Pós-Graduação em Ciência da Computação

AVALIAÇÃO CRITERIOSA DOS ALGORITMOS DE DETECÇÃO DE CONCEPT DRIFTS

Por

SILAS GARRIDO TEIXEIRA DE CARVALHO SANTOS

Dissertação de Mestrado

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

RECIFE/2015

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

SILAS GARRIDO TEIXEIRA DE CARVALHO SANTOS

AVALIAÇÃO CRITERIOSA DOS ALGORITMOS DE DETECÇÃO DE CONCEPT DRIFTS

ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADOR: *ROBERTO SOUTO MAIOR DE BARROS*
COORIENTADOR: PAULO MAURÍCIO GONÇALVES JÚNIOR

RECIFE, FEVEREIRO/2015

Catalogação na fonte
Bibliotecário Jefferson Luiz Alves Nazareno CRB4-1758

S237a Santos, Silas Garrido Teixeira de Carvalho.
Avaliação criteriosa dos algoritmos de detecção de concept drifts. / Silas Garrido Teixeira de Carvalho Santos. – Recife: O Autor, 2015.
96 f.: fig.; tab.

Orientador: Roberto Souto Maior de Barros.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, 2015.
Inclui referências e anexo.

1. Inteligência computacional. 2. Fluxo de dados (Computadores). 3. Algoritmos genéticos I. Barros, Roberto Souto Maior de. (Orientador). II. Titulo.

004.35 CDD (22. ed.) UFPE-MEI 2015-135

Dissertação de Mestrado apresentada por **Silas Garrido Teixeira de Carvalho Santos** à Pós Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**Avaliação Criteriosa dos Algoritmos de Detecção de Concept Drifts**", orientada pelo Prof. **Roberto Souto Maior de Barros** e aprovada pela Banca Examinadora formada pelos professores:

_____

Prof. Frederico Luiz Gonçalves de Freitas
Centro de Informática/UFPE

_____

Prof. Wilson Rosa de Oliveira Júnior
Departamento de Estatística e Informática/UFRPE

_____

Prof. Roberto Souto Maior de Barros
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 27 de fevereiro de 2015.

_____

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

Agradecimentos

Agradeço ao meu orientador, professor Roberto Souto Maior de Barros, por todo o apoio e dedicação durante este período. Com sua vasta experiência soube conduzir as mais diversas situações da melhor maneira possível. Sua ajuda foi de fundamental importância para o meu desenvolvimento.

Agradeço ao meu coorientador, professor Paulo Maurício Gonçalves Júnior, pelo suporte oferecido e pelas ricas discussões ao longo do processo. Suas contribuições foram essenciais para os resultados alcançados até aqui.

Agradeço a todos os professores/funcionários do Centro de Informática (CIn-UFPE) que participaram direta ou indiretamente da minha formação.

Por fim, mas não menos importante, agradeço a todos os parentes e amigos, em especial aos meus pais, Izabel e João; à minha irmã, Isabela; e aos meus tios, Ana e Joabe.

"Bem-aventurado o homem que acha sabedoria, e o homem que adquire conhecimento."
(Bíblia Sagrada, Provérbios 3:13)

Resumo

A extração de conhecimento em ambientes com fluxo contínuo de dados é uma atividade que vem crescendo progressivamente. Diversas são as situações que necessitam desse mecanismo, como o monitoramento do histórico de compras de clientes; a detecção de presença por meio de sensores; ou o monitoramento da temperatura da água. Desta maneira, os algoritmos utilizados para esse fim devem ser atualizados constantemente, buscando adaptar-se às novas instâncias e levando em consideração as restrições computacionais. Quando se trabalha em ambientes com fluxo contínuo de dados, em geral não é recomendável supor que sua distribuição permanecerá estacionária. Diversas mudanças podem ocorrer ao longo do tempo, desencadeando uma situação geralmente conhecida como mudança de conceito (concept drift). Neste trabalho foi realizado um estudo comparativo entre alguns dos principais métodos de detecção de mudanças: ADWIN, DDM, DOF, ECDD, EDDM, PL e STEPD. Para execução dos experimentos foram utilizadas bases artificiais – simulando mudanças abruptas, graduais rápidas, e graduais lentas – e também bases com problemas reais. Os resultados foram analisados baseando-se na precisão, tempo de execução, uso de memória, tempo médio de detecção das mudanças, e quantidade de falsos positivos e negativos. Já os parâmetros dos métodos foram definidos utilizando uma versão adaptada de um algoritmo genético. De acordo com os resultados do teste de Friedman juntamente com Nemenyi, em termos de precisão, DDM se mostrou o método mais eficiente com as bases utilizadas, sendo estatisticamente superior ao DOF e ECDD. Já EDDM foi o método mais rápido e também o mais econômico no uso da memória, sendo superior ao DOF, ECDD, PL e STEPD, em ambos os casos. Conclui-se então que métodos mais sensíveis às detecções de mudanças, e consequentemente mais propensos a alarmes falsos, obtêm melhores resultados quando comparados a métodos menos sensíveis e menos suscetíveis a alarmes falsos.

Palavras-chave: Ambiente com fluxo contínuo de dados (Data stream). Mudanças de conceito (Concept drift). Métodos de detecção de mudanças. Algoritmo genético. Avaliação criteriosa.

Abstract

Knowledge extraction from data streams is an activity that has been progressively receiving an increased demand. Examples of such applications include monitoring purchase history of customers, movement data from sensors, or water temperatures. Thus, algorithms used for this purpose must be constantly updated, trying to adapt to new instances and taking into account computational constraints. When working in environments with a continuous flow of data, there is no guarantee that the distribution of the data will remain stationary. On the contrary, several changes may occur over time, triggering situations commonly known as concept drift. In this work we present a comparative study of some of the main drift detection methods: ADWIN, DDM, DOF, ECDD, EDDM, PL and STEPD. For the execution of the experiments, artificial datasets were used – simulating abrupt, fast gradual, and slow gradual changes – and also datasets with real problems. The results were analyzed based on the accuracy, runtime, memory usage, average time to change detection, and number of false positives and negatives. The parameters of methods were defined using an adapted version of a genetic algorithm. According to the Friedman test with Nemenyi results, in terms of accuracy, DDM was the most efficient method with the datasets used, and statistically superior to DOF and ECDD. EDDM was the fastest method and also the most economical in memory usage, being statistically superior to DOF, ECDD, PL and STEPD, in both cases. It was concluded that more sensitive change detection methods, and therefore more prone to false alarms, achieve better results when compared to less sensitive and less susceptible to false alarms methods.

Keywords: Data streams. Concept drift. Drift detection methods. Genetic algorithm. Rigorous evaluation.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ADOB	Adaptable Diversity-based Online Boosting – Online Boosting baseado em Diversidade Adaptável (em tradução livre)
AG	Algoritmo Genético
AUE	Accuracy Updated Ensemble – Comitê com Atualizações baseadas na Precisão
ADWIN2	Adaptive Windowing 2 – Janelas Adaptativas 2
DDD	Diversity for Dealing with Drifts – Diversidade para Lidar com Mudanças de Conceito
DDM	Drift Detection Method – Método de Detecção de Mudanças de Conceito
DOF	Degree of Drift – Grau de Mudança de Conceito
DWM	Dynamic Weighted Majority – Voto Majoritário Ponderado Dinâmico
ECDD	EWMA for Concept Drift Detection – EWMA para Detecção de Mudanças de Conceito
EDDM	Early Drift Detection Method – Método de Detecção antecipada de Mudanças de Conceito
EWMA	Exponentially Weighted Moving Average – Média Móvel Ponderada Exponencialmente
FCWM	Fixed Cumulative Windows Model – Modelo de Janelas Fixas e Acumulativas
HEOM	Heterogeneous Euclidian/Overlap Metric – Métrica Heterogênea Euclidiana
MOA	Massive Online Analysis – Análise Massiva em Tempo Real
NB	Naive Bayes
PL	Paired Learners – Classificadores Emparelhados
RCD	Recurring Concept Drifts – Mudanças de Conceito Recorrentes
STEPD	Statistical Test of Equal Proportions – Teste Estatístico com Igualdade de Proporções

SVM	Support Vector Machine – Máquina de Vetor de Suporte
WMA	Weighted Majority Algorithm – Algoritmo baseado no Voto Majoritário Ponderado

Sumário

1 Introdução

Extrair conhecimento de dados é um tema abordado em diversos campos de pesquisa, a exemplo de mineração de dados, aprendizagem de máquina e reconhecimento de padrões. Tradicionalmente, essa extração ou aprendizagem é realizada em ambientes estáticos, onde o conjunto de dados estará disponível para o classificador treinar quantas vezes for necessário. Uma outra característica é que o alvo de aprendizagem será fixo. Para realização dessas tarefas vários métodos foram propostos (BURGES, 1998) (HECKERMAN, 1997) e, atualmente, muitos desses problemas são resolvidos de maneira satisfatória.

Uma atividade que vem recebendo cada vez mais demandas, entretanto, é a extração do conhecimento em ambientes com fluxo contínuo de dados. Exemplos de aplicações incluem monitoramento do histórico de compras de clientes, detecção de presença por meio de sensores e o monitoramento da temperatura da água. Comparando com a aprendizagem tradicional, o processamento dos dados em ambientes com fluxo contínuo normalmente impõe restrições sobre o uso da memória, tempo de execução e quantidade de treinamento por instância.

Outra dificuldade comumente encontrada em ambientes com fluxo contínuo de dados é a mudança na distribuição dos dados, geralmente após um período mínimo de estabilidade (GAMA, 2010). Assim, ficamos diante de um problema conhecido como *mudança de conceito* (*concept drift*). A velocidade na qual a mudança ocorre pode ser classificada como abrupta, quando ocorre uma transição do conceito antigo para o novo de forma rápida, ou gradual, quando a transição é lenta.

Para fins de exemplificação, supor que o histórico de um cliente de um determinado cartão de crédito seja armazenado. Esse cliente, durante vários anos, na maior parte do tempo, utilizou esse cartão de crédito para comprar alimentos, sempre numa mesma cidade. A partir das informações extraídas, um perfil do comportamento desse cliente é criado. Entretanto, não é razoável considerar que esse perfil jamais será modificado.

Diante deste cenário é possível considerar dois tipos de situações: 1) mesmo depois de vários anos comprando alimentos numa mesma cidade, abruptamente, esse cliente compra vários eletrônicos em outro pais. Como será mostrado mais adiante, cabe aos métodos de detecção identificar se essa mudança é decorrente de uma fraude ou se o cliente está viajando e aproveitando as ofertas dos eletrônicos. 2) Esse mesmo cliente, gradativamente, vai alterando o seu tipo de alimentação: aos poucos ele vai substituindo comidas gordurosas e industrializadas por alimentos naturais e frescos. Alguns anos depois, sua alimentação está completamente renovada. Assim, novamente cabe aos métodos de detecção identificar que houve uma mudança no comportamento e, com isso, alertar para

que ofertas de produtos não mais consumidos parem de ser enviadas, por exemplo.

Existem diversas abordagens e algoritmos propostos para a detecção de mudanças de conceito. Cada uma das abordagens e algoritmos tem suas próprias características e, em geral, vários parâmetros diferentes que visam aumentar sua eficiência, dependendo das características das bases de dados, bem como do tipo de mudança de conceito que se deseja otimizar. Esses detectores geralmente utilizam as respostas de um ou mais classificadores para identificar a mudança de conceito, indicando uma mudança quando a quantidade de erros de classificação supera os acertos a partir de uma determinada métrica. Algoritmos que adotam esta abordagem incluem: DDM (GAMA et al., 2004), EDDM (BAENA-GARCíA et al., 2006), ADWIN (BIFET; GAVALDA, 2007), ECDD (ROSS et al., 2012), PL (BACH; MALOOF, 2008), FCWM (SEBASTIAO et al., 2010), PHT (PAGE, 1954), STEPD (NISHIDA; YAMAUCHI, 2007) e DOF (SOBHANI; BEIGY, 2011). Adicionalmente, (SEBASTIAO; GAMA, 2009) fazem um estudo comparativo entre alguns desses métodos e destacam vantagens e desvantagens em diferentes situações.

Outra forma bastante comum de lidar com mudanças de conceito é a utilização de comitês de classificadores (*Ensemble Classifiers*). Neste tipo de estratégia, vários classificadores são treinados com um algoritmo base e suas predições individuais são combinadas para formar a predição global. Os métodos para atualização do conjunto (criação e exclusão de classificadores) e sua forma de integrar as predições dos membros individuais para uma predição global variam de acordo com cada algoritmo. Exemplos de comitês de classificadores são DWM (KOLTER; MALOOF, 2007), AUE (BRZEZINSKI; STEFANOWSKI, 2011), WMA (BLUM, 1997), DDD (MINKU; YAO, 2012), ADOB (SANTOS et al., 2014), dentre outros.

Outra abordagem mais específica visa ainda aumentar a eficiência na detecção de mudanças para conceitos que já ocorreram anteriormente (*Recurring Concept Drifts*). Para lidar com estas situações, a abordagem mais comum é armazenar informações ou calcular estatísticas sobre os dados e regularmente compará-las com os dados mais atuais. Caso ocorra uma mudança de conceito, dados atuais são comparados com dados mais antigos na busca de alguma medida de semelhança entre eles. Se os dados possuírem características semelhantes, classificadores ou estatísticas armazenadas podem ser reutilizadas, evitando calculá-las novamente. Exemplos incluem Flora3 (WIDMER; KUBAT, 1996), Splice2 (HARRIES; SAMMUT; HORN, 1998), EB (KATAKIS; TSOUMAKAS; VLAHAVAS, 2010), RCD (GONCALVES JR.; BARROS, 2012; GONCALVES JR.; BARROS, 2013), dentre outras.

1.1 Objetivo

Este estudo busca identificar mais claramente as principais virtudes e fraquezas de alguns dos principais métodos de detecção de mudanças que utilizam os acertos e os erros dos classificadores para identificá-las. Além disso, deseja-se entender melhor como os seus parâmetros e as características dos dados influenciam suas performances. Dessa forma, esses avanços possibilitarão um melhor aproveitamento das tecnologias disponíveis nos diferentes tipos de situações.

1.2 Metodologia

Para realização deste trabalho foram selecionados sete métodos de detecção de mudanças que foram executados em nove bases artificiais e três reais. Para cada uma das bases artificiais foram geradas três versões, contendo mudanças abruptas, graduais rápidas, e graduais lentas, respectivamente. Os resultados foram avaliados em termos de precisão, tempo de execução, consumo de memória, posição média da detecção das mudanças, falsos positivos e falsos negativos.

Para seleção dos parâmetros foi realizada uma adaptação do algoritmo genético – técnica de busca utilizada para resolver problemas de otimização em geral – voltada para os métodos de detecção de mudanças. Mais especificamente um conjunto de hipóteses, que correspondem a cada um dos métodos, são gerados com parâmetros definidos aleatoriamente. Por meio da simulação de fenômenos conhecidos como crossing-over e mutação, as hipóteses evoluem no decorrer das gerações em busca de indivíduos mais aptos (métodos mais precisos).

1.3 Organização

Este trabalho está organizado da seguinte maneira:

- Capítulo 2 faz uma revisão dos métodos utilizados bem como das bases reais e artificiais;
- Capítulo 3 detalha como foi realizada a adaptação do algoritmo genético para otimização dos parâmetros dos métodos detectores de mudanças;
- Capítulo 4 faz uma análise empírica dos métodos baseando-se nos resultados dos experimentos;
- Capítulo 5 compara os resultados de forma empírica e estatística;
- Capítulo 6 apresenta as conclusões.

2 Revisão da Literatura

Neste capítulo fazemos uma revisão dos métodos utilizados nos experimentos, incluindo as formas como eles detectam uma mudança de conceito. As características das bases de dados também são descritas, destacando também a maneira como as mudanças são simuladas nas bases artificiais.

2.1 Mudança de Conceito

Mudança de conceito (*concept drift*) (GAMA, 2010; GONCALVES JR. et al., 2014; SANTOS et al., 2014) é uma situação comum em ambientes com um contínuo fluxo de dados e refere-se à mudança das características (padrão) dos dados. Uma mudança de conceito pode ocorrer "quando um conjunto de exemplos possui rótulos de classe legítimos em um momento e rótulos diferentes em outros momentos" (KOLTER; MALOOF, 2007). Essa situação é chamada de mudança de conceito real.

Uma mudança de conceito virtual ocorre quando "o conceito alvo permanece o mesmo mas a distribuição de dados muda" (DELANY et al., 2005). Mudanças de conceito também podem ocorrer de forma abrupta ou gradual, dependendo da velocidade com que elas acontecem. Uma dificuldade comum relacionada às mudanças de conceito é que os algoritmos devem conseguir diferenciar entre ruídos e conceitos que estão mudando.

2.2 Métodos de Detecção de Mudanças de Conceito

Um detector de mudança de conceitos é um método e/ou algoritmo cuja função é detectar mudanças na distribuição dos dados que estão sendo processados.

Cada um dos métodos detectores utilizado neste trabalho e apresentado em seguida tem como característica executar em paralelo com um classificador base. Este, para cada instância recebida, gera a previsão de uma classe e, posteriormente, compara sua resposta com a resposta correta. Assim, é possível saber se o classificador base acertou ou errou cada previsão.

Baseando-se nessas informações, os métodos detectores podem indicar se houve uma mudança de conceito ou não, geralmente observando a quantidade de erros sequencialmente cometidos pelo classificador base.

Para os nossos experimentos, utilizamos o Naive Bayes (NB) como classificador base, por conta da sua velocidade, simplicidade, disponibilidade, e pelo seu amplo uso em ambientes com fluxo contínuo de dados.

2.2.1 ADWIN

Para detectar uma mudança de conceito, Adaptive Windowing (ADWIN) (BIFET; GAVALDA, 2007) utiliza uma janela (W) que tem seu tamanho ajustado automaticamente. Quanto mais tempo os dados permanecerem com a mesma distribuição de probabilidade, maior será o tamanho de W. Por outro lado, quando mudanças ocorrerem, W será reduzida. Com essa abordagem, o objetivo do ADWIN é manter em W somente instâncias com uma mesma distribuição.

Dado um conjunto de instâncias, para que o tamanho de W seja reduzido, a diferença absoluta das médias $\hat{\mu}_{W_0}$ e $\hat{\mu}_{W_1}$ juntamente com as duas sub-janelas W_0 e W_1 (pertencentes a W), deve ser maior que o valor de ϵ_{cut}. W_0 e W_1 representam, respectivamente, sub-janelas com instâncias mais antigas e mais recentes e $\hat{\mu}_{W_0}$ e $\hat{\mu}_{W_1}$ representam a média dos erros dessas sub-janelas. Os valores de ϵ_{cut}, m e $\delta^{'}$ são determinados por:

$$\epsilon_{cut} = \sqrt{\frac{2}{m}.\sigma_W^2.\log\left(\frac{2}{\delta'}\right)} + \frac{2}{3m}.\log\left(\frac{2}{\delta'}\right)$$

$$m = \frac{1}{1/n_0 + 1/n_1}$$

$$\delta^{'} = \delta/\log(n)$$

onde $\delta \in [0, 1]$ representa o nível de confiança; n_0 e n_1 são os tamanhos de W_0 e W_1, respectivamente; m é a média harmônica de n_0 e n_1; e σ_W^2 é a variância observada nos elementos de W. Se a diferença absoluta entre $\hat{\mu}_{W_0}$ e $\hat{\mu}_{W_1}$ for menor do que ϵ_{cut}, nenhum elemento de W será removido.

Com o objetivo de otimizar o método, (BIFET; GAVALDA, 2007) propuseram Adaptive Windowing 2 (ADWIN2), uma versão computacionalmente menos custosa e que não precisa verificar todas as possibilidade de corte em W buscando diferenças estatísticas na média. Isso é feito adaptando o conteúdo de W para uma estrutura de dados composta por uma sequência de bits, inspirada por um histograma exponencial apresentado por (DATAR et al., 2002). Dessa forma, é possível ter complexidade logarítmica na memória e na atualização de W.

2.2.2 DDM

O framework Drift Detection Method (DDM) (GAMA et al., 2004) acusa uma mudança na distribuição observando a probabilidade do erro (p) e o desvio padrão (s). Para calcular p, a predição ($\hat{y}$) do classificador base deve ser comparada à resposta (y).

Para cada instância i, p_i será reduzido se $\hat{y} = y$ e aumentado se $\hat{y} \neq y$. Os valores de s serão atualizados por

$$s_i = \sqrt{\frac{p_i(1-p_i)}{i}}$$

Uma diminuição constante de p indica que a distribuição é estacionária, entretanto, um crescimento gradual ou abrupto pode indicar uma mudança de conceito. É importante diferenciar quando o aumento em p representa uma mudança ou é apenas um alarme falso, muitas vezes provocado por ruído nos dados.

À medida em que novas instâncias vão chegando, além de manter p e s atualizados, os valores mínimos da probabilidade do erro e do desvio padrão também serão armazenados e correspondem, respectivamente, a p_{min} e s_{min}. Ambos serão atualizados quando $p_i + s_i < p_{min} + s_{min}$. Finalmente, uma mudança de conceito será consumada quando $p_i + s_i \geq p_{min} + \beta.s_{min}$, onde β corresponde ao nível de confiança.

Antes de detectar uma mudança na distribuição, um novo classificador já estará sendo treinado. Isso é possível graças a um estado de alerta (*warning*) que é acionado quando $p_i + s_i \geq p_{min} + \alpha.s_{min}$, porém, com um nível de confiança (α) menor para que o alerta aconteça antes da detecção. Se a mudança for confirmada, o novo classificador, ativado durante o estado de alerta, será substituído pelo antigo.

2.2.3 DOF

Degree of Drift (DoF) (SOBHANI; BEIGY, 2011) é um método cujo objetivo é detectar mudanças de conceitos calculando a diferença entre dois conjuntos de treinamento (D_t e D_{t-1}), através do algoritmo do vizinho mais próximo. Para calcular a distância, os autores propuseram o uso do Heterogeneous Euclidian/Overlap Metric (HEOM) (WILSON; MARTINEZ, 1997). Para cada nova instância t, D_t e D_{t-1} serão atualizados, os vizinhos mais próximos serão encontrados, e seus rótulos serão comparados. Grandes diferenças sugerem que houve uma mudança na distribuição.

Calcular a distância entre as instâncias de D_t e D_{t-1} utilizando HEOM pode ser feito da seguinte maneira:

$$d_{heom}(x_1, x_2) = \sqrt{\sum_{a=1}^{m} heom_a^2(x_1, x_2)}$$

$$heom_a(x_1, x_2) = \begin{cases} 0, & \text{se } a \text{ é categórico e } x_{1_a} = x_{2_a} \\ 1, & \text{se } a \text{ é categórico e } x_{1_a} \neq x_{2_a} \\ \frac{x_{1_a} - x_{2_a}}{range_a}, & \text{se } a \text{ é numérico} \end{cases}$$

onde m representa o número de instâncias em D; e $range_a$ a largura do intervalo dos valores.

Para detectar uma mudança de conceito (SOBHANI; BEIGY, 2011) propuseram o uso da seguinte relação, denominada *Degree of Drift*:

$$DoF = \frac{\sum_{i=1}^{n} \theta(i).disagree(i)}{\sum_{i=1}^{n} \theta(i)}$$

$$disagree(i) = \begin{cases} 0, & \text{se os rótulos dos vizinhos mais próximos de } x_i \text{ não combinarem} \\ 1, & \text{se os rótulos dos vizinhos mais próximos de } x_i \text{ combinarem} \end{cases}$$

onde θ é o inverso do HEOM; e n o número de instâncias em D. Assim, uma mudança será acusada quando o valor de DoF for mais afastado da média do que s desvios padrões, onde s é uma constante, recebida por meio de um parâmetro.

2.2.4 ECDD

Baseado no Exponentially Weighted Moving Average (EWMA) (ROBERTS, 1959), EWMA for Concept Drift Detection (ECDD) (ROSS et al., 2012) é uma modificação para permitir ser utilizada em ambientes com mudanças de conceitos. Originalmente, EWMA foi utilizado para detectar mudanças significativas na média (μ), por meio da fórmula $Z_t > \mu_0 + L\sigma_{Z_t}$, onde μ_0 corresponde à média das instâncias antes da mudança; Z_t corresponde a uma estimativa de μ no tempo t; σ_{Z_t} é o desvio padrão de Z_t; e, por fim, L corresponde a um limiar, responsável por definir a distância necessária para detectar uma mudança. É possível calcular Z_t da seguinte maneira: $Z_t = (1-\lambda)Z_{t-1} + \lambda X_t$, onde X_t é uma variável aleatória e λ um parâmetro que pondera o aumento das instâncias atuais em relação às mais antigas. O desvio padrão de Z_t é calculado por:

$$\sigma_{z_t} = \sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})}\sigma_x$$

onde σ_x é o desvio padrão das variáveis aleatórias.

Sabendo que a variável aleatória X_t segue uma distribuição de Bernoulli e p_t é a probabilidade do erro no tempo t, uma mudança de conceito pode ser considerada observando a variação de p. De maneira diferente do EWMA, onde os valores de p_0 e σ_x são conhecidos a priori, no ECDD $\hat{p}_0$, $\hat{\sigma}_x$ e $\hat{\sigma}_Z$ são estimados da seguinte maneira:

$$\hat{p}_{0,t} = \frac{t}{t+1}\hat{p}_{0,t-1} + \frac{1}{t+1}X_t$$

$$\hat{\sigma}_{x_t} = \hat{p}_{0,t}(1-\hat{p}_{0,t})$$

$$\hat{\sigma}_{Z_t} = \sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})}\,\hat{\sigma}_{x_t}$$

Finalmente, a mudança de conceito será detectada se $Z_t > \hat{p}_{0,t} + L_t . \hat{\sigma}_{Z_t}$, onde Z_t é o mesmo definido em EWMA.

O valor de L geralmente é calculado baseando-se numa taxa média aceitável de detecções de falsos positivos (ARL_0). Uma forma de encontrar uma valor apropriado para L, dado um ARL_0, é utilizando um método de busca, a exemplo do Monte Carlo (VERDIER; HILGERT; VILA, 2008). Entretanto, utilizando essa abordagem, encontrar um valor para L é um processo computacionalmente custoso, especialmente quando p_0 e σ_x não são conhecidos. Como alternativa (ROSS et al., 2012) propuseram uma adaptação na solução encontrada por (SPARKS, 2000) para ser utilizada no ECDD. Essa técnica consiste em aproximar a função $f(p_0; ARL_0)$ utilizando um polinômio e técnicas convencionais de regressão para estimar os coeficientes. Assim, fornecendo o ARL_0 desejado, o valor de L pode ser encontrado em $O(1)$, de acordo com o valor atual de p_0.

2.2.5 EDDM

De maneira similar ao DDM, Early Drift Detection Method (EDDM) (BAENA-GARCíA et al., 2006) detecta mudanças de conceitos baseando-se nos erros e acertos cometidos pelo classificador base no decorrer das instâncias. Entretanto, de forma diferente do primeiro, EDDM encontra a distância entre dois erros para determinar a ocorrência de uma mudança na distribuição. De acordo com (BAENA-GARCíA et al., 2006), essa modificação proporciona uma melhora nos casos de mudanças graduais e mantém boas performances nas mudanças abruptas.

Para realizar as detecções, a distância média entre dois erros ($p_i^{'}$) juntamente com o desvio padrão ($s_i^{'}$) serão armazenados. Quando $p_i^{'}$ e $s_i^{'}$ atingirem seus valores máximos, eles passarão a ser armazenados em $p_{max}^{'}$ e $s_{max}^{'}$, sendo sobrescritos cada vez que um novo máximo for encontrado.

Se $(p_i^{'} + 2.s_i^{'})/(p_{max}^{'} + 2.s_{max}^{'}) < \alpha$, EDDM entrará em estado de alerta e um novo classificador passará a ser treinado em paralelo.

Se o alerta for confirmado e $(p_i^{'} + 2.s_i^{'})/(p_{max}^{'} + 2.s_{max}^{'}) < \beta$, uma mudança de conceito será acusada e o classificador base será substituído por um mais recente, que estava sendo treinado em paralelo. Para a criação desses estados, α precisa ser maior do que β.

2.2.6 PL

Paired Learners (PL) (BACH; MALOOF, 2008) utiliza dois classificadores para detectar uma mudança de conceito: um estável (S) e um reativo (R). O primeiro utiliza, para o seu treinamento, todas as instâncias conhecidas até o momento. O segundo treina somente as últimas W instâncias, sendo W um parâmetro que define o tamanho da janela.

Quando S classifica incorretamente uma instância e R classifica corretamente a mesma instância, um bit é imediatamente inserido numa lista circular (C). Se o número de bits em C for maior que um limiar θ, uma mudança de conceito é detectada.

É importante destacar que se $C \neq \emptyset$, toda vez que S corretamente classificar uma instância, um bit será removido. Logo, mudanças só serão detectadas se uma sequência de acertos e erros acontecerem em R e S, respectivamente. Depois de confirmada uma mudança na distribuição, S será reiniciado. Para acelerar o processo de aprendizagem na nova distribuição, as últimas W instâncias treinadas em R serão atribuídas a S.

Com o objetivo de manter o classificador reativo R com as últimas W instâncias, uma possibilidade é treinar R para cada novo exemplo com todas as W instâncias. A desvantagem dessa abordagem é que o tempo de execução aumenta devido aos constantes treinamentos com todas as W instâncias.

Alternativamente, o classificador base (como por exemplo o Naive Bayes) pode ser modificado para desaprender uma instância. Assim, com a chegada de novos exemplos, o exemplo mais antigo poderá ser esquecido enquanto o mais novo poderá ser incorporado. Essa solução irá reduzir o tempo de execução do método em geral, mas sua implementação não é viável para alguns tipos de classificadores. Segundo seus autores, em ambos os casos a precisão e o uso da memória não são afetados.

2.2.7 STEPD

Statistical Test of Equal Proportions (STEPD), proposto por (NISHIDA; YAMAUCHI, 2007), utiliza um teste estatístico de iguais proporções para detectar mudanças na distribuição. Nessa abordagem duas janelas são criadas: uma com somente as últimas W instâncias e outra com todas as instâncias conhecidas até o momento. Para encontrar diferenças nas janelas, um teste de hipótese é realizado e a hipótese nula diz que ambas as janelas possuem a mesma distribuição. Se essa hipótese for rejeitada, uma mudança de conceito é acusada.

Formalmente, o teste de hipóteses é equivalente ao teste de Qui Quadrado com correção de continuidade de Yates e é definido por:

$$T(r_o, r_r, n_o, n_r) = \frac{|r_o/n_o - r_r/n_r| - 0.5(1/n_o + 1/n_r)}{\sqrt{\hat{p}\,(1-\hat{p})(1/n_o + 1/n_r)}}$$

onde r_o e r_r correspondem ao número de instâncias corretamente classificadas nas janelas antigas e recentes, respectivamente; n_o e n_r correspondem ao total de instâncias em cada janela; e $\hat{p} = (r_o + r_r)/(n_o + n_r)$. A partir do P-Valor ($P$) encontrado, STEPD entrará em estado de alerta se $P < \alpha_w$ e confirmará uma mudança de conceito quando $P < \alpha_d$.

2.3 Bases de Dados

Esta seção descreve brevemente as bases de dados selecionadas para os experimentos realizados neste trabalho.

2.3.1 Bases de Dados Artificiais

Para as bases artificiais foram selecionadas nove bases diferentes, cada uma contendo versões abruptas, graduais rápidas e graduais lentas. As próximas subseções descrevem essas bases.

2.3.1.1 Agrawal

Agrawal, originalmente proposta em (AGRAWAL; IMIELINSKI; SWAMI, 1993), combina um conjunto de informações (atributos) de pessoas com o objetivo de verificar se elas estão aptas a receberem uma certa quantidade de empréstimo. A partir dessas informações, elas são classificadas como pertencentes a um grupo A ou a um grupo B. Os atributos que representam as informações são: salário, comissão, idade, escolaridade, possui carro, código postal, valor do imóvel, quantidade de anos que vive em casa própria, e valor do empréstimo desejado. Desses atributos, três são categóricos e os demais não-categóricos. Para realizar a classificação (AGRAWAL; IMIELINSKI; SWAMI, 1993) propõem dez funções, cada uma com diferentes formas de avaliação.

2.3.1.2 Hyperplane

Por meio de um espaço d-dimensional, um conjunto de pontos x da base de dados Hyperplane (HULTEN; SPENCER; DOMINGOS, 2001) satisfaz $\sum_{i=1}^{d} w_i x_i = w_o$, onde x_i é a i-ésima coordenada de x. Seus rótulos são classificados como positivos quando $\sum_{i=1}^{d} w_i x_i \geq w_o$ e negativos quando $\sum_{i=1}^{d} w_i x_i < w_o$. Uma mudança pode ser simulada mudando a orientação e a posição do hiperplano através do tamanho relativo dos pesos (w). Dessa forma, a mudança pode ser representada por $w_i = w_i + d\sigma$, onde σ é a chance da direção da mudança ser invertida e d representa a quantidade de mudanças aplicada para cada exemplo.

2.3.1.3 LED

A base de dados LED (BIFET et al., 2010) é composta por 24 atributos categóricos, sendo 17 irrelevantes, e uma classe categórica com dez valores possíveis. Ele representa o problema de predizer um dígito mostrado por um visor LED de sete segmentos, no qual cada atributo tem 10% de chance de ser invertido (ruído). Para simular uma mudança de conceito a base de dados simplesmente muda as posições dos atributos.

2.3.1.4 Mixed

Mixed (GAMA et al., 2004) é composta por dois atributos booleanos (v e w) e dois numéricos (x e y). Cada instância pode ser classificada como positiva ou negativa. Para ser positiva é necessário que duas das três seguintes condições sejam satisfeitas: v, w, e $y < 0.5 + 0.3sin(3\pi x)$. Para simular uma mudança de conceito, os rótulos das condições descritas acima são invertidos.

2.3.1.5 Random RBF

Por definição, RandomRBF (BIFET et al., 2009) inicialmente cria n centróides com seus centros, rótulos e pesos, definidos aleatoriamente. Para determinar os valores dos m atributos de cada exemplo, as variáveis levadas em consideração são: 1) a posição do centro (previamente criado) que, quanto maior for seu peso maior será a probabilidade de ser selecionado; e 2) um número aleatório no intervalo de $(-1, +1)$ multiplicado por um valor baseado na distribuição Gaussiana e no desvio padrão do centro selecionado. Uma mudança de conceito é simulada mudando a posição dos centróides.

2.3.1.6 SEA

Inicialmente utilizada em (STREET; KIM, 2001), SEA possui três atributos numéricos, com o terceiro sendo somente para criação de ruído, e duas classes. Todos os números são gerados aleatoriamente num intervalo de 0 a 10. Para que um exemplo seja definido como pertencente à classe 1, $f_1 + f_2 \leq \theta$, onde f_1 e f_2 são o primeiro e o segundo atributo e θ um limiar. Diferentes valores atribuídos a θ irão produzir diferentes contextos. Nessa base de dados, quatro limiares são definidos: 8, 9, 7 e 9.5.

2.3.1.7 Sine

Sine (GAMA et al., 2004) possui dois atributos numéricos (x e y) e suas condições são divididas em dois contextos: Sine1 e Sine2. No primeiro, para que uma instância seja classificada como positiva, ela deve ficar abaixo da curva $y = sin(x)$. No segundo, a condição $y < 0.5 + 0.3sin(3\pi x)$ deve ser satisfeita. Em ambos os casos, a mudança de conceito é simulada invertendo as condições descritas ou alternando entre Sine1 e Sine2.

2.3.1.8 STAGGER

STAGGER (SCHLIMMER; GRANGER, 1986) utiliza três atributos categóricos: tamanho (pequeno, médio ou grande), cor (vermelha, azul ou verde) e formato (circular, triangular ou quadrangular), e cada exemplo é classificado como positivo ou negativo. As mudanças são realizadas alternando entre três diferentes contextos: o primeiro é positivo somente quando tamanho=pequeno e cor=vermelha; o segundo somente quando

cor=verde ou formato=circular; e, finalmente, o terceiro somente quando tamanho=médio ou tamanho=grande.

2.3.1.9 Waveform

A base de dados Waveform (BIFET et al., 2009) é composta por quarenta atributos numéricos, sendo os últimos dezenove utilizados para produzir ruído. O objetivo principal é detectar diferentes formas de ondas. Para simular mudanças de conceitos, as posições dos atributos – que representam um determinado contexto – são invertidas.

2.3.2 Bases de Dados Reais

As subseções a seguir descrevem brevemente as três bases reais selecionadas.

2.3.2.1 Covertype

Este conjunto de dados contém células de informação que correspondem a uma cobertura florestal de 30x30 metros, e foram determinados pelo Sistema de Informações de recursos da Região 2 do *US Forest Service (USFS)*. O objetivo é predizer o tipo de cobertura florestal a partir das variáveis cartográficas. Covertype possui 581.012 instâncias e 54 atributos, incluindo tipos numéricos e categóricos.

2.3.2.2 Electricity

Basicamente, esta base de dados é composta por 45.312 instâncias e oito atributos, apresentados e coletados do mercado de eletricidade australiano de *New South Wales*. Nesse mercado o preço não é fixo, sofrendo variações de acordo com a demanda e a oferta. Os preços são definidos a cada cinco minutos e os rótulos das classes identificam a mudança no preço relacionada com uma média móvel das últimas 24 horas. O objetivo deste problema é predizer se o preço irá aumentar ou diminuir.

2.3.2.3 Poker Hand

A base de dados PokerHand representa o problema de identificar o valor das cinco cartas em um jogo de Poker. Ela é constituída por cinco atributos categóricos, cinco atributos numéricos e a saída é uma classe categórica com dez possíveis valores representando a mão. Exemplo: um par, dois pares, uma sequência, um flush, etc. Apesar de existir uma versão mais simples deste problema (onde as cartas são ordenadas por valor e naipe, e sem duplicações), optamos por utilizar a versão original e mais difícil, composta por 1.000.000 de instâncias, dez atributos numéricos e dez possíveis respostas (dez classes).

2.4 Considerações Finais

Este capítulo introduziu os métodos de detecção de mudanças de conceito e as bases de dados utilizadas nos experimentos desta dissertação.

O Capítulo 3 trata da utilização do algoritmo genético para a otimização dos valores dos parâmetros dos métodos detectores de mudanças de conceito para os diversos tipos de bases de dados utilizadas.

3 Algoritmo Genético para Otimização de Parâmetros

Neste capítulo, mostramos de que forma os parâmetros dos métodos de detecção de mudança foram otimizados fazendo uso dos conceitos do Algoritmo Genético (AG). Os valores dos parâmetros que foram utilizados pelos autores dos diferentes métodos nem sempre são as melhores escolhas a serem feitas. Muitas vezes esses valores foram definidos para atender a um contexto específico, dependendo das bases de dados utilizadas, quantidade de instâncias, número de mudanças de conceito, entre outros. Assim, diferentes situações necessitarão de diferentes parâmetros e o AG pode ser utilizado para auxiliar na busca de valores adequados.

Diversas são as abordagens que utilizam AG para tratar problemas semelhantes. (FRIEDRICHS; IGEL, 2005) propõem o uso de estratégias evolucionárias na otimização dos parâmetros do Support Vector Machine (SVM), mais especificamente no hiper-parâmetro. Os resultados experimentais reportados indicaram uma melhora estatisticamente significante na maioria dos casos. A partir dessas informações, seus autores afirmaram que, para um modelo de seleção do SVM, otimizações evolucionárias geralmente terá um melhor desempenho do que a utilização de métodos baseados em gradiante. (LORENA; CARVALHO, 2008), também no contexto dos SVM's, investigam o ajuste dos parâmetros de classificação multi-classe dividido em múltiplos subproblemas binários. Esse ajuste é realizado em cada SVM binário através do AG e comparado com os parâmetros definidos por padrão. Por fim, (SOARES; ANTUNES; ARAUJO, 2013), em um dos seus experimentos, utilizam o AG para otimizar a performance de um comitê de Redes Neurais. Os alvos de otimização foram os subconjuntos dos modelos e suas respectivas funções de agregação.

Sabendo da propensão do AG para solucionar problemas dessa natureza, além da sua simplicidade e eficiência, optamos por sua utilização para otimizar os parâmetros dos métodos de detecção de mudanças.

O resto deste capítulo está organizado da seguinte forma: A Seção 3.1 fornece uma visão geral de como funciona os passos básicos do AG; a Seção 3.2 mostra como a população é inicializada apresentando uma solução para reduzir a probabilidade de mínimos locais; a Seção 3.3 discorre sobre o alvo de otimização representado pela função objetivo e mostra como o *crossing-over* e a mutação podem ser adaptados para o nosso contexto; e, finalmente, a Seção 3.5 discute sobre os critérios de seleção bem como a importância da realização do elitismo.

3.1 Visão Geral

Um AG é um algoritmo de busca que utiliza conceitos da teoria da evolução tais como seleção natural, crossing-over e mutação. De forma semelhante às especies, que vão evoluindo com o passar dos anos, o objetivo do AG é simular computacionalmente esses fenômenos com o intuito de otimizar problemas multidimensionais.

O Algoritmo 1 apresenta a estrutura básica do AG. Inicialmente a população é criada com p hipóteses, cada uma tendo seus parâmetros selecionados de maneira aleatória. Em seguida, a partir das hipóteses geradas, a função objetivo, que neste caso representa a precisão dos métodos, é calculada. Assim, cada elemento de P será composto por uma hipótese h e sua respectiva precisão.

Algoritmo 1: Algoritmo Genético Básico

Input: limiar t, número de hipóteses p, número de elitismo e, taxa de mutação m

1 $P \leftarrow$ Inicializa população gerando p hipóteses aleatoriamente;
2 **foreach** h ***in*** P **do**
3 | calcular $Fitness(h)$;
4 **end**
5 **for** $i = 1$ ***to*** t **do**
6 | Cria uma nova geração, P_s;
7 | *Seleciona* os e mais precisos métodos de P para adicionar em P_s;
8 | Utilizando o método do *Ranking*, probabilisticamente *seleciona* $\frac{p-e}{2}$ pares de hipóteses de P para realizar $crossing - over$. Adiciona os resultados em P_s;
9 | Escolhe m porcento de membros de P_s para sofrer *mutação*;
10 | $P \leftarrow P_s$;
11 | **foreach** h ***in*** P **do**
12 | | calcular $Fitness(h)$;
13 | **end**
14 **end**
15 **return** hipótese de P com maior aptidão;

Uma vez inicializada a população, o próximo passo será criar novas gerações dessa população com o objetivo de encontrar indivíduos mais aptos. De maneira geral, o aprimoramento dos indivíduos acontecerá por meio de conceitos herdados da *teoria da evolução*, simulando computacionalmente fenômenos como crossing-over e mutação. As próximas Subseções detalham cada um desses passos apresentados no Algoritmo 1.

3.2 Inicialização da População

O primeiro passo para a utilização do AG é a inicialização da população, mostrada na linha 1 do Algoritmo 1. De acordo com (MITCHELL, 1997), um conjunto de p hipóteses (h) é selecionado aleatoriamente, contribuindo para a formação da população. Trazendo para o contexto dos métodos de detecção de mudanças, h corresponde ao conjunto de parâmetros

pertencentes a um determinado método e tendo seus valores definidos aleatoriamente. Dessa forma, um p grande significa uma maior abrangência de hipóteses a serem exploradas, porém, significa também um maior custo computacional.

Buscando deixar o universo das hipóteses o mais distribuído possível, algumas restrições foram definidas. A primeira delas é que os valores de cada parâmetro devem possuir um limite superior (LS) e um limite inferior (LI). A segunda é que os intervalos dos números aleatórios encontrados para cada parâmetro devem ser obtidos usando o algoritmo 2, onde a hipótese h corresponde a uma matriz $p \times M$ com $j \in [1..p]$ linhas e $m \in [1..M]$ colunas, sendo p a quantidade de hipóteses e M o número total de parâmetros do método. É importante observar que para cada coluna m, diferentes valores de LS e LI serão utilizados, pois esses valores são específicos para cada parâmetro do método. A fim de simplificar o pseudo-código, a iteração do m foi omitida.

Algoritmo 2: Geração de hipóteses bem distribuídas

Input: hipótese h, limite inferior LI, limite superior LS, tamanho da população p

1 $a \leftarrow LI$; $j \leftarrow 1$;
2 **while** $a < LS$ **do**
3 $\quad h_{j,m} \leftarrow$ Gera um número aleatório no intervalo a e $a + (LS - LI)/p$;
4 $\quad a \leftarrow a + (LS - LI)/p$;
5 $\quad j \leftarrow j + 1$;
6 **end**
7 **return** h;

Um dos problemas do AG é a possibilidade de convergência para um mínimo local e a utilização dos números aleatórios para inicialização da população potencializa esse problema. Quando esses valores são gerados, existe um risco deles ficarem próximos uns dos outros fazendo com que a evolução do algoritmo fique restrita a essa região, que pode não ser uma boa opção. Utilizando o Algoritmo 2 – que deve ser chamado na linha 1 do Algoritmo 1 – o risco de receber números aleatórios numa mesma região é eliminado. Como será mostrado mais adiante, o conceito de mutação também pode ajudar a minimizar os riscos de mínimos locais.

3.3 Definição da Função Objetivo

De acordo com a teoria Darwiniana, a função objetivo ou aptidão de um indivíduo se refere a sua capacidade de viver e reproduzir em um determinado ambiente. Quanto maior for essa capacidade, mais apto ou mais adaptado estará esse indivíduo em relação aos outros. Um conjunto de características é responsável pela formação da aptidão, e aquelas que contribuem de maneira mais efetiva, naturalmente, possuirão uma maior probabilidade de serem herdadas (GOLDBERG, 1989).

Para simular esse processo no AG, os diferentes valores atribuídos aos parâmetros representarão as características do indivíduo. Já a precisão do método, alcançada a partir dos parâmetros escolhidos, representará a aptidão. Com isso, desejamos encontrar características que maximizem a função objetivo. Apesar da precisão do método ter sido o nosso alvo de otimização, outras opções também poderiam se encaixar neste fim, tais como uso da memória ou tempo de execução. As linhas 3 e 12 do Algoritmo 1 correspondem à chamada da função objetivo, que retornará a precisão do método para aquele conjunto de parâmetros.

3.4 Operadores Genéticos

Um dos fatores mais importantes para a evolução biológica é a variabilidade genética. Por meio dela, as características mais significativas para a sobrevivência das espécies tenderão a se destacar em relação às outras, ou seja, sofrerão uma seleção natural.

Para realização dessa diversificação, um dos principais responsáveis é um fenômeno conhecido como crossing-over. Nele, diferentes cromossomos fazem uma troca entre suas sequências de genes dando origem a novas sequências com outras configurações.

De forma análoga ao campo biológico, o AG também realiza crossing-over na população. Durante o processo, um conjunto de parâmetros fará o papel dos cromossomos e os parâmetros individuais, diferenciados por seus valores, representarão os genes. O objetivo da realização do crossing-over é fazer com que cromossomos com um bom nível de aptidão/precisão troquem informações genéticas com outros semelhantes em busca de uma nova sequência ainda mais apta.

Diversas são as maneiras de simular esse fenômeno computacionalmente e as formas mais comuns são:

- Escolher n genes em posições aleatórias para cruzar; ou
- Definir um ou mais pontos de corte.

Para nossa implementação, optamos pela segunda opção, definindo um único ponto de corte numa posição (k), escolhida aleatoriamente entre 1 e M, sendo M a quantidade de parâmetros no método. Dessa forma, dado dois cromossomos distintos, todos os genes no intervalo k e M farão as trocas de valores, ou seja, os genes que pertenciam ao cromossomo 1 no intervalo k e M passará a pertencer ao cromossomo 2 e vice versa.. A linha 8 do Algoritmo 1 mostra o momento em que o crossing-over é realizado.

Outro fenômeno importante, que também contribui para a evolução das especies, é a mutação. Reagindo a um série de possíveis estímulos, ou até mesmo por conta de um erro no momento da reprodução da célula, a mutação provoca mudanças nos genes

que alteram as características do indivíduo. Essas mudanças podem ser recebidas de duas maneiras distintas: positivamente, fazendo com que a mutação se reproduza, uma vez que a sua característica está contribuindo de forma efetiva para a sobrevivência da espécie; ou negativamente, caso em que a mudança torna um indivíduo menos apto que os outros. Assim, em várias situações, a própria seleção natural se encarrega de não passar adiante mutações prejudiciais (HERRON, 2009).

Inspirado também nas pesquisas de Charles Darwin, a etapa presente no AG denominada mutação tem grande importância no processo de otimização. Uma de suas características é encontrar genes até então desconhecidos na população, podendo contribuir para a geração de indivíduos mais aptos. Essa introdução de genes desconhecidos pode causar também um efeito negativo, com uma redução na aptidão. Nesses casos, os indivíduos afetados tendem a ter uma menor probabilidade de serem selecionados para realizar crossing-over e a característica prejudicial, na maioria das vezes, acaba ficando estagnada.

Outro aspecto importante da mutação é a sua capacidade de evitar a convergência do algoritmo para mínimos locais. Como os genes que sofrem mutação normalmente encontram valores que fogem do escopo dos genes atuais da população, essa nova informação acaba contribuindo para que o algoritmo consiga tomar um rumo diferente do que se encontra.

A simulação da mutação para otimização dos métodos de detecção de mudanças foi realizada da seguinte maneira: em cada iteração do algoritmo, cada gene presente na população tem uma probabilidade, normalmente inferior a 1%, de sofrer mutação. Uma vez que algum gene seja selecionado para sofrer mutação, seu valor atual é alterado para um outro que é escolhido aleatoriamente dentro do limite inferior e superior do parâmetro. No pseudo-código que representa os passos básicos do AG (Algoritmo 1), a mutação ocorre na linha 9.

3.5 Seleção

Para cada geração do AG, é fundamental fazer com que os indivíduos mais aptos tenham uma maior probabilidade de serem selecionados para realização do crossing-over. Como dito anteriormente, isso irá contribuir para a evolução da população. Duas técnicas são comumente utilizadas para este fim: o método da roleta e o método do ranking. No primeiro, a probabilidade de cada indivíduo ser selecionado será diretamente relacionada com a sua aptidão. Já no segundo, a chance de cada indivíduo será proporcional à sua posição ocupada numa tabela ordenada pela aptidão (BRAGA; CARVALHO; LUDERMIR, 2000). Essas informações podem ser observadas nas Tabelas 1 e 2.

Um dos problemas da utilização do método da roleta acontece quando as aptidões dos indivíduos são muito próximas. Com isso, as chances dos indivíduos serem selecionados serão semelhantes, fazendo com que aqueles que alcançarem uma melhor aptidão obtenham

Tabela 1: Método da Roleta: as probabilidades dos cromossomos serem selecionados são proporcionais às suas precisões, e seus valores são encontrados por meio da normalização.

	Aptidão/Precisão	**Chance de Seleção**
Cromossomo A	85,00%	20,63%
Cromossomo B	84,00%	20,39%
Cromossomo C	82,00%	19,90%
Cromossomo D	81,00%	19,66%
Cromossomo E	80,00%	19,42%

Tabela 2: Método do Ranking: as chances são definidas de acordo com as posições. Os valores correspondem a uma porcentagem definida sobre a porcentagem ainda não utilizada. Assim, definindo uma porcentagem de 60%, a chance do primeiro lugar será 60% de 100% (60%), o do segundo lugar será 60% de 40% (24%), e assim por diante. O último ficará com a porcentagem restante.

	Aptidão/Precisão	**Chance de Seleção**
Cromossomo A	85,00%	60,00%
Cromossomo B	84,00%	24,00%
Cromossomo C	82,00%	09,60%
Cromossomo D	81,00%	03,84%
Cromossomo E	80,00%	02,56%

uma vantagem pouco relevante em relação aos outros para realização do crossing-over. Nos métodos de detecção de mudanças, esse é um tipo de problema que pode acontecer com bastante frequência. No geral, a modificação dos parâmetros não causam uma variação brusca nas precisões, motivo pelo qual as diferenças nos valores tendem a ser pequenas. Buscando evitar essas situações, optamos pelo uso do método do ranking, como é mostrado na linha 8 do Algoritmo 1. As Tabelas 1 e 2 retratam também como cada método se comporta quando os valores da aptidão são próximos.

Por fim, quando toda a população está disponível para realizar crossing-over, existe um risco de que indivíduos com boas características se misturem com outros e a percam. Para minimizar esse risco, antes de iniciar o crossing-over, um ou mais indivíduos com as melhores aptidões já serão selecionados para a próxima geração sem sofrer qualquer modificação nos seus genes (linha 7 do Algoritmo 1). Esse processo é conhecido na literatura como elitismo. É importante ressaltar que os indivíduos que foram elitizados na última geração não necessariamente serão também na próxima. Se na última geração, após o crossing-over, forem encontradas aptidões que superam às dos últimos elitizados, na próxima geração eles serão substituídos.

3.6 Considerações Finais

Este capítulo apresentou de maneira detalhada como o AG foi adaptado para otimizar os parâmetros dos métodos de detecção de mudanças.

O Capítulo 4 mostrará a metodologia utilizada para realização dos experimentos e fará uma análise empírica dos resultados para cada método.

4 Análise Empírica

Neste capítulo, faremos uma análise dos métodos baseando-se nos resultados apresentados tanto nas bases reais quanto nas bases artificiais. Para cada um dos oito métodos verificamos suas precisões, tempo de execução, uso de memória, quantidade de alarmes falsos e precisão na posição das detecções das mudanças. Em todas as bases artificiais foram inseridas mudanças abruptas, graduais rápidas e graduais lentas.

A seção 4.1 fala sobre a metodologia utilizada para chegar aos resultados obtidos bem como a forma como as bases artificiais foram configuradas; já as seções 4.2 a 4.8 detalham os resultados obtidos pelos métodos ADWIN, DDM, DoF, ECDD, EDDM, PL, e STEPD, respectivamente.

4.1 Metodologia e Configurações

Levando em consideração o ambiente com fluxo contínuo de dados, um procedimento de avaliação comumente utilizado neste contexto é o *Test-Then-Train*. Nele, na medida em que as instâncias vão chegando, primeiramente elas são testadas e, somente depois, treinada.

Uma definição mais detalhada pode ser feita da seguinte maneira: antes de conhecer a instância que acaba de chegar o classificador realiza a sua predição baseando-se em zero ou mais instâncias já conhecidas; logo após, a precisão será atualizada de acordo com o erro ou acerto cometido; por fim, a instância será treinada pelo classificador e exercerá influência na sua decisão na próxima iteração do algoritmo.

Para seleção dos parâmetros dos métodos de detecção de mudanças, o AG foi utilizado em cada um deles nas diferentes bases. Naturalmente, os valores encontrados a priori pelo AG estão sobreajustados (overfitting) às situações para as quais foram submetidos. Por exemplo: os parâmetros otimizados encontrados pelo AG para o DDM, na base Sine, obterá uma precisão muito boa para ela, porém, é provável que não tenha resultados tão bons em outras bases.

Com o objetivo de encontrar parâmetros mais gerais, ou seja, que mantenha boas precisões em diferentes cenários, para cada método foi calculada a média dos valores obtidos em todas as bases com mudanças abruptas, graduais lentas, graduais rápidas e nas bases reais. Dessa forma, os métodos terão seus próprios valores de parâmetros variando de acordo com a característica da base e não pela base em si.

Por fim, os parâmetros utilizados no AG, encontrados a partir de testes preliminares, foram os seguintes: tamanho da população = 20; número de elitismo por geração = 2; taxa

do método do ranking = 0,2; probabilidade de algum gene na população sofrer mutação (para cada geração) = 0,1; e, finalmente, número de gerações = 10.

Para avaliar o uso da memória, foi aplicada uma métrica que computa a quantidade de memória utilizada pelo método em bytes/hora. Já o tempo de execução é medido em segundos. Tanto no cálculo da precisão quanto no uso da memória e tempo de execução, para cada método com as bases artificiais, os experimentos foram repetidos 40 vezes e o valor final corresponde à média dessas repetições, utilizando intervalos com 95% de confiança.

Em todas as bases artificiais foram utilizadas 10.000 instâncias, cada uma contendo quatro mudanças de conceito nas posições 2.000, 4.000, 6.000 e 8.000. Para mudanças abruptas, a transição de uma distribuição para outra ocorre em apenas uma instância. Nos casos das mudanças graduais rápidas e graduais lentas elas ocorrem, respectivamente, a cada 100 e 500 instâncias. Não existe nenhum consenso na literatura com relação a definição desses valores, ficando as escolhas a cargo dos autores.

Algumas das bases artificiais possuem uma ou mais características parametrizáveis. Essas características, bem como os valores utilizados, são descritos e organizados da seguinte forma:

- Nas bases Agrawal, LED, e SEA é possível definir a porcentagem de ruído e, para os experimentos, os valores definidos foram, respectivamente, 5%, 10% e 10%;
- Na base Hyperplane as parametrizações podem ser feitas no número de classes, número de atributos, magnitude da mudança e porcentagem de ruídos, e os valores utilizados foram 2, 10, 1 e 5%, respectivamente;
- Na base RandomRBF é possível definir o número de classes, número de atributos e número de centróides, e os valores escolhidos foram 6, 10 e 50, respectivamente; e
- As bases Mixed, Sine, Stagger e Waveform não possuem nenhuma característica parametrizável.

Representando problemas reais, as bases Covertype, Electricity e PokerHand são utilizadas.

Finalmente, para identificar a precisão dos métodos na detecção das mudanças, é calculada a média das instâncias em que cada método, dentro das 40 repetições, detectou a nova distribuição. Para essa finalidade definimos que uma detecção será considerada se em pelo menos 30 das 40 repetições o método acusar que houve uma mudança. Detecções antes dos pontos de mudanças e/ou repetidas serão consideradas alarmes falsos. Para as bases reais foram somente computadas a quantidade de detecções.

Todos os testes foram realizados num computacor com processador Intel Core 2 Duo E4500, 3GB de memória principal e utilizando o sistema operacional Ubuntu 14.04 LTS 64 bits.

4.2 ADWIN

4.2.1 Otimizando Parâmetros

No ADWIN, é possível manipular dois parâmetros: 1) um nível de confiança para que uma janela W tenha seu valor reduzido – $\delta \in (0, 1)$; e 2) a frequência mínima de instâncias (f) necessárias para que uma redução em W ocorra. Os valores definidos por padrão, foram: $\delta = 0{,}002$ e $f = 32$.

Inicialmente, para as bases com mudanças abruptas, foram encontrados pelo AG os seguintes parâmetros: $\delta = 0{,}58$ e $f = 70$. É possível observar que o δ sofreu um acréscimo consideravelmente grande em relação ao seu valor padrão. Dessa forma, ADWIN ficará mais propenso a sofrer reduções na sua janela e, consequentemente, aumentar o seu grau de sensibilidade para detecções de novos conceitos.

Com essa modificação, a base na qual o ADWIN foi mais beneficiada foi Agrawal, com um ganho de 0,71%. Por outro lado, a base RandomRBF foi a que sofreu uma maior redução, com uma perda de 0,14%. No geral, o ADWIN na base RandomRBF demandou mais tempo para realizar as detecções de mudanças além de disparar vários falsos positivos. Já o ADWIN na base Agrawal obteve um dos menores índices de falsos positivos juntamente com uma boa precisão nas detecções das mudanças. Esses fatores foram determinantes para as respectivas perdas e ganhos. Finalmente, o ganho médio do ADWIN nas bases abruptas foi de 0,12%.

Com as mudanças graduais rápidas, os parâmetros encontrados foram $\delta = 0{,}57$ e $f = 158$. O aumento considerável na frequência pode ser justificado pelo maior tempo de transição entre um conceito e outro. Assim, um f grande pode contribuir para redução de alarmes falsos durante esse período.

Novamente, Agrawal foi a base onde o ADWIN obteve o maior aumento na precisão (0,70%). Já na Mixed obteve a maior queda, com perda de 0,29%. Apesar do aumento no valor de f trazer ganhos, como dito anteriormente, o seu tamanho também pode estar contribuindo para a queda de rendimento do ADWIN em algumas bases. Sabendo que o aumento no δ também causa uma maior predisposição para alarmes falsos, um falso positivo momentos antes de alguns dos pontos de mudança fará com que a detecção seja acusada somente depois do período de transição, que é de 100 instâncias, diminuindo a eficiência do ADWIN. Esse é o tipo de problema que não aconteceria caso f fosse menor. Calculando a média de ganhos e perdas nestas bases em relação aos parâmetros padrões,

houve uma melhora de 0,03%.

Nas bases graduais lentas, os parâmetros encontrados foram $\delta = 0{,}48$ e $f = 156$. Neste contexto, um f grande traz mais vantagens do que desvantagens, haja vista a quantidade de instâncias necessárias para realizar a transição (500). O δ também sofreu uma pequena redução se comparado aos casos anteriores, tornando o ADWIN um pouco menos sensível. A base mais beneficiada com os parâmetros continuou sendo a Agrawal, com um ganho de 0,60%, enquanto que a maior queda foi na RandomRBF (0,10%). De todos os três tipos de mudanças, no geral, foi nas bases graduais lentas onde o ADWIN obteve os maiores ganhos na precisão, com uma média positiva de 0,16%.

Com as bases reais, os valores médios selecionados pelo AG foram $\delta = 0{,}63$ e $f = 10$. A partir dos parâmetros encontrados, é possível inferir que as mudanças nas distribuições é uma característica que prevalece nessas bases, sendo necessário um δ alto para detectá-las mais facilmente e um f pequeno para reduzir o período de inatividade das detecções. Utilizando os novos valores, o ADWIN apresentou ganhos de 3,98% e 3,39%, nas bases Covertype e Electricity, respectivamente, e na PokerHand manteve a mesma média apresentada com os parâmetros padrões.

4.2.2 Precisão

De acordo com os valores apresentados na Tabela 4, o ADWIN alcançou suas maiores precisões médias nas bases com mudanças abruptas (79,38%), seguido pelas bases com mudanças graduais rápidas (79,07%), graduais lentas (77,28%) e bases reais (73,74%). Individualmente, foi nas bases Stagger, com os três tipos de mudanças, onde o ADWIN obteve os melhores resultados. Já nas bases RandomRBF e PokerHand, ele alcançou os resultados mais baixos.

4.2.3 Tempo de Execução

A Tabela 5 mostra os tempos utilizados nas diferentes bases para o NB, juntamente com o ADWIN, chegarem aos valores apresentadas. Das bases artificiais, LED foi a que o ADWIN precisou de mais tempo, enquanto na SEA seu desempenho foi o melhor. Uma comparação rápida entre as características de ambas justifica essa diferença: enquanto LED possui 24 atributos booleanos e 10 classes, SEA possui apenas 3 atributos numéricos e 2 classes. Nas bases reais, o rendimento do ADWIN foi mais lento na Covertype e na PokerHand. Na Electricity, até pelo número de instâncias bem inferior, o ADWIN conseguiu o seu melhor tempo.

Tabela 3: Precisão do ADWIN com 95% de confiança (parâmetros padrões)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	72,33±0,33	72,12±0,33	71,24±0,33	-
Hyperplane	83,03±0,35	82,87±0,36	81,73±0,37	-
LED	68,10±0,25	68,00±0,24	67,43±0,25	-
Mixed	89,61±0,22	89,26±0,19	84,76±0,22	-
RandomRBF	48,83±0,35	48,84±0,35	48,85±0,38	-
SEA	84,71±0,23	84,71±0,23	84,67±0,23	-
Sine	89,94±0,34	89,58±0,33	85,60±0,24	-
Stagger	98,72±0,09	98,32±0,08	93,83±0,12	-
Waveform	78,05±0,36	77,72±0,34	76,01±0,33	-
Covertype	-	-	-	82,22
Electricity	-	-	-	81,57
PokerHand	-	-	-	50,07
Média	79,26	79,05	77,12	71,29

Tabela 4: Precisão do ADWIN com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	73,04±0,31	72,82±0,31	71,84±0,32	-
Hyperplane	83,69±0,30	83,35±0,34	82,05±0,35	-
LED	68,10±0,25	67,98±0,24	67,47±0,25	-
Mixed	89,55±0,21	88,97±0,19	84,69±0,22	-
RandomRBF	48,69±0,35	48,72±0,35	48,75±0,39	-
SEA	84,73±0,23	84,72±0,23	84,69±0,23	-
Sine	89,86±0,36	89,33±0,34	85,57±0,26	-
Stagger	98,65±0,10	98,10±0,08	94,33±0,16	-
Waveform	78,08±0,35	77,66±0,33	76,15±0,34	-
Covertype	-	-	-	86,20
Electricity	-	-	-	84,96
PokerHand	-	-	-	50,07
Média	79,38	79,07	77,28	73,74

Tabela 5: Tempo de execução do ADWIN com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	0,48±0,02	0,49±0,02	0,50±0,02	-
Hyperplane	0,41±0,02	0,40±0,02	0,39±0,02	-
LED	1,02±0,03	1,02±0,03	1,00±0,03	-
Mixed	0,28±0,02	0,28±0,02	0,28±0,02	-
RandomRBF	0,53±0,02	0,52±0,02	0,52±0,02	-
SEA	0,27±0,02	0,26±0,02	0,27±0,02	-
Sine	0,29±0,02	0,28±0,02	0,28±0,02	-
Stagger	0,27±0,02	0,28±0,02	0,29±0,02	-
Waveform	0,68±0,02	0,68±0,02	0,68±0,02	-
Covertype	-	-	-	99,71
Electricity	-	-	-	2,30
PokerHand	-	-	-	95,99
Média	0,47	0,47	0,47	66,00

4.2.4 Memória

Observando os valores da Tabela 6, é possível identificar o mesmo padrão de comportamento presente na tabela 5: das bases artificiais, LED foi a mais custosa para o ADWIN, enquanto SEA foi a que menos lhe demandou uso de memória por hora. A mesma

justificativa apresentada para o tempo de execução se aplica à memória.

Nas bases reais, porém, o uso de memória do ADWIN no Covertype foi muito superior ao uso no PokerHand, segundo maior. Apesar de o PokerHand possuir uma quantidade de instâncias superior (1.000.000 vs. 581.012) e uma quantidade de classes maior (10 vs. 7), a quantidade de atributos é inferior (11 vs. 54) e, ao que tudo indica, é um dos principais responsáveis por essa diferença no consumo da memória.

Tabela 6: Uso de memória do ADWIN em Bytes por hora com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	2,28±0,08	2,35±0,07	2,39±0,09	-
Hyperplane	1,61±0,08	1,56±0,08	1,54±0,08	-
LED	10,20±0,26	10,16±0,26	10,04±0,26	-
Mixed	0,84±0,05	0,84±0,06	0,83±0,05	-
RandomRBF	2,50±0,10	2,47±0,09	2,47±0,09	-
SEA	0,75±0,05	0,74±0,04	0,76±0,05	-
Sine	0,84±0,05	0,84±0,05	0,83±0,05	-
Stagger	0,80±0,05	0,80±0,05	0,84±0,05	-
Waveform	4,03±0,13	4,04±0,13	4,16±0,13	-
Covertype	-	-	-	1290,31
Electricity	-	-	-	8,04
PokerHand	-	-	-	540,72
Média	2,65	2,64	2,65	613,02

4.2.5 Precisão na Detecção das Mudanças e Alarmes Falsos

A partir das posições médias e da quantidade de falsos positivos e falsos negativos apresentados na Tabela 7, é possível extrair algumas informações a respeito do desempenho do ADWIN nas diferentes bases.

No geral, SEA foi a base onde o ADWIN mais apresentou falsos negativos e também onde obteve menos falsos positivos. Sua pequena quantidade de atributos (3), juntamente com a baixa quantidade de classes (2), torna sua mudança de conceito bastante sutil, mesmo nos casos abruptos. Assim, identificar uma nova distribuição pode não ser uma tarefa tão simples e a utilização de parâmetros que deixem o ADWIN mais sensível pode ser desejável especificamente para este problema.

Por outro lado, na Hyperplane o ADWIN detectou todas as mudanças de forma relativamente rápida, porém, foi onde apresentou mais falsos positivos. Nas bases reais (Tabela 8), uma quantidade muito grande de mudanças foram detectadas na Covertype pelo ADWIN, enquanto nenhuma foi detectada na PokerHand.

Tabela 7: Posição média das detecções, falsos positivos (FP), e falsos negativos (FN) do ADWIN com as bases artificiais

	PM1	PM2	PM3	PM4	FP
Agrawal_A	2061.00	4145.75	6135.25	8125.00	8
Agrawal_{GR}	2080.50	4217.25	6215.13	8180.00	1
Agrawal_{GL}	2085.00	4380.83	6443.24	8316.50	4
Hyperplane_A	2084.50	4063.25	6079.75	8078.00	156
Hyperplane_{GR}	2098.50	4151.75	6138.00	8157.50	140
Hyperplane_{GL}	2115.00	4708.75	6585.75	8440.50	84
LED_A	FN	4114.50	6081.50	8289.69	5
LED_{GR}	FN	4143.75	6108.50	FN	2
LED_{GL}	FN	4228.75	FN	FN	10
Mixed_A	2036.50	4035.00	6033.75	8062.25	1
Mixed_{GR}	2029.75	4089.75	FN	8039.75	26
Mixed_{GL}	3536.50	4991.50	6821.50	FN	80
RandomRBF_A	2689.00	FN	6286.50	8220.75	15
RandomRBF_{GR}	2728.75	FN	6309.00	8360.75	7
RandomRBF_{GL}	2975.68	FN	6506.50	8515.00	0
SEA_A	FN	FN	6952.82	FN	0
SEA_{GR}	FN	FN	6968.72	FN	0
SEA_{GL}	FN	FN	7031.58	FN	0
Sine_A	2024.50	4047.50	6047.50	8073.50	18
Sine_{GR}	2022.25	4082.25	6271.61	8032.25	21
Sine_{GL}	3059.75	5119.75	7423.50	FN	74
Stagger_A	2034.25	4039.75	6029.00	8059.00	0
Stagger_{GR}	2021.50	4081.50	FN	8031.50	38
Stagger_{GL}	3194.75	5595.50	6633.50	FN	93
Waveform_A	2054.00	4097.25	6187.75	8217.50	14
Waveform_{GR}	2066.00	4129.25	6215.75	FN	9
Waveform_{GL}	2319.25	FN	FN	FN	25

Tabela 8: Quantidade de detecções realizadas pelo ADWIN nas bases reais

	Covertype	Electricity	PokerHand
Detecções	1398	105	0

4.3 DDM

4.3.1 Otimizando Parâmetros

Basicamente, o DDM conta com três características parametrizáveis: 1) a quantidade de instâncias mínimas para que o DDM seja ativado e possa realizar as detecções (n); 2) o fator multiplicador de desvios padrões que irá influenciar na entrada do método no estado de alerta (α_w); e 3) o fator multiplicador que irá concretizar a detecção (α_d). Por padrão, os valores definidos são, respectivamente, 30, 2 e 3. Com a utilização do AG para realizar a busca dos parâmetros, os valores encontrados diferenciam-se para as bases com mudanças abruptas, graduais rápidas, graduais lentas e nas bases reais.

Nas bases com mudanças abruptas, os parâmetros encontrados foram: $n = 140$, $\alpha_w = 1{,}73$, e $\alpha_d = 2{,}42$. A base na qual o método mais se beneficiou com a alteração destes valores foi a Sine, com um aumento de 1,06% na precisão em relação aos parâmetros definidos por padrão. Com a diminuição dos valores de α_w e α_d, DDM passa a ser mais sensível às mudanças, porém, está mais propenso a detectar falsos positivos (consequência

da maior sensibilidade). No caso da base de dados Sine, uma maior sensibilidade e, consequentemente, uma detecção mais eficiente das mudanças, pesou mais positivamente do que um provável aumento de falsos positivos. O mesmo não aconteceu com o DDM na base RandomRBF, que teve uma redução na precisão de 0,19%. Entretanto, no geral, o desempenho do DDM nas bases com mudanças abruptas teve um aumento médio de 0,45% na precisão, e oito das nove bases foram favorecidas.

Com as mudanças graduais rápidas, os valores definidos foram $n = 119$, $\alpha_w = 2{,}0$, e $\alpha_d = 2{,}35$. Com esses novos valores, o DDM ficará ainda mais sensível para detecções quando comparado aos parâmetros escolhidos para as bases com mudanças abruptas. Entretanto, entrará em estado de alerta menos vezes pelo fato de α_w ser maior. As bases na qual o DDM foi mais favorecido com os novos valores foram Hyperplane e LED, com uma melhora de 0,75% e 0,73% nas suas precisões. A maior perda foi na RandomRBF, com uma queda de 0,33%. Com a mudança nos parâmetros, o DDM nas bases com mudanças graduais rápidas teve um aumento médio de 0,36%.

Já com as bases de mudanças graduais lentas, foram selecionados os seguintes valores: $n = 165$, $\alpha_w = 2{,}32$, e $\alpha_d = 2{,}87$. Com estes valores é possível observar que o nível de sensibilidade é inferior em relação às velocidades de mudanças anteriores e um pouco superior em relação aos parâmetros padrões. Mesmo a sensibilidade sendo algo benéfico, seu uso demasiado, sobretudo em mudanças que levam muitas instâncias para acontecer, como é o caso das graduais lentas, fará com que uma quantidade excessiva de alarmes falsos sejam disparados em momentos de transição.

Um parâmetro que pode ajudar nesse equilibrio é o n: uma vez que o seu valor corresponde ao intervalo mínimo de instâncias para ocorrer uma ou mais detecções, uma escolha adequada manterá uma distância razoável entre elas. Com isso, o método terá mais tempo para se estabilizar, diminuindo a quantidade de disparos, inclusive em momentos de transição. Aplicando os valores encontrados pelo AG, o DDM obteve maior vantagem na base LED, com 0,54% de ganho. Novamente, o DDM foi menos favorecido na RandomRBF, tendo sua precisão reduzida em 0,18%.

De todas as bases utilizadas para o experimento, RandomRBF é a que apresenta um maior nível de dificuldade para convergência, fazendo com que o classificador base necessite de mais tempo para realizar uma generalização satisfatória. Por essa característica, a detecção de alarmes falsos e, consequentemente, a reinicialização do classificador, causará um prejuízo maior. Devido a essa particularidade, um DDM menos sensível às detecções de mudanças – porém mais robusto aos ruídos – seria mais favorável à RandomRBF. Como os parâmetros escolhidos correspondem à média dos valores encontrados pora cada base, a característica que irá prevalecer não é a melhor para RandomRBF, que é exceção. Com os novos valores, as bases com mudanças graduais lentas tiveram um aumento médio de 0,17% na precisão.

Para as bases de dados reais, foram encontrados $n = 52$, $\alpha_w = 1{,}42$, e $\alpha_d = 1{,}99$. De todos os parâmetros utilizados até então, esses são os mais sensíveis às detecções. Uma possível explicação pode estar associada à quantidade de mudanças na distribuição: tanto na base Electricity quanto na Covertype esse número é bem superior às quatro mudanças definidas nas bases artificiais. Assim, com os novos valores, o DDM teve um aumento de 4,16% e 2,52% nas precisões das bases Electricity e Covertype, respectivamente. Já na PokerHand, que mantém uma característica mais uniforme em relação às mudanças, o DDM teve uma queda de 0,10%. No geral, com as bases reais houve um ganho médio de 2,19%.

Tabela 9: Precisão do DDM com 95% de confiança (parâmetros padrões)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	72,32±0,33	72,20±0,30	71,51±0,32	-
Hyperplane	83,53±0,34	83,22±0,31	81,88±0,37	-
LED	68,16±0,34	67,99±0,42	67,39±0,35	-
Mixed	89,76±0,19	89,37±0,18	84,98±0,22	-
RandomRBF	49,11±0,37	49,14±0,38	49,22±0,36	-
SEA	85,14±0,26	85,05±0,26	84,93±0,27	-
Sine	88,77±0,85	89,18±0,55	86,08±0,32	-
Stagger	99,18±0,08	98,09±0,10	94,60±0,15	-
Waveform	77,47±0,44	77,25±0,41	75,76±0,40	-
Covertype	-	-	-	86,61
Electricity	-	-	-	82,58
PokerHand	-	-	-	50,07
Média	79,27	79,05	77,37	73,09

Tabela 10: Precisão do DDM com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	72,88±0,31	72,84±0,31	71,62±0,33	-
Hyperplane	84,04±0,31	83,97±0,30	82,16±0,38	-
LED	69,01±0,26	68,72±0,25	67,93±0,27	-
Mixed	89,96±0,23	89,46±0,17	84,83±0,20	-
RandomRBF	48,92±0,37	48,81±0,36	49,04±0,35	-
SEA	85,56±0,24	85,47±0,24	85,13±0,26	-
Sine	89,83±0,40	89,62±0,30	86,16±0,30	-
Stagger	99,22±0,08	97,98±0,09	94,76±0,12	-
Waveform	78,10±0,40	77,85±0,37	76,21±0,34	-
Covertype	-	-	-	89,13
Electricity	-	-	-	86,74
PokerHand	-	-	-	49,97
Média	79,72	79,41	77,54	75,28

4.3.2 Precisão

De acordo com os resultados apresentados na Tabela 10, é possível notar nas bases artificiais que, à medida que os momentos de transição vão ficando maiores, a precisão vai diminuindo. Essa característica pode ser observada através das médias dos resultados do DDM nas mudanças abruptas, graduais rápidas e graduais lentas, que são, respectivamente, 79,72%, 79,41%, e 77,54%. Entretanto, observando individualmente

cada base, esse comportamento nem sempre ocorre, principalmente quando as mudanças abruptas e graduais rápidas são comparadas.

Diversos são os fatores que podem contribuir para que os resultados não sigam um padrão de comportamento naturalmente esperado, dentre os quais destacamos:

- Os parâmetros escolhidos para um determinado método pode ter influenciado mais positivamente na sua precisão do que para o outro. Como o AG é encontra valores aproximadamente ótimos, esse tipo de situação é algo que pode acontecer com uma frequência relativamente alta; e
- A quantidade de instâncias necessárias para a transição entre as distribuições nas bases com mudanças abruptas e graduais rápidas não são tão grandes.

Logo, em ambos os casos o nível de dificuldade é parecido e influenciam de forma semelhante na precisão, diferentemente das mudanças graduais lentas.

4.3.3 Tempo de Execução

A Tabela 11 apresenta os tempos de execução das diferentes bases de dados. Nas bases artificiais com mudanças abruptas, o DDM obteve os melhores tempos em Stagger e SEA, ambos com 0,26s. O seu menor tempo foi na base LED, utilizando 1,05s. Esse padrão também se repete nas mudanças graduais rápidas e graduais lentas, com pequenas alterações: o melhor desempenho do DDM continua sendo na SEA, porém, agora de forma isolada. Na LED seu tempo continua sendo o pior. Uma maior quantidade de classes juntamente com uma maior quantidade de atributos e algum nível de ruído possivelmente contribui para o aumento do tempo de execução. Nas bases reais, o elevado número de instâncias, principalmente na Covertype e PokerHand, justificam o aumento considerável desses tempos quando comparados às bases artificiais.

Tabela 11: Tempo de execução do DDM com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	0,50±0,03	0,49±0,02	0,49±0,02	-
Hyperplane	0,44±0,02	0,40±0,02	0,40±0,02	-
LED	1,05±0,03	1,02±0,03	1,03±0,03	-
Mixed	0,28±0,02	0,28±0,02	0,27±0,02	-
RandomRBF	0,55±0,02	0,51±0,02	0,50±0,02	-
SEA	0,26±0,02	0,24±0,02	0,25±0,02	-
Sine	0,29±0,02	0,28±0,02	0,27±0,02	-
Stagger	0,26±0,02	0,26±0,01	0,28±0,02	-
Waveform	0,75±0,03	0,69±0,02	0,69±0,03	-
Covertype	-	-	-	100,46
Electricity	-	-	-	2,31
PokerHand	-	-	-	98,68
Média	0,49	0,46	0,46	67,15

4.3.4 Memória

Seguindo um padrão de comportamento semelhante aos resultados do tempo de execução, a Tabela 12 mostra a quantidade de memória em bytes utilizada por hora. Das bases artificiais, o DDM consome mais memória com a LED, enquanto na SEA seu uso é o mais reduzido. Já nas bases reais, o DDM obteve o maior e o menor consumo na Covertype e Electricity, respectivamente.

Tabela 12: Uso de memória do DDM em Bytes por hora com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	2,47±0,12	2,25±0,09	2,22±0,10	-
Hyperplane	1,75±0,08	1,49±0,08	1,49±0,08	-
LED	11,03±0,31	10,17±0,30	10,53±0,39	-
Mixed	0,77±0,05	0,73±0,05	0,73±0,04	-
RandomRBF	2,79±0,13	2,36±0,09	2,31±0,08	-
SEA	0,68±0,04	0,62±0,04	0,62±0,04	-
Sine	0,81±0,05	0,73±0,05	0,72±0,04	-
Stagger	0,68±0,04	0,68±0,04	0,72±0,04	-
Waveform	5,22±0,18	4,32±0,17	4,27±0,18	-
Covertype	-	-	-	1311,93
Electricity	-	-	-	8,06
PokerHand	-	-	-	584,10
Média	2,91	2,59	2,62	634,70

4.3.5 Precisão na Detecção das Mudanças e Alarmes Falsos

Para analisar a precisão na detecção das mudanças e a quantidade de alarmes falsos no DDM, os valores apresentados na Tabela 13 são levados em consideração. De todas as bases artificiais, o DDM mostrou-se mais propenso aos falsos positivos com a Hyperplane. Apesar de conseguir detectar as mudanças relativamente cedo nos casos abruptos e graduais rápidos, o método teve dificuldade para se adaptar às novas distribuições e continuou disparando detecções, mesmo após os pontos de mudanças (falsos positivos). LED foi a única base que o DDM não detectou falsos positivos, porém, apresentou um elevado número de falsos negativos, principalmente com mudanças graduais lentas.

A partir dos resultados apresentados, uma importante relação pode ser destacada: a rápida detecção das mudanças e a baixa quantidade de alarmes falsos estão diretamente ligadas a uma boa precisão, como pode ser visto nas bases Stagger e Mixed abruptas.

Com as bases reais, DDM apresentou o maior número de detecções na Covertype, seguido por Electricity e PokerHand. A quantidade de detecções juntamente com a quantidade de atributos é algo que parece estar relacionado ao uso de memória. Mesmo a base PokerHand possuindo mais instâncias que a Covertype, a memória consumida pelo DDM na segunda é quase quatro vezes maior, enfatizando a relação apresentada.

Tabela 13: Posição média das detecções, falsos positivos (FP), e falsos negativos (FN) do DDM com as bases artificiais

	PM1	PM2	PM3	PM4	FP
Agrawal_A	2056.75	4300.50	6241.00	8195.00	27
Agrawal_{GR}	2064.75	4242.50	6209.00	8155.75	36
Agrawal_{GL}	2074.71	4538.33	FN	8445.59	7
Hyperplane_A	2072.50	4074.25	6079.25	8072.50	240
Hyperplane_{GR}	2074.75	4091.50	6101.25	8064.75	252
Hyperplane_{GL}	2091.50	4481.25	6340.25	8382.00	86
LED_A	2776.92	4090.25	6101.25	FN	0
LED_{GR}	2676.11	4094.75	6105.75	FN	0
LED_{GL}	FN	FN	6209.00	FN	0
Mixed_A	2025.50	4025.75	6026.00	8023.75	12
Mixed_{GR}	2016.41	4018.75	6011.03	8013.00	2
Mixed_{GL}	2580.75	5154.50	7474.75	FN	79
RandomRBF_A	2298.25	FN	6459.25	8374.75	12
RandomRBF_{GR}	2297.25	FN	6426.25	8295.25	19
RandomRBF_{GL}	2626.50	FN	6760.00	8585.88	1
SEA_A	2566.50	4285.13	6161.25	8468.25	28
SEA_{GR}	2612.00	4323.25	6166.50	8552.75	20
SEA_{GL}	FN	4727.67	6477.22	FN	2
Sine_A	2031.00	4022.50	6039.75	8046.75	83
Sine_{GR}	FN	4027.25	6101.00	8030.50	43
Sine_{GL}	2431.25	FN	FN	8024.41	71
Stagger_A	2011.50	4010.50	6010.00	8010.75	3
Stagger_{GR}	FN	4112.50	FN	8026.50	74
Stagger_{GL}	2465.25	5420.25	FN	8081.05	84
Waveform_A	2056.75	4046.25	6290.75	8356.00	49
Waveform_{GR}	2060.25	4044.25	6263.50	FN	44
Waveform_{GL}	2873.24	FN	FN	FN	20

Tabela 14: Quantidade de detecções realizadas pelo DDM nas bases reais

	Covertype	Electricity	PokerHand
Detecções	1442	110	15

4.4 DOF

4.4.1 Otimizando Parâmetros

O método DoF possui dois tipos de parâmetros: o tamanho das janelas (w) que serão comparadas em busca de possíveis mudanças na distribuição; e o limite do desvio padrão (s) para que uma mudança seja acusada. Os valores utilizados pelos autores nos seus experimentos foram $w = 100$ e $s = 4.0$.

Fazendo uso do AG para selecionar os melhores parâmetros para as bases com mudanças abruptas, os valores escolhidos foram $w = 135$ e $s = 2{,}0$. De maneira geral, a diminuição no s permite que as reduções no grau de semelhança entre as janelas sejam notadas mais rapidamente, e assim a detecção ser acusada mais cedo.

Com as modificações, o DoF foi mais beneficiado na base Waveform, com um aumento de 6,57% na precisão. Já com a Mixed, teve uma queda de 3,06%, sendo a maior de todas. Uma das características da base Mixed é que as mudanças de conceito são simuladas com a inversão dos rótulos: tudo que foi aprendido pelo classificador base como

sendo positivo passa a ser negativo e vice versa. Isso permitirá que o novo conceito afete toda a generalização realizada até o momento, tornando mais fácil a sua detecção. Logo, o aumento da sensibilidade do DoF na base Mixed Abrupta, quando comparado ao DoF com os parâmetros padrões, aumenta o índice de falsos positivos sem trazer nenhum benefício evidente nas detecções das mudanças, justificando, dessa forma, a queda na sua precisão. Mesmo com essa baixa, os novos valores definidos pelo AG foram responsáveis por um aumento médio de 2,81% na precisão do DoF nas bases com mudanças abruptas.

Já nas bases com mudanças graduais rápidas, os parâmetros definidos foram: $w = 120$ e $s = 2{,}14$. É perceptível que os valores encontrados são semelhantes aos utilizados com as mudanças abruptas. Porém, como o período de transição aumenta, uma maior sensibilidade tende a trazer maiores benefícios. O DoF apresentou um maior aumento na base Sine, com crescimento de 14,34%. Esse ganho considerável pode ser justificado pelo fato de nenhuma das mudanças terem sido detectadas com os parâmetros padrões e, com os novos valores, todas passaram a ser. O DoF com a Mixed, novamente, sofreu a maior perda, com redução de 1,28%. A média geral, incluindo as perdas e os ganhos, foi de 4,42%.

Com as mudanças graduais lentas, os valores selecionados foram $w = 120$ e $s = 1{,}84$. O DoF obteve o maior aumento com a base Sine (13,85%), enquanto na RandomRBF, única base que teve queda de rendimento, apresentou uma perda de 1,50%. Os valores padrões mostraram-se bastante ineficientes, sobretudo na medida em que os períodos de transição vão aumentando. Assim, superando as médias gerais anteriores, a atual apresentou uma melhora de 5,72%.

Para as bases reais, foram definidos $w = 104$ e $s = 1{,}54$. Com o aumento na sensibilidade do DoF, seu desempenho foi mais beneficiado na base Covertype – que possui maiores mudanças na distribuição –, com um aumento de 12,24%. Já na PokerHand, cuja característica é a pouca variabilidade, o DoF sofreu uma redução de 2,08%, influenciada principalmente pelos falsos positivos. Com isso, o DoF nas bases reais obteve um aumento médio de 4,78% na sua precisão.

4.4.2 Precisão

Observando os valores apresentados na Tabela 16, o DoF nas bases abruptas obteve as maiores médias de precisão (77,33%), seguido das bases graduais rápidas (76,93%), graduais lentas (75,53%) e reais (69,21%). No geral, o DoF na base Stagger, com os três tipos de mudanças, obteve as maiores precisões. Stagger é composta por três atributos categóricos e duas classes. Dessa forma, sua simplicidade juntamente com a fácil detecção nas mudanças, faz com que os métodos obtenham uma fácil convergência, elevando suas precisão. Por fim, os conjuntos de dados no qual o DoF obteve os piores desempenhos foram RandomRBF e PokerHand.

Tabela 15: Precisão do DOF com 95% de confiança (parâmetros padrões)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	67,67±0,38	67,66±0,38	67,62±0,38	-
Hyperplane	73,22±0,50	73,23±0,50	73,19±0,50	-
LED	62,75±0,40	62,64±0,32	62,51±0,23	-
Mixed	87,83±0,18	85,50±0,63	74,46±0,80	-
RandomRBF	48,44±0,35	49,12±0,42	49,48±0,38	-
SEA	84,59±0,24	84,60±0,24	84,60±0,24	-
Sine	84,47±0,60	72,24±2,70	71,18±2,23	-
Stagger	92,50±0,25	88,92±0,66	76,62±0,11	-
Waveform	69,18±0,59	68,68±0,30	68,57±0,30	-
Covertype	-	-	-	67,01
Electricity	-	-	-	76,19
PokerHand	-	-	-	50,07
Média	77,10	76,90	75,26	64,16

Tabela 16: Precisão do DOF com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	71,34±0,35	70,96±0,42	70,82±0,38	-
Hyperplane	79,47±0,71	79,45±0,68	79,34±0,66	-
LED	68,01±0,27	67,75±0,39	66,94±0,40	-
Mixed	84,77±1,52	84,22±1,94	83,32±1,14	-
RandomRBF	48,10±0,37	48,22±0,34	47,98±0,36	-
SEA	84,96±0,25	84,88±0,26	84,89±0,25	-
Sine	86,25±0,96	86,58±0,95	85,03±0,81	-
Stagger	97,30±0,38	94,71±0,86	87,08±0,19	-
Waveform	75,75±0,47	75,63±0,45	74,35±0,43	-
Covertype	-	-	-	79,25
Electricity	-	-	-	80,38
PokerHand	-	-	-	47,99
Média	77,33	76,93	75,53	69,21

4.4.3 Tempo de Execução

De acordo com a Tabela 17, das bases artificiais, LED foi a que demandou um maior tempo para o DoF. Em seguida vieram Waveform, Agrawal e RandomRBF. Uma relação interessante é que as quatro bases que exigiram mais tempo do DoF também são as com menores precisões (Tabela 16). Como dito anteriormente, esses dois aspectos podem estar relacionados com a quantidade de atributos e classes das bases. Por outro lado, o DoF foi mais rápido com as bases SEA, Stagger, Sine e Mixed, respectivamente. Seguindo o padrão dos tempos que foram apresentados até então, o DoF demorou mais com a base real Covertype, e foi o mais rápido com a Electricity.

4.4.4 Memória

A maneira como os valores da Tabela 18 são distribuídos novamente tem uma forte correlação com os valores da Tabela 17: LED foi a base que requisitou maior consumo de memória do DoF, enquanto na SEA ele foi o mais econômico.

Uma informação apresentada que chama a atenção é o fato da média de uso de

Tabela 17: Tempo de execução do DOF com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	1,01±0,03	0,99±0,04	1,02±0,03	-
Hyperplane	0,89±0,03	0,81±0,03	0,84±0,03	-
LED	1,77±0,04	1,69±0,04	1,78±0,04	-
Mixed	,68±0,03	0,62±0,03	0,66±0,03	-
RandomRBF	1,00±0,03	0,91±0,03	0,94±0,03	-
SEA	0,64±0,03	0,58±0,02	0,60±0,03	-
Sine	,68±0,03	0,63±0,03	0,64±0,03	-
Stagger	0,66±0,03	0,61±0,03	0,64±0,02	-
Waveform	1,30±0,04	1,23±0,03	1,26±0,03	-
Covertype	-	-	-	140,76
Electricity	-	-	-	3,76
PokerHand	-	-	-	115,73
Média	0,96	0,90	0,93	86,75

memória do DoF nas bases abruptas (14,57%), ser maior que suas médias nas bases com mudanças graduais rápidas (12,68%) e graduais lentas (13,56%). Especificamente no caso do DoF, o tamanho da janela pode estar exercendo uma grande parcela de contribuição nestes resultados: a cada iteração do algoritmo, para cada instância presente na janela mais recente, o seu vizinho mais próximo será procurado na janela mais antiga, utilizando a distância euclidiana. Assim, mesmo pequenas diferenças nesse tamanho (135 nas bases abruptas vs. 120 nas graduais rápidas e lentas) serão suficientes para alterar a quantidade de memória utilizada.

Tabela 18: Uso de memória do DOF em Bytes por hora com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	13,61±0,42	12,35±0,43	13,14±0,32	-
Hyperplane	10,83±0,34	9,05±0,30	9,50±0,32	-
LED	46,45±0,93	41,33±0,84	45,21±0,88	-
Mixed	5,89±0,24	4,92±0,21	5,32±0,26	-
RandomRBF	13,15±0,37	11,07±0,38	11,68±0,34	-
SEA	5,07±0,20	4,17±0,17	4,38±0,18	-
Sine	5,82±0,24	4,90±0,21	5,04±0,21	-
Stagger	5,42±0,22	4,61±0,21	4,96±0,19	-
Waveform	24,89±0,70	21,68±0,55	22,81±0,59	-
Covertype	-	-	-	4859,07
Electricity	-	-	-	33,47
PokerHand	-	-	-	1239,36
Média	14,57	12,68	13,56	2043,97

4.4.5 Precisão na Detecção das Mudanças e Alarmes Falsos

Analisando os valores apresentados na Tabela 19, representando as bases artificiais, o DoF na base SEA apresentou a maior quantidade de falsos negativos. Já as bases Sine, Mixed e RandomRBF foram as únicas onde o DoF conseguiu detectar todas as mudanças, sendo mais eficiente na primeira. Na base Stagger com mudanças graduais lentas o DoF disparou a maior quantidade de falsos positivos. O longo período de transição entre os

diferentes contextos (500 instâncias) fez com que o DoF acusasse consecutivas detecções nesse espaço de tempo.

Nas bases reais, a maior sensibilidade do DoF mostrou-se bastante prejudicial quando aplicada ao PokerHand. Apesar de ser uma base com aparente estabilidade, o DoF detectou 709 mudanças. Observando as Tabelas 15 e 16, a queda de rendimento com os parâmetros ajustados pode ser justificada pela quantidade excessiva de detecções.

Tabela 19: Posição média das detecções, falsos positivos (FP), e falsos negativos (FN) do DOF com as bases artificiais

	PM1	PM2	PM3	PM4	FP
Agrawal_A	2338.06	FN	FN	FN	6
Agrawal_{GR}	2317.71	FN	FN	FN	4
Agrawal_{GL}	2708.11	FN	6882.58	FN	18
Hyperplane_A	2258.33	FN	FN	FN	8
Hyperplane_{GR}	2194.29	FN	FN	FN	6
Hyperplane_{GL}	2492.31	FN	6917.65	FN	17
LED_A	2369.21	4226.00	6363.24	FN	9
LED_{GR}	2413.33	4244.21	FN	FN	5
LED_{GL}	2625.00	FN	6512.00	FN	24
Mixed_A	2210.57	4327.00	6127.67	8315.38	9
Mixed_{GR}	2276.57	4112.00	6204.71	8140.65	10
Mixed_{GL}	2125.16	4200.00	6116.57	8156.67	21
RandomRBF_A	2194.50	4367.75	6297.25	8388.25	6
RandomRBF_{GR}	2202.00	4240.00	6216.00	8286.49	7
RandomRBF_{GL}	2325.00	4864.00	6333.00	8396.92	38
SEA_A	FN	FN	FN	FN	3
SEA_{GR}	FN	FN	FN	FN	4
SEA_{GL}	2865.00	FN	6768.75	FN	22
Sine_A	2184.87	4252.75	6260.91	8239.47	6
Sine_{GR}	2196.00	4239.00	6260.57	8149.09	4
Sine_{GL}	2096.47	4290.00	6366.49	8137.50	14
Stagger_A	2030.00	4344.25	6080.00	8330.00	10
Stagger_{GR}	FN	FN	6135.00	FN	49
Stagger_{GL}	FN	FN	6243.00	FN	80
Waveform_A	2269.00	FN	FN	FN	6
Waveform_{GR}	2124.00	FN	FN	FN	5
Waveform_{GL}	2306.25	FN	FN	FN	18

Tabela 20: Quantidade de detecções realizadas pelo DOF nas bases reais

	Covertype	Electricity	PokerHand
Detecções	280	26	709

4.5 ECDD

4.5.1 Otimizando Parâmetros

Originalmente no ECDD, seus autores disponibilizam três tipos de parametrização: a quantidade média de alarmes falsos por instâncias (ARL_o); o lambda (λ), que representa os pesos que serão atribuídos para as instâncias mais recentes em relação às mais antigas, influenciando nas detecções; e o limiar de alerta (w), que fará o método entrar em estado

de alerta (criando um classificador alternativo) quando houver os primeiros sinais de mudança na distribuição.

Entretanto, calcular os possíveis valores para o ARL_o enquanto o ECDD está ativo é bastante custoso, aumentando consideravelmente seu tempo de execução. Como solução, seus autores calcularam antecipadamente estes valores, considerando que λ=0.2. Assim, utilizaremos como parâmetros somente os ARL_o já calculados (100, 400 e 1000), considerando $\lambda = 0.2$, e w. Como no artigo do ECDD nenhum valor padrão foi definido para os parâmetros, para fins experimentais utilizamos $ARL_o = 100$ e $w = 0{,}5$.

Para as bases com mudanças abruptas, foram selecionados pelo AG $ARL_o = 400$ e $w = 0{,}25$. Com o aumento do ARL_o, o ECDD passa a ser menos sensível às detecções, ao contrário do que aconteceu com todos os outros métodos analisados até então. Com os novos parâmetros, o ECDD foi beneficiado em todas as bases, obtendo o maior aumento com a LED (17,75%) e o menor na Stagger (0,07%). A quantidade excessiva de alarmes falsos, quando utilizado o $ARL_o = 100$, ao que tudo indica estava prejudicando o desempenho do método com as constantes intervenções no classificador base.

Nas bases com mudanças graduais rápidas, os parâmetros definidos foram $ARL_o = 400$ e $w = 0{,}38$. Novamente foi na base LED onde o ECDD obteve o maior aumento de precisão (17,52%), enquanto Stagger foi a única onde o ECDD teve perda de rendimento (0,04%). De todas as bases, Stagger é a mais tolerante aos falsos positivos em função de sua rápida convergência, ou seja, o classificador base consegue se recuperar facilmente das detecções incorretas acusadas pelo ECDD. Por esse motivo, neste caso, as vantagens oferecidas pela maior sensibilidade – como a detecção mais rápida das mudanças – acaba influenciando mais na precisão do que as desvantagens.

Com as mudanças graduais lentas, os valores encontrados foram $ARL_o = 400$ e $w = 0{,}35$. É possível observar que em todos os tipos de mudanças analisadas até o momento, w vem sofrendo uma redução. Na teoria, isso significa que o ECDD entrará em estado de alerta mais cedo e, caso a mudança se confirme, o classificador alternativo estará com mais instâncias treinadas da nova distribuição (melhorando sua eficiência). Entretanto, quanto mais cedo o ECDD entrar em estado de alerta, maior é o risco de o classificador alternativo incluir no seu treinamento instâncias da distribuição antiga, sobretudo nos casos de mudanças graduais. Mantendo os padrões anteriores, o DoF com os novos parâmetros foi mais beneficiado na base LED, com um aumento de 17,09%. Já na base Stagger, apesar de também ter tido sua performance melhorada, o DoF obteve um aumento de somente 0,89%.

Por fim, nas bases reais, os parâmetros selecionados foram $ARL_o = 400$ e $w = 0{,}34$. Com os novos valores, o DoF na base PokerHand foi o mais beneficiada, com um aumento de performance de 3,22%. Com o $ARL_o = 100$, nesta mesma base, o DoF detectou uma quantidade bem superior de mudanças na distribuição (1821 detecções), enquanto com o

novo valor apenas uma única mudança foi acusada. Essa provavelmente é a principal causa na melhora do desempenho do método. Nas outras bases houve melhoras no desempenho do DoF de 0,72% (Electricity) e 0,71% (Covertype).

Tabela 21: Precisão do ECDD com 95% de confiança (parâmetros padrões)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	61,65±0,31	61,53±0,33	60,62±0,25	-
Hyperplane	70,88±0,41	70,59±0,40	68,83±0,41	-
LED	49,02±0,17	48,99±0,16	48,86±0,13	-
Mixed	81,96±0,48	81,04±0,46	76,46±0,37	-
RandomRBF	44,06±0,20	44,05±0,20	44,01±0,18	-
SEA	75,19±0,47	75,17±0,47	74,95±0,43	-
Sine	85,30±0,43	84,43±0,38	79,90±0,37	-
Stagger	99,14±0,10	97,91±0,11	93,56±0,11	-
Waveform	65,62±0,45	65,25±0,43	63,60±0,39	-
Covertype	-	-	-	88,74
Electricity	-	-	-	86,72
PokerHand	-	-	-	46,84
Média	70,31	69,88	67,87	74,10

Tabela 22: Precisão do ECDD com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	71,30±0,28	71,10±0,28	69,94±0,28	-
Hyperplane	83,32±0,33	83,09±0,31	81,54±0,33	-
LED	66,77±0,44	66,51±0,46	65,95±0,43	-
Mixed	89,01±0,25	88,12±0,20	84,06±0,23	-
RandomRBF	48,60±0,34	48,61±0,35	48,60±0,34	-
SEA	84,32±0,26	84,34±0,25	84,08±0,28	-
Sine	89,62±0,29	88,78±0,28	85,25±0,26	-
Stagger	99,21±0,09	97,87±0,12	94,45±0,15	-
Waveform	77,63±0,33	77,08±0,32	76,02±0,32	-
Covertype	-	-	-	89,45
Electricity	-	-	-	87,44
PokerHand	-	-	-	50,06
Média	78,86	78,39	76,65	75,65

4.5.2 Precisão

De acordo com os resultados apresentados na Tabela 22, dentre as bases artificias, o ECDD obteve os melhores e os piores desempenhos com as bases Stagger e RandomRBF, respectivamente. Já nas bases reais, os melhores resultados foram alcançados na Covertype e Electricity. Um fator curioso é que com a base RandomRBF o ECDD manteve praticamente a mesma precisão nos três tipos de mudanças. Como a sua precisão é sempre baixa nessa base, nos momentos de transição de contexto, independente da quantidade de instâncias que sejam necessárias, sua média de acertos e erros continuará baixa e seguindo o mesmo padrão de outros momentos. Dessa forma, o tipo de mudança não exerce muita influência no desempenho do método.

4.5.3 Tempo de Execução

A Tabela 23 mostra os tempos, em segundos, utilizados pelo ECDD e pelo NB nas diferentes bases de dados. Com LED e Waveform, foram constatados os piores resultados do ECDD nas bases artificiais. Por outro lado, com as bases Mixed, SEA, Sine e Stagger, o seu tempo de execução foi relativamente baixo. Por fim, nas bases reais, o melhor desempenho alcançado pelo ECDD foi na Electricity, com 2,56s, muito superior ao Covertype e ao Pokerhand, com 103,49s e 100,34s, respectivamente. O fato da mudança ser abrupta, gradual rápida ou gradual lenta não influenciou muito nos tempos. Essa informação pode ser constatada observando a pouca variação dos resultados dos métodos nos diferentes casos.

Tabela 23: Tempo de execução do ECDD com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	0,54±0,02	0,52±0,02	0,53±0,02	-
Hyperplane	0,48±0,02	0,47±0,02	0,51±0,02	-
LED	1,10±0,03	1,10±0,03	1,09±0,03	-
Mixed	0,32±0,02	0,31±0,02	0,32±0,02	-
RandomRBF	0,60±0,03	0,61±0,02	0,61±0,02	-
SEA	0,31±0,02	0,32±0,02	0,30±0,02	-
Sine	0,33±0,02	0,33±0,02	0,34±0,02	-
Stagger	0,32±0,02	0,33±0,02	0,33±0,02	-
Waveform	0,84±0,03	0,81±0,03	0,82±0,03	-
Covertype	-	-	-	103,49
Electricity	-	-	-	2,56
PokerHand	-	-	-	100,34
Média	0,54	0,53	0,54	68,80

4.5.4 Memória

A partir dos resultados apresentados na Tabela 24, nas bases reais, LED foi a que mais demandou memória do ECDD, seguida por Waveform e RandomRBF. Em contrapartida, o ECDD foi mais eficientes nas bases SEA, Stagger e Sine, respectivamente. Nas reais, o ECDD gastou a maior e a menor quantidade de memória nas bases Covertype e Electricity, respectivamente. Como observado no tempo de execução, o consumo da memória pelo ECDD parece não ter relação com os tipos das mudanças. Os fatores que mais devem contribuir para o aumento desses valores são as quantidades de atributos e classes das bases de dados.

4.5.5 Precisão na Detecção das Mudanças e Alarmes Falsos

Uma das coisas que mais chamam atenção na Tabela 25 é a quantidade de alarmes falsos. Apesar dos novos parâmetros terem reduzido o nível de sensibilidade do ECDD, ainda assim esse nível continua relativamente alto. Esse é principal motivo que fez com que quase todas as mudanças fossem detectadas. RandomRBF foi a única base que não teve

Tabela 24: Uso de memória do ECDD em Bytes por hora com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	2,85±0,11	2,76±0,10	2,80±0,10	-
Hyperplane	2,20±0,09	2,11±0,09	2,32±0,09	-
LED	12,01±0,31	11,98±0,32	11,85±0,30	-
Mixed	1,00±0,06	0,97±0,06	0,98±0,06	-
RandomRBF	3,31±0,16	3,28±0,13	3,33±0,12	-
SEA	0,91±0,06	0,91±0,05	0,88±0,06	-
Sine	1,02±0,06	1,04±0,06	1,07±0,07	-
Stagger	0,92±0,05	0,96±0,06	0,98±0,06	-
Waveform	6,12±0,17	5,90±0,23	6,00±0,17	-
Covertype	-	-	-	1399,27
Electricity	-	-	-	9,76
PokerHand	-	-	-	596,38
Média	3,37	3,32	3,36	668,47

nenhuma das suas mudanças detectadas e, consequentemente, foi aonde o ECDD apresentou o menor número de falsos positivos. Já com a base Hyperplane, o ECDD detectou o maior número de falsos positivos, porém, aumentou sua eficiência na identificação das mudanças. Nas bases reais (Tabela 26), o ECDD detectou o maior número de mudanças na Covertype (1446), enquanto na PokerHand apenas uma (1) foi indicada. Nesta última, por sinal, foi apresentada uma redução significativa na quantidade de detecções com os novos parâmetros, contribuindo para a sua melhora de desempenho.

4.6 EDDM

4.6.1 Otimizando Parâmetros

No EDDM é possível parametrizar três características: a quantidade mínima de erros (m) para que o EDDM comece a detectar possíveis mudanças; o α, que determinará quando o método entrará em estado de alerta; e o β, responsável por consumar a mudança de conceito. Durante seus experimentos, os autores utilizaram $m = 30$, $\alpha = 0{,}95$, e $\beta = 0{,}90$.

Utilizando o AG para selecionar os melhores parâmetros para as bases com mudanças abruptas, os valores foram $m = 174$, $\alpha = 0{,}90$, e $\beta = 0.87$. Com a redução dos valores de α e β, o EDDM passa a ser menos sensível às detecções. Com essas alterações, o EDDM foi mais beneficiado na base RandomRBF, com um ganho de 0,88%. O fato da RandomRBF ser uma base de difícil convergência para o classificador faz com que muitos erros sejam cometidos. Dessa forma, com os constantes erros, o EDDM fica com uma falsa impressão de que uma mudança está ocorrendo, gerando muitos falsos positivos. Com isso, a redução da sensibilidade beneficia diretamente essa base, que tem a menor precisão. Já na base Hyperplane foi aonde o EDDM sofreu a maior queda, com uma redução de 0,64% no seu desempenho.

Tabela 25: Posição média das detecções, falsos positivos (FP), e falsos negativos (FN) do ECDD com as bases artificiais

	PM1	PM2	PM3	PM4	FP
Agrawal_A	2026.25	4140.50	6129.00	8082.00	523
Agrawal_{GR}	2083.00	4230.50	6190.50	8134.75	526
Agrawal_{GL}	2221.25	4310.50	6367.50	8239.50	457
Hyperplane_A	2034.00	4030.50	6031.75	8024.25	608
Hyperplane_{GR}	2057.75	4168.00	6085.50	8092.50	580
Hyperplane_{GL}	2187.25	4297.75	6208.75	8147.75	561
LED_A	2350.75	4159.00	6144.25	8276.00	169
LED_{GR}	2395.75	4479.50	6420.75	8373.50	165
LED_{GL}	2766.25	4904.50	7058.75	8921.75	126
Mixed_A	2013.00	4013.00	6013.00	8013.00	437
Mixed_{GR}	2855.75	4873.50	6449.75	8579.75	319
Mixed_{GL}	3204.75	5163.00	7468.00	FN	241
RandomRBF_A	FN	FN	FN	FN	34
RandomRBF_{GR}	FN	FN	FN	FN	35
RandomRBF_{GL}	FN	FN	FN	FN	42
SEA_A	2217.50	4234.50	6143.75	8202.25	395
SEA_{GR}	2267.00	4243.00	6161.75	8292.75	390
SEA_{GL}	2354.25	4229.25	6214.00	8313.50	404
Sine_A	2010.25	4011.50	6013.75	8014.25	588
Sine_{GR}	3044.75	4442.00	6203.00	8583.75	429
Sine_{GL}	3219.00	4172.00	7300.50	FN	327
Stagger_A	2010.00	4010.00	6010.00	8004.25	13
Stagger_{GR}	3180.75	5516.50	6474.00	FN	134
Stagger_{GL}	2503.75	4796.50	6915.75	FN	232
Waveform_A	2031.00	4019.75	6041.25	8160.75	618
Waveform_{GR}	2272.50	4238.25	6214.75	8295.00	567
Waveform_{GL}	2300.25	4388.25	6251.50	8418.00	527

Tabela 26: Quantidade de detecções realizadas pelo ECDD nas bases reais

	Covertype	Electricity	PokerHand
Detecções	1446	113	1

Para as bases graduais rápidas, os parâmetros definidos foram: $m = 180$, $\alpha = 0{,}89$, e $\beta = 0{,}87$. Novamente a sensibilidade foi reduzida em relação aos parâmetros padrões, com a diminuição dos valores de α e β. O aumento de m também influencia nas detecções e nos alarmes falsos: quanto maior esse número, maior o intervalo mínimo entre uma detecção e outra e, consequentemente, menos detecções. Um m grande é particularmente interessante em bases graduais, pois reduz a quantidade de detecções repetidas nos períodos de transição. Com o ajuste dos parâmetros, o EDDM foi mais beneficiada, novamente, na RandomRBF, com um ganho de 0,89%. Na Hyperplane o EDDM também teve a maior queda (0,68%). Com a redução da sensibilidade, as mudanças do Hyperplane passaram a ser detectadas com um atraso muito maior, influenciando diretamente no seu desempenho.

Com as bases graduais lentas, foram encontrados pelo AG os seguintes parâmetros: $m = 156$, $\alpha = 0{,}90$, e $\beta = 0{,}87$. Mantendo o mesmo padrão dos outros casos, esses valores diminuem a sensibilidade e reduzem a quantidade de alarmes falsos do EDDM. Novamente, com os novos parâmetros o EDDM teve o maior aumento com a RandomRBF: 0,95%. Já na base Stagger obteve a maior queda, com uma redução de 0,36% na sua precisão.

Enquanto nos outros casos a redução da sensibilidade pareceu não ter muita influência na Stagger, com um período de transição maior foi diferente. A forma mais sutil de realizar as mudanças aumentou os casos de falsos negativos no EDDM, reduzindo sua precisão.

Finalmente, nas bases reais, os parâmetros utilizados foram $m = 43$, $\alpha = 0{,}94$, e $\beta = 0{,}92$. Com esses valores, apesar do método entrar em estado de alerta mais tarde, as detecções necessitarão de um quantidade um pouco menor de instâncias para ocorrer quando comparado ao EDDM com parâmetros padrões. Com esse pequeno aumento na sensibilidade, o desempenho do EDDM nas bases Electricity e PokerHand foi reduzido em 0,10% e 0,06%, respectivamente. Em contrapartida, seu desempenho cresceu na Covertype, com um aumento de 0,39%.

Tabela 27: Precisão do EDDM com 95% de confiança (parâmetros padrões)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	71,07±0,38	71,22±0,34	70,37±0,39	-
Hyperplane	83,86±0,28	83,86±0,28	82,41±0,31	-
LED	67,31±0,31	67,30±0,30	67,15±0,33	-
Mixed	88,86±0,28	88,93±0,22	85,18±0,23	-
RandomRBF	48,54±0,39	48,56±0,39	48,47±0,41	-
SEA	84,84±0,25	84,81±0,25	84,74±0,26	-
Sine	89,54±0,37	89,36±0,28	86,10±0,25	-
Stagger	98,52±0,05	98,31±0,06	95,08±0,15	-
Waveform	77,05±0,35	76,92±0,34	75,94±0,36	-
Covertype	-	-	-	85,01
Electricity	-	-	-	85,74
PokerHand	-	-	-	50,07
Média	78,84	78,81	77,27	73,61

Tabela 28: Precisão do EDDM com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	71,59±0,35	71,62±0,33	71,23±0,32	-
Hyperplane	83,22±0,35	83,18±0,34	82,07±0,38	-
LED	67,63±0,30	67,57±0,29	67,49±0,31	-
Mixed	88,83±0,26	88,80±0,21	85,02±0,21	-
RandomRBF	49,42±0,38	49,45±0,39	49,42±0,40	-
SEA	85,19±0,30	85,17±0,29	85,01±0,28	-
Sine	89,49±0,37	89,23±0,32	86,17±0,25	-
Stagger	98,52±0,05	98,32±0,06	94,72±0,13	-
Waveform	77,18±0,37	77,07±0,34	75,91±0,37	-
Covertype	-	-	-	85,40
Electricity	-	-	-	85,64
PokerHand	-	-	-	50,01
Média	79,01	78,93	77,45	73,68

4.6.2 Precisão

A Tabela 28 mostra as precisões alcançadas pelo EDDM nas diferentes bases de dados. Nas artificiais, o pior desempenho geral do EDDM foi na base RandomRBF, enquanto na Stagger alcançou seu melhor resultado. Já nas reais, o EDDM obteve o melhor e o pior rendimento com as bases Electricity e PokerHand, respectivamente.

4.6.3 Tempo de Execução

De acordo com os resultados apresentados na Tabela 29, RandomRBF foi a base artificial que o EDDM precisou de mais tempo, enquanto na SEA ele foi o mais rápido. Com a redução da sensibilidade, foi perceptível uma redução geral no tempo de execução. Com a redução na quantidade de detecções, menor será a quantidade de vezes que o método entrará em estado de alerta e, consequentemente, mais rápido será o processo. Nas bases reais, Covertype e PokerHand foram as que demandaram mais tempo do EDDM. Já na Electricity, o EDDM precisou de apenas 2,17s.

Tabela 29: Tempo de execução do EDDM com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	0,51±0,02	0,50±0,03	0,51±0,03	-
Hyperplane	0,40±0,02	0,40±0,02	0,42±0,02	-
LED	1,03±0,03	1,01±0,03	1,03±0,03	-
Mixed	0,28±0,02	0,27±0,02	0,28±0,02	-
RandomRBF	0,52±0,02	0,50±0,02	0,52±0,02	-
SEA	0,25±0,02	0,25±0,01	0,25±0,02	-
Sine	0,29±0,02	0,28±0,02	0,28±0,02	-
Stagger	0,26±0,02	0,26±0,01	0,27±0,02	-
Waveform	0,71±0,03	0,69±0,03	0,70±0,03	-
Covertype	-	-	-	99,79
Electricity	-	-	-	2,17
PokerHand	-	-	-	99,16
Média	0,47	0,46	0,47	67,02

4.6.4 Memória

Seguindo um padrão semelhante aos resultados apresentados do tempo de execução, a Tabela 30 mostra a quantidade de memória utilizada pelo EDDM. RandomRBF foi a base artificial que o EDDM mais utilizou memória, enquanto na SEA foi onde ele mais economizou.

A redução da sensibilidade também foi benéfica à quantidade de memória utilizada. Menos detecções significa menos classificadores sendo treinados em paralelo nos estados de alerta. Já nas bases reais, o EDDM precisou de mais memória na Covertype e menos com a Electricity.

4.6.5 Precisão na Detecção das Mudanças e Alarmes Falsos

Observando os valores da Tabela 31, é possível constatar que o EDDM detectou o maior número de falsos positivos na base Hyperplane. Já a RandomRBF foi a única que o EDDM não detectou falsos positivos. Em contrapartida, foi identificado também um grande número de falsos negativos.

Comparando com os resultados apresentados nos parâmetros padrões, um dos motivos que levaram o EDDM na base RandomRBF ter um ganho considerável na precisão

Tabela 30: Uso de memória do EDDM em Bytes por hora com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	2,41±0,11	2,31±0,13	2,42±0,12	-
Hyperplane	1,51±0,07	1,47±0,08	1,57±0,08	-
LED	10,50±0,32	10,06±0,26	10,45±0,30	-
Mixed	0,76±0,05	0,73±0,05	0,74±0,04	-
RandomRBF	2,50±0,11	2,35±0,09	2,48±0,10	-
SEA	0,64±0,04	0,61±0,04	0,63±0,04	-
Sine	0,78±0,05	0,74±0,05	0,74±0,05	-
Stagger	0,67±0,04	0,67±0,04	0,69±0,05	-
Waveform	4,48±0,19	4,16±0,18	4,41±0,19	-
Covertype	-	-	-	1283,55
Electricity	-	-	-	7,16
PokerHand	-	-	-	574,10
Média	2,69	2,57	2,68	621,60

com os parâmetros ajustados foi a quantidade de falsos positivos. Como dito anteriormente, falsos positivos em bases de difícil convergência tendem a ter um grande impacto negativo no resultado final. Com o ajuste dos parâmetros, apesar do aumento de falsos positivos, a redução dos falsos negativos acabou tendo um impacto maior na precisão. Por fim, nas bases reais, o EDDM detectou um maior número de mudanças na Covertype, enquanto na PokerHand ele foi mais estável.

4.7 PL

4.7.1 Otimizando Parâmetros

Basicamente o PL conta com dois tipos de parametrização: o tamanho da janela (w) que o classificador reativo terá; e o limiar (θ), responsável por definir quando uma mudança de conceito acontecerá. Originalmente, para esses parâmetros, os autores utilizam diversos valores. Para fins experimentais selecionamos $w = 12$ e $\theta = 0{,}2$ – que foram um dos valores utilizados pelos seus autores – para usar como padrão.

Nas bases artificiais com mudanças abruptas, os valores selecionados pelo AG foram $w = 66$ e $\theta = 0{,}13$. Comparando com os valores utilizados por padrão, apesar da diminuição do θ, com o aumento de w a detecção necessitará de mais sequências de acertos e erros para ocorrer. Isso acontece porque θ é proporcional ao tamanho de w e, para que a comparação seja feita de maneira adequada, os valores devem ser analisados conjuntamente. Com a redução da sensibilidade, o PL na base LED foi o mais beneficiado, com um ganho de 12,16% na precisão. Já na Stagger, único caso com queda de desempenho, o PL sofreu uma redução de 0,74%.

Utilizando as bases com mudanças graduais rápidas, os parâmetros definidos foram: $w = 67$ e $\theta = 0{,}16$. Com o aumento no θ e o w sendo praticamente o mesmo, quando

Tabela 31: Posição média das detecções, falsos positivos (FP), e falsos negativos (FN) do EDDM com as bases artificiais

	PM1	PM2	PM3	PM4	FP
Agrawal_A	2207.50	FN	FN	FN	12
Agrawal_{GR}	2222.00	FN	FN	FN	8
Agrawal_{GL}	2192.25	FN	FN	FN	13
Hyperplane_A	2201.75	4146.00	6168.75	8228.00	230
Hyperplane_{GR}	2202.00	4150.50	6201.25	8242.00	201
Hyperplane_{GL}	2159.50	4293.25	6394.00	8293.50	175
LED_A	FN	4440.25	6301.75	FN	10
LED_{GR}	FN	4437.25	6320.25	FN	9
LED_{GL}	FN	4453.50	6386.58	FN	11
Mixed_A	2068.50	4063.75	6063.50	8065.50	53
Mixed_{GR}	2082.50	4064.75	6062.75	8063.50	26
Mixed_{GL}	2868.25	4617.43	6476.00	FN	53
RandomRBF_A	FN	FN	FN	8686.36	0
RandomRBF_{GR}	FN	FN	FN	8520.97	0
RandomRBF_{GL}	FN	4655.76	7088.75	8797.19	0
SEA_A	3141.76	4654.12	6303.00	8822.50	30
SEA_{GR}	3170.26	4617.25	6433.00	8918.00	22
SEA_{GL}	FN	4629.50	6404.00	FN	39
Sine_A	2039.75	4046.75	6095.25	8107.50	153
Sine_{GR}	2156.84	4063.00	6136.75	8134.50	89
Sine_{GL}	2767.50	FN	6155.00	8084.29	86
Stagger_A	2042.00	4055.00	6039.75	8042.50	0
Stagger_{GR}	2038.75	4035.50	6017.00	8009.46	4
Stagger_{GL}	3029.25	4568.00	FN	8096.18	88
Waveform_A	2150.50	4111.75	6633.25	FN	127
Waveform_{GR}	2182.50	4110.25	6707.25	FN	108
Waveform_{GL}	2336.25	4240.25	FN	FN	73

Tabela 32: Quantidade de detecções realizadas pelo EDDM nas bases reais

	Covertype	Electricity	PokerHand
Detecções	1392	109	10

comparado aos valores utilizados nas bases abruptas, o EDDM tende a ter um aumento na sensibilidade. Novamente o PL na base LED teve o maior aumento na precisão (12,83%) e na Stagger perda de 0,39%. Com os parâmetros ajustados, apesar das mudanças de conceito levarem mais instâncias para serem detectadas, os falsos positivos foram reduzidos consideravelmente. Dessa forma, as bases mais difíceis – na qual os classificadores necessitam de várias instâncias para se recuperarem dos falsos positivos – tiveram os maiores aumentos, como é o caso da LED e RandomRBF.

Para os casos com mudanças graduais lentas, os valores dos parâmetros foram $w = 117$ e $\theta = 0{,}14$. Assim, o PL fica ainda menos sensível do que nos casos anteriores. Analisando as diferenças das precisões com os parâmetros padrões e com os parâmetros ajustados, o mesmo padrão das bases com mudanças abruptas e graduais rápidas permanece: PL com LED obteve o maior aumento (13,19%) e PL com Stagger foi o único com redução na precisão (0,34%). Como Stagger é uma base na qual os falsos positivos não a prejudicam tanto por conta da sua rápida recuperação, um PL mais sensível é preferível.

Por fim, com as bases reais, foram definidos $w = 68$ e $\theta = 0{,}14$, valores semelhantes

aos das bases abruptas e graduais rápidas. Entretanto, PokerHand foi a única base beneficiada com os novos parâmetros, com um ganho de 0,34%. Covertype e Electricity tiveram uma perda de 4,48% e 3,52%, respectivamente.

No geral, os novos valores fizeram com que o PL nas bases reais tivesse uma queda média de 2,55%. Um dos motivos que levaram a esse baixo rendimento está relacionado com a forma que os parâmetros finais foram definidos: calculando a média dos parâmetros encontrados pelo AG para cada base. Apesar dessa metodologia evitar sobreajuste nas bases, existe um risco dos valores encontrados serem bastante diferentes uns dos outros. Assim, o meio termo encontrado pela média pode não ser o mais adequado para satisfazer todos os métodos. Em contrapartida, os experimentos mostraram que essa é uma situação que não acontece com muita frequência, sobretudo quando as bases são agrupadas pelas características.

Tabela 33: Precisão do PL com 95% de confiança (parâmetros padrões)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	65,15±0,37	64,97±0,37	63,81±0,39	-
Hyperplane	79,24±0,54	79,03±0,52	77,27±0,51	-
LED	56,35±1,41	56,27±1,38	55,36±1,33	-
Mixed	88,08±0,47	87,29±0,46	82,72±0,37	-
RandomRBF	36,14±0,39	36,12±0,38	36,01±0,37	-
SEA	83,51±0,45	83,46±0,45	83,30±0,46	-
Sine	89,86±0,32	89,04±0,33	85,02±0,30	-
Stagger	99,12±0,08	98,41±0,08	95,39±0,10	-
Waveform	75,00±0,45	74,73±0,43	73,23±0,42	-
Covertype	-	-	-	88,25
Electricity	-	-	-	87,84
PokerHand	-	-	-	46,28
Média	74,72	74,37	72,46	74,12

Tabela 34: Precisão do PL com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	71,79±0,34	72,13±0,32	71,68±0,36	-
Hyperplane	84,61±0,25	84,47±0,24	82,87±0,30	-
LED	68,51±0,48	69,10±0,27	68,55±0,26	-
Mixed	89,42±0,21	89,04±0,19	85,23±0,19	-
RandomRBF	43,85±0,32	45,23±0,32	47,23±0,36	-
SEA	85,55±0,27	85,03±0,25	84,75±0,23	-
Sine	89,89±0,28	89,47±0,30	86,25±0,30	-
Stagger	98,38±0,08	98,02±0,09	95,05±0,10	-
Waveform	78,32±0,34	77,84±0,33	76,45±0,33	-
Covertype	-	-	-	83,77
Electricity	-	-	-	84,32
PokerHand	-	-	-	46,62
Média	78,92	78,93	77,56	71,57

4.7.2 Precisão

Observando os valores apresentados na Tabela 34, nas bases artificiais, o PL obteve os piores rendimentos com as bases RandomRBF, LED e Agrawal, respectivamente. Já

em Stagger, Sine e Mixed obteve suas maiores precisões. Nas bases reais, no PokerHand o PL teve seu pior resultado. Apesar desta ser uma base mais difícil do que as outras, a queda de rendimento do PL se deve também à quantidade de falsos positivos detectados, mesmo depois da redução da sensibilidade. Nas bases Electricity e Covertype o PL teve rendimentos próximos, sendo levemente superior na primeira.

4.7.3 Tempo de Execução

De acordo com a Tabela 35, LED e Waveform foram as bases onde o PL necessitou de um maior tempo. Já Stagger, Mixed e Sine foram onde o PL alcançou seus melhores resultados. Nas bases reais, o PL foi mais rápido e mais lento, respectivamente, na Electricity e Covertype. Com os novos parâmetros, o PL teve seu tempo aumentado em 100% dos casos.

O principal motivo para esses acréscimos está relacionado com o aumento de w. Além do classificador reativo passar a armazenar mais instâncias, a quantidade de erros do classificador estável para que uma mudança seja detectada também será maior. Dessa forma, haverá um aumento natural na manipulação das estruturas de dados responsáveis por controlar essas ações e, como consequência, refletirá no tempo de execução.

Tabela 35: Tempo de execução do PL com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	0,89±0,03	0,87±0,03	1,09±0,03	-
Hyperplane	0,74±0,03	0,74±0,03	0,91±0,03	-
LED	1,95±0,04	1,95±0,04	2,30±0,04	-
Mixed	0,45±0,02	0,46±0,03	0,57±0,03	-
RandomRBF	0,95±0,04	0,95±0,03	1,15±0,03	-
SEA	0,40±0,02	0,39±0,02	0,50±0,02	-
Sine	0,46±0,03	0,47±0,02	0,57±0,03	-
Stagger	0,39±0,02	0,41±0,02	0,52±0,02	-
Waveform	1,36±0,04	1,38±0,04	1,63±0,04	-
Covertype	-	-	-	160,11
Electricity	-	-	-	3,64
PokerHand	-	-	-	116,35
Média	0,84	0,85	1,03	93,37

4.7.4 Memória

Possuindo várias semelhanças com a Tabela 35, a Tabela 36 mostra a quantidade de memória utilizada pelo PL nas diferentes bases de dados. No geral, das bases artificiais, LED foi a que mais demandou memória do PL, seguida por Waveform e RandomRBF. Os casos em que o PL necessitou de menos memória foi com SEA e Stagger. Já nas reais, Electricity e Covertype foram as que exigiram o menor e o maior consumo do PL. Da mesma forma como aconteceu anteriormente, a mudança nos parâmetros fez com que o consumo de memória do PL aumentasse, em todos os casos. Novamente, o aumento no

valor de w é o principal responsável, pois várias informações serão armazenadas por um tempo maior.

Tabela 36: Uso de memória do PL em Bytes por hora com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	7,61±0,25	7,64±0,23	12,42±0,33	-
Hyperplane	5,34±0,18	5,40±0,18	8,83±0,26	-
LED	39,84±0,70	40,64±0,73	59,25±0,87	-
Mixed	2,03±0,11	2,11±0,14	3,48±0,17	-
RandomRBF	8,34±0,29	8,48±0,24	12,97±0,33	-
SEA	1,59±0,10	1,57±0,09	2,71±0,12	-
Sine	2,09±0,11	2,15±0,11	3,45±0,15	-
Stagger	1,69±0,10	1,79±0,10	2,99±0,14	-
Waveform	17,25±0,42	17,75±0,44	27,41±0,52	-
Covertype	-	-	-	4067,24
Electricity	-	-	-	21,30
PokerHand	-	-	-	880,01
Média	9,53	9,73	14,83	1656,18

4.7.5 Precisão na Detecção das Mudanças e Alarmes Falsos

A Tabela 37 mostra as posições médias das detecções e a quantidade de falsos positivos negativos do PL. Como destacado anteriormente, quanto menor a quantidade de falsos positivos, maior a probabilidade de falsos negativos; e quanto mais falsos positivos mais rápida as detecções acontecerão e menor a probabilidade de falsos negativos. Essa relação pode ser controlada através da sensibilidade do método, com o objetivo de encontrar um meio termo.

Para ratificar essas observações, Agrawal foi a base com maior número de falsos positivos detectados pelo PL, porém, foi nela também que o PL detectou as mudanças mais rapidamente. Já na SEA, onde o PL encontrou a menor quantidade de falsos positivos, foi também onde ele encontrou a maior quantidade de falsos negativos. Finalmente, nas bases reais, a quantidade excessiva de detecções realizadas pelo PL em PokerHand e Covertype influenciaram diretamente na queda do seu desempenho quando comparado ao PL com parâmetros padrões.

4.8 STEPD

4.8.1 Otimizando Parâmetros

Três características são passíveis de parametrização no STEPD: o tamanho da janela (w) para que as w instâncias mais recentes sejam comparadas com as restantes; o limiar de alerta (α_w), que funciona para o método como uma suspeita de que uma mudança está prester a acontecer; e o limiar de detecção (α_d) que, quando ultrapassado, acusará uma

Tabela 37: Posição média das detecções, falsos positivos (FP), e falsos negativos (FN) do PL com as bases artificiais

	PM1	PM2	PM3	PM4	FP
Agrawal_A	2037.25	4049.00	6048.00	8056.25	2037
Agrawal_{GR}	2048.75	4054.25	6063.75	8066.25	1330
Agrawal_{GL}	2078.00	4115.25	6126.50	8144.50	685
Hyperplane_A	2057.75	4056.00	6053.75	8056.00	690
Hyperplane_{GR}	2079.25	4065.25	6062.25	8070.75	477
Hyperplane_{GL}	2112.75	4155.00	6117.25	8129.50	333
LED_A	2126.25	4060.25	6056.25	8169.00	170
LED_{GR}	2316.00	4094.75	6085.00	FN	17
LED_{GL}	2551.54	4244.75	6159.25	FN	2
Mixed_A	2041.75	4041.75	6041.75	8040.50	72
Mixed_{GR}	2046.75	4042.25	6044.50	8044.75	31
Mixed_{GL}	2077.75	4070.50	6075.25	8064.25	56
RandomRBF_A	2087.25	4085.50	6110.00	8076.50	1942
RandomRBF_{GR}	2161.00	4155.25	6185.25	8110.75	1132
RandomRBF_{GL}	2319.75	4247.50	6321.25	8280.75	549
SEA_A	FN	4317.30	6111.75	8365.45	5
SEA_{GR}	FN	FN	6403.95	FN	0
SEA_{GL}	FN	FN	6771.82	FN	0
Sine_A	2042.25	4044.25	6044.75	8042.00	57
Sine_{GR}	2045.50	4046.50	6046.50	8045.00	22
Sine_{GL}	2081.50	4093.25	6064.25	8061.75	45
Stagger_A	2054.00	4053.25	6050.00	8053.50	10
Stagger_{GR}	2063.25	4057.50	6058.00	8060.50	4
Stagger_{GL}	2112.50	4108.00	6101.00	8099.25	24
Waveform_A	2027.75	4036.50	6140.00	8084.25	182
Waveform_{GR}	2026.50	4044.25	6367.37	8193.75	56
Waveform_{GL}	3302.82	FN	FN	FN	32

Tabela 38: Quantidade de detecções realizadas pelo PL nas bases reais

	Covertype	Electricity	PokerHand
Detecções	1269	102	2497

mudança na distribuição nos dados. Para fins experimentais, os valores utilizados foram w = 100, α_w = 0,05 e α_d = 0,02.

Nas bases artificiais com mudanças abruptas, a média dos parâmetros selecionados pelo AG foram: w = 70, α_w = 0,06 e α_d = 0,05. Com esses novos parâmetros, o STEPD passa a ser mais sensível às mudanças. Um dos principais responsáveis pela maior sensibilidade é o α_d. O seu aumento faz com que o teste estatístico utilizado para detectar as mudanças seja menos rigoroso e, consequentemente, realize mais detecções.

Com essas modificações o STEPD com a base Hyperplane foi o mais beneficiado, com um ganho de 0,79%. Já o STEPD com a LED foi o mais prejudicada, com uma perda de 5,90%. Uma provável explicação para essa perda está na quantidade de falsos positivos: com a troca dos parâmetros, esse número cresceu consideravelmente. Entretanto, das nove bases abruptas, somente em duas o STEPD teve queda de desempenho.

Para as bases com mudanças graduais rápidas, os parâmetros definidos foram w = 153, α_w = 0,07 e α_d = 0,05. Neste caso, apesar do aumento de α_d – em relação aos valores padrões – indicar uma menor rigidez no teste estatístico para detecção de

mudanças, o aumento de w fará com que esses testes sejam mais confiáveis por conta da maior quantidade de instâncias sendo analisadas. Em contrapartida, o valor de w implica diretamente no intervalo mínimo entre uma detecção e outra: quanto maior ele for, maior será esse intervalo. Com os novos parâmetros, o STEPD na base Agrawal foi o mais beneficiado, com um aumento de 0,49% na precisão. Já o STEPD com a LED, novamente, obteve o maior prejuízo, com uma perda de 3,72%.

Com as mudanças graduais lentas, os valores encotrados foram $w = 170$, $\alpha_w = 0{,}06$ e $\alpha_d = 0{,}04$. É possível observar que à medida que os intervalos entre as mudanças vão ficando maiores, maiores também vão sendo os valores de w selecionados pelo AG. De certa forma, um w grande em situações com longos períodos de mudança evita disparos de detecções, sobretudo quando a sensibilidade é aumentada. Dos casos analisadas com as mudanças dos parâmetros, o que obteve um maior ganho foi o STEPD na base Stagger, com 1,09%. Já o STEPD com LED foi o mais prejudicado, com uma perda de 2,31%.

Por fim, nas bases reais, os parâmetros escolhidos foram $w = 21$, $\alpha_w = 0{,}06$ e $\alpha_d = 0{,}04$. Comparando com os valores selecionados com as mudanças graduais lentas, a redução de w indica detecções de mudanças com menos precisão e em curtos espaços de tempo. Isso beneficia principalmente o STEPD nas bases Covertype e Electricity, pois estas possuem uma quantidade razoável de mudanças. Dessa forma, ambos os casos tiveram um aumento de 3,50% e 2,99%, respectivamente. Já o STEPD na PokerHand não teve a sua precisão alterada com os novos valores.

Tabela 39: Precisão do STEPD com 95% de confiança (parâmetros padrões)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	71,90±0,33	71,81±0,32	71,20±0,34	-
Hyperplane	82,82±0,39	82,71±0,37	81,40±0,46	-
LED	68,02±0,31	67,78±0,32	66,21±0,34	-
Mixed	89,96±0,20	89,46±0,18	84,68±0,23	-
RandomRBF	49,56±0,39	49,57±0,39	49,54±0,39	-
SEA	84,60±0,24	84,60±0,24	84,60±0,24	-
Sine	90,17±0,35	89,69±0,30	85,31±0,28	-
Stagger	99,14±0,09	98,06±0,08	93,46±0,31	-
Waveform	78,19±0,35	77,78±0,34	76,02±0,33	-
Covertype	-	-	-	82,12
Electricity	-	-	-	81,71
PokerHand	-	-	-	50,07
Média	79,37	79,05	76,94	71,30

4.8.2 Precisão

Analisando os resultados apresentados na Tabela 40, Stagger foi a base artificial onde o STEPD obteve o melhor desempenho geral. Já na base RandomRBF o STEPD obteve os resultados mais baixos. É importante ressaltar que os valores apresentados pelo STEPD nas bases de dados são proporcionais aos seus níveis de dificuldade, motivo pelo qual

Tabela 40: Precisão do STEPD com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	72,56±0,36	72,30±0,34	71,22±0,34	-
Hyperplane	83,61±0,32	83,06±0,36	81,87±0,35	-
LED	62,12±1,22	64,06±0,76	63,90±0,64	-
Mixed	90,14±0,20	89,32±0,17	84,10±0,20	-
RandomRBF	48,48±0,42	48,65±0,42	48,92±0,42	-
SEA	85,01±0,23	84,83±0,24	84,71±0,23	-
Sine	90,35±0,34	89,47±0,28	85,16±0,31	-
Stagger	99,19±0,08	97,93±0,07	94,55±0,18	-
Waveform	78,37±0,34	77,87±0,35	76,24±0,34	-
Covertype	-	-	-	85,62
Electricity	-	-	-	84,70
PokerHand	-	-	-	50,07
Média	78,87	78,61	76,74	73,46

algumas delas sempre aparecem com uma precisão baixa. Nas bases reais, a maior média do STEPD foi obtida em Covertype (85,62%), e a menor em PokerHand (50,07%).

4.8.3 Tempo de Execução

De acordo com os tempos apresentados na Tabela 41, das bases artificiais, LED foi a que requisitou maior tempo do STEPD. Por outro lado, com SEA e Stagger o STEPD obteve os menores tempos, com a primeira apresentando uma pequena vantagem em relação à segunda. A modificação dos parâmetros não influenciou muito nos tempos de execução do STEPD nas bases artificiais. Essa afirmação pode ser justificada pelo fato de que a maior diferença foi de apenas 0,09%, na base LED Abrupta. Já nas bases reais, o maior ganho no tempo foi obtido pelo STEPD através da base Covertype (13,86%).

Tabela 41: Tempo de execução do STEPD com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	0,54±0,02	0,60±0,02	0,59±0,03	-
Hyperplane	0,45±0,02	0,50±0,02	0,51±0,02	-
LED	1,06±0,03	1,15±0,04	1,16±0,03	-
Mixed	0,33±0,02	0,38±0,02	0,42±0,02	-
RandomRBF	0,56±0,02	0,62±0,02	0,63±0,02	-
SEA	0,30±0,02	0,35±0,02	0,37±0,02	-
Sine	0,32±0,02	0,37±0,02	0,39±0,02	-
Stagger	0,30±0,02	0,37±0,02	0,38±0,02	-
Waveform	0,72±0,03	0,77±0,03	0,79±0,03	-
Covertype	-	-	-	102,60
Electricity	-	-	-	2,31
PokerHand	-	-	-	101,30
Média	0,51	0,57	0,58	68,74

4.8.4 Memória

Observando os valores da Tabela 42, LED foi a base artificial onde o STEPD teve o maior consumo de memória. SEA e Stagger foram as menos custosas para o STEPD,

respectivamente. Nas bases reais, Covertype e Electricity demandaram o maior e o menor consumo do STEPD, respectivamente.

Como dito anteriormente, a diferença nos consumos de memória entre as bases artificiais e reais se dá, principalmente, pela quantidade de instâncias que cada uma apresenta. A base Electricity, com 45.312 instâncias, é a que mais se aproxima das 10.000 instâncias das bases artificiais e, como consequência, é onde o STEPD apresenta um consumo mais parecido com o ocorrido nas bases artificiais.

Tabela 42: Uso de memória do STEPD em Bytes por hora com 95% de confiança (parâmetros ajustados)

	Abrupta	**Gradual Rápida**	**Gradual Lenta**	**Real**
Agrawal	2,63±0,10	3,31±0,12	3,33±0,15	-
Hyperplane	1,78±0,09	2,38±0,11	2,41±0,11	-
LED	10,62±0,28	13,10±0,52	13,15±0,41	-
Mixed	1,00±0,06	1,37±0,08	1,52±0,09	-
RandomRBF	2,71±0,10	3,59±0,15	3,62±0,14	-
SEA	0,85±0,05	1,21±0,07	1,32±0,08	-
Sine	0,96±0,05	1,34±0,07	1,41±0,08	-
Stagger	0,90±0,05	1,28±0,07	1,38±0,07	-
Waveform	4,35±0,14	5,31±0,22	5,46±0,23	-
Covertype	-	-	-	1382,22
Electricity	-	-	-	8,01
PokerHand	-	-	-	573,07
Média	2,87	3,65	3,73	654,43

4.8.5 Precisão na Detecção das Mudanças e Alarmes Falsos

A Tabela 43 apresenta as posições médias das detecções de mudanças bem como a quantidade de falsos positivos e falsos negativos. SEA foi a base onde o STEPD apresentou a maior quantidade de falsos negativos e a menor quantidade de falsos positivos. Com os parâmetros padrões, nenhuma mudança e nenhum falso positivo tinham sido detectados. Com o ajuste, uma mudança passou a ser detectada no SEA abrupto e gradual rápido. Para melhorar esse quadro seria necessário utilizar parâmetros ainda mais sensíveis.

Já a base LED, mesmo causando o maior número de falsos positivos – indicando uma alta sensibilidade –, não foi suficiente para o STEPD detectar todas as mudanças. Finalmente, nas bases reais, o aumento na sensibilidade do STEPD fez com que a quantidade de detecções também aumentasse, nos três casos. Essa modificação foi benéfica uma vez que, nas bases reais, o STEPD obteve um ganho médio de desempenho de 2,16%, como apresentado na Tabela 40.

4.9 Comparação entre Otimizações

A Tabela 45 mostra a média dos ganhos e perdas dos métodos nos diferentes tipos de mudanças bem como a média dessas médias. É possível observar que o ECDD foi,

Tabela 43: Posição média das detecções, falsos positivos (FP), e falsos negativos (FN) do STEPD com as bases artificiais

	PM1	PM2	PM3	PM4	FP
Agrawal_A	2033.50	4316.36	FN	8232.00	11
Agrawal_{GR}	2057.25	4406.22	6484.67	8302.29	8
Agrawal_{GL}	FN	FN	FN	FN	16
Hyperplane_A	2046.50	4050.00	6062.25	8073.50	145
Hyperplane_{GR}	2074.75	4107.50	6119.25	8096.00	123
Hyperplane_{GL}	2117.25	4352.50	6419.50	8274.50	121
LED_A	2519.50	4084.25	6096.50	8906.75	390
LED_{GR}	2904.57	4155.00	6226.25	FN	248
LED_{GL}	FN	4275.50	6450.50	FN	191
Mixed_A	2018.25	4018.75	6018.50	8017.00	2
Mixed_{GR}	2017.50	4024.50	6038.75	8024.50	59
Mixed_{GL}	2246.75	4334.00	6290.25	8296.75	206
RandomRBF_A	2471.71	FN	6799.12	8514.84	19
RandomRBF_{GR}	2529.43	FN	6568.95	FN	11
RandomRBF_{GL}	FN	FN	FN	FN	9
SEA_A	FN	FN	6528.65	FN	2
SEA_{GR}	FN	FN	6694.69	FN	2
SEA_{GL}	FN	FN	FN	FN	0
Sine_A	2015.25	4017.00	6022.00	8021.50	4
Sine_{GR}	2034.50	4019.50	6044.00	8043.50	46
Sine_{GL}	2165.75	4478.75	6342.00	8368.75	190
Stagger_A	2010.50	4010.75	6010.00	8010.25	0
Stagger_{GR}	2174.25	4194.25	6265.75	8260.25	149
Stagger_{GL}	2154.25	4423.75	6087.50	8521.25	233
Waveform_A	2031.00	4030.00	6142.70	FN	8
Waveform_{GR}	2055.25	4038.75	FN	FN	1
Waveform_{GL}	2953.00	FN	FN	FN	25

Tabela 44: Quantidade de detecções realizadas pelo STEPD nas bases reais

	Covertype	Electricity	PokerHand
Detecções	1352	95	1

com grande margem de diferença, o método mais beneficiado com a otimização, obtendo um ganho geral de 6,85%. Como dito anteriormente, a excessiva quantidade de alarmes falsos detectadas com os parâmetros padrões fizeram com que a precisão do método fosse bastante prejudicada. Com o ajuste dos parâmetros utilizando o AG, o ECDD passou a ser menos sensível e, consequentemente, sua precisão aumentou. Outra importante observação é com relação às bases reais: o aumento nela foi bastante inferior quando comparado aos aumentos das bases artificiais. Uma possível justificativa para esse fato pode ser por conta da quantidade de mudanças. As bases reais, no geral, apresentam um número de mudanças de conceito bem superior às quatro definidas nas bases artificiais. Com isso, um ECDD bastante sensível tende a ser melhor aproveitado.

EDDM e STEPD foram os métodos que obtiverem um menor ganho com a otimização dos parâmetros. No caso do EDDM, é possível que os parâmetros definidos por padrão por seus autores obtenham, de fato, um bom resultado em diferentes situações. Ainda assim o AG obteve uma melhora geral na precisão de 0,14%. Já no STEPD, o principal responsável pelas perdas médias do desempenho foram as precisões discrepantes obtidas

Tabela 45: Média dos ganhos e perdas das precisões dos métodos após a otimização dos parâmetros

	ADWIN	DDM	DOF	ECDD	EDDM	PL	STEPD
Abrupta	0,12	0,45	0,22	**8,55**	0,16	4,21	-0,50
Gradual Rápida	0,03	0,36	0,04	**8,50**	0,13	4,56	-0,44
Gradual Lenta	0,16	0,17	0,26	**8,79**	0,18	5,11	-0,19
Real	2,46	2,19	**5,05**	1,55	0,08	-2,55	2,16
μ	0,69	0,79	1,39	**6,85**	0,14	2,83	0,26

especificamente pelas bases RandomRBF e LED, puxando a média para baixo. Se as precisões obtidas nas 30 bases selecionadas forem analisadas individualmente, é possível identificar ganho em 19 delas.

Por fim, observando as médias gerais da Tabela 45, é possível destacar o bom desempenho do AG na otimização dos parâmetros dos métodos de detecção de mudanças. Dessa forma, afirmamos que o seu uso pode ser uma boa alternativa para solução de problemas desta natureza.

4.10 Considerações Finais

Este capítulo descreveu a metodologia utilizada para os experimentos e as respectivas configurações dos métodos e bases de dados; comparou os resultados apresentados pelos métodos com parâmetros definidos por padrão e parâmetros ajustados; e, por fim, discutiu de forma empírica os resultados dos métodos em termos de precisão, tempo de execução, uso de memória, posição média de detecção, falsos positivos, e falsos negativos.

O Capítulo 5 faz uma análise comparativa de forma empírica e estatística entre os métodos, de acordo com os seus respectivos resultados.

5 Comparação Empírica e Estatística

Este capítulo apresenta uma análise comparativa, predominantemente empírica, dos resultados apresentados pelos diferentes métodos de detecção de mudança. As características aqui analisadas são a precisão de cada método, os tempos de execução, e os respectivos consumos de memória.

De acordo com as recomendações apresentadas em (DEMSAR, 2006), os resultados foram comparados estatisticamente por meio do teste de Friedman. Para esse teste, a hipótese nula indica que todos os métodos são estatisticamente equivalentes. Caso seja rejeitada, a hipótese alternativa diz que existe diferença estatística em um ou mais métodos, mas não especifica em quais. Para realizar essa tarefa, o uso de um pós-teste é recomendado. Como a intenção é comparar todos os métodos entre si, em pares, optamos pela utilização do Nemenyi.

5.1 Precisão

De acordo com a Tabela 46, nas bases com mudanças abruptas, DDM foi o método com melhor precisão média, seguido pelo ADWIN e EDDM. Já o que obteve a pior precisão foi o DoF. Um dos principais motivos para a baixa precisão do DoF está associado com a grande quantidade de falsos negativos bem como o seu alto tempo médio na detecção das mudanças.

Já nas bases com mudanças graduais rápidas, DDM continua sendo o método com melhor precisão média. Apesar de não obter as maiores médias em todas as bases, DDM apresenta uma regularidade que sustenta a sua liderança. Além disso, suas detecções de mudanças são realizadas rapidamente, com quantidade aceitável de falsos negativos e positivos. O método que obteve a pior precisão média foi novamente o DoF, com uma diferença de 2,48% para o DDM.

No caso das bases com mudanças graduais lentas, PL foi o método com melhor desempenho, porém, com uma superioridade de apenas 0,02% para o DDM, que foi o segundo colocado. Comparando ambos, é possível observar que o PL, apesar de apresentar uma quantidade maior de falsos positivos, consegue detectar as mudanças graduais lentas de forma mais rápida que o DDM. Esse deve ser o principal motivo para sua ligeira vantagem.

Por outro lado, DoF, juntamente com ECDD, obtiveram as piores precisões médias. De todos os métodos, ECDD foi o que apresentou a maior quantidade de falsos positivos. Essa característica prejudicou o seu desempenho nas bases artificiais, fazendo dele o segundo método com pior precisão.

Tabela 46: Precisão média dos métodos em porcentagem (%) com 95% de confiança nas bases artificiais

	NB	ADWIN	DDM	DOF	ECDD	EDDM	PL	STEPD
Agrawal_A	67,67±0,38	73,04±0,31	72,88±0,31	71,34±0,35	71,30±0,28	71,59±0,35	71,79±0,34	72,56±0,36
Hyp_A	73,22±0,50	83,69±0,30	84,04±0,31	79,47±0,71	83,32±0,33	83,22±0,35	84,61±0,25	83,61±0,32
LED_A	62,53±0,23	68,10±0,25	69,01±0,26	68,01±0,27	66,77±0,44	67,63±0,30	68,51±0,48	62,12±1,22
Mixed_A	64,35±0,12	89,55±0,21	89,96±0,23	84,77±1,52	89,01±0,25	88,83±0,26	89,42±0,21	90,14±0,20
RBF_A	49,64±0,39	48,69±0,35	48,92±0,37	48,10±0,37	48,60±0,34	49,42±0,38	43,85±0,32	48,48±0,42
SEA_A	84,60±0,24	84,73±0,23	85,56±0,24	84,96±0,25	84,32±0,26	85,19±0,30	85,55±0,27	85,01±0,23
Sine_A	64,10±0,22	89,86±0,36	89,83±0,40	86,25±0,96	89,62±0,29	89,49±0,37	89,89±0,28	90,35±0,34
Stagger_A	76,62±0,11	98,65±0,10	99,22±0,08	97,30±0,38	99,21±0,09	98,52±0,05	98,38±0,08	99,19±0,08
Wave_A	68,69±0,30	78,08±0,35	78,10±0,40	75,75±0,47	77,63±0,33	77,18±0,37	78,32±0,34	78,37±0,34
μ	67,94	79,38	**79,72**	77,33	78,86	79,01	78,92	78,87
Agrawal_{GR}	67,66±0,38	72,82±0,31	72,84±0,31	70,96±0,42	71,10±0,28	71,62±0,33	72,13±0,32	72,30±0,34
Hyp_{GR}	73,23±0,50	83,35±0,34	83,97±0,30	79,45±0,68	83,09±0,31	83,18±0,34	84,47±0,24	83,06±0,36
LED_{GR}	62,54±0,23	67,98±0,24	68,72±0,25	67,75±0,39	66,51±0,46	67,57±0,29	69,10±0,27	64,06±0,76
Mixed_{GR}	64,39±0,12	88,97±0,19	89,46±0,17	84,22±1,94	88,12±0,20	88,80±0,21	89,04±0,19	89,32±0,17
RBF_{GR}	49,64±0,39	48,72±0,35	48,81±0,36	48,22±0,34	48,61±0,35	49,45±0,39	45,23±0,32	48,65±0,42
SEA_{GR}	84,60±0,24	84,72±0,23	85,47±0,24	84,88±0,26	84,34±0,25	85,17±0,29	85,03±0,25	84,83±0,24
Sine_{GR}	64,14±0,22	89,33±0,34	89,62±0,30	86,58±0,95	88,78±0,28	89,23±0,32	89,47±0,30	89,47±0,28
Stagger_{GR}	76,64±0,11	98,10±0,08	97,98±0,09	94,71±0,86	97,87±0,12	98,32±0,06	98,02±0,09	97,93±0,07
Wave_{GR}	68,68±0,30	77,66±0,33	77,85±0,37	75,63±0,45	77,08±0,32	77,07±0,34	77,84±0,33	77,87±0,35
μ	67,95	79,07	**79,41**	76,93	78,39	78,93	78,93	78,61
Agrawal_{GL}	67,62±0,38	71,84±0,32	71,62±0,33	70,82±0,38	69,94±0,28	71,23±0,32	71,68±0,36	71,22±0,34
Hyp_{GL}	73,19±0,50	82,05±0,35	82,16±0,38	79,34±0,66	81,54±0,33	82,07±0,38	82,87±0,30	81,87±0,35
LED_{GL}	62,51±0,23	67,47±0,25	67,93±0,27	66,94±0,40	65,95±0,43	67,49±0,31	68,55±0,26	63,90±0,64
Mixed_{GL}	64,31±0,11	84,69±0,22	84,83±0,20	83,32±1,14	84,06±0,23	85,02±0,21	85,23±0,19	84,10±0,20
RBF_{GL}	49,60±0,40	48,75±0,39	49,04±0,35	47,98±0,36	48,60±0,34	49,42±0,40	47,23±0,36	48,92±0,42
SEA_{GL}	84,60±0,24	84,69±0,23	85,13±0,26	84,89±0,25	84,08±0,28	85,01±0,28	84,75±0,23	84,71±0,23
Sine_{GL}	64,14±0,22	85,57±0,26	86,16±0,30	85,03±0,81	85,25±0,26	86,17±0,25	86,25±0,30	85,16±0,31
Stagger_{GL}	76,62±0,11	94,33±0,16	94,76±0,12	87,08±0,19	94,45±0,15	94,72±0,13	95,05±0,10	94,55±0,18
Wave_{GL}	68,57±0,30	76,15±0,34	76,21±0,34	74,35±0,43	76,02±0,32	75,91±0,37	76,45±0,33	76,24±0,34
μ	67,91	77,28	77,54	75,53	76,65	77,45	**77,56**	76,74
Cov	67,01	86,20	89,13	79,25	89,45	85,40	83,77	85,62
Elec	76,19	84,96	86,74	80,38	87,44	85,64	84,32	84,70
Poker	50,07	50,07	49,97	47,99	50,06	50,01	46,62	50,07
μ	64,42	73,74	75,28	69,21	**75,65**	73,68	71,57	73,46
Rank	6,97	3,67	**2,27**	6,37	5,40	3,97	3,32	4,05

Nas bases reais, a alta sensibilidade do ECDD fez dele o melhor método. Diferente das bases artificiais – na qual apenas quatro mudanças de conceito acontecem em 10.000 instâncias – nas bases reais, sobretudo na Covertype e Electricity, esse número é bem maior. Assim, o ECDD alcança as melhores médias. Em contrapartida, DoF continua sendo o método com as piores precisões em quase todas as bases testadas.

Para ter uma visão mais apropriada das diferenças entre os métodos foi executado o teste de Friedman, cuja hipótese nula diria que todos os métodos são estatisticamente semelhantes. Tendo essa hipótese rejeitada, foi utilizado o pós-teste de Nemenyi que apontou quais métodos apresentaram superioridade estatística. Os resultados podem ser resumidos da seguinte maneira:

- DDM e PL apresentaram superioridade estatística em relação a ECDD e DoF; e

- ADWIN, EDDM, e STEPD apresentaram superioridade estatística em relação ao DoF.

O gráfico abaixo (Fig. 1) apresenta de forma mais completa como a precisão de cada método se posicionou em relação a todos os outros métodos testados.

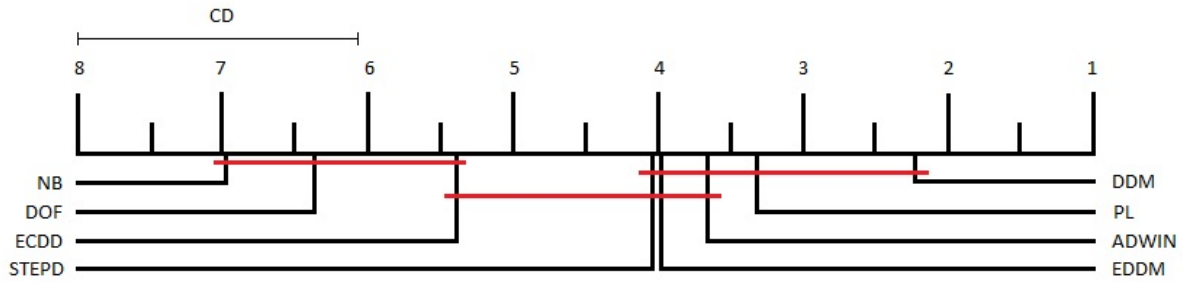

Figura 1: Comparação entre as precisões dos métodos utilizando o teste de Nemenyi com 95% de confiança.

5.2 Tempo de Execução

Observando os tempos consumidos pelos métodos nas bases com mudanças abruptas apresentados na Tabela 47, os detectores mais rápidos foram ADWIN e EDDM, ambos com média de 0,47s. DoF foi o mais lento, com média de 0,96s. Um dos motivos para o tempo comparativamente mais elevado do DoF está relacionado com os constantes cálculos da distância euclidiana, necessários para definir os vizinhos mais próximos a cada instância.

Já nas bases com as mudanças graduais rápidas, os métodos mais rápidos foram DDM e EDDM, ambos com 0,46s. A simplicidade desses métodos, associada à utilização de estrutura de dados comparativamente mais simples, contribuem para um bom tempo de execução. Com média de 0,90, DoF novamente foi o método que precisou de mais tempo para processar as 10.000 instâncias.

Nas bases com mudanças graduais lentas, DDM continuou sendo o método mais rápido, apresentando um tempo médio de 0,46s. Em seguida vem ADWIN e EDDM, os dois apresentando média de 0,47s. O mais lento neste cenário foi o PL, com tempo médio de 1,03s. O seu tempo proporcionalmente mais elevado pode ser justificado pelo fato do PL possuir dois classificadores sendo treinados em paralelo: um contendo somente as w instâncias mais recentes e o outro contendo todas as instâncias até o último ponto de mudança.

Mantendo um padrão semelhante àquele observado nas bases artificiais abruptas, nas bases reais o método mais rápido foi o ADWIN, seguido por EDDM e DDM. Os que precisaram de mais tempo foram o PL e o DoF, respectivamente. Quando comparados aos tempos das bases artificiais, os tempos das bases reais são bem superiores. O principal

Tabela 47: Tempo de execução médio em segundos (s) com 95% de confiança nas bases artificiais

	NB	ADWIN	DDM	DOF	ECDD	EDDM	PL	STEPD
Agrawal_A	0,32±0,02	0,48±0,02	0,50±0,03	1,01±0,03	0,54±0,02	0,51±0,02	0,89±0,03	0,54±0,02
Hyp_A	0,24±0,02	0,41±0,02	0,44±0,02	0,89±0,03	0,48±0,02	0,40±0,02	0,74±0,03	0,45±0,02
LED_A	0,75±0,03	1,02±0,03	1,05±0,03	1,77±0,04	1,10±0,03	1,03±0,03	1,95±0,04	1,06±0,03
Mixed_A	0,16±0,01	0,28±0,02	0,28±0,02	0,68±0,03	0,32±0,02	0,28±0,02	0,45±0,02	0,33±0,02
RBF_A	0,31±0,02	0,53±0,02	0,55±0,02	1,00±0,03	0,60±0,03	0,52±0,02	0,95±0,04	0,56±0,02
SEA_A	0,13±0,01	0,27±0,02	0,26±0,02	0,64±0,03	0,31±0,02	0,25±0,02	0,40±0,02	0,30±0,02
Sine_A	0,16±0,01	0,29±0,02	0,29±0,02	0,68±0,03	0,33±0,02	0,29±0,02	0,46±0,03	0,32±0,02
Stagger_A	0,14±0,01	0,27±0,02	0,26±0,02	0,66±0,03	0,32±0,02	0,26±0,02	0,39±0,02	0,30±0,02
Wave_A	0,45±0,02	0,68±0,02	0,75±0,03	1,30±0,04	0,84±0,03	0,71±0,03	1,36±0,04	0,72±0,03
μ	**0,30**	0,47	0,49	0,96	0,54	0,47	0,84	0,51
Agrawal_{GR}	0,32±0,01	0,49±0,02	0,49±0,02	0,99±0,04	0,52±0,02	0,50±0,03	0,87±0,03	0,60±0,02
Hyp_{GR}	0,23±0,01	0,40±0,02	0,40±0,02	0,81±0,03	0,47±0,02	0,40±0,02	0,74±0,03	0,50±0,02
LED_{GR}	0,74±0,02	1,02±0,03	1,02±0,03	1,69±0,04	1,10±0,03	1,01±0,03	1,95±0,04	1,15±0,04
Mixed_{GR}	0,15±0,01	0,28±0,02	0,28±0,02	0,62±0,03	0,31±0,02	0,27±0,02	0,46±0,03	0,38±0,02
RBF_{GR}	0,30±0,02	0,52±0,02	0,51±0,02	0,91±0,03	0,61±0,02	0,50±0,02	0,95±0,03	0,62±0,02
SEA_{GR}	0,13±0,01	0,26±0,02	0,24±0,02	0,58±0,02	0,32±0,02	0,25±0,01	0,39±0,02	0,35±0,02
Sine_{GR}	0,15±0,01	0,28±0,02	0,28±0,02	0,63±0,03	0,33±0,02	0,28±0,02	0,47±0,02	0,37±0,02
Stagger_{GR}	0,15±0,01	0,28±0,02	0,26±0,01	0,61±0,03	0,33±0,02	0,26±0,01	0,41±0,02	0,37±0,02
Wave_{GR}	0,46±0,02	0,68±0,02	0,69±0,02	1,23±0,03	0,81±0,03	0,69±0,03	1,38±0,04	0,77±0,03
μ	**0,29**	0,47	0,46	0,90	0,53	0,46	0,85	0,57
Agrawal_{GL}	0,32±0,02	0,50±0,02	0,49±0,02	1,02±0,03	0,53±0,02	0,51±0,03	1,09±0,03	0,59±0,03
Hyp_{GL}	0,23±0,02	0,39±0,02	0,40±0,02	0,84±0,03	0,51±0,02	0,42±0,02	0,91±0,03	0,51±0,02
LED_{GL}	0,75±0,02	1,00±0,03	1,03±0,03	1,78±0,04	1,09±0,03	1,03±0,03	2,30±0,04	1,16±0,03
Mixed_{GL}	0,16±0,01	0,28±0,02	0,27±0,02	0,66±0,03	0,32±0,02	0,28±0,02	0,57±0,03	0,42±0,02
RBF_{GL}	0,30±0,02	0,52±0,02	0,50±0,02	0,94±0,03	0,61±0,02	0,52±0,02	1,15±0,03	0,63±0,02
SEA_{GL}	0,13±0,01	0,27±0,02	0,25±0,02	0,60±0,03	0,30±0,02	0,25±0,02	0,50±0,02	0,37±0,02
Sine_{GL}	0,15±0,01	0,28±0,02	0,27±0,02	0,64±0,03	0,34±0,02	0,28±0,02	0,57±0,03	0,39±0,02
Stagger_{GL}	0,16±0,01	0,29±0,02	0,28±0,02	0,64±0,02	0,33±0,02	0,27±0,02	0,52±0,02	0,38±0,02
Wave_{GL}	0,45±0,02	0,68±0,02	0,69±0,03	1,26±0,03	0,82±0,03	0,70±0,03	1,63±0,04	0,79±0,03
μ	**0,29**	0,47	0,46	0,93	0,54	0,47	1,03	0,58
Cov	98,08	99,71	100,46	140,76	103,49	99,74	160,11	102,60
Elec	1,37	2,30	2,31	3,76	2,56	2,17	3,64	2,31
Poker	64,49	95,99	98,68	115,73	100,34	99,16	116,35	101,30
μ	**54,65**	66,00	67,15	86,75	68,80	67,02	93,37	68,74
Rank	**1,00**	3,00	3,08	7,60	5,40	2,97	7,40	5,55

motivo para esse aumento significativo é a quantidade de instâncias das bases artificiais: 581.012 na CoverType, 45.312 na Electricity, e 1.000.000 na PokerHand.

Utilizando o teste de Friedman nos resultados apresentados na Tabela 47, a hipótese nula de que todos os métodos têm tempos semelhantes é rejeitada. Dando continuidade, o Nemenyi apontou as seguintes diferenças:

- EDDM, ADWIN, e DDM foram estatisticamente mais rápidos do que ECDD, STEPD, PL, e DoF;
- ECDD foi estatisticamente mais rápido do que PL e DoF; e
- STEPD foi estatisticamente mais rápido do que o DoF.

O gráfico abaixo (Fig. 2) apresenta de forma mais completa como o tempo de execução de cada método se posicionou em relação a todos os outros métodos testados.

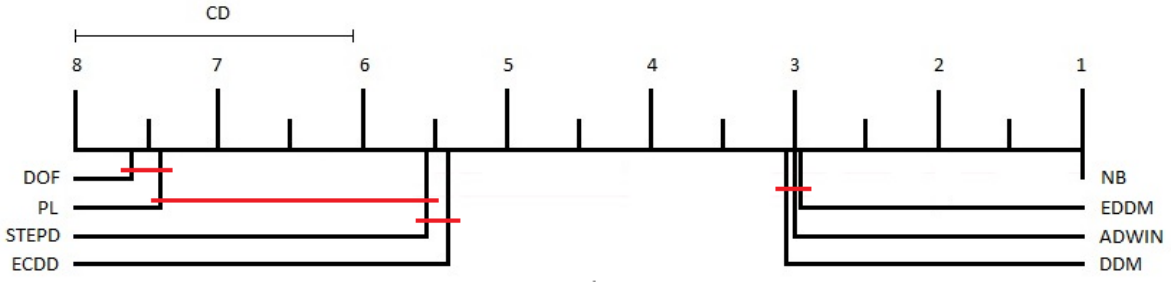

Figura 2: Comparação entre os tempos de execução dos métodos utilizando o teste de Nemenyi com 95% de confiança.

5.3 Memória

A partir dos valores da Tabela 48, nas bases com mudanças abruptas, ADWIN foi o método mais econômico em termos de uso de memória, seguido de perto por EDDM e DDM. Como citado anteriormente, a simplicidade desses métodos tem uma grande influência nestes valores. Por outro lado, DoF e PL apresentaram os maiores consumos de memória, nesta ordem.

Nas bases com mudanças graduais rápidas, EDDM, DDM e ADWIN obtiveram as menores médias de consumo de memória, também nesta ordem. Já DoF e PL ficaram com os piores desempenhos.

Já nas bases com mudanças graduais lentas, o mesmo padrão de comportamento dos outros casos é observado: DDM, ADWIN e EDDM foram os métodos mais econômicos em termos de memória, enquanto PL e DoF foram os mais custosos. Um dos motivos para o aumento de consumo de memória do PL, além do classificador que roda em paralelo como dito anteriormente, está ligado ao tamanho da janela (W). Quanto maior esse valor, mais instâncias precisarão ser armazenadas para que uma mudança de conceito seja detectada e, como consequência, mais memória será consumida.

Por fim, nas bases reais, ADWIN foi o método mais eficiente no consumo de memória. Em seguida vieram EDDM, DDM e ECDD. Já os mais custosos foram DoF e PL.

Analisando os resultados da perspectiva das bases de dados, foi necessário usar uma quantidade elevada de memória para processar todas as instâncias de Covertype. Já em Electricity, o consumo foi muito próximo aos das bases artificiais. Seguindo a mesma lógica utilizada na análise do tempo de execução, a primeira variável que irá influenciar no

Tabela 48: Uso da memória em bytes por hora com 95% de confiança nas bases artificiais

	NB	ADWIN	DDM	DOF	ECDD	EDDM	PL	STEPD
$Agrawal_A$	0,86±0,05	2,28±0,08	2,47±0,12	13,61±0,42	2,85±0,11	2,41±0,11	7,61±0,25	2,63±0,10
Hyp_A	0,44±0,03	1,61±0,08	1,75±0,08	10,83±0,34	2,20±0,09	1,51±0,07	5,34±0,18	1,78±0,09
LED_A	5,86±0,19	10,20±0,26	11,03±0,31	46,45±0,93	12,01±0,31	10,50±0,32	39,84±0,70	10,62±0,28
$Mixed_A$	0,13±0,01	0,84±0,05	0,77±0,05	5,89±0,24	1,00±0,06	0,76±0,05	2,03±0,11	1,00±0,06
RBF_A	0,80±0,04	2,50±0,10	2,79±0,13	13,15±0,37	3,31±0,16	2,50±0,11	8,34±0,29	2,71±0,10
SEA_A	0,09±0,01	0,75±0,05	0,68±0,04	5,07±0,20	0,91±0,06	0,64±0,04	1,59±0,10	0,85±0,05
$Sine_A$	0,13±0,01	0,84±0,05	0,81±0,05	5,82±0,24	1,02±0,06	0,78±0,05	2,09±0,11	0,96±0,05
$Stagger_A$	0,12±0,01	0,80±0,05	0,68±0,04	5,42±0,22	0,92±0,05	0,67±0,04	1,69±0,10	0,90±0,05
$Wave_A$	1,79±0,07	4,03±0,13	5,22±0,18	24,89±0,70	6,12±0,17	4,48±0,19	17,25±0,42	4,35±0,14
μ	**1,14**	2,65	2,91	14,57	3,37	2,69	9,53	2,87
$Agrawal_{GR}$	0,87±0,04	2,35±0,07	2,25±0,09	12,35±0,43	2,76±0,10	2,31±0,13	7,64±0,23	3,31±0,12
Hyp_{GR}	0,42±0,02	1,56±0,08	1,49±0,08	9,05±0,30	2,11±0,09	1,47±0,08	5,40±0,18	2,38±0,11
LED_{GR}	5,85±0,18	10,16±0,26	10,17±0,30	41,33±0,84	11,98±0,32	10,06±0,26	40,64±0,73	13,10±0,52
$Mixed_{GR}$	0,13±0,01	0,84±0,06	0,73±0,05	4,92±0,21	0,97±0,06	0,73±0,05	2,11±0,14	1,37±0,08
RBF_{GR}	0,79±0,04	2,47±0,09	2,36±0,09	11,07±0,38	3,28±0,13	2,35±0,09	8,48±0,24	3,59±0,15
SEA_{GR}	0,09±0,01	0,74±0,04	0,62±0,04	4,17±0,17	0,91±0,05	0,61±0,04	1,57±0,09	1,21±0,07
$Sine_{GR}$	0,12±0,01	0,84±0,05	0,73±0,05	4,90±0,21	1,04±0,06	0,74±0,05	2,15±0,11	1,34±0,07
$Stagger_{GR}$	0,12±0,01	0,80±0,05	0,68±0,04	4,61±0,21	0,96±0,06	0,67±0,04	1,79±0,10	1,28±0,07
$Wave_{GR}$	1,82±0,07	4,04±0,13	4,32±0,17	21,68±0,55	5,90±0,23	4,16±0,18	17,75±0,44	5,31±0,22
μ	**1,13**	2,64	2,59	12,68	3,32	2,57	9,73	3,65
$Agrawal_{GL}$	0,87±0,05	2,39±0,09	2,22±0,10	13,14±0,32	2,80±0,10	2,42±0,12	12,42±0,33	3,33±0,15
Hyp_{GL}	0,43±0,03	1,54±0,08	1,49±0,08	9,50±0,32	2,32±0,09	1,57±0,08	8,83±0,26	2,41±0,11
LED_{GL}	5,88±0,18	10,04±0,26	10,53±0,39	45,21±0,88	11,85±0,30	10,45±0,30	59,25±0,87	13,15±0,41
$Mixed_{GL}$	0,13±0,01	0,83±0,05	0,73±0,04	5,32±0,26	0,98±0,06	0,74±0,04	3,48±0,17	1,52±0,09
RBF_{GL}	0,80±0,04	2,47±0,09	2,31±0,08	11,68±0,34	3,33±0,12	2,48±0,10	12,97±0,33	3,62±0,14
SEA_{GL}	0,09±0,01	0,76±0,05	0,62±0,04	4,38±0,18	0,88±0,06	0,63±0,04	2,71±0,12	1,32±0,08
$Sine_{GL}$	0,13±0,01	0,83±0,05	0,72±0,04	5,04±0,21	1,07±0,07	0,74±0,05	3,45±0,15	1,41±0,08
$Stagger_{GL}$	0,13±0,01	0,84±0,05	0,72±0,04	4,96±0,19	0,98±0,06	0,69±0,05	2,99±0,14	1,38±0,07
$Wave_{GL}$	1,77±0,08	4,16±0,13	4,27±0,18	22,81±0,59	6,00±0,17	4,41±0,19	27,41±0,52	5,46±0,23
μ	**1,14**	2,65	2,62	13,56	3,36	2,68	14,83	3,73
Cov	1410,18	1290,31	1311,93	4859,07	1399,27	1283,55	4067,24	1382,22
Elec	2,10	8,04	8,06	33,47	9,76	7,16	21,30	8,01
Poker	201,19	540,72	584,10	1239,36	596,38	574,10	880,01	573,07
μ	**537,82**	613,02	634,70	2043,97	668,47	621,60	1656,18	654,43
Rank	**1,17**	3,25	3,22	7,90	5,42	2,70	7,10	5,25

consumo de memória será a quantidade de instâncias, como é possível observar na Tabela 48.

É importante ressaltar que o consumo de memória pelos métodos possui várias semelhanças com seu tempo de execução, e que muitas das justificativas utilizadas para um também se aplica ao outro.

De acordo com os resultados apontados pelo teste de Friedman, a hipótese nula de que todos os métodos possuem consumo de memória equivalente é rejeitada. Seguindo com o pós-teste de Nemenyi, os resultados encontrados foram:

- EDDM, DDM, e ADWIN estatisticamente usaram menos memória do que STEPD, ECDD, PL e DoF; e
- STEPD e ECDD estatisticamente usaram menos memória do que o DoF.

O gráfico abaixo (Fig. 3) apresenta de forma mais completa como a quantidade de memória utilizada por cada método se posicionou em relação a todos os outros métodos testados.

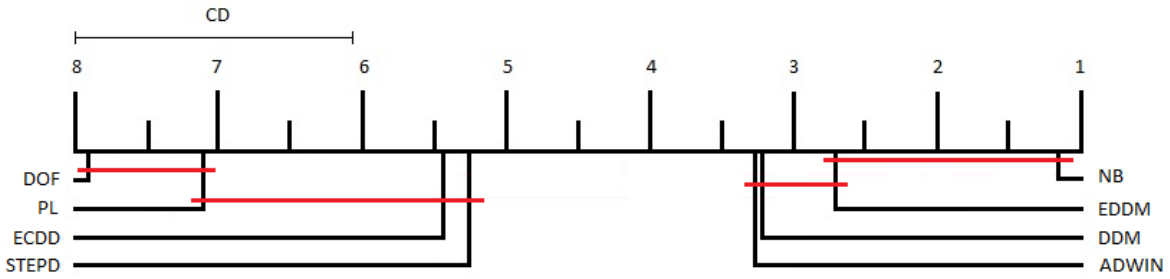

Figura 3: Comparação entre os consumos de memória dos métodos utilizando o teste de Nemenyi com 95% de confiança.

5.4 Considerações Finais

Este capítulo realizou uma comparação empírica e estatística dos métodos de detecção de mudanças em termos de precisão, tempo de execução e uso da memória. Em termos gerais, DDM foi o método que apresentou o menor rank médio na precisão enquanto EDDM obteve o menor rank no tempo de execução e no uso da memória.

Fazendo uma comparação dos resultados encontrados nesta dissertação com os resultados apresentados por (GONCALVES JR. et al., 2014), o DDM obteve o melhor desempenho geral em ambos os casos. Além disso, DoF também aparece como método com pior desempenho. Apesar da semelhança em muitos resultados, deve ser levado em consideração que diferentes metodologias foram utilizadas, tanto para selecionar os parâmetros quanto para avaliar os métodos. Portanto, uma comparação mais minuciosa seria inviabilizada.

Por fim, é importante ressaltar que os tempos de execução e também os respectivos consumos de memória de todos os métodos são pequenos o suficiente para que estes não devam ser fatores muito relevantes para a escolha do método de detecção de mudanças de conceito a ser utilizado, exceto em situações de extrema carência de recursos e/ou da utilização de bases de dados que exijam o processamento de um número muito grande de instâncias em um curto espaço de tempo.

O Capítulo 6 apresenta as conclusões deste trabalho bem como as contribuições e as propostas de trabalhos futuros.

6 Conclusões

Neste trabalho foi realizada uma avaliação criteriosa de alguns dos principais métodos de detecção de mudanças de conceito em ambientes com fluxo contínuo de dados. Os métodos avaliados foram: ADWIN, DDM, DoF, ECDD, EDDM, PL e STEPD. Essa avaliação incluiu a análise da precisão bem como o tempo de execução, o uso da memória, quantidade de alarmes falsos, e a precisão do ponto de detecção das mudanças. Para seleção dos parâmetros dos métodos detectores foi utilizada uma versão adaptada do algoritmo genético para este contexto.

Inicialmente, no Capítulo 3, uma série de adaptações que foram realizadas na estrutura básica do algoritmo genético tradicional são detalhadas. Para geração da população, foi definido um algoritmo que produz hipóteses de forma bem distribuída, reduzindo o risco de mínimos locais. Nesse capítulo é mostrado também como o uso do método do Ranking pode ajudar a selecionar indivíduos mais adequados para a realização do crossing-over.

No Capítulo 4 foi realizada uma análise empírica de cada um dos métodos. Para validação das métricas utilizou-se um procedimento denominado *Test-Then-Train*. Nove bases artificias foram escolhidas – Agrawal, Hyperplane, LED, Mixed, RandomRBF, SEA, Sine, Stagger, e Waveform – cada uma contendo quatro mudanças de conceito com velocidades abruptas, graduais rápidas, e graduais lentas. Foram selecionadas também três bases reais – Covertype, Electricity e PokerHand.

O desempenho dos métodos foi comparado utilizando-se os parâmetros padrões e também os parâmetros ajustados pelo AG. Em termos de precisão, ECDD foi o método mais beneficiado, com um ganho de 7,91% em relação aos métodos com parâmetros definidos por padrão. STEPD foi o único com perdas, tendo uma queda de 0,13%. Mesmo assim, das 30 instâncias de bases de dados existentes, somente em 11 existiram perdas. O que mais contribuiu para o desempenho médio negativo do STEPD foi a elevada perda em alguns casos pontuais, distorcendo o resultado final.

No tempo de execução dos métodos, STEPD foi o que obteve o maior ganho com o ajuste dos parâmetros, economizando, em média, 0,67s. Por outro lado, DoF foi o método que passou a utilizar mais tempo, ficando cerca de 2,68s mais lento. Dos sete métodos analisados, quatro apresentaram perdas utilizando os novos parâmetros: ADWIN, DoF, ECDD e PL.

Finalmente, comparando a quantidade da memória medido em Bytes por hora (B/h) dos métodos e utilizando os parâmetros padrões e os parâmetros ajustados, todos os métodos aumentaram o seu consumo com os novos valores. DoF foi o método com maior aumento, utilizando em média 137,48 B/h a mais. Já o EDDM, apesar de também ter ficado

mais custoso, foi o que teve o menor aumento, consumindo em média 2,22 B/h a mais.

De maneira geral, os parâmetros encontrados pelo AG buscavam quase sempre métodos mais sensíveis às mudanças de conceito. Apesar dessa abordagem aumentar a quantidade de falsos positivos, o aumento na velocidade da detecção das mudanças acaba tendo uma maior influência. As exceções estão nos casos em que a base é de difícil convergência, por exemplo RandomRBF e LED, onde um falso positivo em geral se torna mais prejudicial do que o normal. As constantes interrupções aliadas à difícil recuperação do classificador nessas bases contribuem para essa característica.

A utilização da média como função de agregação dos resultados encontrados pelo AG, apesar de ter obtido um desempenho satisfatório, não é a melhor solução. A principal justificativa para essa afirmação é motivada pelo fato de que a média não consegue lidar com valores discrepantes. Dessa forma, um único resultado pode prejudicar todos os outros, como aconteceu no DoF com mudanças graduais rápidas. Nesse método, os parâmetros encontrados pelo AG na base RandomRBF foi totalmente diferente dos encontrados nas outras bases. Com isso, o meio termo definido pela média não foi muito útil como parâmetro geral.

O Capítulo 5 apresentou uma análise comparativa empírica dos métodos de detecção de mudanças de conceito e verificou em termos de precisão, tempo de execução e uso da memória quais foram superiores estatisticamente. Para esse fim foi utilizado o teste de Friedman juntamente com o pós-teste de Nemenyi. Em termos de precisão, DDM obteve a maior média geral e foi considerado estatisticamente superior ao DoF e ao ECDD. Em relação ao tempo de execução e ao uso de memória, ADWIN foi, na média, o mais rápido e também o mais econômico no uso da memória, sendo estatisticamente superior a DoF, ECDD, PL, e STEPD.

Levando em consideração as três métricas – precisão, tempo de execução e uso da memória – pode-se dizer que o DDM foi o método com melhor desempenho. Além de ter obtido o melhor resultado médio em termos de precisão, seu desempenho nas outras métricas ficou muito próximo do melhor. Já o DoF pode ser considerado o pior: além de obter a pior precisão média, obteve também o maior consumo de memória e o segundo maior tempo de execução.

6.1 Contribuições

As principais contribuições desta dissertação são:

- A identificação de situações em que a utilização de um determinado método de detecção de mudança de conceito pode ser mais adequada do que usar outros métodos;
- A identificação dos métodos que obtêm os melhores rendimentos em termos de

precisão, tempo de execução, e uso de memória, tanto de forma geral quanto em bases de dados contendo diferentes tipos de mudanças de conceito (abrupta, gradual rápida, e gradual lenta);

- Uma aproximação das influências que cada um dos parâmetros exerce nos referidos métodos, bem como da melhor forma de configurá-los para os diferentes contextos;
- A utilização do algoritmo genético para otimização dos valores dos parâmetros dos métodos detectores de mudanças de conceito, com uma abordagem facilmente generalizável para outros métodos.

Parte do conteúdo apresentado nesta dissertação serviu de base para a escrita dos seguintes artigos, um deles já publicado em um periódico e o outro apresentado em uma conferência, ambos internacionais e bem classificados no qualis da CAPES:

- **A comparative study on concept drift detectors** (Paulo M. Gonçalves Jr., Silas G. T. de Carvalho Santos, Roberto S. M. Barros, Davi C. L. Vieira) - Publicado no periódico **Expert Systems with Applications** em Dezembro de 2014;
- **Optimizing the Parameters of Concept Drift Detection Methods using a Genetic Algorithm** (Silas Garrido Teixeira de Carvalho Santos, Roberto Souto Maior de Barros e Paulo Mauricio Gonçalves Júnior) - Apresentado na **International Conference on Tools with Artificial Intelligence (ICTAI)** em Novembro de 2015.

Outros estudos motivados pelo tema de pesquisa, mas que não estão ligados diretamente aos resultados desta dissertação, também foram publicados em conferências internacionais de boa qualidade:

- **Speeding Up Recovery from Concept Drifts** (Silas Garrido Teixeira de Carvalho Santos, Paulo Mauricio Gonçalves Júnior, Geyson Daniel dos Santos Silva e Roberto Souto Maior de Barros) - Apresentado na **European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)** em Setembro de 2014;
- **A Lightweight Concept Drift Detection Ensemble** (Bruno Iran Ferreira Maciel, Silas Garrido Teixeira de Carvalho Santos e Roberto Souto Maior de Barros) - Apresentado na **International Conference on Tools with Artificial Intelligence (ICTAI)** em Novembro de 2015.

6.2 Trabalhos Futuros

Existem muitas possibilidades de trabalhos futuros que podem ser desenvolvidas como continuação deste trabalho. Dentre elas incluímos:

- Estudar a possibilidade de ajustar dinamicamente os valores dos parâmetros de um ou mais métodos detectores de mudanças de conceito, dependendo da distribuição dos dados, visando aumentar sua precisão;
- Buscar outras funções para agregação dos resultados do algoritmo genético para melhorar a escolha de valores para os parâmetros dos referidos métodos, que sejam mais adequados a cada tipo de base de dados;
- Utilizar uma meta heurística para encontrar os melhores parâmetros para o AG;
- Verificar a viabilidade de criação de um *framework* que possa identificar de maneira automática determinadas situações nos dados e que, baseando-se nessas informações, possa selecionar o método mais adequado para detectar as mudanças de conceito;
- Experimentar com a utilização de um comitê de métodos detectores, executando em paralelo e tendo suas respostas agregadas, visando identificações mais precisas das mudanças de conceito;
- Testar a utilização de outras distâncias – diferente da euclidiana – no DoF com o intuito de reduzir o seu alto tempo de execução.

Referências

AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. Database mining: A performance perspective. *IEEE Trans. on Knowl. and Data Eng.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 5, n. 6, p. 914–925, dez. 1993. ISSN 1041-4347. Disponível em: <http://dx.doi.org/10.1109/69.250074>. Citado na página 26.

BACH, S. H.; MALOOF, M. A. Paired learners for concept drift. *2013 IEEE 13th International Conference on Data Mining*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 23–32, 2008. ISSN 1550-4786. Citado 2 vezes nas páginas 18 e 24.

BAENA-GARCíA, M. et al. Early drift detection method. In: *International Workshop on Knowledge Discovery from Data Streams*. [s.n.], 2006. (IWKDDS '06), p. 77–86. Disponível em: <http://eprints.pascal-network.org/archive/00002509/>. Citado 2 vezes nas páginas 18 e 24.

BIFET, A.; GAVALDA, R. Learning from time-changing data with adaptive windowing. In: *Proceedings of the Seventh SIAM International Conference on Data Mining*. Lake Buena Vista, Florida, USA: SIAM, 2007. (SDM '07), p. 443–448. Citado 2 vezes nas páginas 18 e 21.

BIFET, A. et al. MOA: Massive online analysis. *Journal of Machine Learning Research*, MIT Press, Cambridge, MA, USA, v. 11, p. 1601–1604, August 2010. ISSN 1532-4435. Disponível em: <http://portal.acm.org/citation.cfm?id=1859890.1859903>. Citado na página 26.

BIFET, A. et al. New ensemble methods for evolving data streams. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2009. (KDD '09), p. 139–148. ISBN 978-1-60558-495-9. Disponível em: <http://dx.doi.org/10.1145/1557019.1557041>. Citado 2 vezes nas páginas 27 e 28.

BLUM, A. Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain. *Machine Learning*, Springer Netherlands, v. 26, n. 1, p. 5–23, 1997. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A:1007335615132>. Citado na página 18.

BRAGA, A. de P.; CARVALHO, A. de L. F.; LUDERMIR, T. *Redes neurais artificiais: teoria e aplicações*. LTC Editora, 2000. ISBN 9788521612186. Disponível em: <http://books.google.com.br/books?id=cUgEaAEACAAJ>. Citado na página 34.

BRZEZINSKI, D.; STEFANOWSKI, J. Accuracy updated ensemble for data streams with concept drift. In: CORCHADO, E.; KURZYNSKI, M.; WOZNIAK, M. (Ed.). *Hybrid Artificial Intelligent Systems*. Springer Berlin / Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6679). p. 155–163. Disponível em: <http://dx.doi.org/10.1007/978-3-642-21222-2_19>. Citado na página 18.

BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 2, n. 2, p. 121–167,

jun. 1998. ISSN 1384-5810. Disponível em: <http://dx.doi.org/10.1023/A:1009715923555>. Citado na página 17.

DATAR, M. et al. Maintaining stream statistics over sliding windows: (extended abstract). In: *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002. (SODA '02), p. 635–644. ISBN 0-89871-513-X. Disponível em: <http://dl.acm.org/citation.cfm?id=545381.545466>. Citado na página 21.

DELANY, S. J. et al. A case-based technique for tracking concept drift in spam filtering. *Knowledge-Based Systems*, v. 18, n. 4-5, p. 187–195, 2005. ISSN 0950-7051. AI-2004, Cambridge, England, 13th-15th December 2004. Disponível em: <http://dx.doi.org/10.1016/j.knosys.2004.10.002>. Citado na página 20.

DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, JMLR.org, v. 7, p. 1–30, January 2006. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=1248547.1248548>. Citado na página 71.

FRIEDRICHS, F.; IGEL, C. Evolutionary tuning of multiple {SVM} parameters. *Neurocomputing*, v. 64, n. 0, p. 107 – 117, 2005. ISSN 0925-2312. Trends in Neurocomputing: 12th European Symposium on Artificial Neural Networks 2004. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925231204005223>. Citado na página 30.

GAMA, J. *Knowledge Discovery from Data Streams*. 1st. ed. [S.l.]: Chapman & Hall/CRC, 2010. ISBN 1439826110, 9781439826119. Citado 2 vezes nas páginas 17 e 20.

GAMA, J. et al. Learning with drift detection. In: BAZZAN, A.; LABIDI, S. (Ed.). *Advances in Artificial Intelligence – SBIA 2004*. Springer Berlin / Heidelberg, 2004, (Lecture Notes in Computer Science, v. 3171). p. 66–112. Disponível em: <http://dx.doi.org/10.1007/978-3-540-28645-5_29>. Citado 3 vezes nas páginas 18, 21 e 27.

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201157675. Citado na página 32.

GONCALVES JR., P. M.; BARROS, R. S. M. A comparison on how statistical tests deal with concept drifts. In: ARABNIA, H. R. et al. (Ed.). *Proceedings of the 2012 International Conference on Artificial Intelligence*. Las Vegas, Nevada, USA: CSREA Press, 2012. (ICAI '12, v. 2), p. 832–838. ISBN 1-60132-218-6. Disponível em: <http://world-comp.org/proc2012/icai.html>. Citado na página 18.

GONCALVES JR., P. M.; BARROS, R. S. M. RCD: A recurring concept drift framework. *Pattern Recognition Letters*, v. 34, n. 9, p. 1018–1025, July 2013. ISSN 0167-8655. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2013.02.005>. Citado na página 18.

GONCALVES JR., P. M. et al. A comparative study on concept drift detectors. *Expert Systems with Applications*, v. 41, n. 18, p. 8144 – 8156, December 2014. ISSN 0957-4174. Disponível em: <http://dx.doi.org/10.1016/j.eswa.2014.07.019>. Citado 2 vezes nas páginas 20 e 77.

HARRIES, M. B.; SAMMUT, C.; HORN, K. Extracting hidden context. *Machine Learning*, Springer Netherlands, v. 32, n. 2, p. 101–126, 1998. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A:1007420529897>. Citado na página 18.

HECKERMAN, D. Bayesian networks for data mining. *Data Min. Knowl. Discov.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 1, n. 1, p. 79–119, jan. 1997. ISSN 1384-5810. Disponível em: <http://dx.doi.org/10.1023/A:1009730122752>. Citado na página 17.

HERRON, S. *Análise Evolutiva*. Artmed Editora, 2009. ISBN 9788536319575. Disponível em: <http://books.google.com.br/books?id=JhZb-cuEh1oC>. Citado na página 34.

HULTEN, G.; SPENCER, L.; DOMINGOS, P. Mining time-changing data streams. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2001. (KDD '01), p. 97–106. ISBN 1-58113-391-X. Disponível em: <http://dx.doi.org/10.1145/502512.502529>. Citado na página 26.

KATAKIS, I.; TSOUMAKAS, G.; VLAHAVAS, I. Tracking recurring contexts using ensemble classifiers: an application to email filtering. *Knowledge and Information Systems*, Springer-Verlag, v. 22, p. 371–391, 2010. ISSN 0219-1377. Disponível em: <http://dx.doi.org/10.1007/s10115-009-0206-2>. Citado na página 18.

KOLTER, J. Z.; MALOOF, M. A. Dynamic weighted majority: An ensemble method for drifting concepts. *Journal of Machine Learning Research*, JMLR.org, v. 8, p. 2755–2790, December 2007. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=1314498.1390333>. Citado 2 vezes nas páginas 18 e 20.

LORENA, A. C.; CARVALHO, A. C. P. L. F. de. Evolutionary tuning of svm parameter values in multiclass problems. *Neurocomput.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 71, n. 16-18, p. 3326–3334, out. 2008. ISSN 0925-2312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2008.01.031>. Citado na página 30.

MINKU, L. L.; YAO, X. DDD: A new ensemble approach for dealing with concept drift. *IEEE Transactions on Knowledge and Data Engineering*, v. 24, n. 4, p. 619–633, April 2012. ISSN 1041-4347. Disponível em: <http://dx.doi.org/10.1109/TKDE.2011.58>. Citado na página 18.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072. Citado na página 31.

NISHIDA, K.; YAMAUCHI, K. Detecting concept drift using statistical testing. In: CORRUBLE, V.; TAKEDA, M.; SUZUKI, E. (Ed.). *Proceedings of the 10th International Conference on Discovery Science*. Berlin, Heidelberg: Springer-Verlag, 2007. (DS '07, v. 4755), p. 264–269. ISBN 978-3-540-75487-9. Disponível em: <http://portal.acm.org/citation.cfm?id=1778942.1778972>. Citado 2 vezes nas páginas 18 e 25.

PAGE, E. S. Continuous inspection schemes. *Biometrika*, v. 41, n. 1/2, p. 100–115, June 1954. Disponível em: <http://www.jstor.org/stable/2333009>. Citado na página 18.

ROBERTS, S. W. Control chart tests based on geometric moving averages. *Technometrics*, v. 1, n. 3, p. 239–250, 1959. Citado na página 23.

ROSS, G. J. et al. Exponentially weighted moving average charts for detecting concept drift. *Pattern Recognition Letters*, v. 33, n. 2, p. 191 – 198, 2012. ISSN 0167-8655. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167865511002704>. Citado 3 vezes nas páginas 18, 23 e 24.

SANTOS, S. et al. Speeding up recovery from concept drifts. In: CALDERS, T. et al. (Ed.). *Machine Learning and Knowledge Discovery in Databases*. Springer Berlin Heidelberg, 2014, (Lecture Notes in Computer Science, v. 8726). p. 179–194. ISBN 978-3-662-44844-1. Disponível em: <http://dx.doi.org/10.1007/978-3-662-44845-8_12>. Citado 2 vezes nas páginas 18 e 20.

SCHLIMMER, J. C.; GRANGER, R. H. Incremental learning from noisy data. *Machine Learning*, Springer Netherlands, v. 1, n. 3, p. 317–354, 1986. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A:1022810614389>. Citado na página 27.

SEBASTIAO, R.; GAMA, J. A study on change detection methods. In: *4th Portuguese Conf. on Artificial Intelligence, Lisbon*. [S.l.: s.n.], 2009. Citado na página 18.

SEBASTIAO, R. et al. Monitoring incremental histogram distribution for change detection in data streams. In: GABER, M. et al. (Ed.). *Knowledge Discovery from Sensor Data*. Springer Berlin / Heidelberg, 2010, (Lecture Notes in Computer Science, v. 5840). p. 25–42. ISBN 978-3-642-12518-8. Disponível em: <http://dx.doi.org/10.1007/978-3-642-12519-5_2>. Citado na página 18.

SOARES, S.; ANTUNES, C. H.; ARAUJO, R. Comparison of a genetic algorithm and simulated annealing for automatic neural network ensemble development. *Neurocomput.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 121, p. 498–511, dez. 2013. ISSN 0925-2312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2013.05.024>. Citado na página 30.

SOBHANI, P.; BEIGY, H. New drift detection method for data streams. In: BOUCHACHIA, A. (Ed.). *Adaptive and Intelligent Systems*. Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6943). p. 88–97. ISBN 978-3-642-23856-7. Disponível em: <http://dx.doi.org/10.1007/978-3-642-23857-4_12>. Citado 3 vezes nas páginas 18, 22 e 23.

SPARKS, R. S. Cusum charts for signalling varying location shifts. *Journal of Quality Technology*, v. 32, n. 2, p. 157–171, abr. 2000. Citado na página 24.

STREET, W. N.; KIM, Y. A streaming ensemble algorithm (SEA) for large-scale classification. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2001. (KDD '01), p. 377–382. ISBN 1-58113-391-X. Disponível em: <http://dx.doi.org/10.1145/502512.502568>. Citado na página 27.

VERDIER, G.; HILGERT, N.; VILA, J.-P. Adaptive threshold computation for cusum-type procedures in change detection and isolation problems. *Comput. Stat. Data Anal.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 52, n. 9, p. 4161–4174, maio 2008. ISSN 0167-9473. Disponível em: <http://dx.doi.org/10.1016/j.csda.2008.01.026>. Citado na página 24.

WIDMER, G.; KUBAT, M. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, Springer Netherlands, v. 23, n. 1, p. 69–101, 1996. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A:1018046501280>. Citado na página 18.

WILSON, D. R.; MARTINEZ, T. R. Improved heterogeneous distance functions. *J. Artif. Int. Res.*, AI Access Foundation, USA, v. 6, n. 1, p. 1–34, jan. 1997. ISSN 1076-9757. Disponível em: <http://dl.acm.org/citation.cfm?id=1622767.1622768>. Citado na página 22.

Anexos

ANEXO A – Gráficos das Precisões

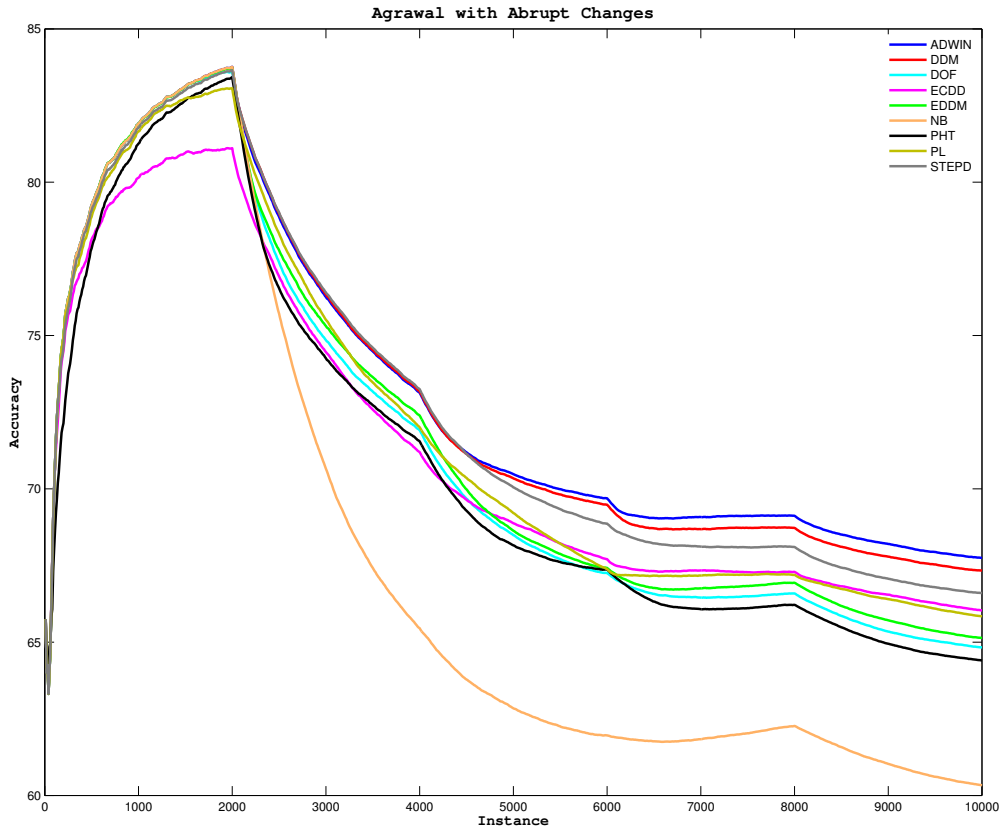

Figura 4: Agrawal com mudanças abruptas

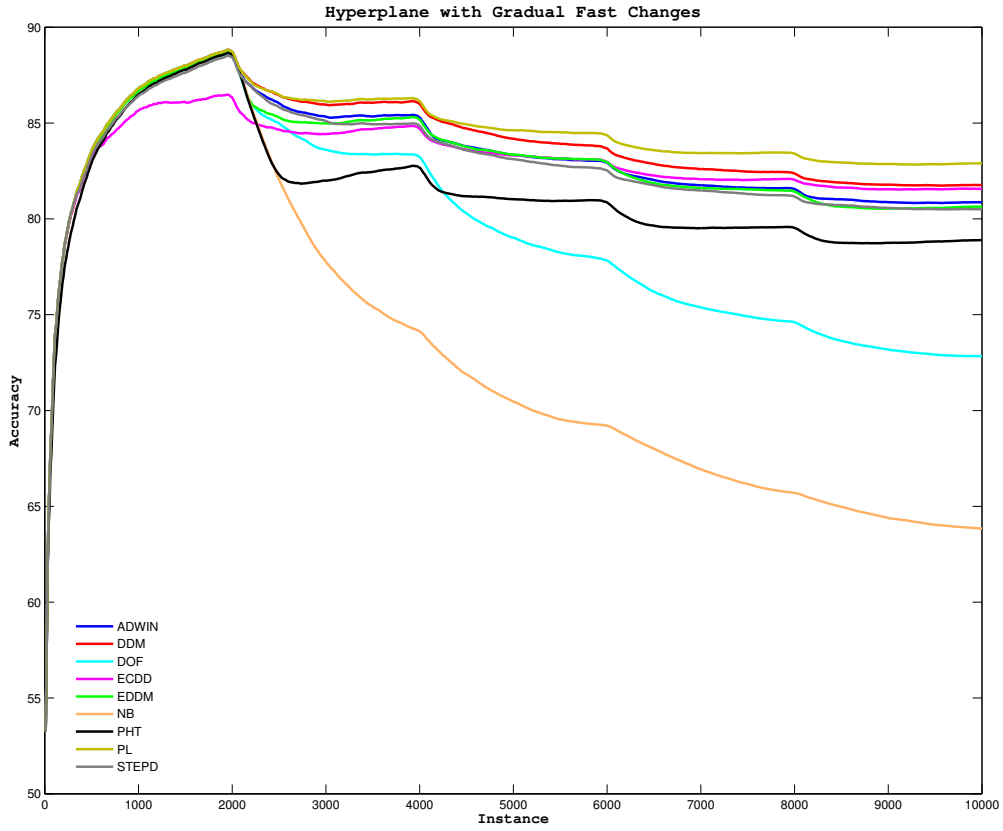

Figura 5: Hyperplane com mudanças graduais rápidas

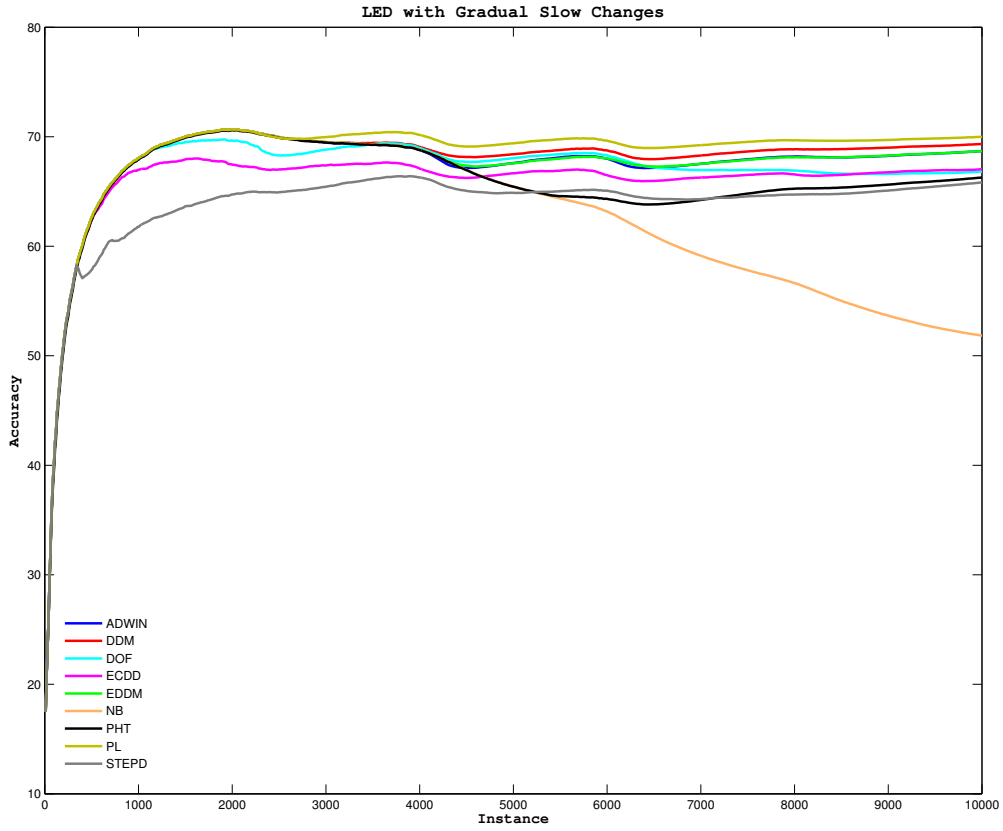

Figura 6: LED com mudanças graduais lentas

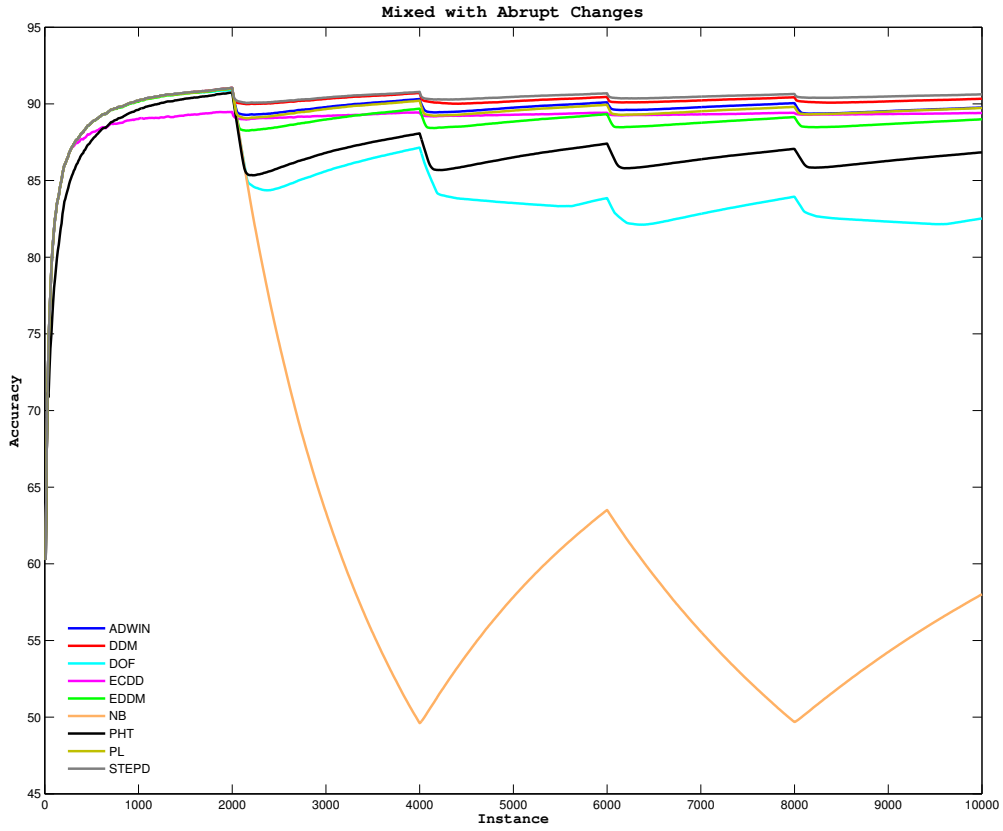

Figura 7: Mixed com mudanças abruptas

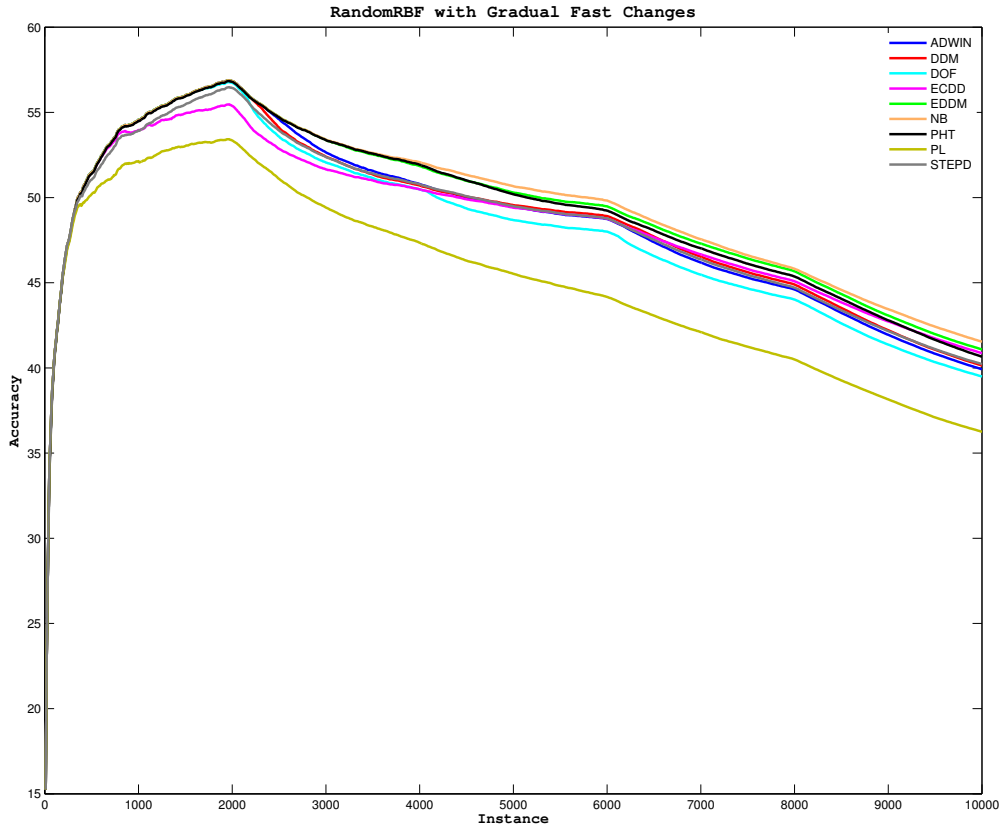

Figura 8: RandomRBF com mudanças graduais rápidas

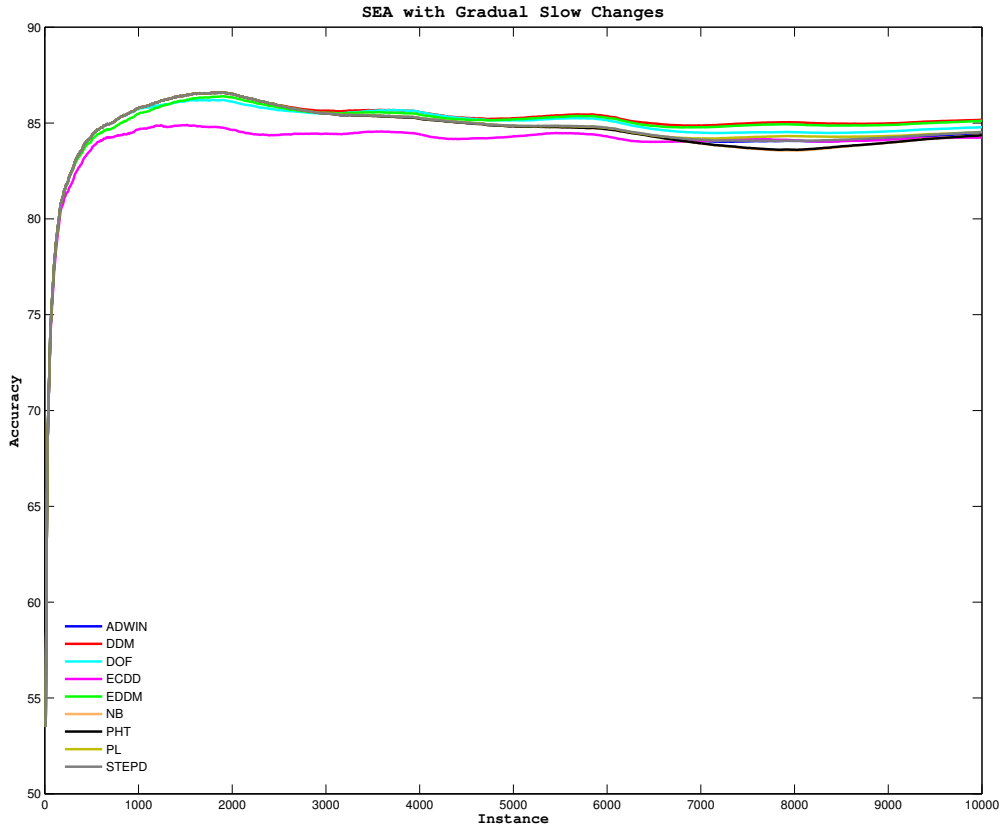

Figura 9: SEA com mudanças graduais lentas

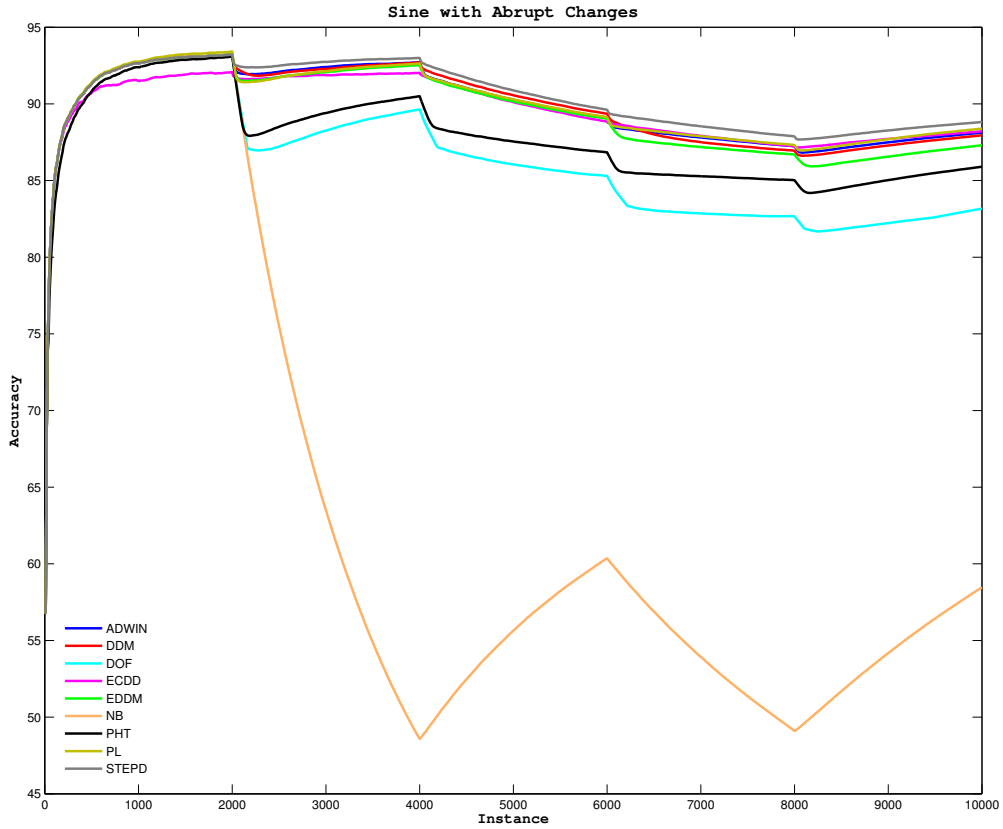

Figura 10: Sine com mudanças abruptas

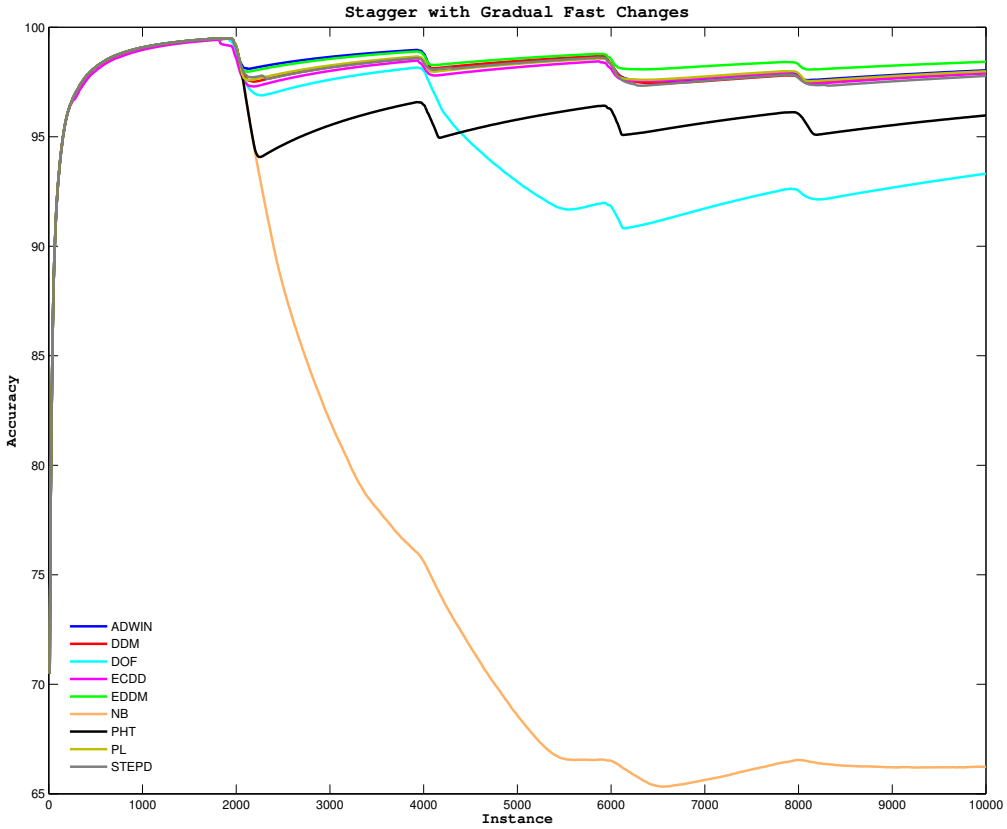

Figura 11: Stagger com mudanças graduais rápidas

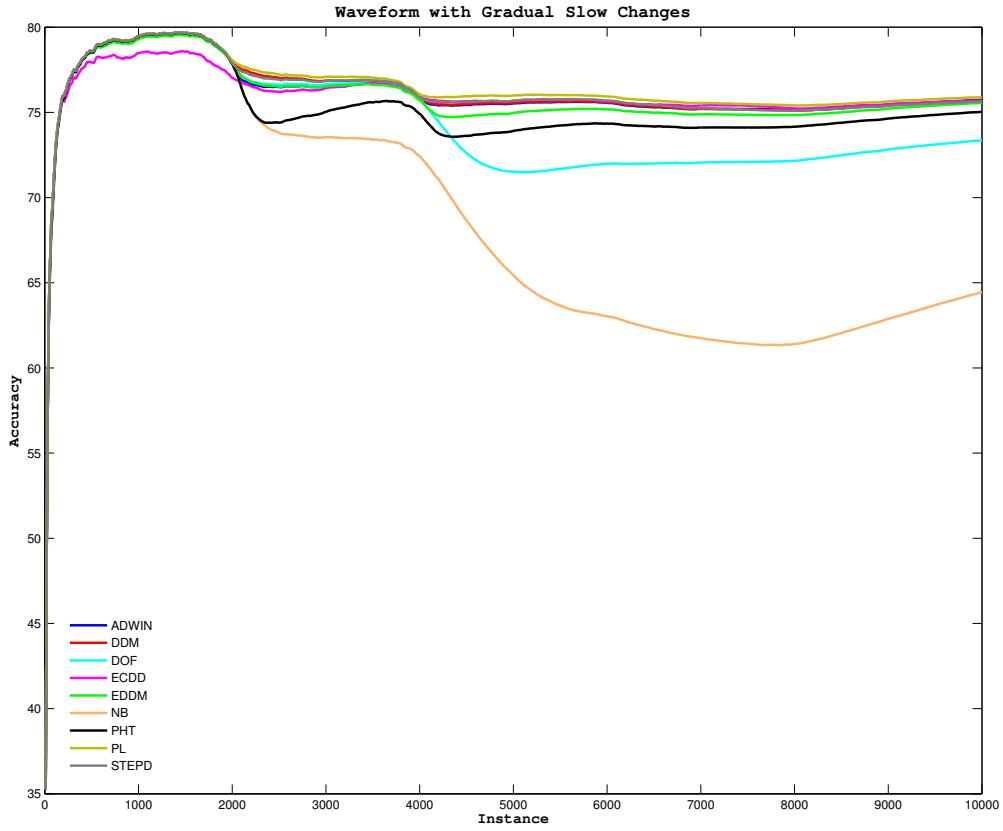

Figura 12: Waveform com mudanças graduais lentas

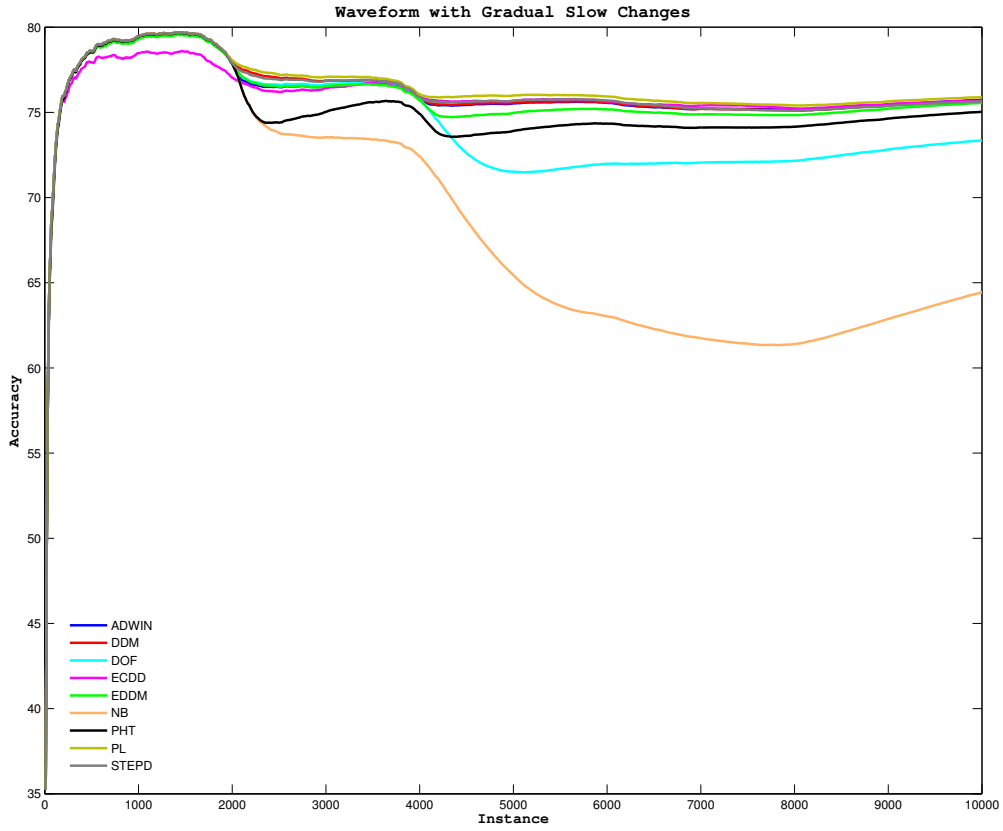

Figura 12: Waveform com mudanças graduais lentas