melhores desempenhos em séries que as mudanças de conceitos eram abruptas.

# 6

## CONCLUSÃO

O presente estudo apresentou três métodos para detecção de mudança de conceito baseado em inteligência de enxames e dois sistemas adaptativos que utilizam técnicas de otimização dinâmica. Os experimentos computacionais foram executados para verificar se mecanismos de detecção baseados em enxames podem trabalhar juntos com mecanismos de otimização dinâmica. Os experimentos foram realizados em três grupos de séries: séries com mudanças de conceitos gerados artificialmente; séries reais com mudanças de conceito conhecidas; e séries financeiras reais sem o conhecimento de mudanças.

Os principais resultados mostraram que os métodos propostos são competitivos em tarefas de previsão de séries temporais e detecção de mudança de conceito. Com estes resultados, respondemos as perguntas de pesquisa desse trabalho da seguinte forma:

Com relação as perguntas relacionadas a detecção de mudanças de conceito por enxames (bloco 1), obtivemos as seguintes constatações: (i) monitorar mais de um modelo melhora a capacidade de detecção? Sim, monitorar mais características de um enxame melhora a acurácia de detecção, diminuindo significativamente detecções falso positivas sem afetar os atrasos de detecção; (ii) monitorar mais características de um enxame beneficia a detecção? Não, o método baseado no comportamento do enxame, como IDPSO-ELM-B, não se mostrou competitivo em relação aos métodos da literatura; (iii) Até que ponto o monitoramento dessas soluções beneficiam a detecção? Constatou-se que algoritmos baseados no erro do modelo preditivo, possuem melhores detecções em séries em que as mudanças de conceitos são mais abruptas, pois o erro do modelo é degradado mais rapidamente.

Com relação as perguntas de pesquisa sobre o uso de partículas para adaptação a mudança de conceito (bloco 2), obtive-se as seguintes constatações: (i) O PSO pode ser usado para obter uma melhora na capacidade de adaptação ao novo conceito em previsão de séries temporais? Sim, utilizar as partículas treinadas previamente, melhoram na adaptação de um novo conceito, uma vez que o desempenho de predição melhora significativamente; (ii) Quais tipos de mudança de conceito se beneficiam? Constatou-se que independente do tipo de mudança de conceito, o método baseado em partículas para adaptação (P-IDPSO-NN-DDSV) alcançou as melhores colocações no Ranking de Friedman em relação ao desempenho de previsão. (iii) Até que ponto elas podem se beneficiar? Constatou-se que elas se beneficiam em conceitos não recorrentes, ou

seja, em mudanças ainda não vistas anteriormente.

Por fim, em relação as perguntas referentes ao uso de enxames passados (bloco 3), obtivemos as seguintes constatações: (i) o uso de soluções de enxames passados pode levar a uma aceleração na adaptação de conceitos recorrentes em previsão de séries temporais? Sim, comprovou-se que o uso de soluções passadas por meio do método M-IDPSO-NN-DDSV acelera o processo de adaptação em conceitos recorrentes, obtendo melhores colocações no Ranking de Friedman e apresentando diferença estatística dos demais; (ii) Em que medida isso pode ajudar na adaptação? Constatou-se que usar enxames passados melhora significativamente o desempenho de previsão, quando comparado com métodos sem adaptação; (iii) Outros tipos de mudanças de conceito podem se beneficiar disso? Sim, constatou-se que se estratégias de diversidade são aplicadas nas soluções armazenadas, elas podem ser usadas para diferentes tipos de mudanças de conceitos.

### 6.1 Limitações do Método Proposto

Apesar dos resultados terem mostrado desempenho superior dos métodos propostos, há algumas limitações que diminuem a acurácia de detecção e acurácia de previsão. Essas limitações estão listadas a seguir:

1. Com relação a detecção de mudança de conceito, os algoritmos propostos não se mostraram tão eficientes em séries em que as mudanças de conceitos são graduais. O principal motivo para essa limitação é que as abordagens baseiam-se no erro do modelo preditivo, então se o modelo não se degrada de forma intensa, o módulo de detecção não identifica essas mudanças. Esse por sua vez é um grande problema para a previsão de séries temporais, pois não recomenda-se esperar a degradação do modelo preditivo, uma vez que as previsões são usadas para tomada de decisões.
2. Com relação a previsão de séries temporais, os algoritmos propostos dependem de uma grande quantidade de dados para poderem ser atualizados. Portanto, quando ocorre uma mudança de conceito os métodos necessitam coletar dados referentes ao novo conceito, enquanto não possuem dados suficientes as previsões são geradas por um modelo alternativo. Este por sua vez não garante que as previsões serão as melhores possíveis, pois o modelo não foi treinado com dados do novo conceito. Uma das formas de garantir melhores previsões seria desenvolver uma forma de armazenar dados *on-line* que garanta que o modelo tenha muitos dados para treinamento.

### 6.2 Trabalhos Futuros

Como trabalhos futuros, alguns aspectos ainda podem ser investigados, como: (i) experimentar novas abordagens de gerenciamento de dados de forma *on-line*, de modo a acelerar

o processo de adaptação do modelo. Uma das possibilidades para isto é combinar o sistema de janela deslizante com a coleta de dados baseados nos níveis de detecção; (ii) investigar como acelerar o processo de treinamento do modelo regressivo. Em um retreinamento, em vez de reinicializar todas as soluções do enxame, reinicializar apenas as piores soluções, desse modo acelerando o processo de busca ao aproveitar as melhores soluções; (iii) investigar outras topologias de PSO, de modo evitar que as partículas se dispersem tanto e afetem a detecção de mudança de conceito; (iv) investigar o resíduo do modelo preditivo, esse resíduo pode conter informações valiosas para a detecção da mudança de conceito. Além disso se o resíduo for ajustado, pode melhorar a acurácia de previsão do modelo preditivo; (v) investigar como usar os modelos criados pelo PSO para criar um comitê de modelos de previsão, uma vez que, após a mudança, os modelos apresentam diferentes capacidades preditivas, tornando viável uma possível combinação.

## Referências

- ADAMS, R. P.; MACKAY, D. J. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.
- ADHIKARI, R. A neural network based linear ensemble framework for time series forecasting. *Neurocomputing*, Elsevier, v. 157, p. 231–242, 2015.
- ANDRZEJAK, R. G. et al. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, v. 64, n. 6, p. 061907, 2001.
- BABU, C. N.; REDDY, B. E. A moving-average filter based hybrid arima-ann model for forecasting time series data. *Applied Soft Computing*, Elsevier, v. 23, p. 27–38, 2014.
- BARROS, A. C. d. A. Otimização de redes neurais para a previsão de séries temporais. Universidade de Pernambuco, 2005.
- BEIRANVAND, V.; BAKAR, A. A.; OTHMAN, Z. A comparative survey of three ai techniques (nn, pso, and ga) in financial domain. In: IEEE. *Computing and Convergence Technology (ICCCT), 2012 7th International Conference on*. [S.l.], 2012. p. 332–337.
- BORACCHI, G.; ROVERI, M. Exploiting self-similarity for change detection. In: IEEE. *Neural Networks (IJCNN), 2014 International Joint Conference on*. [S.l.], 2014. p. 3339–3346.
- BOX, G. E. et al. *Time series analysis: forecasting and control*. [S.l.]: John Wiley & Sons, 2015.
- BRITO, R. F. de; OLIVEIRA, A. L. Sliding window-based analysis of multiple foreign exchange trading systems by using soft computing techniques. In: IEEE. *International Joint Conference on Neural Networks*. [S.l.], 2014. p. 4251–4258.
- CAVALCANTE, R. C. An adaptive learning system for time series forecasting in the presence of concept drift. Universidade Federal de Pernambuco, 2017.
- CAVALCANTE, R. C. et al. Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, Elsevier, v. 55, p. 194–211, 2016.
- CAVALCANTE, R. C.; MINKU, L. L.; OLIVEIRA, A. L. Fedd: Feature extraction for explicit concept drift detection in time series. In: IEEE. *Neural Networks (IJCNN), 2016 International Joint Conference on*. [S.l.], 2016. p. 740–747.
- CAVALCANTE, R. C.; OLIVEIRA, A. L. An autonomous trader agent for the stock market based on online sequential extreme learning machine ensemble. In: IEEE. *Int. Joint Conf. on Neural Networks*. [S.l.], 2014. p. 1424–1431.
- CAVALCANTE, R. C.; OLIVEIRA, A. L. An approach to handle concept drift in financial time series based on extreme learning machines and explicit drift detection. In: IEEE. *Neural Networks (IJCNN), 2015 International Joint Conference on*. [S.l.], 2015. p. 1–8.
- COWPERTWAIT, P. S.; METCALFE, A. V. *Introductory time series with R*. [S.l.]: Springer Science & Business Media, 2009.

- DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, v. 7, p. 1–30, 2006.
- DING, Y. et al. Global nonlinear kernel prediction for large data set with a particle swarm-optimized interval support vector regression. *IEEE transactions on neural networks and learning systems*, IEEE, v. 26, n. 10, p. 2521–2534, 2015.
- DITZLER, G. et al. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, IEEE, v. 10, n. 4, p. 12–25, 2015.
- FIRMINO, P. R. A.; NETO, P. S. de M.; FERREIRA, T. A. Error modeling approach to improve time series forecasters. *Neurocomputing*, Elsevier, v. 153, p. 242–254, 2015.
- FOULADGAR, N.; LOTFI, S. A novel approach for optimization in dynamic environments based on modified cuckoo search algorithm. *Soft Computing*, Springer, v. 20, n. 7, p. 2889–2903, 2016.
- FRIEDMAN, M. A comparison of alternative tests of significance for the problem of m rankings. In: *The Annals of Mathematical Statistics*. [S.l.: s.n.], 1940. v. 11, p. 86–92.
- GAMA, J. et al. Learning with drift detection. In: SPRINGER. *Brazilian Symposium on Artificial Intelligence*. [S.l.], 2004. p. 286–295.
- GAMA, J. et al. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, ACM, v. 46, n. 4, p. 44, 2014.
- GUO, L.; LIEW, A. W.-C. Online-offline extreme learning machine with concept drift tracking for time series data. In: IEEE. *Digital Image Computing: Techniques and Applications (DICTA), 2016 International Conference on*. [S.l.], 2016. p. 1–6.
- HAMILTON, J. D. *Time series analysis*. [S.l.]: Princeton university press Princeton, 1994. v. 2.
- HAN, F.; YAO, H.-F.; LING, Q.-H. An improved evolutionary extreme learning machine based on particle swarm optimization. *Neurocomputing*, Elsevier, v. 116, p. 87–93, 2013.
- HAN, H.-G. et al. An adaptive-pso-based self-organizing rbf neural network. *IEEE transactions on neural networks and learning systems*, IEEE, 2016.
- HECHT-NIELSEN, R. et al. Theory of the backpropagation neural network. *Neural Networks*, v. 1, n. Supplement-1, p. 445–448, 1988.
- HENDTLASS, T.; MOSER, I.; RANDALL, M. Dynamic problems and nature inspired meta-heuristics. In: *Biologically-Inspired Optimisation Methods*. [S.l.]: Springer, 2009. p. 79–109.
- HOSSEINI, M. J.; AHMADI, Z.; BEIGY, H. Using a classifier pool in accuracy based tracking of recurring concepts in data stream classification. *Evolving Systems*, Springer, v. 4, n. 1, p. 43–60, 2013.
- HU, X.; EBERHART, R. C. Adaptive particle swarm optimization: detection and response to dynamic systems. In: IEEE. *Evol. Computation, 2002. CEC'02. Proc. of the 2002 Congress on*. [S.l.], 2002. v. 2, p. 1666–1670.

- HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: theory and applications. *Neurocomputing*, Elsevier, v. 70, n. 1, p. 489–501, 2006.
- JANSON, S.; MIDDENDORF, M. A hierarchical particle swarm optimizer for noisy and dynamic environments. *Genetic Programming and Evolvable Machines*, Springer, v. 7, n. 4, p. 329–354, 2006.
- KENNEDY, J. Particle swarm optimization. In: *Encyclopedia of machine learning*. [S.l.]: Springer, 2011. p. 760–766.
- KUNCHEVA, L. I.; GUNN, I. A. A concept-drift perspective on prototype selection and generation. In: IEEE. *Neural Networks (IJCNN), 2016 International Joint Conference on*. [S.l.], 2016. p. 16–23.
- LAMOUREUX, C. G.; LASTRAPES, W. D. Heteroskedasticity in stock return data: Volume versus garch effects. *The journal of finance*, Wiley Online Library, v. 45, n. 1, p. 221–229, 1990.
- LAVIELLE, M.; TEYSSIERE, G. Detection of multiple change-points in multivariate time series. *Lithuanian Mathematical Journal*, Springer, v. 46, n. 3, p. 287–306, 2006.
- LEE, L.-W. et al. Handling forecasting problems based on two-factors high-order fuzzy time series. *IEEE Transactions on fuzzy Systems*, IEEE, v. 14, n. 3, p. 468–477, 2006.
- MAVROVOUNIOTIS, M.; LI, C.; YANG, S. A survey of swarm intelligence for dynamic optimization: algorithms and applications. *Swarm and Evolutionary Computation*, Elsevier, v. 33, p. 1–17, 2017.
- MINKU, L. L.; YAO, X. Ddd: A new ensemble approach for dealing with concept drift. *IEEE transactions on knowledge and data engineering*, IEEE, v. 24, n. 4, p. 619–633, 2012.
- MONTANA, D. J.; DAVIS, L. Training feedforward neural networks using genetic algorithms. In: *IJCAI*. [S.l.: s.n.], 1989. v. 89, p. 762–767.
- MORETTIN, P. A. *Econometria financeira: um curso em séries temporais financeiras*. [S.l.]: Edgard Blücher, 2008.
- MORETTIN, P. A.; TOLOI, C. *Análise de séries temporais*. [S.l.]: Blucher, 2006.
- MUDA, L.; BEGAM, M.; ELAMVAZUTHI, I. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*, 2010.
- NASIRI, B.; MEYBODI, M.; EBADZADEH, M. History-driven particle swarm optimization in dynamic and uncertain environments. *Neurocomputing*, Elsevier, v. 172, p. 356–370, 2016.
- NGUYEN, T. T.; YANG, S.; BRANKE, J. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, Elsevier, v. 6, p. 1–24, 2012.
- OLIVEIRA, A. L.; MEIRA, S. R. Detecting novelties in time series through neural networks forecasting with robust confidence intervals. *Neurocomputing*, Elsevier, v. 70, n. 1, p. 79–92, 2006.

- OLIVEIRA, G. H. F. M. et al. Time series forecasting in the presence of concept drift: A pso-based approach. In: IEEE. *Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on*. [S.l.], 2017. p. 1–8.
- OLIVEIRA, J. F. de; LUDERMIR, T. B. A hybrid evolutionary decomposition system for time series forecasting. *Neurocomputing*, Elsevier, v. 180, p. 27–34, 2016.
- PALIT, A. K.; POPOVIC, D. *Computational intelligence in time series forecasting*. 2006.
- RAKITIANSKAIA, A.; ENGELBRECHT, A. P. Training neural networks with pso in dynamic environments. In: IEEE. *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. [S.l.], 2009. p. 667–673.
- RAKITIANSKAIA, A. S.; ENGELBRECHT, A. P. Training feedforward neural networks with dynamic particle swarm optimisation. *Swarm Intelligence*, Springer, v. 6, n. 3, p. 233–270, 2012.
- RICHTER, H. Detecting change in dynamic fitness landscapes. In: IEEE. *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. [S.l.], 2009. p. 1613–1620.
- ROSS, G. J. et al. Exponentially weighted moving average charts for detecting concept drift. *Pattern Recognition Letters*, Elsevier, v. 33, n. 2, p. 191–198, 2012.
- SONG, Y. et al. A fuzzy kernel c-means clustering model for handling concept drift in regression. In: IEEE. *Fuzzy Systems (FUZZ-IEEE), 2017 IEEE International Conference on*. [S.l.], 2017. p. 1–6.
- SOUZA, E. et al. An unsupervised particle swarm optimization approach for opinion clustering. In: IEEE. *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*. [S.l.], 2016. p. 307–312.
- VELLASQUES, E.; SABOURIN, R.; GRANGER, E. A high throughput system for intelligent watermarking of bi-tonal images. *Applied Soft Computing*, Elsevier, v. 11, n. 8, p. 5215–5229, 2011.
- VELLASQUES, E.; SABOURIN, R.; GRANGER, E. Fast intelligent watermarking of heterogeneous image streams through mixture modeling of pso populations. *Applied Soft Computing*, Elsevier, v. 13, n. 6, p. 3130–3148, 2013.
- WAGNER, N. et al. Time series forecasting for dynamic environments: the dyfor genetic program model. *IEEE transactions on evolutionary computation*, IEEE, v. 11, n. 4, p. 433–452, 2007.
- YAMAUCHI, K. Incremental learning and model selection under virtual concept drifting environments. In: IEEE. *Neural Networks (IJCNN), The 2010 International Joint Conference on*. [S.l.], 2010. p. 1–8.
- YEH, W.-C. New parameter-free simplified swarm optimization for artificial neural network training and its application in the prediction of time series. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, v. 24, n. 4, p. 661–665, 2013.
- ZHANG, G. P. An investigation of neural networks for linear time-series forecasting. *Computers & Operations Research*, Elsevier, v. 28, n. 12, p. 1183–1202, 2001.

---

ZHANG, G. P. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, Elsevier, v. 50, p. 159–175, 2003.

ZHANG, Y.; XIONG, X.; ZHANG, Q. An improved self-adaptive pso algorithm with detection function for multimodal function optimization problems. *Mathematical Problems in Engineering*, Hindawi Publishing Corporation, v. 2013, 2013.

ZLIOBAITE, I. et al. Active learning with drifting streaming data. *IEEE transactions on neural networks and learning systems*, IEEE, v. 25, n. 1, p. 27–39, 2014.