Pós-Graduação em Ciência da Computação

José Luis Martínez Pérez

Comitê de métodos estatísticos para detecção de mudanças de conceito

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

RECIFE

2018

José Luis Martínez Pérez

Comitê de métodos estatísticos para detecção de mudanças de conceito

Trabalho apresentado ao Programa de 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: Prof. Roberto Souto Maior de Barros

RECIFE
2018

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

P438c Pérez, José Luis Martínez
Comitê de métodos estatísticos para detecção de mudanças de conceito / José Luis Martínez Pérez. – 2018.
105 f.: il., fig., tab.

Orientador: Roberto Souto Maior de Barros.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2018.
Inclui referências e apêndices.

1. Ciência da computação. 2. Aprendizagem online. I. Barros, Roberto Souto Maior de (orientador). II. Título.

004 CDD (23. ed.) UFPE- MEI 2018-056

José Luis Martínez Pérez

Comitê de Métodos Estatísticos para Detecção de Mudanças de Conceito

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação

Aprovado em: 28/02/2018.

BANCA EXAMINADORA

_____

Prof. Dr. Germano Crispim Vasconcelos
Centro de Informática/UFPE

_____

Prof. Dr. Paulo Mauricio Gonçalves Junior
Instituto Federal de Pernambuco / Campus Recife

_____

Prof. Dr. Roberto Souto Maior de Barros
Centro de Informática / UFPE
(Orientador)

Dedico este trabalho especialmente a minha avó, assim como, a minha família e namorada.

Agradecimentos

No decorrer da vida tem pessoas que influenciam nossa formação pessoal e profissional, tendo a certeza que faltaram muitos nomes das mesmas por mencionar e solicitando minhas mais sinceras desculpas, agradeço para aqueles que neste momento chegam a meus pensamentos:

A minha avó Norma por exigir mais esforço nos meus estudos, lembrando-me os tempos em que me levou à escola; a meus pais Nuvia e Luis, por mostrar-me sempre que embora o caminho tenha obstáculo nunca se deve desistir; a minha namorada Ivis de la Caridad Sánchez por ensinar-me que com paciência, dedicação e deus no coração se continua em frente.

A meu orientador Professor Roberto Souto Maior de Barros, primeiramente agradecer por me aceitar como orientando, pelos ensinamentos nestes dois anos de mestrado, assim como a paciência mantida. Muitas graças ao senhor por exigir de me entrega e dedicação no trabalho.

Aos integrantes da turma de pesquisa, a Bruno Maciel pelas ideias sugeridas, especialmente a Silas Garrido pelo tempo dedicado na revisão da dissertação. Assim como por sempre estar quando precise, com um conselho na hora; a aquelas pessoas que me permitiram sentir em recife como na casa de meus pais, Yarima Sánchez, Sonia Perreira, Juan Hidalgo, Laura Palomino, obrigado por todo amigos.

A meus irmãos Yudi e Yuni por me ensinar que todo começa com um primeiro passo; a meus meninos de coração Mairon e evelim, assim como a minhas tias (nelsis, nuelvis, normi) das quais nunca falto um conselho quando precise.

A Dailys e Yaicel, muito obrigado por ter me dado a oportunidade de continuar desenvolvendo profissionalmente.

Aos professores e funcionários do programa de pós-graduação em ciências da computação do Centro de informática (CIN-UFPE). Assim como á Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) por fornecer uma bolsa de estudos de Mestrado.

"Eu acredito, que às vezes são as pessoas que ninguém espera nada, que fazem as coisas que ninguém consegue imaginar."

(Alan Turing)

Resumo

O notável aumento da quantidade de informação proveniente das tecnologias nos impossibilita de continuar usando os sistemas de aprendizagem tradicionais (batch). Por isso, precisa-se de algoritmos online, que devem ser atualizados constantemente, adaptando-se rapidamente às novas instâncias (dados). Além disto, os dados chegam em grande velocidade (fluxo de dados) e suas distribuições podem mudar com o tempo, gerando um evento chamado de mudança de conceito (Concept drift), o qual degrada o modelo de aprendizagem. A extração de conhecimento (KDD) em fluxos de dados com a presença de mudanças de conceito é uma das tarefas mais desafiadoras nas áreas de mineração de dados e aprendizado de máquina. Muitos algoritmos de aprendizagem de máquina, conhecidos como classificadores base, utilizam o aprendizado indutivo supervisionado e, para melhorar sua acurácia, são usados com detectores de mudanças de conceito, também chamados de métodos adaptativos. O algoritmo de aprendizagem ideal combina a robustez ao ruído com a sensibilidade às mudanças de conceito. Fundamentando-se nas alegações acima colocadas, nesta investigação foi implementado um algoritmo para detecção de mudanças de conceito (ANOVA_C), cujo objetivo constitui prover e demonstrar empiricamente que a metodologia de construção de um detector baseado na combinação de vários testes estatísticos para notificar as mudanças de conceitos é uma boa alternativa para melhorar os resultados. O comitê de métodos estatísticos ANOVA_C utiliza no processo de notificação das mudanças os resultados de três provas estatísticas (ANOVA padrão, Brown_Forysthe, e O'Brien) combinadas mediante dois sistemas de votação: para o estado de alerta (warning) é usado o voto majoritário com a informação fornecida pelos três testes estatísticos e para as mudanças de conceito usa-se a regra "O primeiro que encontra é o primeiro que reporta", usando somente a informação fornecida pelos métodos estatísticos de Brown-Forysthe e O'Brien. A experimentação realizada com os classificadores bases Hoeffding Tree (HT) e Naive Bayes (NB) usando 24 bases de dados artificiais e nove reais demonstraram a eficiência da proposta. No que se refere à avaliação da proposta, ANOVA_C atingiu os melhores valores de acurácia e foi o mais balanceado na análise das detecções de mudanças de conceitos, o que foi confirmado por ser o melhor posicionado na avaliação utilizando a métrica Matthews Correlation Coefficient (MCC).

Palavras-chaves: Mudanças de conceito. Comitê de métodos estatísticos. Fluxo de dados. Aprendizagem online.

Abstract

The remarkable increase in the amount of information coming from technology makes it impossible to continue using the traditional learning systems (batch). Therefore, we need online algorithms, which must be updated constantly, adapting quickly to new instances (data). In addition, the data arrives at high speed (data streams) and their distributions may change over time, generating an event called concept drift, which degrades the learning model. Knowledge Discovery from databases (KDD) in data streams with the presence of concept drift is one of the most challenging tasks in the areas of data mining and machine learning. Many machine learning algorithms, known as base classifiers, use supervised inductive learning and, to improve their accuracy, they are used with concept drift detectors, also called adaptive methods. The ideal learning algorithm combines the robustness to noise with sensitivity to the concept drift. Based on the above claims, in this investigation an algorithm was implemented to detect concept drifts (ANOVA_C). Its purpose is to provide and demonstrate empirically that the methodology of constructing a detector based on a combination of several statistical tests to notify concept drift is a good alternative to improve the results. The statistical methods committee ANOVA_C uses in the process of notification of changes the results of three statistical tests (Standard ANOVA, Brown_Forysthe, and O'Brien) combined by two voting systems: to warning status the majority vote is used with the information provided by the three statistical tests and for concept drift the "Early-find-early-report" rule is adopted, using only the information provided by the Brown-Forysthe and O'Brien statistical methods. The experimentation results with Hoeffding Tree (HT) and Naive Bayes (NB) as bases classifiers using 24 artificial and nine real-world databases demonstrated the efficiency of the proposal. Regarding the evaluation of the proposal, ANOVA_C achieved the best accuracy values and was the most balanced in the analysis of concept drift detections, which was confirmed as it was the best positioned in the evaluation using the Matthews Correlation Coefficient (MCC).

Key-words: Concept Drift. Committee of statistical methods. Data stream. Online Learning.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AD All Detectors Detect Drift.

ADWIN Adaptive Windowing.

ALHD At Least Half of the Detectors Detect Drift.

ALO At Least One Detects Drift.

AM Aprendizagem de Máquina.

DDE A Lightweight Concept Drift Detection Ensemble.

DDM Drift Detection Method.

e-Detector A Selective Detector Ensemble for Concept Drift Detection.

ECDD EWMA for Concept Drift Detection.

EDDM Early Drift Detection Method.

EHCD² Ensembles of Heterogeneous Concept Drift Detectors - Experimental Study.

EWMA Exponentially Weighted Moving Average.

FHDDM Fast Hoeffding Drift Detection Method for Evolving Data Stream.

FPDD Fisher Proportions Drift Detector.

FSDD Fisher Square Drift Detector.

FTDD Fisher Test Drift Detector.

HT Hoeffding Tree.

JRE Java Runtime Environment.

KDD Knowledge Discovery in Databases.

KNN K-nearest neighboors.

MCC Matthews Correlation Coefficient.

MOA Massive Online Analysis.

NB Naive Bayes.

PAC Probably Approximately Correct.

PL Paired Learners.

RDDM Reactive Drift Detection Method.

SDK Kit de desenvolvimento de software.

STEPD Statistical Test of Equal Proportions.

SVM Support Vector Machine.

Weka Waikato Environment for Knowledge Analysis.

WSTD Wilcoxon Rank Sum Test Drift Detector.

Sumário

1 INTRODUÇÃO

O mundo hoje está em constante desenvolvimento e evolução das tecnologias. O incremento desproporcional da informação proveniente das mesmas é cada vez maior, e seu armazenamento, organização e recuperação são automatizadas mediante sistemas de bases de dados. Devido à enorme quantidade de dados e sua produção contínua, não é possível para os seres humanos analisar ou utilizar sistemas de lote tradicionais (batch), onde todos os dados devem ser coletados antes. Por quanto, se faz necessária a implementação online da resposta. A *extração de conhecimento* (Knowledge Discovery in Databases (KDD)) a partir de grandes *fluxos de dados* é uma das tarefas mais desafiadoras na área de *mineração de dados* e *aprendizado de máquina*.

O KDD é o processo pelo qual se identificam de forma não trivial padrões válidos, novos, potencialmente úteis e compreensíveis que se encontram nos dados (FAYYAD et al., 1996). O conceito de KDD pode ser confundido com o conceito de mineração de dados (do inglês, Data Mining), porém não tem o mesmo significado, o último fazendo parte do primeiro. A mineração de dados é o processo de descoberta de padrões de dados. Trata-se de um tema complexo e está relacionado com várias disciplinas como: estatística, recuperação de informação, reconhecimento de padrões e aprendizagem de máquina.

A aprendizagem de máquina (do inglês, Machine Learning) é a disciplina que estuda os métodos utilizados para programar computadores de forma que aprendam (ORALLO et al., 2004), ou seja, que possam melhorar seu desempenho para realizar uma tarefa automaticamente, com base na experiência. O aprendizado para a criação do modelo a partir de dados previamente observados é chamado aprendizado indutivo. O mesmo pode ser dividido em aprendizado supervisionado, não-supervisionado e semi-supervisionado. No aprendizado supervisionado todos os exemplos têm um rótulo, diferente do aprendizado não-supervisionado, que não tem exemplos rotulados. Por sua parte, o aprendizado semi-supervisionado posiciona-se entre os aprendizados anteriores, englobando exemplos onde os rótulos são conhecidos e exemplos que não possuem rótulos.

As técnicas usadas no Aprendizado de Máquina podem ser divididas em dois grandes grupos: descritivas para o aprendizado não-supervisionado e as preditivas ligadas ao aprendizado supervisionado. As primeiras estão em correspondência com as tarefas de detecção de agrupamentos e correlações, e as últimas com as tarefas de classificação e regressão dependendo se os rótulos são discretos ou contínuos respectivamente.

O KDD nos ambientes com fluxo contínuo de dados é uma atividade que vem crescendo progressivamente. Exemplos de aplicações incluem: internet; telefonia móvel; monitoramento do histórico de compras de clientes; detecção de presença por meio de sensores; análise de tráfego TCP/IP; e monitoramento da temperatura da água (GAMA; GABER,

2007). Assim, algoritmos utilizados para análise dos dados de maneira online devem ser atualizados constantemente, adaptando-se rapidamente aos dados (as novas instâncias) que vem chegando em grande velocidade (fluxos de dados) (DU; SONG; JIA, 2014), cujo comportamento das distribuições podem mudar ao longo do tempo, gerando um fenômeno conhecido como *mudança de conceito* (Concept drift), afetando o rendimento do modelo de aprendizagem (GONÇALVES et al., 2014; PESARANGHADER; VIKTOR, 2016; BRZEZINSKI; STEFANOWSKI, 2016).

As mudanças de conceito são categorizadas de diferentes maneiras em ambiente de fluxos de dados. Com respeito à velocidade das mudanças, podem ser: abruptas (sudden drift, concept shift) ou graduais (gradual or incremental drift) (MINKU; WHITE; YAO, 2010). Outra forma de categorização pode ser determinada pelas reações que elas provocam nas distribuições dos dados. Nesse contexto, as mudanças podem ser reais ou virtuais (GONÇALVES JR.; BARROS, 2013).

A existência de mudanças de conceito em fluxo de dados é frequentemente observada em problemas do mundo real e, devido ao impacto que podem causar no modelo de aprendizagem (GAMA et al., 2014), em geral, os algoritmos de aprendizagem incremental devem incorporar mecanismos adicionais para manipular mudança de conceitos (BRZEZINSKI; STEFANOWSKI, 2014).

Muitas são as pesquisas realizadas para tentar fazer uma melhor detecção das mudanças de conceito em fluxos de dados. Estas propõem, em sua maioria, algoritmos para monitorar os resultados das predições de um classificador base, com o objetivo de identificar e sinalizar possíveis mudanças nas distribuições dos dados (BRZEZINSKI; STEFANOWSKI, 2016). Os chamados métodos adaptativos (ŽLIOBAITĖ et al., 2015) incluem: Adaptive Windowing (ADWIN) (BIFET; GAVALDÀ, 2007), Drift Detection Method (DDM) (GAMA et al., 2004), Early Drift Detection Method (EDDM) (BAENA-GARCIA et al., 2006), EWMA for Concept Drift Detection (ECDD) (ROSS et al., 2012), Statistical Test of Equal Proportions (STEPD) (NISHIDA; YAMAUCHI, 2007), Fast Hoeffding Drift Detection Method for Evolving Data Stream (FHDDM) (PESARANGHADER; VIKTOR, 2016), Reactive Drift Detection Method (RDDM) (BARROS et al., 2017), Wilcoxon Rank Sum Test Drift Detector (WSTD) (BARROS; HIDALGO; CABRAL, 2018), etc.

Os métodos podem enfrentar diversas dificuldades, pode-se citar, que DDM em cenários nos quais amostras apresentam mudanças de conceito graduais e muito longas tende a perder precisão (SALPERWYCK; BOULLÉ; LEMAIRE, 2015). Portanto, foi proposto um novo método EDDM, para melhorar as detecções do DDM nas mudanças de conceitos graduais e manter uma boa performance nas abruptas, mas o mesmo não funciona bem na presença de mudanças de conceitos abruptas (DU et al., 2014). Já com objetivo de ter um melhor comportamento nos cenários onde DDM e EDDM apresentam problemas, principalmente quando as mudanças de conceitos são muito longas foi proposto o RDDM.

Já o STEPD também não está isento de problemas. A utilização do teste estatístico

de proporções de duas amostras independentes para a detecção de mudanças de conceito, mesmo para os casos de amostra com o tamanho pequeno e/ou desbalanceadas, gera perda de acurácia (MEHTA; PATEL, 1996). Apesar dos autores terem identificado o problema, optaram pela não utilização do teste exato de Fisher (mais recomendado em tais situações) (AGRESTI, 1992) devido ao seu alto custo computacional. Os métodos Fisher Proportions Drift Detector (FPDD), Fisher Square Drift Detector (FSDD) e Fisher Test Drift Detector (FTDD) apresentados em (CABRAL, 2017; CABRAL; BARROS, 2018) propõem uma eficiente implementação do teste exato de Fisher para resolver os inconvenientes presentes no STEPD.

É difícil, na atualidade, encontrar um modelo de detecção de mudança de conceito eficiente em todas as situações. As investigações anteriores mostram que certos detectores são melhores em alguns cenários que outros. Isto conduziu que utilizar uma combinação de detectores (Comitê ou ensemble) seja uma alternativa a ter em conta, baseados na filosofia de que a opinião conjunta de vários especialistas tem melhor probabilidade de acertar na decisão do que a opinião de um único especialista. Já foram propostos alguns métodos para a criação de combinações de detectores, embora não exista uma maneira clara de saber que método de comitê de detectores é melhor que outro, ou quando usar um determinado método. Exemplos de algoritmos de comitê de detectores incluem: A Selective Detector Ensemble for Concept Drift Detection (e-Detector) (DU et al., 2014), A Lightweight Concept Drift Detection Ensemble (DDE) (MACIEL; SANTOS; BARROS, 2015), Ensembles of Heterogeneous Concept Drift Detectors - Experimental Study (WOŹNIAK et al., 2016) chamado de EHCD^2 nesta pesquisa.

Os comitês de detectores podem ter associados os mesmos problemas que detectores individuais, sendo o alto custo computacional (tempo e memória) um importante ponto para analisar. Além do mais, que os detectores para formar comitês homogêneos ou heterogêneos podem ser muito diversos quanto à estrutura de informar a mudança, contando com diferentes estados de alarmes, pelo qual se deve ser muito cuidadoso na escolha dos mesmos. Outro problema associado é o tempo em que os detectores notificam a mudança, o qual pode ser muito diferente. Em geral a maioria dos comitês estudados até o momento só usam a notificação de mudança dos detectores que os conformam, sem ter muito presente o tratamento dos alarmes de alertas de mudanças (presente em muitos detectores atuais). Pelo acima exposto, se pode perceber que buscar a melhor maneira de combinar estruturalmente os detectores assim como as suas saídas pode ser muito custoso.

A união de detectores deve resultar em um método mais preciso nas detecções das mudanças de conceitos, sendo capaz de aumentar o rendimento do classificador. Diante do exposto anteriormente, constitui uma problemática que ainda a ciência não tem dado respostas definitivas. Com o objetivo de avançar um passo mais neste sentido, na investigação atual surge a seguinte questão que constituem o *problema de investigação*: A construção de um detector baseado na combinação das respostas de vários testes estatísticos, seria

uma boa escolha para detectar mudanças de conceitos abruptas e graduais?

É válido ressaltar que a nova proposta não será uma combinação de detectores (DU et al., 2014; MACIEL; SANTOS; BARROS, 2015) que inclua vários detectores baseados em métodos estatísticos (NISHIDA; YAMAUCHI, 2007; BARROS; HIDALGO; CABRAL, 2018; CABRAL; BARROS, 2018). Sendo a estrutura de construção dos comitês de detectores a principal diferença com a nova proposta, já que a mesma equivalerá a um detector, só que usando para a notificação dos alarmes vários testes estatísticos, pelo que será conhecida na pesquisa como comitê de testes estatísticos.

1.1 Objetivos

O *objetivo geral* da investigação é a construção de um detector baseado na combinação das respostas de vários testes estatísticos aumentando a precisão dos modelos de classificação ou predição, considerando as restrições computacionais comuns nestes cenários, e que seja eficiente na manipulação de mudanças de conceito abruptas e graduais.

Para alcançar a realização deste objetivo geral são propostos como *objetivos específicos* os seguintes:

- Analisar os testes a usar na construção do detector bem como a diversidade dos mesmos.
- Construção do detector baseado na combinação das respostas dos testes estatísticos escolhidos.

Como *ideia a ser investigada* (Hipótese), se considera que o desenvolvimento de um detector baseado na combinação das respostas de vários testes estatísticos será uma boa alternativa para tratar com mudanças de conceitos abruptas e graduais, aumentando a precisão dos modelos de classificação ou predição.

1.2 Metodologia

Metodologias de pesquisa teóricas e empíricas foram usadas para alcançar o objetivo geral. Portanto, uma análise do estado da arte foi realizada, logrando deixar estabelecidos os conceitos teóricos e empíricos que regem o desenvolvimento da pesquisa. Assim como assinaladas as desvantagens ou necessidades das abordagens existentes para a detecção de mudanças de conceito. Além disso, o estudo detalhado das estratégias e métodos a seguir para enfrentar estas deficiências com a finalidade de prover a nova proposta. Proposta que é validada experimentalmente usando metodologias apropriadas de avaliação e usando um amplo conjunto de dados relevantes, comparando o método proposto com as abordagens relacionadas mais conhecidas.

Consequentemente, as *tarefas de investigação* que foram realizadas para cumprir o objetivo da dissertação são apresentadas a seguir:

1. Análise do estado da arte dos detectores de mudanças de conceitos e dos testes estatísticos paramétricos e não paramétricos para criar um levantamento bibliográfico;
2. Investigação dos testes estatísticos selecionados para a realização de um novo enfoque de detecção de mudança de conceito;
3. Investigação do funcionamento detalhado do STEPD e WSTD, a fim de usar e fazer adaptação necessária do trabalho das janelas para o novo enfoque;
4. Levantamento bibliográfico sobre bases de dados reais e artificiais que contenham mudanças de conceitos em fluxos de dados;
5. Identificação das metodologias de avaliação de algoritmos adequadas para a aprendizagem online com mudança de conceito, assim como das métricas para avaliar o rendimento de detectores de mudança de conceito;
6. Implementação de um novo enfoque de detecção combinando os métodos estatísticos escolhidos;
7. Realização dos experimentos comparativos, utilizando o ambiente Massive Online Analysis (MOA) (BIFET et al., 2010), com bases de dados reais e artificiais, a fim de avaliar o desempenho dos detectores de mudanças de conceito propostos com respeito a acurácia e outras métricas a serem apresentadas;
8. Análise dos resultados dos experimentos comparativos e conclusões da pesquisa.

1.3 Organização do Trabalho

O presente documento está estruturado em sete capítulos, os quais estão descritos a seguir:

- No capítulo 2 são introduzidos os conceitos associados às áreas de pesquisa que sustentam a presente investigação. O capítulo faz uma breve revisão sobre os principais conceitos de aprendizagem de máquina, com foco em fluxos de dados não estacionários e mudanças de conceitos considerados na aprendizagem incremental.
- O capítulo 3 apresenta um estudo acerca dos testes estatísticos utilizados no trabalho. Além disso, são definidos os princípios básicos para a realização de um teste de hipóteses e destacadas as principais diferenças entre os testes a serem usados.
- O capítulo 4 faz apresentação da implementação dos testes ANOVA padrão, Brown_Forysthe e O'Brien. Cada proposta é apresentada mediante a descrição geral de seu pseudocódigo, enfatizando nas explicações as diferenças dos mesmos.

- O capítulo 5 propõe um novo enfoque para a manipulação de mudanças de conceitos baseado na combinação de vários testes de hipótese. Cada teste é apresentado através de uma descrição geral e um pseudo-código detalhando o processo de detecção de mudanças de conceito.

- O capítulo 6 apresenta os experimentos realizados a fim de comparar a nova proposta com alguns dos principais métodos da atualidade. É descrita a configuração dos experimentos, seguida pela apresentação das bases de dados utilizadas nos testes. Para finalizar, os algoritmos são comparados através de suas acurácias e por meio da análise de suas detecções de mudanças de conceito.

- Por fim, o capítulo 7 tece as conclusões dessa dissertação. Esta última parte contém o detalhamento das considerações finais a respeito desta pesquisa, além de descrever alguns possíveis trabalhos futuros.

2 FLUXOS DE DADOS E MUDANÇAS DE CONCEITOS

Este capítulo, depois de uma minuciosa pesquisa no estado da arte, descreve os principais conceitos focados na área de pesquisa que abordará a investigação. Além disso, estabelece o porquê da necessidade de novos métodos para o tratamento de mudanças de conceitos, sendo o foco da investigação.

2.1 Preliminar

Na última década o desenvolvimento de modelos analíticos automatizados para a produção de ações inteligentes em tempo real teve um aumento considerável, devido à necessidade de analisar grandes volumes de dados proveniente da internet e que a cada dia são mais complexos. Aprendizagem de Máquina (AM) e Machine Learning são métodos de análise de dados que automatizam o desenvolvimento de modelos analíticos para ter um desempenho melhorado baseado na experiência (ORALLO et al., 2004). Usando algoritmos que aprendem interativamente a partir de dados, o aprendizado de máquina permite que os computadores encontrem conhecimentos ocultos sem serem explicitamente programados para procurar algo específico (DIETTERICH, 2003).

Quanto às técnicas utilizadas na AM, estas podem ser divididas em dois grandes grupos: preditivas ou descritivas (BLANCO; INOCENCIO, 2014). Dessa forma, os algoritmos que realizam a indução de modelos preditivos, através das tarefas de classificação ou regressão, seguem o paradigma do aprendizado supervisionado. Por outro lado, a meta do aprendizado não supervisionado é a exploração e descrição de um conjunto de dados por meio das tarefas de agrupamento, associação ou sumarização (FACELI et al., 2011).

Brevemente serão explicados os tipos de aprendizagem mais usados na atualidade. O aprendizado supervisionado compreende a abstração de um modelo de conhecimento a partir dos dados apresentados na forma de pares ordenados (entrada, saída desejada). Por entrada entenda-se o conjunto de valores de variáveis de entrada do algoritmo para um determinado caso (GOLDSCHMIDT; PASSOS, 2005). A saída desejada corresponde ao valor esperado que o algoritmo possa produzir, sempre que receber os valores especificados na entrada. No aprendizado não supervisionado não existe a informação da saída desejada. Os algoritmos partem dos dados, procurando estabelecer relacionamentos entre eles. Anteriormente mencionado, o aprendizado semi-supervisado está sendo cada vez mais estudado, assim como o aprendizado por reforço. O primeiro é uma área de pesquisa relativamente nova em AM, representa a junção do aprendizado supervisionado e não supervisionado, e tem o potencial de reduzir a necessidade de dados rotulados quando

somente um pequeno conjunto de exemplos rotulados está disponível. Por sua parte o segundo baseia-se na avaliação de um reforço ou recompensa para o conjunto de ações realizadas (DÍAZ, 2014) e é muito usado na área da robótica.

Outra classificação muito usada na literatura para dividir AM é determinada por protocolos de treinamentos. Os três mais conhecidos são: estocástico, lote e online (DUDA; HART; STORK, 2001). No protocolo estocástico, os dados do conjunto de treinamento são apresentados aos classificadores de forma aleatória. Já no aprendizado em lote, todos os exemplos de treinamento são apresentados aos preditores antes do início da aprendizagem. Por último, no treinamento online, cada instância é apresentada apenas uma única vez aos algoritmos de aprendizagem, sem o armazenamento dos exemplos de treinamento.

2.2 Fluxos de dados

No contexto de AM assim como no problema de classificação, um fluxo de dados é comumente definido como uma sequência de dados muito longa (e provavelmente infinita) que flui continuadamente em alta velocidade (SANTOS; BARROS; GONÇALVES JR., 2015), apresentado a seguir: $M = (\vec{x}_1, y_1); (\vec{x}_2, y_2); ...$ sendo $(\vec{x}_i, y_i)$ os exemplos ou instâncias que chegam ao longo do tempo (BIFET et al., 2009; KRAWCZYK et al., 2017), e $\vec{x}_i \in \overrightarrow{X}$ um vetor no qual cada componente é chamado de atributo e $y_i \in Y$ é seu rótulo correspondente, retirado de um conjunto finito Y de possibilidades (SUN et al., 2016). Assumindo a existência de uma função objetivo $f(\vec{x}_i) = y_i$, a tarefa de aprendizagem incremental é obter um modelo $\hat{f}$ que aproxime f, de modo que $\hat{f}$ maximize a precisão na previsão (TROYANO; RUIZ; SANTOS, 2005; FRÍAS-BLANCO et al., 2016). Muitas vezes, também é assumido que os exemplos são regulados por uma função de densidade de probabilidade $P(\overrightarrow{X}, Y)$. É Considerado conceito como o término que faz referência à função $P(\overrightarrow{X}, Y)$ do problema em um ponto determinado no tempo (MINKU; WHITE; YAO, 2010).

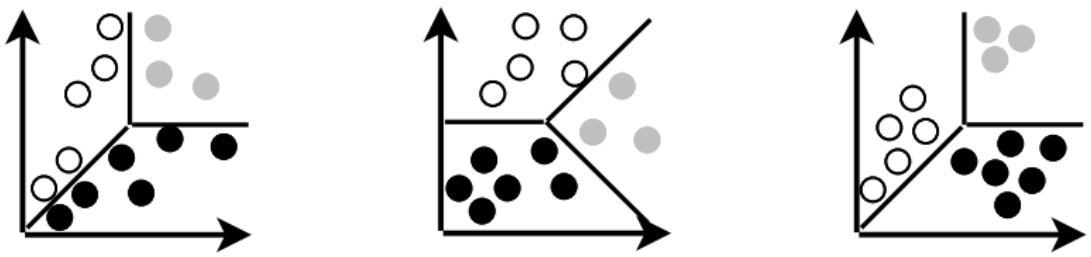
Os fluxos de dados não estão isentos de restrições, sobretudo pelo fato da quantidade de instâncias que serão processadas serem consideradas potencialmente infinitas (BIFET, 2009). Portanto, é impossível armazená-los completamente em memória, sendo só uma pequena parte processada e armazenada enquanto o resto deve ser descartado. Mesmo que fosse possível armazenar toda a informação, não seria viável processá-la em sua totalidade. Outra restrição a ter em conta é que a velocidade de chegada dos dados é alta, pelo que devem ser processados em tempo real e daí serem descartados (DU; SONG; JIA, 2014). O anterior exposto demonstra as limitações de quantidade de memória e do tempo de processamento que vai ter um algoritmo para o processamento de um fluxo de dados (BABCOCK et al., 2002; AGGARWAL; PHILIP, 2007). Outra restrição não menos importante e foco de muitas pesquisas nos últimos anos, assim como desta, é a possível variação no tempo da função de distribuição de probabilidade que gera os dados (RUTKOWSKI et al., 2015). Essa variação pode tornar os dados irrelevantes, desencadeando um fenô-

meno conhecido como mudança de conceito (*concept drift*). Devido à importância dentro do aprendizado incremental e complexidade de manipular as mesmas, será dedicada a seguinte seção para descrever suas principais características.

2.3 Mudança de conceitos

Conhecendo a definição de um fluxo de dados e também de forma resumida o significado de mudança de conceito, pelo já exposto em seções anteriores, é possível definir mais formalmente que uma mudança na função de distribuição do problema $P(\overrightarrow{X}, Y)$ (também conhecidos como contexto) seja chamada mudança de conceito (GAMA et al., 2004), sendo os fluxos muito suscetíveis à mesma (READ et al., 2012).

Embora as mudanças de conceito possam ser essencialmente causadas por alteração nas probabilidades, é comum encontrá-las categorizadas na literatura (Figura 1) (KHAMASSI et al., 2015) como *mudanças reais* (FAN et al., 2004) quando a probabilidade condicional $P(Y \mid \overrightarrow{X})$ muda como ilustra a Figura 1(b), com ou sem mudanças em $P(\overrightarrow{X})$ (GAMA et al., 2014; DELANY et al., 2005; ŽLIOBAITĖ, 2010). A outra categorização um pouco menos abordada nas pesquisas é a que acontece quando só a probabilidade a priori $P(\overrightarrow{X})$ muda (KRAWCZYK et al., 2017), como mostra a Figura 1(c), chamadas na bibliografia de *mudanças virtuais*.



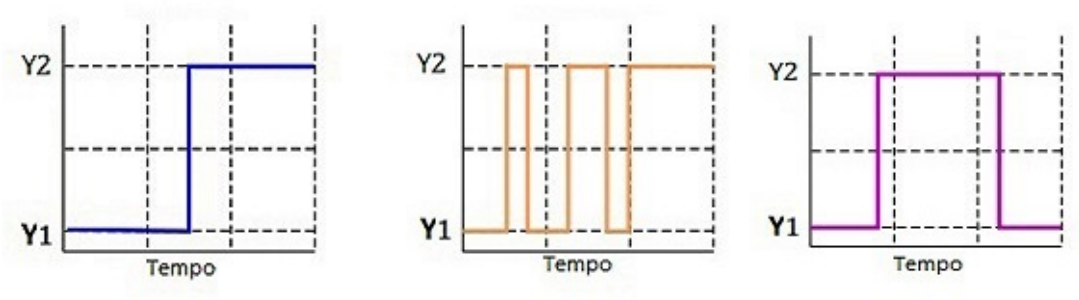
(a) (b) (c)

Figura 1 – Tipos de mudanças de conceitos. **(a)** Dados sem mudança, **(b)** Mudança real, **(c)** Mudança virtual.

Na prática estas categorias de mudanças geralmente não são relevantes já que todas impactam de forma similar no modelo de aprendizagem. Como os erros produzidos pelo modelo se incrementam com respeito aos exemplos atuais, é necessário atualizar constantemente o modelo. Segundo Woźniak et al. (2016), considerando a tarefa de classificação, a mudança de conceito real é mais importante.

Outra taxonomia muito usada na literatura para categorizar as mudanças de conceitos em fluxos de dados é de acordo com a velocidade. Essas mudanças podem ser divididas em abruptas, quando a distribuição é modificada num único intervalo de tempo como descreve visualmente a figura 2(a), ou graduais, onde as modificações nos conceitos são mais sutis (figura 2(b)) e necessitam de um maior número de instâncias para ocorrerem (MINKU;

WHITE; YAO, 2010). Alguns autores dividem a mudança gradual em moderada e lenta, dependendo da velocidade da mesma (STANLEY, 2003). Também é valido mencionar as mudanças recorrentes, embora sejam menos estudadas e seu tratamento não seja foco desta investigação. As mesmas são definidas como uma mudança que acontece temporalmente e que retorna a seu estado normal depois de um intervalo de tempo como pode ser percebido figura 2(c).



(a) (b) (c)

Figura 2 – Tipos de mudanças de conceitos de acordo à velocidade. **(a)** Mudança abrupta, **(b)** Mudança gradual, **(c)** Mudança recorrente.

O tratamento das mudanças de conceito é um problema muito desafiador, mas a rápida detecção e manipulação das mesmas constitui o foco de várias pesquisas, já que as mudanças de conceito com frequência impactam significativamente no rendimento do modelo de aprendizagem, de maneira que o modelo pode não mais ser válido. Portanto, é aconselhável que os algoritmos de aprendizagem incremental incorporem mecanismos adicionais para se adaptar às mudanças de conceito. O algoritmo de aprendizagem deve ser capaz de capturar e manter-se atualizado com respeito aos possíveis padrões transitórios subjacentes no fluxo de dados (WANG et al., 2003). Assim, o modelo de aprendizagem deve ser constantemente atualizado, aprendendo da nova informação e esquecendo experiências que representam conceitos passados (BLANCO; INOCENCIO, 2014).

As metodologias para enfrentar as mudanças de conceito nos ambientes de fluxos de dados usadas pelos pesquisadores são muitas, mas vale mencionar cinco das principais (GONÇALVES JR.; BARROS, 2013; MACIEL; SANTOS; BARROS, 2015), sendo elas:

- Adaptação de classificadores originalmente propostos para funcionarem em modo batch;
- Detecção de mudanças de conceito, seguida pela criação de um novo classificador a fim de representar o novo contexto;
- Combinação de classificadores;
- Combinação de detectores;

- Manutenção dos dados relativos aos conceitos aprendidos pelos classificadores com a ideia de lidar com mudanças de conceitos recorrentes.

De forma geral, um sistema que manipule mudanças de conceito deve ser capaz de adaptar-se rapidamente à mudança, ser robusto na distinção entre uma verdadeira mudança de conceito e ruído, assim como reconhecer e tratar conceitos. A distinção entre uma verdadeira mudança de conceito e o ruído é um problema de grande dificuldade. Alguns algoritmos podem ser muito susceptíveis ao ruído, realizando uma interpretação errônea do mesmo como uma mudança de conceito. Enquanto outros podem ser muito robustos, mas adaptar-se muito lentamente à mudança. Na seguinte seção são apresentados vários métodos que constituem parte do estado da arte na detecção de mudanças de conceitos.

2.4 Manipulação de mudanças de conceito

Na seção anterior foi detalhado que 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. Nesta seção será realizado um levantamento bibliográfico dos algoritmos (classificadores, detectores e ensemble de detectores) que serão usados no Capítulo 6 para a comparação com a nova proposta, ou que contribuem de alguma forma em sua construção. Cada um dos métodos detectores utilizados neste trabalho e descrito a seguir tem como característica sua execução em paralelo com um classificador base, que comumente são chamados de métodos adaptativos. O classificador, 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 conceitos ou não, geralmente observando a quantidade de erros sequencialmente cometidos pelo classificador base (BRZEZINSKI; STEFANOWSKI, 2016). Em geral, os detectores trabalham com dois níveis de alarmes: warning e drift (ATTAR et al., 2012), onde, o drift representa um nível maior de mudança na distribuição analisada e simboliza que, de fato, ocorreu uma modificação de conceito. Assim sendo, quando o nível de warning é sinalizado, uma nova instância do classificador base é criada e mantida em paralelo com o classificador antigo. Caso o nível de drift seja alcançado, o detector exclui o antigo classificador e mantém apenas o novo. Por outro lado, caso o sinal de warning passe a ser considerado um alarme falso, a nova instância do classificador é excluída.

A seguir apresentamos seis detectores que são parte importante dos enfoques providos para o tratamento de mudança de conceitos e encontram-se disponíveis no framework para a mineração de fluxos de dados Massive Online Analysis (MOA). Além disso, pela sua importância na investigação foram incluídos o detector WSTD e três comitês (ensembles) de detectores, embora estes não estejam ainda disponíveis na distribuição oficial do MOA.

2.4.1 Adaptive Windowing (ADWIN)

ADWIN (BIFET; GAVALDÀ, 2007) tem como ideia básica usar uma janela deslizante de instâncias (W) para detectar mudança de conceitos. O tamanho de W é ajustado automaticamente, sendo de maior tamanho quando os dados permanecerem na mesma distribuição de probabilidade por mais tempo. Com o objetivo de manter em W somente instâncias com uma mesma distribuição, W será reduzida quando mudanças ocorrerem.

Duas sub-janelas dinamicamente ajustáveis (W_0 e W_1) obtidas da divisão de W são armazenadas, representando os dados antigos e recentes. Quando a diferença entre as médias ($\hat{\mu}_{W_0}$ e $\hat{\mu}_{W_1}$) é maior que um dado limite (ϵ_{cut}, apresentado na equação 2.1) notifica-se a mudança.

$$\begin{aligned} \epsilon_{cut} &= \sqrt{\frac{2}{m}.\sigma^2_W.\log\left(\frac{2}{\delta'}\right)} + \frac{2}{3m}.\log\left(\frac{2}{\delta'}\right) \\ \delta' &= \delta/\log(n) \\ m &= \frac{1}{\frac{1}{n_0}+\frac{1}{n_1}} \end{aligned} \tag{2.1}$$

No ADWIN é calculada a média harmônica (m) de n_0 e n_1, os quais referem-se aos tamanhos de W_0 e W_1 respectivamente. Além disso, σ^2_W 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.

Para garantir o ajuste do método a implementação apresenta dois parâmetros, o nível de confiança ($\delta \in (0,1)$), e a frequência mínima (f) de instâncias necessitadas para diminuir o tamanho da janela. Os valores padrão no MOA são $\delta = 0.002$ e $f = 32$.

2.4.2 Drift Detection Method (DDM)

DDM (GAMA et al., 2004) usa o erro da predição de um algoritmo de aprendizagem incremental como uma variável aleatória correspondente a experimentos de Bernoulli. É assumida uma distribuição binomial e considera-se que para um número grande de exemplos (instâncias) esta aproxima-se à distribuição normal, onde o erro na predição (p_i) e seu desvio padrão $s_i = \sqrt{p_i(1-p_i)/i}$ são calculados para cada instância (i). Se estabelece que o erro do algoritmo de aprendizagem (p_i) vai diminuir se o número de exemplos aumenta e a distribuição dos exemplos se mantém estacionária. Por outro lado, um aumento significativo no número de erros do algoritmo sugere que a distribuição das classes está mudando e, portanto, o modelo de decisão atual é inapropriado. O método para a detecção de mu-

danças armazena duas variáveis no treinamento do algoritmo de aprendizagem, (p_{min}) e (s_{min}), que são atualizadas quando uma nova instância causa que $p_i + s_i < p_{min} + s_{min}$.

DDM apresenta três estados:

- *em-controle* (do inglês, in-control) quando $p_i + s_i < p_{min} + w \times s_{min}$, a ser declarado estável o sistema, os autores assumem que a mesma distribuição contém os exemplos gerados.
- *Alerta* (do inglês, warning) quando $p_i + s_i \geq p_{min} + w \times s_{min}$, o nível avisa que o erro está aumentando, mas ainda não chegou ao nível considerado significativamente alto para declarar a mudança.
- *Mudança* (do inglês, drift) a mudança é detectada, já que $p_i + s_i \geq p_{min} + d \times s_{min}$ é satisfeita, portanto os valores p_{min} e s_{min} são reiniciados.

Os parâmetros do DDM por padrão no MOA para os níveis *warning* e *drift* são: $w = 2.0$, $d = 3.0$ respectivamente, e $n = 30$, onde n é o mínimo número de instâncias antes que a detecção das mudanças seja permitida. Este detector apresenta bom rendimento na detecção em cenários onde as mudanças abruptas e graduais não são muito lentas, mas nos cenários onde as mudanças graduais são muito lentas apresenta algumas dificuldades (BAENA-GARCIA et al., 2006).

2.4.3 EWMA for Concept Drift Detection (ECDD)

ECDD (ROSS et al., 2012) é baseado na ideia de Exponentially Weighted Moving Average (EWMA) (ROBERTS, 1959) que foi proposto para detectar um incremento na média (μ) de uma sequência de variáveis aleatórias, considerando que a média e o desvio padrão do fluxo são conhecidos. Matematicamente, o EWMA é formulado como ilustra-se na equação 2.2, onde $Z_t = (1-\lambda)Z_{t-1} + \lambda X_t$ é uma estimativa de μ no tempo (t). Já μ_0 e σ_{Z_t} são a média das instâncias antes da mudança e o desvio padrão de Z_t, respectivamente. O cálculo deste último é apresentado na equação 2.3. É importante observar que L é o limiar encarregado de definir a distância necessária para detectar uma mudança. Também, é valido mencionar o uso da variável aleatória X_t e o parâmetro que pondera o aumento das instâncias atuais em relação às mais antigas λ, presente no cálculo Z_t.

$$Z_t > \mu_0 + L\sigma_{Z_t} \tag{2.2}$$

$$\sigma_{Z_t} = \sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})\sigma_x} \tag{2.3}$$

Pelo disposto acima, e conhecendo que σ_x usado na equação 2.3 é o desvio padrão das variáveis aleatórias, e que X_t segue uma distribuição de Bernoulli com p_t probabilidade de

erro em t, os autores propõem o ECDD, já que uma variação observada na probabilidade pode ser uma mudança de conceito.

O ECDD detecta mudanças de conceito através da inequação 2.4. Os valores de $\hat{p}_0$, $\hat{\sigma}_Z$ e $\hat{\sigma}_x$ não são conhecidos a priori, sendo estimados como mostra o sistema de equações 2.5.

$$Z_t > \hat{p}_{0,t} + L_t \times \hat{\sigma}_{Z_t} \tag{2.4}$$

Dada uma taxa média aceitável de detecções de falsos positivos (ARL_0) por instâncias é procurado um valor adequado de L. Devido ao alto custo computacional que é encontrar o valor de L pelos métodos convencionais (VERDIER; HILGERT; VILA, 2008), os autores adaptam no ECDD o proposto por (SPARKS, 2000)

$$\begin{aligned} \hat{p}_{0,t} &= \frac{t}{t+1}\hat{p}_{0,t-1} + \frac{1}{t+1}X_t \\ \hat{\sigma}_{Z_t} &= \sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})\hat{\sigma}_{x_t}} \\ \hat{\sigma}_{x_t} &= \hat{p}_{0,t}(1-\hat{p}_{0,t}) \end{aligned} \tag{2.5}$$

A implementação de ECDD no MOA tem os parâmetros com seus respectivos valores padrões: $ARL_0 = 400$, o peso das instâncias $\lambda = 0.2$ (usado para diferenciar as instâncias recentes das antigas), o limite de alerta $w = 0.5$ (indica quando o método entra em nível de warning) e por último $n = 30$ que é o número mínimo de instâncias antes que a mudança possa ser detectada.

2.4.4 Fast Hoeffding Drift Detection Method (FHDDM)

FHDDM (PESARANGHADER; VIKTOR, 2016) usa uma janela deslizante de tamanho n (igual a 200 na implementação do MOA) dos resultados da classificação e a inequação de Hoeffding (HOEFFDING, 1963) para a detecção das mudanças de conceito. A janela conterá valores 0 e 1 correspondendo ao erro ou acerto do classificador respectivamente. Conforme as entradas são processadas, a probabilidade de observar $1s$ (p_t^1) na janela deslizante no tempo (t) é calculada. Adicionalmente, mantém a probabilidade máxima de ocorrência de $1s$ (p_{max}^1) atualizada como é apresentado na equação 2.6 para t.

$$if\, p_{max}^1 < p_t^1 \Rightarrow p_t^1 \rightarrow p_{max}^1 \tag{2.6}$$

Baseado no modelo de aprendizado Probably Approximately Correct (PAC) (MITCHELL, 1997) no detector FHDDM os autores demostram que a possibilidade de acontecer uma mudança de conceito aumenta caso p^1_{max} não mude e p^1_t diminua ao longo do tempo. Eventualmente uma diferença significativa entre p^1_{max} e p^1_t indica a ocorrência da mudança de conceito no fluxo de dados, como apresenta-se na equação 2.7.

$$\Delta p = p^1_{max} - p^1_t \geq \varepsilon d \Rightarrow Drift := True \tag{2.7}$$

O valor ε_d é calculado usando a probabilidade do erro δ (padrão 10^{-7}) fornecida pelo conceito de Hoeffding bound (HOEFFDING, 1963; MARON; MOORE, 1993) cujo teorema pode ser encontrado no anexo A.

2.4.5 Statistical Test of Equal Proportions (STEPD)

STEPD (NISHIDA; YAMAUCHI, 2007) tem como ideia principal usar uma abordagem de duas janelas para detectar as mudanças de conceitos. A janela recente contendo somente as últimas w instâncias, e a antiga com todas as instâncias conhecidas até o momento. Os autores usam na realidade para construir o detector um teste de hipóteses para comparação entre proporções de duas amostras independentes como mostra-se na equação 2.8.

O detector para realizar a comparação entre as precisões das duas janelas, usa o número de predições corretas sobre as n_o instâncias da janela antiga. O resultado é armazenado na variável r_o. Na comparação são excluídas as w instâncias recentes. O r_r é definida como o número de predições corretas sobre as w (n_r) instâncias, da janela atual, e o valor de $\hat{p}$ $= (r_o + r_r)/(n_o + n_r)$.

$$T(r_o, r_r, n_o, n_r) = \frac{|r_o/n_o - r_r/n_r| - 0,5(1/n_o + 1/n_r)}{\sqrt{\hat{p}\,(1-\hat{p})(1/n_o + 1/n_r)}} \tag{2.8}$$

Conhecendo que o detector trabalha conforme à suposição de igualdade entre as precisões de um classificador para w instâncias recentes e a precisão total computada desde o começo do processo de aprendizagem, a partir do momento que não tenha mudança do conceito. Além disso, é suposto que um declínio significativo na precisão da janela recente assinala uma eventual mudança de conceito.

O teste de hipóteses apresentado na equação 2.8 é equivalente ao teste de Qui Quadrado com correção de continuidade de Yates. O critério de decisão é determinado pelo valor p encontrado através da tabela da distribuição normal padrão, usando o consequente de $T(r_o, r_r, n_o, n_r)$. Sendo o valor p menor do que o nível de significância tomado (α_w para $warning$ e α_d para $drift$), a hipótese nula ($r_o/n_o = r_r/n_r$) será rejeitada e o detector de-

clara os estados de *warning* ou em $drift$ (de acordo com o valor de *alpha* (α) comparado) (NISHIDA; YAMAUCHI, 2007).

A implementação do método encontra-se no framework MOA, com valores padrões $w = 30$, $\alpha_w = 0.05$ e $\alpha_d = 0.003$. Segundo os autores do método, a estatística que mede a diferença entre as proporções dos acertos não é eficiente nos casos onde os tamanhos das janelas são extremamente pequenos.

2.4.6 Wilcoxon Rank Sum Test Drift Detector (WSTD)

O WSTD (BARROS; HIDALGO; CABRAL, 2018) faz uso para a detecção de mudanças de conceito do teste estatístico da soma dos ranks de Wilcoxon, o qual é usado estatisticamente para determinar se duas amostras independentes provêm de populações com a mesma distribuição nos dados (LARSON; FARBER, 2010).

Os autores como referência de partida para o desenvolvimento se baseiam no detector de mudanças de conceito STEPD. Como o método de origem, o WSTD monitora as predições do classificador base usando as duas janelas (recente e antiga), embora, a janela antiga usa um tamanho fixo (valor recomendado de 4000) e não todos os exemplos como o STEPD. Outra notável diferença é o teste estatístico usado na proposta. É válido o destacar da implementação realizada em WSTD, a adaptação e simplificação matemática dos cálculos dos ranks, usando a fórmula para calcular a soma dos elementos das séries aritméticas (AS) - progressões aritméticas finitas, tornando dispensável o uso de uma ordenação explícita como acontece no teste estatístico original. O anterior foi possível devido que as observações da classificação são binárias (0 ou 1).

$$z = \frac{R - \frac{n_r(n_o+n_r+1)}{2}}{\sqrt{\frac{n_o n_r(n_o+n_r+1)}{12}}} \tag{2.9}$$

Mostrando mais explicitamente na equação 2.9 a forma em que são comparadas as distribuições entre as duas janelas, onde R é a soma dos postos para a menor amostra. O critério de decisão que será levado em conta para rejeitar ou aceitar a hipótese nula depois de calculado o valor p, e que estabelece os estados de *warning* e $drift$, são iguais aos de STEPD.

2.4.7 A Lightweight Concept Drift Detection Ensemble (DDE)

DDE (MACIEL; SANTOS; BARROS, 2015) é um algoritmo de comitê de detectores que trabalha em função da probabilidade da ocorrência da mudança de conceito fornecida por três detectores individuais. Até certo ponto, a natureza desta proposta é semelhante à do método de detecção de mudanças de conceito Paired Learners (PL) (BACH; MALOOF, 2008), e tenta melhorar a predição sem afetar o tempo de execução. Em DDE os três detectores que o conformam compartilham a informação do mesmo classificador base.

A implementação conta de um importante parâmetro referente à sensibilidade (*sens*) com valor padrão 1 na implementação do MOA, sendo assim, unicamente necessária a predição de um detector para acionar o estado de alerta (warning) ou mudança. Os outros possíveis valores para *sens* (2 e 3) podem ser usados com o objetivo de obter melhores previsões em fluxos onde detecções de falsos positivos são muito prejudiciais aos resultados finais. Em tais casos, respectivamente, é aguardada a previsão de dois ou três detectores do ensemble para sinalizar os estados.

Outro parâmetro não menos importante é $maxWait$, exclusivamente usado quando o valor de *sens* é maior que 1. O parâmetro vai a conter o número específico de instâncias que podem ser aguardadas depois que um único detector informa a mudança, para que os restantes confirmem a mesma. Os autores estabelecem 100 instâncias como valor padrão. O não acontecimento da confirmação da detecção, é tomado como um falso positivo e o detector base volta ao estado estável.

2.4.8 Ensembles of Heterogeneous Concept Drift Detectors (EHCD^2)

Os autores de Ensembles of Heterogeneous Concept Drift Detectors - Experimental Study (EHCD^2) apresentam três regras para combinar detectores heterogêneos. Como se mostra na equação 2.10, eles assumem que se tem um conjunto de n detectores de mudanças.

$$\prod = \{D_1, D_2, ..., D_n\} \tag{2.10}$$

Para garantir a heterogeneidade, cada um dos detectores escolhidos tem a possibilidade de emitir um de três sinais possíveis, sendo:

$$D_i = \begin{cases} 0 & \text{se a mudança não é detectada} \\ 1 & \text{se o nível de warning } (alerta) \text{ é atingido} \\ 2 & \text{se a mudança é detectada} \end{cases} \tag{2.11}$$

As regras de decisão do comitê apresentadas somente levam em consideração a notificação do estado de mudança de conceito dos detectores que conformam o comitê. Essas regras podem ser enumeradas da seguinte maneira:

- At Least One Detects Drift (ALO): O comitê assume a ocorrência da mudança de conceito se ao menos um dos detectores informa a ocorrência da mudança.
- At Least Half of the Detectors Detect Drift (ALHD): A mudança de conceito é declarada se a metade dos detectores do comitê a informam.
- All Detectors Detect Drift (AD): Todos os detectores do comitê notificam a ocorrência da mudança.

Para cada uma das regras anteriores foi construído um comitê de detectores de mudanças de conceito. Segundo os autores Woźniak et al. (2016), os comitês apresentados nesta parte de experimentação, para provar o funcionamento das regras, não têm ótimo rendimento já que a escolha dos detectores que conformam os comitês foi realizada sem uma profunda análise, pois seu objetivo somente consistiu em propor novas metodologias para a criação de comitês (ensembles).

2.4.9 A Selective Detector Ensemble (e_Detector)

O comitê de detectores e-Detector (DU et al., 2014) foi desenvolvido para localizar mudanças de conceitos abruptas e graduais. Inicialmente se aplicam técnicas para juntar os detectores de acordo com a sua diversidade, e selecionar o número de detectores para fazer parte do comitê (ZHOU; WU; TANG, 2002). O e-Detector consiste das fases de aprendizagem online e a detecção de mudança de conceito.

O ensemble baseia-se na regra "O primeiro que encontra é o primeiro que reporta" para assumir os três possíveis estados:

- *em-controle*: Todos os detectores base do ensemble reconhecem que o fluxo é estável.
- *Alerta*: Se ao menos um dos detectores base notifica sinal de alerta.
- *Mudança*: Se precisa apenas que um dos detectores base notifique a mudança de conceito.

2.5 Classificadores

Nesta seção são apresentados os classificadores usados em nossos experimentos. Foi decidido utilizar HT e NB como classificadores bases (modelos de aprendizagem ou preditores) dentre outras propostas possíveis como K-nearest neighboors (KNN) (FUKUNAGA; NARENDRA, 1975), Support Vector Machine (SVM) (CORTES; VAPNIK, 1995) etc., por serem classificadores rápidos, disponíveis e amplamente usados em ambientes com fluxo contínuos de dados.

2.5.1 Naive Bayes (NB)

O NB (JOHN; LANGLEY, 1995) é um algoritmo de classificação conhecido por sua simplicidade e seu baixo custo computacional. O classificador aplica o Teorema de Bayes para realizar as predições, assumindo que todas as entradas são independentes (GAMALLO; GARCIA; FERNÁNDEZ-LANZA, 2013). Em termos compreensíveis, o classificador NB assume que a presença de uma característica particular em uma classe não está relacionada com a presença de qualquer outro recurso, pelo que é apresentado como ingênuo (OGURI,

2006). Como mostra-se na equação 2.12 para cada instância i não rotulada, o NB prediz uma classe C, com base na probabilidade a posteriori da classe C, dada a instância i.

Apesar de sua simplicidade e ingenuidade, o classificador é um dos mais utilizados no mundo para o aprendizado de máquinas (OGURI, 2006), reportando bom desempenho em várias tarefas de classificação.

$$C = \arg\max_{c^j \in C} p(c^j) \prod_{i=1}^{n} p(x_i|c^j) \tag{2.12}$$

2.5.2 Hoeffding Tree (HT ou VFDT)

Sendo um algoritmo incremental, capaz de formar uma árvore de decisão indutiva e aprender a partir de um fluxo de dados, o HT (HULTEN; SPENCER; DOMINGOS, 2001) assume que os exemplos da distribuição com os dados analisados não se alteram ao longo do tempo.

Este classificador explora o fato de que uma pequena amostra pode ser suficiente para a escolha de um atributo ótimo para a divisão. Tal ideia é apoiada, matematicamente, pelo Hoeffding Bound (HOEFFDING, 1963), o qual quantifica o número de instâncias necessárias para alcançar um certo nível de confiança. Partindo de ter n observações independentes da variável aleatória r cujo intervalo é R, o hoeffding bound afirma que, com probabilidade $1 - \delta$, a média real da variável é pelo menos $\bar{r} - \varepsilon$, onde ε está dado pela equação 2.13.

$$\varepsilon = \sqrt{\frac{R^2(\ln 1/\delta)}{2n}} \tag{2.13}$$

O que torna o Hoeffding Bound atraente é a capacidade de obter os mesmos resultados, independentemente da distribuição de probabilidade que gera as observações. No entanto, o número de observações necessárias para alcançar certos valores de δ e ϵ são diferentes em distribuições de probabilidade. A heurística utilizada para escolher os atributos de teste é o ganho de informação. Para uma melhor compreensão do algoritmo o pseudo-código do algoritmo é apresentado no anexo A.

2.6 Considerações Finais

Neste capítulo foram introduzidos vários conceitos inerentes à área de aprendizagem de máquina. Também foram apresentados algoritmos de detecção das mudanças de conceitos a serem usados na investigação como base do método novo, assim como para a comparação de rendimentos. Por último foi apresentado uma descrição dos classificadores aos quais o novo método de detecção ajudará a melhorar sua precisão.

3 TESTES ESTATÍSTICOS

Requerimentos de que a distribuição dos dados atenda ao pressuposto de normalidade e que as variâncias sejam estatisticamente iguais (homogêneas) são necessários na análise estatística paramétrica, especialmente na análise de variância (ANOVA). Contudo, caso uma dessas pressuposições não seja atendida, é necessário submeter os dados a transformações. Neste capítulo será explicado o teste ANOVA de um fator, assim como os testes de Brown-Forysthe e O'Brien que trabalham com dados transformados. Também são definidos os fundamentos básicos para realizar um teste de hipóteses e apresentadas as principais diferenças entre os testes paramétricos e não paramétricos.

3.1 Fundamentos dos Testes de Hipóteses

O teste de hipóteses ou teste de significância (GRAYBILL; IYER; BURDICK, 1998) é uma regra que tem como propósito fundamental verificar se os parâmetros de duas ou mais populações são estatisticamente iguais ou não, através de análises baseadas em suas amostras (DAVIS; MUKAMAL, 2006). Deste modo, esse modelo simplifica os questionamentos referentes às amostras analisadas em apenas duas hipóteses: nula (H_0) e alternativa (H_1). A H_0 é a declaração que está sendo testada. Normalmente, a H_0 é uma alegação de que não há nenhuma consequência ou nenhuma desigualdade. A H_1 é a alegação que você quer ser capaz de concluir sua veracidade, baseado nas evidências fornecidas pelos dados da amostra.

Em outras palavras, o teste baseado nos dados amostrais determina se devemos rejeitar a hipótese nula, para o qual usa uma de dois variantes possíveis na hora de aceitar ou rejeitar a hipótese. A primeira expõe o uso da Região crítica (Rc) definida como o conjunto de valores assumidos pela variável aleatória ou estatística de teste para os quais a hipótese nula é rejeitada, obtida para o nível de significância denotado com alfa (α). O α representa a probabilidade de incorretamente rejeitar a hipótese nula quando esta é verdadeira. Se o valor da estatística do teste cair dentro da região crítica, rejeita-se H_0. Ao rejeitar a hipótese nula (H_0) existe uma forte evidência de sua falsidade. Ao contrário, quando aceitamos, dizemos que não houve evidência amostral significativa no sentido de permitir a rejeição de H_0. A segunda variante usa um valor p definido como a probabilidade de obter uma estatística de teste igual ou mais extrema que a estatística observada a partir de uma amostra populacional, assumindo–se a hipótese nula como verdadeira. Na literatura, o valor p também é conhecido como a probabilidade de significância (BUSSAB, 2004). Se o valor p for menor que o nível de significância, então você pode rejeitar a hipótese nula.

O esquema comumente seguido para fazer um teste de hipóteses consta dos seguintes

passos (ALLUA; THOMPSON, 2009):

1. Definir as hipóteses H_0 e H_1 com base no problema;
2. Calcular a estatística do teste;
3. Definição de um nível de significância (α);
4. Obter a região crítica ou o valor p;
5. Estabelecer a regra de decisão do teste;
6. Análise e conclusão apoiada pelas hipóteses nula e/ou alternativa.

Pelo fato de serem usados resultados amostrais para realizar a inferência sobre a população, é possível ter erros associados, caracterizados como de tipo I e tipo II. O erro do tipo I é definido como a probabilidade de rejeitar H_0 quando ela é verdadeira. Por sua parte, o erro de tipo II é determinado como uma probabilidade que depende do valor real do parâmetro que está sendo testado. A mesma será grande para pequenos afastamentos e pequena no caso contrário.

3.2 Testes Paramétricos e Não Paramétricos

Quando se decide usar um teste de hipóteses, se deve tomar uma decisão prévia que é se utilizar os testes paramétricos ou os testes não paramétricos. Autores como (SIEGEL; JR, 1975; BLUMAN, 2014) apresentam os testes paramétricos como uma projeção para inferências que dependam do conhecimento de parâmetros populacionais, tais como: médias, variâncias, desvio padrão e proporções. Porquanto, várias suposições têm que ser atingidas para sua aplicação, como por exemplo a necessidade de que as amostras a serem testadas sejam suficientemente grandes para se assumir o conhecimento de algum parâmetro populacional. Além disso, pode-se aplicar um teste paramétrico sempre que as populações que originaram as amostras analisadas cumpram ao critério da normalidade.

Por outro lado, os testes estatísticos não paramétricos podem ser utilizados em inferências que não precisem do conhecimento dos parâmetros populacionais (SIEGEL; JR, 1975; BLUMAN, 2014). Assim, a execução de um teste não paramétrico não depende do conhecimento de nenhum parâmetro populacional. Tampouco tem a necessidade que as distribuições das populações envolvidas obedeçam ao critério da normalidade.

3.3 Anova

Uma potente ferramenta estatística que se deve aos estudos do estatístico-genético Ronald Aylmer Fisher, autor do livro Statistics Methods for Research Workers (FISHER, 1925), é

a Análise da Variância (ANOVA, do inglês Analysis of Variance) (MASSART et al., 1997). A mesma é usada para o controle de processos na indústria, assim como nos laboratórios de análises para o controle de métodos analíticos. As diferentes aplicações do ANOVA podem ser agrupadas na estimação dos componentes de variação de um processo e na comparação de diversos conjuntos de dados (BOQUÉ; MAROTO, 2004).

ANOVA permite a separação das diversas fontes de variação como são o erro aleatório na medição e o fator controlado. Além disso, acontecem situações onde ambas as fontes são aleatórias e o ANOVA pode ser facilmente aplicado nestes casos. Aqueles experimentos que utilizam uma única variável independente (fator) e uma variável dependente se analisam mediante variância chamada de um fator (*one way*), e trata-se de comparar grupos ou amostras que se diferenciam sistematicamente em um só fator. A diferença para vários grupos ou amostras se atribuem a diferentes combinações de dois fatores, o ANOVA correspondente é chamado de dois fatores (*two way*). Trata-se de comparar grupos ou amostras que diferem sistematicamente em dois fatores.

Três tipos de hipóteses devem ser atingidos para de forma satisfatória usar o ANOVA:

1. ***Cada grupo de dados deve ser independente do resto:*** Entre as observações não há conexão alguma que não seja explicada pelos fatores controlados. O suposto não é tão claramente correto, mas se pode manter razoavelmente se os indivíduos são tomados de forma aleatória e a medição se faz separadamente para cada um.

2. ***Os resultados em cada grupo devem seguir uma distribuição normal:*** O suposto menos válido de obter em um cenário do mundo real.

3. ***A variância de cada grupo de dados não deve diferir de forma significativa:*** Conhecido como suposto de homoscedasticidade ou de igualdade de variâncias. Como os métodos de medida produzem variações de diferentes magnitudes e os valores esperados estão relacionados com os desvios típicos, manter o suposto se faz pouco viável. Vários são os métodos para conseguir que tal suposto seja satisfeito: número igual de indivíduos nos grupos, transformação das observações originais entre outras.

É importante ressaltar que se aceitam leves desvios das condições ideais, como que se pode tolerar certa distância do suposto de normalidade com mínimo efeito prático sobre as propriedades do ANOVA.

No ANOVA, a variação na resposta separa-se na variação entre os diferentes níveis do fator (os diferentes grupos) e a variação entre indivíduos dentro de cada nível. Assumindo que as médias dos grupos são iguais, a variação entre grupos é comparável à variação entre indivíduos. Se a primeira tem um valor muito maior que a segunda, se pode indicar que as médias em realidade não são iguais. Como uma explicação mais direta podemos dizer que o objetivo principal do ANOVA é contrastar se existem diferenças entre as diferentes

médias dos níveis das variáveis (fatores). Tem que ser conhecido que a quantidade de médias que serão comparadas deve ser maior ou igual a duas.

Na pesquisa só será usada a prova ANOVA padrão de um fator, que se baseia na comparação das somas de quadrados médios por conta da variabilidade entre grupos assim como à variabilidade dentro dos grupos. Ambas somas são estimações independentes da variabilidade global, de forma que, se o quociente entre a primeira e a segunda é grande, será maior a probabilidade de rejeitar a hipótese nula. Este quociente segue uma distribuição F-Snedecor com seus graus de liberdade. A prova tem a hipótese nula (H_0) que assume que as médias dos k grupos são todas iguais, enquanto que a hipótese alternativa (H_1) diz que ao menos uma das médias é diferente.

Como passo prévio para o cálculo de ANOVA temos a somas de quadrados (equações 3.1, 3.2, 3.3), conhecendo que k é o número de grupos como foi expressado na hipótese nula, n_i é o número de dados pertencentes a cada grupo com $i = 1, 2, ..., k$, $\bar{x}_i$ e S_i^2 são a média e variância de cada grupo respectivamente, e $\bar{x}$ é a média global. Para facilitar o manuseio dos dados, os mesmos geralmente são organizados em uma tabela de análise de variância como comumente se conhece, com o formato especificado na tabela 1:

Soma de quadrado entre grupo:

$$SQ(grupo) = \sum n_i(\bar{x}_i - \bar{x})^2 \tag{3.1}$$

Soma de quadrado do erro:

$$SQ(erro) = \sum(n_i - 1)S_i^2 \tag{3.2}$$

Soma de quadrado total:

$$SQ(total) = \sum(x - \bar{x})^2$$

$$SQ(total) = SQ(grupo) + SQ(erro) \tag{3.3}$$

Os quadrados médios calculam-se como mostram as equações 3.4, 3.5, 3.6.

Quadrado médio do grupo:

$$QM(grupo) = \frac{SQ(grupo)}{k-1} \tag{3.4}$$

Quadrado médio do erro:

$$QM(erro) = \frac{SQ(erro)}{n-k} \tag{3.5}$$

Quadrado médio total:

$$QM(total) = \frac{SQ(total)}{n-1} \tag{3.6}$$

Onde os Graus de liberdade (gl) são calculados da seguinte forma:

gl do numerador= $k - 1$

gl do denominador= $n - k$

O valor do estatístico de contraste F (também chamado F-statistic) é obtido da seguinte maneira:

$$F = \frac{QM(grupo)}{QM(erro)} \tag{3.7}$$

Apresentados todos os cálculos que são realizados no ANOVA até calcular o valor F, valor que tem que ser comparado com o valor $F_{critico}$ para os graus de liberdade e o nível de significância escolhido (α), sendo rejeitada H_0 se $F > F_{critico}$, ou seja, existem diferenças entre dois ou mais grupos. Outro enfoque para definir que hipóteses cumpre-se é usando o valor p (também chamado de nível descritivo ou probabilidade de significância). Satisfeito que o valor $p < \alpha$ se pode rejeitar a H_0, pelo que se pode interpretar que o valor p é o menor nível de significância com que se rejeitaria a hipótese nula.

Tabela 1 – Análises de Variância.

Fontes de variação	Soma de quadrados	Graus de liberdade	Quadrado médio	Valor de F
Grupos	SQ(grupo)	k-1	QM(grupo)	
Erro	SQ(erro)	n-k	QM(erro)	$F = \frac{QM(grupo)}{QM(erro)}$
Total	SQT	N-1		

3.4 Brown-Forsythe

A prova de Levene apresentada em (LEVENE, 1960) é considerada o teste padrão de homogeneidade de variância. Modificações da prova como é o teste Brown-Forsythe (BROWN; FORSYTHE, 1974) melhoram o procedimento para provar a homogeneidade (igualdade) das variâncias (LIM; LOH, 1996).

Em diversas bibliografias podemos encontrar que o teste Brown-Forsythe é chamado de W_{50}. Nesta prova a variável transformada r_{ij} é obtida como a diferença absoluta entre a resposta original y_{ij} e a mediana de seu grupo $\tilde{y}_i$ (NORDSTOKKE; ZUMBO, 2010). De forma mais explicita a realização do teste é formada pelos seguintes passos:

1. A mediana é calculada para cada grupo.
2. O valor da mediana é subtraído a cada valor do grupo que pertence (equação 3.8).
3. O teste de ANOVA como foi apresentado na seção anterior é aplicado usando as variáveis transformadas, e obtida a correspondente F-statistic que para a prova de Brown-Forsythe é chamado na bibliografia também como W-Statistic. O anterior não deve ser confundido com o coeficiente de concordância (W-statistic) usado para avaliar a concordância entre avaliadores.

$$r_{ij} = \mid y_{ij} - \tilde{y}_i \mid \tag{3.8}$$

Onde y_{ij} é a j-ésima observação do i-ésimo grupo.

O valor p neste teste geralmente é comparado contra um nível de significância igual a 5%, se o primeiro valor é o menor, as variâncias das populações não são iguais, o que quer dizer que H_0 é rejeitada.

3.5 O'brien

O teste de O'Brien (O'BRIEN, 1979; O'BRIEN, 1981) é uma das provas mais sensitivas na análise da Homogeneidade de variâncias. A ideia básica do teste é transformar a variável original de modo que a variável transformada reflita a variação da variável original (ABDI, 2007). O teste foi desenvolvido com a H_0 que as amostras vêm de populações com a mesma variância. Por quanto, a H_1 nega essa suposição (os exemplos vêm de populações com diferentes variâncias).

As operações computacionais para este teste são bastante simples. Nesta conjuntura, é usada uma função dos desvios em relação à média das amostras. Cada variável original v_{ij} é transformada usando a equação 3.9 (para grupos do mesmo tamanho), onde as médias de cada sub-grupo i é $\bar{v}_i = \sum_{j=1}^{n_i} v_{ij}/n_i$ com variância $s_i^2 = \sum_{j=1}^{n_i} \left(v_{ij} - \bar{v}_i\right)^2 / n_{i-1}$.

$$r_{ij} = \frac{\left(n_i - 1.5\right) n_i \left(v_{ij} - \bar{v}_i\right) - 0.5 s_i^2 \left(n_i - 1\right)}{\left(n_i - 1\right)\left(n_i - 2\right)} \tag{3.9}$$

O modelo estatístico padrão na análise de variância (ANOVA) é aplicado nas variáveis r_{ij} efetivamente. No entanto, hipóteses mais específicas, como contrastes, efeitos simples, etc., ou desenhos extremamente desequilibrados podem exigir algumas variações do ANOVA padrão. O teste estatístico é calculado baseado no F-statistic como mostra a equação 3.10, onde t é o número de amostras, $\bar{r}_i$ a média do n_i desvios absolutos dos grupos i, e $\bar{r}$ é a média total de $N = \sum_{i=1}^{t} n_i$ desvios absolutos.

$$F - statistic = \frac{\frac{\sum_{i=1}^{t} n_i (\bar{r}_i - \bar{r})^2}{(t-1)}}{\frac{\sum_{i=1}^{t} \sum_{j=1}^{n_i} (r_{ij} - \bar{r}_i)^2}{(N-t)}} \tag{3.10}$$

A região de rejeição é obtida mediante a comparação do valor crítico com o F-statistic calculado (enfoque clássico: $F-statistic \geq F_{\alpha,(df_1=t-1,df_2=N-t)}$) ou mediante a comparação valor p com α (enfoque p_{value}, $p_{value} \leq \alpha$), onde $df_1 = t-1$ and $df_2 = N-t$ são os graus de liberdade.

3.6 Considerações Finais

Este capítulo oferece uma detalhada introdução aos testes de hipóteses, assim como os pressupostos que regem a sua utilização. Finalizando com a apresentação dos testes de hipóteses ANOVA, Brown-Forysthe e O'Brien, que serão implementados em capítulos posteriores.

4 MÉTODOS DETECTORES DE MUDANÇA DE CONCEITOS SADD, BFDD e OBDD

A necessidade de ter métodos eficientes para a detecção de mudança de conceito levou à implementação de três métodos, fazendo a exploração das funcionalidades dos testes estatísticos ANOVA de um fator (ou ANOVA padrão), Brown-Forysthe e O'Brien. Na implementação, os cálculos correspondentes a cada teste foram reduzidos otimizando os mesmos para ambientes binários, variáveis com valores entre 0 e 1, obtidos da saída de um classificador.

Além disso, o trabalho se concentrou em teste com amostras de um mesmo tamanho. Para conseguir que os tamanhos das mesmas coincidam foi aplicado um cálculo de proporção da amostra maior com respeito à menor procedimento também adotado por (CABRAL, 2017).

Ademais os algoritmos apresentados neste capítulo são baseados no detector de mudança de conceito WSTD (BARROS; HIDALGO; CABRAL, 2018) e de igual forma são usados para a aprendizagem em cenários totalmente supervisionados onde todos os rótulos das classes são conhecidos com anterioridade e estão disponíveis rapidamente para serem comparados com a saída do modelo de classificação. Nas próximas seções serão apresentados os pseudo-códigos das implementações dos testes ANOVA padrão, Brown-Forysthe e O'Brien, assim, como suas explicações.

4.1 Implementação do detector SADD

Uma implementação do ANOVA padrão foi realizada e chamada de SADD (pelo seu nome em inglês, Standard Anova Drift Detection) na investigação, embora conhecendo que o teste ANOVA pode ser afetado pela dependência temporal dos dados, assim como pela ausência de normalidade e não cumprimento da homogeneidade de variância. As causas dos problemas anteriores podem ser comumente encontradas nos fluxos de dados. Para reduzir na implementação os efeitos que degradam o teste, foram usadas amostras de tamanhos grandes, além do fato que os tamanhos das mesmas foram ajustados proporcionalmente.

O teste estatístico ANOVA de um fator explicado no capítulo capítulo 3 foi implementado como apresenta-se no algoritmo 1. Nele é calculada a média de ambas as janelas, usando o número de predições incorretas e o número total das predições de cada janela, não sendo utilizada na soma as predições corretas pois contém valor zero. Lembrando que as predições são as saídas de um classificador base (capítulo 2).

Algoritmo 1: Implementação do detector SADD

Input: Data Stream S, Recent Window Size w, Drift Level α_d, Warning Level α_w, Older Window Size w_2

$storedPreds \leftarrow$ **new byte** $[w]$
$storedPreds_2 \leftarrow$ **new byte** $[w_2]$
$n_o \leftarrow n_r \leftarrow n_p \leftarrow w_o \leftarrow w_r \leftarrow w_p \leftarrow r_o \leftarrow r_r \leftarrow r_p \leftarrow 0$
$changeDetected \leftarrow$ **false**
foreach *instance* ***in*** s **do**
    if $changeDetected$ **then**
        reset $storedPreds, storedPreds_2$
        $n_o \leftarrow n_r \leftarrow w_o \leftarrow w_r \leftarrow r_o \leftarrow r_r \leftarrow 0$
        $changeDetected \leftarrow$ **false**
    Updates predictions in *older* and *recent* windows
    Updates stats of both windows: $n_o, n_r, w_o, w_r, r_o, r_r$
    $isWarningZone \leftarrow$ **false**
    if $n_o \geq w$ **then**
        $w_p \leftarrow$ **round**$(w_o \times w/n_o)$
        $r_p \leftarrow (w - w_p)$
        $averwp \leftarrow w_p/n_p$
        $averwr \leftarrow w_r/n_r$
        $averall \leftarrow (averwp + averwp)/2$
        $variano \leftarrow (averwp^2 * r_p) + ((1 - averwp)^2 * w_p)$
        $varianr \leftarrow (averwr^2 * r_r) + ((1 - averwr)^2 * w_r)$
        $SumSqinter \leftarrow (n_p * (averwp - averall)^2) + (n_r * (averwr - averall)^2)$
        $SumSqinside \leftarrow variano + varianr$
        $dfree \leftarrow (n_p + n_r) - 2$
        $f_{value} \leftarrow (SumSqinter * dfree)/SumSqinside$
        if $Double.isNaN(f_{value})$ **then**
            $f_{value} \leftarrow 0.0$
        $p_{value} \leftarrow$ **FProbability** $(|f_{value}|, 1, dfree)$
        $p_{value} \leftarrow 2\times\ p_{value}$
        if $p_{value} < \alpha_d$ **then**
            $changeDetected \leftarrow$ **true**
        else if $p_{value} < \alpha_w$ **then**
            $isWarningZone \leftarrow$ **true**

A implementação do método é baseada no detector de mudança de conceito WSTD tal como indicado acima, mantendo os mesmos 4 parâmetros básicos (entrada do algoritmo 1). Entretanto, os valores numéricos dos mesmos diferenciam-se da seguinte maneira: $w = 100$ (tamanho da janela recente), $\alpha_d = 0.001$ (nível de notificação da mudança), $\alpha_w = 0.5$ (nível de notificação de alerta), $w_2 = 200$ (tamanho da janela antiga), sendo a variável S a que representa o fluxo de dados. No esboço apresentado na figura 3 se mostra de forma resumida o fluxo de trabalho levado pelos três detectores implementados neste

capítulo.

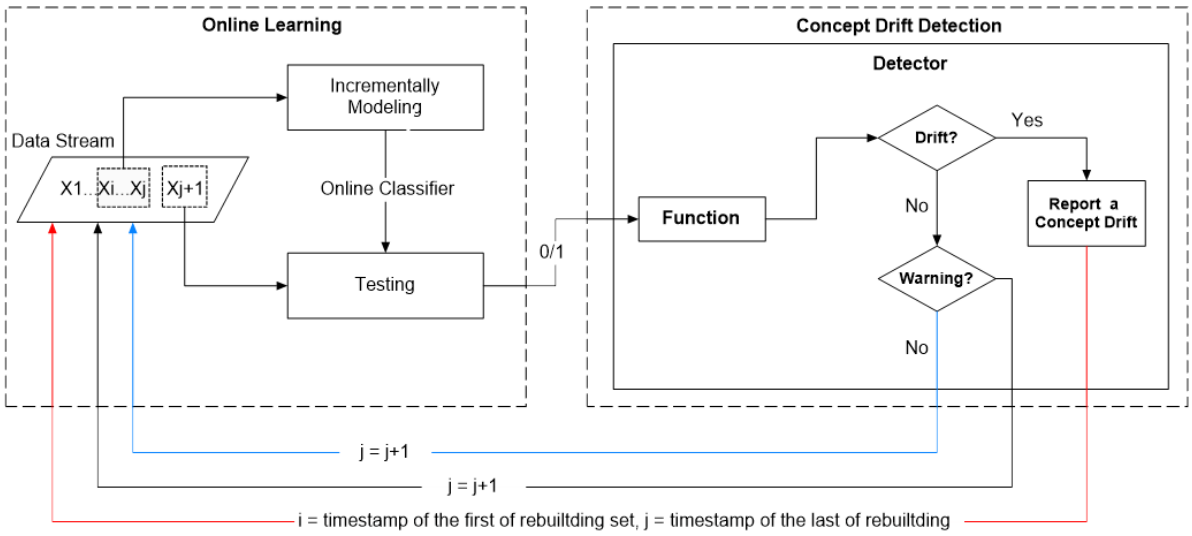


Figura 3 – Fluxo de trabalho seguido pelos detectores SADD, BFDD e OBDD.

Continuando a explicação do pseudo-código, nas linhas **1-4** é feita a inicialização das variáveis utilizadas pelo algoritmo. Em seguida, nas linhas **5-32**, é apresentada a parte principal do algoritmo. Nas linhas **6-9** é realizado o ajuste após a notificação da ocorrência de uma mudança de conceito. Em seguida, nas linhas **10-11** se abstrai as atualizações das predições armazenadas nas janelas antiga e recente, assim como das variáveis relacionadas com as predições armazenadas em cada janela. Já a linha **13** garante que as detecções ocorram após $2 \times w$ instâncias serem processadas, ou seja, que as detecções ocorram depois que a janela antiga possuir, no mínimo, w instâncias.

Nas linhas **14-15**, através das equações 4.1 e 4.2, o tamanho da janela antiga é ajustado proporcionalmente para se tornar igual ao da janela recente. As variáveis w_p e r_p representam, respectivamente, o número de erros e acertos da janela antiga com o tamanho proporcional. Além disso, w_o e r_o são os números de predições incorretas e corretas, respectivamente, sobre todos os n_o exemplos, com exceção dos w exemplos recentes.

$$w_p = w_o \times w/n_o \tag{4.1}$$

$$r_p = w - w_p \tag{4.2}$$

O ajuste permite não descartar nenhum dado antigo e assumiu-se que esse ajuste não comprometeria a precisão dos testes estatísticos a que estão sendo usados nesta investigação.

As linhas **16-17** apresentam o cálculo das médias para os valores proporcionais da janela antiga calculados, assim como a média dos dados da janela recente e uma média total que é resumida como a média das médias anteriormente apresentadas (linha **18**). Em seguida, nas linhas **19-20** é calculado o numerador da equação de variância.

Já nas linhas **21-22** são obtidas as somas de quadrados que são necessárias para o cálculo de quadrados médios. Os graus de liberdade também são calculados (linha **23**) e usados na linha **24** na obtenção do f_{value} (F-statistic) como foi apresentado na equação 3.7, valor que é filtrado nas linhas **25-26** devido a que valores indefinidos podem resultar do cálculo, provocado por possíveis divisões por zero.

A obtenção do p_{value} é realizada na linha **27** usando a função *FProbability()* presente no pacote Waikato Environment for Knowledge Analysis (Weka) (WITTEN et al., 2016) incluído no MOA. O p_{value} (valor p) é posteriormente avaliado nas linhas **28-32** para definir o estado em correspondência aos valores atingidos.

4.2 Implementação do detector BFDD

O teste estatístico Brown-Forysthe (capítulo 3) foi implementado como apresentado no algoritmo 2. Nele a definição explícita dos valores da mediana e os graus de liberdade do numerador para eliminar cálculos matemáticos levam à redução do tempo de execução. BFDD (Brown-Forysthe Drift Detector), como é chamada a nova proposta, mantém os mesmos 4 parâmetros de entrada que a proposta SADD como mostra-se no algoritmo 2. Porém, os valores numéricos dos mesmos foram ajustados para o melhor desempenho do teste, ficando da seguinte maneira fixados: $w = 150$ (tamanho da janela recente), $\alpha_d = 0.001$ (nível de notificação da mudança), $\alpha_w = 0.5$ (nível de notificação de alerta), $w_2 = 200$ (tamanho da janela antiga).

O desenvolvimento do algoritmo BFDD é conduzido pela mesma implementação realizada no método SADD nas linhas **1-15**. Nesse ponto começa a ser introduzida a estatística do método representando a principal diferença entre os algoritmos. Então, a continuidade é da seguinte forma: nas linhas **18-33** definem-se os valores da mediana e o valor transformado que será usado para o cálculo de F-statistic abstraído na linha **34**, sendo o resultado armazenado na variável f_{value}. O mesmo resultado é filtrado na linha **35-36** atribuindo o valor zero a f_{value} quando um valor indefinido acontece. Finalizando, a obtenção do p_{value} é realizada na linha **37** usando a função *FProbability()*. O p_{value} é posteriormente avaliado nas linhas **38-42** para colocar o estado em relação aos valores atingidos.

Algoritmo 2: Implementação do detector BFDD

Input: Data Stream S, Recent Window Size w, Drift Level α_d, Warning Level α_w, Older Window Size w_2

$storedPreds \leftarrow$ **new byte** $[w]$
$storedPreds_2 \leftarrow$ **new byte** $[w_2]$
$n_o \leftarrow n_r \leftarrow n_p \leftarrow w_o \leftarrow w_r \leftarrow w_p \leftarrow r_o \leftarrow r_r \leftarrow r_p \leftarrow 0$
$changeDetected \leftarrow$ **false**
foreach *instance* ***in*** s **do**
    if $changeDetected$ **then**
        reset $storedPreds, storedPreds_2$
        $n_o \leftarrow n_r \leftarrow w_o \leftarrow w_r \leftarrow r_o \leftarrow r_r \leftarrow 0$
        $changeDetected \leftarrow$ **false**
    Updates predictions in *older* and *recent* windows
    Updates stats of both windows: $n_o, n_r, w_o, w_r, r_o, r_r$
    $isWarningZone \leftarrow$ **false**
    if $n_o \geq w$ **then**
        $w_p \leftarrow$ **round**$(w_o \times w/n_o)$
        $r_p \leftarrow (w - w_p)$
        if $r_p > w_p$ **then**
            $median_p \leftarrow 0.0$
            $tranfw_p \leftarrow 1.0$
        else if $r_p < w_p$ **then**
            $median_p \leftarrow 1.0$
            $tranfw_p \leftarrow 0.0$
        else
            $median_p \leftarrow 0.5$
            $tranfw_p \leftarrow 0.5$
        if $r_r > w_r$ **then**
            $median_r \leftarrow 0.0$
            $tranfw_r \leftarrow 1.0$
        else if $r_r < w_r$ **then**
            $median_r \leftarrow 1.0$
            $tranfw_r \leftarrow 0.0$
        else
            $median_p \leftarrow 0.5$
            $tranfw_p \leftarrow 0.5$
        Calculation of F-statistic using the transformed values
        if $Double.isNaN(f_{value})$ **then**
            $f_{value} \leftarrow 0.0$
        $p_{value} \leftarrow$ **FProbability** $(|f_{value}|, 1, dfree)$
        $p_{value} \leftarrow 2\times \ p_{value}$
        if $p_{value} < \alpha_d$ **then**
            $changeDetected \leftarrow$ **true**
        else if $p_{value} < \alpha_w$ **then**
            $isWarningZone \leftarrow$ **true**

4.3 Implementação do detector OBDD

A implementação do teste estatístico de O'Brien (capítulo 3), como um algoritmo para a detecção de mudança de conceito é feita de forma semelhante ao detector BFDD apresentado na secção 4.2. A implementação é igualmente baseada no WSTD, conservando a mesma quantidade de parâmetros de entrada e numericamente inicializados igual que em BFDD. Esta proposta é conhecida como OBDD (O'Brien Drift Detector).

Nas linhas **1-15** o teste de O'Brien é implementado da mesma forma que o ANOVA de um fator (algoritmo 1). Já nas linhas **16-27** se apresenta a definição da mediana como mostra-se no pseudo-código do algoritmo 3. A transformação das variáveis originais são expostas nas linhas **28-31** sendo a principal diferença entre os dois métodos (capítulo 3). Na pesquisa foram feitas variações no teste original O'Brien em busca de melhores resultados, sendo usada a mediana ($\tilde{v}_i$) em substituição da média ($\bar{v}_i$) para o cálculo dos valores transformados (equação 4.3).

$$r_{ij} = \frac{(n_i - 1.5)\, n_i\, (v_{ij} - \tilde{v}_i) - 0.5 s_i^2\, (n_i - 1)}{(n_i - 1)\, (n_i - 2)} \tag{4.3}$$

A linha **32** abstrai o cálculo do F-statistic, cujo resultado é armazenado na variável f_{value}, sendo filtrado nas linhas **33-34**, devido que uma divisão por zero pode acontecer e provocar a indefinição do valor de f_{value}. Por esta razão, é atribuído um valor a f_{value} nestes casos. Finalmente nas linhas **35-40** o p_{value} é calculado e comparado para estabelecer um estado.

Algoritmo 3: Implementação do detector OBDD

Input: Data Stream S, Recent Window Size w, Drift Level α_d, Warning Level α_w, Older Window Size w_2

$storedPreds \leftarrow$ **n**ew byte $[w]$
$storedPreds_2 \leftarrow$ **new byte** $[w_2]$
$n_o \leftarrow n_r \leftarrow n_p \leftarrow w_o \leftarrow w_r \leftarrow w_p \leftarrow r_o \leftarrow r_r \leftarrow r_p \leftarrow 0$
$changeDetected \leftarrow$ **false**
foreach *instance* ***in*** s **do**
    if $changeDetected$ **then**
        reset $storedPreds, storedPreds_2$
        $n_o \leftarrow n_r \leftarrow w_o \leftarrow w_r \leftarrow r_o \leftarrow r_r \leftarrow 0$
        $changeDetected \leftarrow$ **false**
    Updates predictions in *older* and *recent* windows
    Updates stats of both windows: $n_o, n_r, w_o, w_r, r_o, r_r$
    $isWarningZone \leftarrow$ **false**
    if $n_o \geq w$ **then**
        $w_p \leftarrow$ **round**$(w_o \times w/n_o)$
        $r_p \leftarrow (w - w_p)$
        if $r_p > w_p$ **then**
            $median_p \leftarrow 0.0$
        else if $r_p < w_p$ **then**
            $median_p \leftarrow 1.0$
        else
            $median_p \leftarrow 0.5$
        if $r_r > w_r$ **then**
            $median_r \leftarrow 0.0$
        else if $r_r < w_r$ **then**
            $median_r \leftarrow 1.0$
        else
            $median_r \leftarrow 0.5$
        $tranf_{r_p} \leftarrow ((r_p \times (r_p - 1.5) \times median_p^2) - (0.5 \times var_{w_p}))/((r_p - 1) \times (r_p - 2))$
        $tranf_{w_p} \leftarrow$
        $((w_p \times (w_p - 1.5) \times (1 - median_p)^2) - (0.5 \times var_{w_p}))/((w_p - 1) \times (w_p - 2))$
        $tranf_{r_r} \leftarrow ((r_r \times (r_r - 1.5) \times median_r^2) - (0.5 \times var_{w_r}))/((r_r - 1) \times (r_r - 2))$
        $tranf_{w_r} \leftarrow$
        $((w_r \times (w_r - 1.5) \times (1 - median_r)^2) - (0.5 \times var_{w_r}))/((w_r - 1) \times (w_r - 2))$
        Calculation of F-statistic using the transformed values
        if $Double.isNaN(f_{value})$ **then**
            $f_{value} \leftarrow 0.0$
        $p_{value} \leftarrow$ **FProbability** $(|f_{value}|, 1, dfree)$
        $p_{value} \leftarrow 2\times \ p_{value}$
        if $p_{value} < \alpha_d$ **then**
            $changeDetected \leftarrow$ **true**
        else if $p_{value} < \alpha_w$ **then**
            $isWarningZone \leftarrow$ **true**

4.4 Considerações Finais

No capítulo foram apresentados SADD, BFDD e OBDD, três detectores de mudanças de conceitos baseados nos testes estatísticos ANOVA de um fator, Brown-Forysthe e O'Brien, com o objetivo de comparar com o comitê de métodos estatísticos que utiliza os mesmos testes, assim como de atingir melhores resultados em comparação com as propostas do estado da arte selecionadas. Os três testes acima mencionados foram escolhidos pelo fato de anteriormente não serem utilizados na construção de detectores de mudanças de conceitos. Além disso pela versatilidade deles e sua compatibilidade, pelo fato que Brown-Forysthe e O'Brien utilizarem ANOVA de um fator em seus cálculos, permitindo uma fácil integração para a construção da nova proposta a ser apresentada no capítulo 5. Os algoritmos fazem uma implementação atrativa dos testes, onde os mesmos são reduzidos com a definição de valores fixos, resumindo e eliminando cálculos matemáticos, atingindo assim uma diminuição do tempo de execução. A comparação e análises dos resultados obtidos pelos três detectores apresentados são expostos no capítulo 6.

5 MÉTODO PROPOSTO PARA A DETECÇÃO DE MUDANÇA DE CONCEITOS EM FLUXOS DE DADOS

Na atualidade, a construção de métodos para a detecção de mudanças de conceito que trabalham independentemente do algoritmo de aprendizagem está tendo um perceptível incremento. Também conhecidos como métodos adaptativos na literatura, onde a maioria destas propostas verifica mudanças nas distribuições dos dados com o uso de um teste estatístico (ROSS; TASOULIS; ADAMS, 2011; BLANCO; INOCENCIO, 2014), modificados para cenários de fluxos de dados (BARROS; HIDALGO; CABRAL, 2018). No entanto, poucos detectores do estado da arte combinam vários testes estatísticos na hora de detectar mudanças de conceitos (ALIPPI; BORACCHI; ROVERI, 2011; ALIPPI; BORACCHI; ROVERI, 2017). Este capítulo apresenta um novo método para a detecção de mudanças de conceitos que inclui vários testes estatísticos na hora de notificar as mudanças de conceitos.

5.1 Implementação proposta ANOVA_C

A detecção das mudanças de conceitos tem sido abordada de várias maneiras, exemplos são apresentados a seguir: Gama et al. (2004), Baena-Garcia et al. (2006), Bifet e GAVALDÀ (2007), Nishida e Yamauchi (2007), Ross et al. (2012), Žliobaitė et al. (2015), pesaranghader e Viktor (2016), Barros et al. (2017), Barros, Hidalgo e Cabral (2018). Como foi exposto anteriormente, o uso de testes estatísticos para o tratamento das mesmas está sendo amplamente explorado. Embora os testes estatísticos usados em muitos dos métodos de detecção de mudanças de conceitos atuais trabalham baixo a suposição que as observações no fluxo de dados são distribuídas independentemente. Assim como que a probabilidade de um erro, dada uma classe são iguais (i.e $P(\frac{erro}{A} = \frac{erro}{B})$ onde A e B são as classes). Porém, muitas propostas violam os princípios necessários para a realização satisfatória dos testes. Por exemplo o DDM, EDDM, ADWIN e STEPD, como foi apresentado em (ŽLIOBAITĖ et al., 2015).

Ademais, tendências usadas nas tarefas de classificação são exportadas à construção de métodos detectores de mudanças de conceitos. Uma tendência que na atualidade está apresentando ótimos resultados é o desenvolvimento de comitês de classificadores, baseados na filosofia de que a opinião conjunta de vários especialistas tem melhor probabilidade de acertar na decisão do que a opinião de um único especialista, servindo como inspiração à construção de comitês de detectores, embora, geralmente, dita construção seja associada a problemas como alto consumo de tempo e memória. Além do mais, que a decisão

de escolher os detectores que conformam o comitê e como buscar a melhor maneira de combiná-los estruturalmente assim como as suas saídas pode ser muito custoso.

Com objetivo de propor um novo enfoque para a detecção e tratamento das mudanças de conceitos baseado em métodos estatísticos, com a mesma filosofia dos comitês de detectores, mas que leve em conta as premissas necessárias para a utilização dos testes estatísticos e que diminua o custo computacional associado à construção do comitê, é desenvolvido um comitê de métodos estatísticos chamado de ANOVA_C.

O ANOVA_C reúne os três métodos estatísticos que foram apresentados no capítulo 3. Na figura 4 se apresenta o seu funcionamento. Esta nova proposta adota a mesma configuração usada no algoritmo SADD. Como no SADD, são usadas duas janelas (antiga e recente) com os valores da quantidade de predições armazenadas $w_2 = 200$ e $w = 100$, respectivamente. É válido destacar que o ajuste do tamanho da janela recente, no lugar do comumente utilizado valor 30, foi definido como maior que 50, baseado em garantir a não degradação do método quando existem dependências temporais nos dados (ŽLIOBAITĖ et al., 2015). Também os valores padrões fixados para mudar os estados alerta (*warning*) ou mudança (*drift*) são mantidos os mesmos usados no SADD, sendo $\alpha_w = 0.5$ e $\alpha_d = 0.001$, respectivamente. Também a nova proposta será usada em ambiente de a aprendizagem totalmente supervisionada.

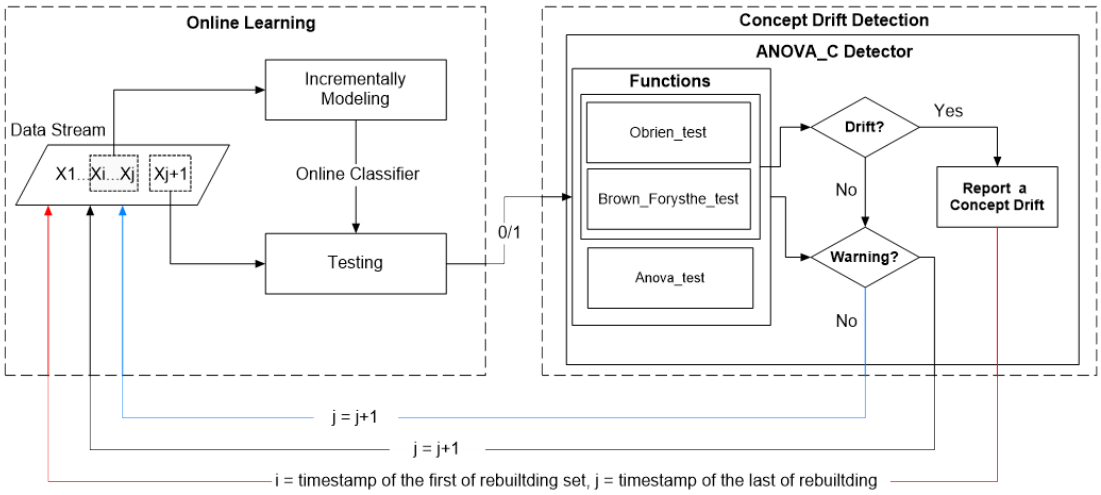


Figura 4 – Fluxo de trabalho seguido pelo comitê de métodos estatísticos ANOVA_C.

O pseudo-código do fluxo geral de execução do método ANOVA_C é apresentado no algoritmo 4. A implementação do ANOVA_C foi realizado de forma geral semelhante ao método SADD apresentado na seção 4.1. Embora, nota-se que o ANOVA_C adiciona na linha **4** a declaração e inicialização (com o valor 1) das variáveis ($vote_{Aw}$, $vote_{Bw}$, $vote_{Bd}$, $vote_{Ow}$ e $vote_{Od}$) que são usadas para armazenar o valor que indica se certo estado (alerta

ou drift) foi atingido, dado pela notificação de cada teste estatístico. É importante destacar que o valor 1 na variável indica que o estado não foi alcançado de acordo com os resultados obtidos pelo teste estatístico correspondente, caso contrário conterá zero.

Algoritmo 4: Implementação do detector ANOVA_C

Input: Data Stream S, Recent Window Size w, Drift Level α_d, Warning Level α_w, Older Window Size w_2

```
1  storedPreds ← new byte [w]
2  storedPreds_2 ← new byte [w_2]
3  n_o ← n_r ← n_p ← w_o ← w_r ← w_p ← r_o ← r_r ← r_p ← 0
4  vote_Aw ← vote_Bw ← vote_Bd ← vote_Ow ← vote_Od ← 1
5  changeDetected ← false
6  foreach instance in s do
7      if changeDetected then
8          reset storedPreds, storedPreds_2
9          n_o ← n_r ← w_o ← w_r ← r_o ← r_r ← 0
10         changeDetected ← false
11     Updates predictions in older and recent windows
12     Updates stats of both windows: n_o, n_r, w_o, w_r, r_o, r_r
13     isWarningZone ← false
14     if n_o ≥ w then
15         w_p ← round(w_o × w/n_o)
16         r_p ← w − w_p
17         Anova_test()
18         Brown_forysthe_test()
19         Obrien_test()
20         if vote_Bd + vote_Od ≤ 1 then
21             changeDetected ← true
22         else if vote_Aw + vote_Bw + vote_Ow ≤ 1 then
23             isWarningZone ← true
```

A principais diferenças entre o comitê de métodos estatísticos e o SADD se podem perceber a partir da linha **17**. O ANOVA_C faz as chamadas às funções que implementam os testes estatísticos de ANOVA, Brown-Forysthe, O'Brien, sendo apresentados seus pseudo-códigos nos algoritmos 5, 6 e 7, respectivamente. As funções chamadas de Anova_test, Brown_forysthe_test, Obrien_test alteram os valores das variáveis $vote_{Aw}, vote_{Bw}, vote_{Ow}$ (para informar o estado de alerta) e $vote_{Bd}, vote_{Od}$ (para o estado de mudança) conforme lhes correspondam. Nota-se que para a função Anova_test não é incluída uma variável para informar estado de mudança, já que a mesma não é usada pelo comitê de métodos estatísticos pela influência negativa que produz. Dado que sua inclusão gera alta quantidade de Falsos Positivos (FP, conceito a ser explicado no capítulo 6) principalmente quando as variâncias dos dados armazenados nas janelas comparadas são desiguais (ROGAN; KESELMAN, 1977).

As linhas finais do algoritmo (**20-23**) representam a sinalização dos estados de $drift$ e $warning$. E válido destacar que o $drift$ é notificado mediante a regra "o primeiro que encontra é o primeiro que reporta" (DU et al., 2014). Usando somente a implementação das duas funções que contém os testes estatísticos de Brown_Forysthe e O'Brien, respectivamente. O que significa que, assim que qualquer das duas funções reporte o drift, o comitê adota esse estado. A regra pode ser entendida como a chamada At Least One Detects Drift (ALO) apresentada em EHCD^2 (WOŹNIAK et al., 2016), já que possui o mesmo funcionamento, conforme descrito no capítulo 2.

Já o $warning$ é informado usando o voto majoritário (KUNCHEVA, 2004). Neste caso as três funções podem influenciar o resultado, mas se ao menos duas notificam o estado de alerta o mesmo é anunciado pelo comitê de métodos estatísticos ANOVA_C.

5.1.1 Implementação da função Anova_test

No algoritmo 5 se exibe a implementação da função Anova_test() que desenvolve o teste ANOVA de um fator que foi apresentado no capítulo 3. Alguns ajustes matemáticos na implementação foram usados com objetivo de permitir um melhor funcionamento do algoritmo, assim como sua adaptabilidade a entornos binários (variáveis contêm valor 0 ou 1).

A função Anova_test() é declarada na linha **1** do algoritmo 5. É uma função que não contém argumentos de entrada, nem retorna nenhum valor, seu objetivo é modificar a variável definida com alcance global $vote_{Aw}$. A implementação do teste estatístico ANOVA padrão contida na função foi realiza da mesma forma feita no detector SADD apresentado na sub-seção 4.1.

Neste caso o p_{value} pertencente a f_{value} para os graus de liberdade correspondentes e obtido como mostra a linha **17** pela função *FProbability()* e somente é usado para determinar estado de alerta (warning) como mostram as linhas **20-21**. Esta é a única informação fornecida por esta função para o comitê de métodos estatísticos ANOVA_C. Nota-se que na linha **19** a variável $vote_{Aw}$ será inicializada cada vez que o método seja chamado.

5.1.2 Implementação da função Brown_forysthe_test

Na literatura é afirmado que o teste de ANOVA, mesmo que não seja atingido o suposto de normalidade, funciona muito bem, a menos que uma ou mais das distribuições sejam muito assimétricas ou se as variâncias forem suficientemente diferentes. As transformações do conjunto de dados original podem compensar estas dificuldades. O teste Brown_forysthe surgiu com a necessidade de ter uma prova estatística mais robusta, em situações onde são transgredidos os pré-requisitos para aplicar o ANOVA padrão.

Algoritmo 5: Implementação da função Anova_test().

```
1  Function Anova_test()
2      /*Calculation average in proportion window, recent windows and the average
        total*/
3      averwp ← w_p/n_p
4      averwr ← w_r/n_r
5      averall ← (averwp + averwp)/2
6      /*Calculation of the numerator of the variance equation*/
7      variano ← (averwp^2 * r_p) + ((1 − averwp)^2 * w_p)
8      varianr ← (averwr^2 * r_r) + ((1 − averwr)^2 * w_r)
9      /*Calculation Sum of squares and mean squares*/
10     SumSqinter ← (n_p * (averwp − averall)^2) + (n_r * (averwr − averall)^2)
11     SumSqinside ← variano + varianr
12     /*Calculation of degrees of freedom and mean squares*/
13     dfree ← (n_p + n_r) − 2
14     f_value ← (SumSqinter * dfree)/SumSqinside
15     if Double.isNaN(f_value) then
16         f_value ← 0.0
17     p_value ← FProbability (|f_value|, 1, dfree)
18     p_value ← 2× p_value
19     vote_Aw ← 1
20     if p_value < α_w then
21         vote_Aw ← 0
```

O algoritmo 6 faz a implementação de uma função que contém o teste Brown_forysthe que foi apresentado no capítulo 3, para sua utilização pelo comitê estatístico. A declaração da função Brown_forysthe_test() é realizada na linha **1**. O cálculo da mediana e do valor da variável transformada são necessários os algoritmo, sendo apresentado nas linhas **3-20** a definição dos valores das mesmas da mesmo jeito como mostrado na implementação do detector BFDD na sub-seção 4.2.

A linha **21** abstrai o cálculo do valor de F-statistic que é realizado da mesma forma como descrito no algoritmo 5, só que os dados usados são os resultantes da transformação dos dados originais através da equação 3.8. Além disso, as linhas **22-23** garantem que o f_{value} não conterá valores inválidos. Também é modificado o sistema de notificação como exibem as linhas **28-31**, sendo adicionada a restrição para informar a mudança de conceito. Finalmente, a função aporta duas notificações (alarmes) para o comitê de testes estatísticos, representadas pelas variáveis $vote_{Bd}$ e $vote_{Bw}$ para os estados de mudança e alerta, respectivamente.

Algoritmo 6: Implementação da função Brown_forysthe_test().

```
1  Function Brown_forysthe_test()
2      /* Definition of the values of the mean and the transformed variable in
         proportion window*/
3      if r_p > w_p then
4          median_p ← 0.0
5          tranfw_p ← 1.0
6      else if r_p < w_p then
7          median_p ← 1.0
8          tranfw_p ← 0.0
9      else
10         median_p ← 0.5
11         tranfw_p ← 0.5
12     if r_r > w_r then
13         median_r ← 0.0
14         tranfw_r ← 1.0
15     else if r_r < w_r then
16         median_r ← 1.0
17         tranfw_r ← 0.0
18     else
19         median_p ← 0.5
20         tranfw_p ← 0.5
21     /*Calculation of F-statistic using the transformed values*/
22     if Double.isNaN(f_value) then
23         f_value ← 0.0
24     p_value ← FProbability (|f_value|, 1, dfree)
25     p_value ← 2× p_value
26     vote_Bd ← 1
27     vote_Bw ← 1
28     if p_value < α_d/2 then
29         vote_Bd ← 0
30     else if p_value < α_w then
31         vote_Bw ← 0
```

5.1.3 Implementação da função Obrien_test

Uma proposta sugestiva é o teste de O'Brien (O'BRIEN, 1979; O'BRIEN, 1981). Segundo Hatchavanich (2014) o mesmo é robusto frente à não-normalidade da distribuição dos dados. Além disso, é competitivo com outros testes em termos de poder, e pode ser facilmente aplicado em diferentes modelos ANOVA com tamanhos de amostra iguais ou diferentes. A investigação propõe modificar a equação 3.9 apresentada no capítulo 3, como já foi dito no capítulo 4. É válido destacar que, empiricamente, demostra-se que a

substituição da média $\bar{v}_i$ pela mediana $\tilde{v}_i$ na equação implica obter melhores resultados (equação 4.3).

Algoritmo 7: Implementação da função Obrien_test().

```
1  Function Obrien_test()
2      /*Calculation median in proportion window*/
3      if r_p > w_p then
4          median_p ← 0.0
5      else if r_p < w_p then
6          median_p ← 1.0
7      else
8          median_p ← 0.5
9      if r_r > w_r then
10         median_r ← 0.0
11     else if r_r < w_r then
12         median_r ← 1.0
13     else
14         median_r ← 0.5
15     tranf_{r_p} ← ((r_p × (r_p − 1.5) × median_p^2) − (0.5 × var_{w_p}))/((r_p − 1) × (r_p − 2))
16     tranf_{w_p} ← ((w_p×(w_p−1.5)×(1−median_p)^2)−(0.5×var_{w_p}))/((w_p−1)×(w_p−2))
17     tranf_{r_r} ← ((r_r × (r_r − 1.5) × median_r^2) − (0.5 × var_{w_r}))/((r_r − 1) × (r_r − 2))
18     tranf_{w_r} ← ((w_r×(w_r−1.5)×(1−median_r)^2)−(0.5×var_{w_r}))/((w_r−1)×(w_r−2))
19     /*Calculation of F-statistic using the transformed values*/
20     if Double.isNaN(f_value) then
21         f_value ← 0.0
22     p_value ← FProbability (|f_value|, 1, dfree)
23     p_value ← 2× p_value
24     vote_Od ← 1
25     vote_Ow ← 1
26     if p_value ≤ α_d then
27         vote_Od ← 0
28     else if p_value < α_w then
29         vote_Ow ← 0
```

O algoritmo 7 apresenta o pseudo-código da nova função implementada. Na linha **1** é definida a função Obrien_test(). Em seguida, os valores das medianas necessárias no cálculo são definidos nas linhas **3-8** para a proporção dos dados da janela antiga. Do mesmo modo, são definidos esses valores para os dados da janela recente (linhas **9-14**). Em seguida, uma transformação é aplicada às variáveis originais, onde a nova equação 4.3 é introduzida (linhas **15-18**).

A linha **19** abstrai o cálculo da F-statistic (f_{value}), que é explicado no algoritmo 5. Os dados usados no cálculo são os resultantes da transformação dos dados originais através

da equação 4.3. Uma filtragem ao f_{value} é realizado para garantir que não contém um valor indefinido NaN (*Not a Number*).

O sistema de notificação que é usado pela função Obrien_test é mostrado nas linhas **26-29**. Se pode perceber que a função aporta duas notificações para o comitê de métodos estatísticos, com o uso das variáveis $vote_{Od}$ e $vote_{Ow}$ para os estados de drift e warning, respectivamente.

5.2 Considerações Finais

O capítulo apresentou uma implementação estruturada do novo método para a detecção de mudança ANOVA_C. Na implementação se minimizaram cálculos com a definição dos valores da média, grau de liberdade do numerador, etc. A equação 4.3 foi aportada pela pesquisa já que testada empiricamente obteve melhores resultados que a original. O método declara o estado de warning (alerta) usando o voto majoritário contando com a informação fornecida pelos três testes estatísticos implementados. Ademais, o estado da ocorrência da mudança é declarado baseado na regra "o primeiro que encontra é o primeiro que reporta" (DU et al., 2014) mas usando somente a informação fornecida pelos métodos estatísticos de Brown-Forysthe e O'Brien.

6 ESTUDO EMPÍRICO E RESULTADOS

Neste capítulo, o framework de desenvolvimento Massive Online Analysis (MOA) (BIFET et al., 2010) é usado na pesquisa para comparar as propostas deste trabalho contra métodos anteriormente explicados no capítulo 2. O framework foi desenvolvido em linguagem de programação Java na Universidade de Waikato na Nova Zelândia e conta com licença GPL. MOA tem muita aceitação pelos pesquisadores e é fortalecida por sua fácil integração com o framework de mineração tradicional WEKA (BOUCKAERT et al., 2010). Para o melhor entendimento e uso do framework MOA é aconselhada a leitura do anexo B e B onde são expostas as facilidades do mesmo assim como foram usadas na pesquisa.

MOA é principalmente usado em tarefas de classificação ou agrupamento (*clustering*). Além disso, pode ser usado nas tarefas de regressão, tratamento de Outliers, e detecção de mudanças de conceitos (anexo B). Outras utilidades é que proporciona amostras e sequências de dados já que tem uma boa coleção de geradores de sequências de dados artificiais (anexo B, tabela 10), assim como a integração de bases de dados de cenários reais (anexo B, tabela 11).

Para avaliação do modelo de aprendizagem foi usada a metodologia Prequential (predictive sequential), que é uma técnica comumente usada tanto para sequências de dados que apresentam mudanças de conceito como para as que não tem. Na mesma o modelo é avaliado com cada nova instância que chega e depois usada para o treino (HIDALGO, 2017). O método no começo da avaliação tem grandes probabilidade de ter resultados ruims já que o modelo não está ainda bem treinado. É válido ressaltar que Prequential tem três variações: a janela deslizante (sliding window), o fator de desvanecimento (fading factors) e a usada nesta pesquisa (secção 6.3.1) chamada de janela básica (Basic Window ou Interleaved Test-Then-Train) (GAMA et al., 2014; LEMAIRE; SALPERWYCK; BONDU, 2015).

O computador usado na experimentação conta de um processador Core i7 4500U 1.8 GHz, sendo a memória RAM de 8 GB dual channel DDR 3 1600 MHz, disco rígido de 1 TB SATA 5400 RPM rodando um sistema operacional Ubuntu Desktop 15.10 64 bits. Os algoritmos propostos foram implementados em Java usando IDE Eclipse 4.6.0 (`https://www.eclipse.org/`). Além disso, é válido ressaltar que foi usado o algoritmo genético (AG) apresentado em (SANTOS; BARROS; GONÇALVES JR., 2015) para procurar melhores parâmetros aos métodos resultantes desta pesquisa. AG foi utilizado com bases de dados de vários tamanhos em todos os cenários de mudanças de conceitos, mas não foram encontrados melhor combinação dos mesmos que os inicialmente propostos pelo autor. Importante conhecer que as bases usadas pelo AG são diferentes as utilizadas na seção de experimentação desta investigação com o objetivo que os parâmetros obtidos foram

gerais e não adaptados as bases específicas usadas na pesquisa. Não menos importante foi a utilização da ferramenta (MOAManager) para gestão de experimentos realizados no MOA. Desenvolvida por Bruno Maciel e apresentada na qualificação de teses de doutorado. A ferramenta permite a realização dos experimentos, assim como a fácil extração, manipulação e análise dos dados resultantes (anexo B).

6.1 Bases de dados

Anteriormente foi exposto que o MOA permite gerar fluxos de dados artificias. Na pesquisa foram usados um total de quatro geradores artificias que serão explicados na sub-seção 6.1.1 e nove bases de dados reais que são de fácil integração no framework. É possível encontrar a explicação das mesmas na sub-seção 6.1.2.

6.1.1 Bases de dados artificiais

É necessário para a pesquisa construir sequências de dados com mudança de conceitos controlados para fazer as avaliações e comparações entre métodos. Entenda-se por mudanças de conceitos controladas aquelas onde conhecemos o número de mudanças, em que momento exato acontecem e o tempo em que a mudança vai sendo produzida no fluxo.

Portanto, para induzir as mudanças nos fluxos é usada uma função sigmoide $f(t) = \frac{1}{(1+e^{-s(t-t_0)})}$ (figura 5), presente no MOA. Essa função determina a mudança de uma origem de dados a outra.

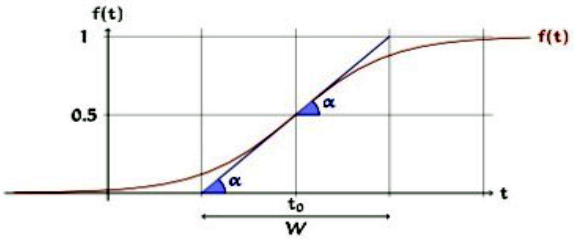


Figura 5 – Função sigmoide

Na função se tem o parâmetro t_0 definido como momento no que acontece a mudança, assim como α usado para ajustar a aspereza com a que se produz. Também pode ser usado para o ajuste W (quantidade de instâncias em que demora acontecer a mudança). Durante este intervalo serão misturados na sequência de dados instâncias de ambos conceitos. Outra possibilidade é gerar várias sequências de dados de forma independente e depois juntá-las de forma consecutiva em uma única sequência.

6.1.1.1 *Agrawal*

O gerador Agrawal (AGRAWAL; IMIELINSKI; SWAMI, 1993) pode construir sequências de dados usando 10 funções diferentes com 9 atributos. Dos quais o nível de escolaridade, marca do carro e código postal são categóricos. Já o salário, comissão, idade, valor do imóvel, idade do imóvel, e o valor do empréstimo desejado não são. A classificação pode ser em duas classes A e B, simulando a hipotética concessão de um empréstimo bancário. Agrawal é uma excelente opção para gerar mudanças de conceitos já que a classificação pode ser realizada com até dez funções diferentes, e o rótulo de cada instância poderá variar de acordo com a função utilizada. Além disso, foi usado para comprovar a escalabilidade em algoritmos de aprendizagem com árvores de decisão. Para uma melhor análise ler anexo B a tabela 12.

6.1.1.2 *LED*

LED (BREIMAN et al., 1984) é um gerador que produz uma sequência de dados com 24 atributos binários categóricos, dos quais 17 são irrelevantes. A tarefa a partir da sequência é prever o dígito mostrado em um visor LED de 7 segmentos onde cada atributo tem 10% de possibilidades de estar invertido. Uma mudança de conceito pode ser incluída indicando o número d de atributos com mudanças. Este conjunto de dados é originário do livro CART. Uma implementação em linguagem de programação C foi doada para o UCI (ASUNCION; NEWMAN, 2007) Machine Learning Repository por David Aha.

6.1.1.3 *Mixed*

Mixed (GAMA et al., 2004) gera uma sequência onde cada instância pode ser classificada como negativa ou positiva baseada nas análises dos quatro atributos com que conta, dos quais dois são booleanos (v, w) e dois numéricos (x, y). A classe será positiva quando ao menos duas das três seguintes condições forem cumpridas: os valores de v e w devem ser verdadeiros, e $y < 0.5 + 0.3 \times \sin(3\pi x)$. As mudanças de conceitos são simuladas mediante a inversão dos rótulos resultado da classificação, onde, para que uma instância seja considerada positiva, pelo menos duas das seguintes condições tem que ser cumpridas: os valores de v e w deverão ser falsos e $y \geq 0,5 + 0,3 \times \sin(3\pi x)$.

6.1.1.4 *Waveform*

A sequência construída pelo gerador Waveform (BIFET et al., 2010) foi apresentada também por Breiman et al. (1984) de forma semelhante à base LED. Existem dois geradores disponíveis para a sequência. A primeira versão, wave21, possui 21 atributos numéricos com ruído. A segunda, wave40, contem 19 atributos a mais que a primeira versão, sendo os mesmos irrelevantes. Em ambas as versões, o objetivo é diferenciar entre 3 classes diferentes de ondas que são resultado de uma combinação de 2 ou 3 ondas bases.

Existe a possibilidade de introduzir mudanças em um determinado número d de atributos para induzir uma mudança de conceito. Cada instância é gerada com ruído (média 0, variância 1). A taxa do erro de Bayes pode ser usada para derivar uma expressão analítica da sequência, sabendo que o erro Bayes ideal é 14% para uma amostra de teste de tamanho 5000.

6.1.2 Bases de dados reais

As nove bases de dados reais usadas nesta pesquisa encontram-se disponíveis em quase sua totalidade no endereço `http://archive.ics.uci.edu/ml/index.php`, pertencente ao UCI Machine Learning Repository (FRANK; ASUNCION, 2010). A escolha das mesmas foi feita baseado em ter na experimentação bases que diferem em tamanho assim como em números de classes.

6.1.2.1 *Airlines*

A base de dados binária Airlines (BIFET et al., 2013; SANTOS; BARROS; GONÇALVES JR., 2015) contém 539.383 instâncias. O objetivo nesta base é predizer se os voos estão atrasados ou não, com base em um conjunto de informações do voo: nome da companhia aérea, hora de partida, número do voo, duração, aeroporto de saída e chegada, e dia da semana.

6.1.2.2 *Cars*

A base de dados Car evaluation (Cars) contem 1.728 instâncias e 6 atributos. O objetivo é avaliar a aceitabilidade de um carro baseado no preço (compra e manutenção), assim como especificações técnicas e de conforto (número de portas, capacidade em termos de pessoas para transportar, o tamanho do porta-bagagens, segurança estimada do carro). Os rótulos das quatro possíveis classes são mencionados a seguir: inaceitável, aceitável, bom, e muito bom.

6.1.2.3 *Connect_4*

A Connect_4 é uma base de dados composta por 67.557 instâncias e 42 atributos (GODASE; ATTAR, 2012; WANKHADE; DONGRE; THOOL, 2012). A base de dados contém todas as combinações possíveis permitidas no jogo Connect_4 onde nenhum dos jogadores ganhou ainda e em qual o próximo movimento não é forçado. A base de dados tem três classes (win, loss, draw) e não tem presença de valores faltantes.

6.1.2.4 *CovSorted*

A base de dados Covertype (BIFET; HOLMES; PFAHRINGER, 2010) contém células de dados de 30×30 metros da Região 2 do Serviço Florestal de US. Tem 581.012 instâncias e 54

atributos, tanto numéricos como categóricos, o objetivo é prever o tipo de cobertura da floresta. Na investigação foi usada uma versão (CovSorted) apresentada por Ienco et al. (2013) onde a base é ordenada pelo atributo de elevação. Na CovSorted se induz mudanças de conceitos graduais na distribuição da classe: dependendo da elevação, alguns tipos de vegetação desaparecem enquanto outros começam a aparecer.

6.1.2.5 *Letter Recognition*

A versão original de Letter Recognition (FRANK; ASUNCION, 2010) é uma base de dados multi-classes para classificação, que possui 26 classes, 16 atributos e 20.000 instâncias. O objetivo é identificar letras maiúsculas do alfabeto inglês representadas por pixels retangulares em preto e branco.

6.1.2.6 *PokerHand*

Pokerhand (GONÇALVES JR.; BARROS, 2013; BIFET; HOLMES; PFAHRINGER, 2010) é uma base de dados onde se representa o problema de identificar o valor de uma mão de cinco cartas no jogo do Poker. Tem 5 atributos categóricos e 5 numéricos e o valor de uma mão (por exemplo, um par, dois pares, etc.) é uma classe categórica com 10 valores possíveis. Nesta versão modificada, disponível no site MOA, as cartas são ordenadas pelo rank e o terno, sendo removidas as duplicadas, resultando em 829.201 instâncias.

6.1.2.7 *Rialto*

Rialto (LOSING; HAMMER; WERSING, 2016) é uma base de dados que contém 82,250 instâncias, dez (10 classes) das coloridas construções próximas ao famoso Ponte de Rialto em Venice foram codificadas em um histograma normalizado de 27 dimensões RGB (27 atributos). As imagens são obtidas de espaço de tempos em capturas de vídeo por Webcam com posição fixa. As gravações cobrem 20 dias consecutivos durante maio-junho 2016. A mudança contínua do clima e das condições de iluminação afetam a representação.

6.1.2.8 *Usenet2*

Usenet2 (ASUNCION; NEWMAN, 2007; KATAKIS; TSOUMAKAS; VLAHAVAS, 2008b; KOHAIL, 2011) é uma versão da base de dados Usenet1. Esta base de dados é uma coleção de 20 grupos de notícias, é um fluxo de 1500 instâncias, dividido em cinco períodos de tempo. Os períodos contêm 300 instâncias. Depois da finalização de cada período a mudança de conceito acontece. A Simulação de um fluxo de mensagens de diferentes grupos de notícias que são apresentados sequencialmente a um usuário, quem então os etiqueta como interessantes (+) ou lixo (-), de acordo com seus interesses pessoais. É notado que a primeira base de dados Usenet1 é um conjunto de dados muito mais diversificado, com todas as categorias alterando a classe em cada período (KATAKIS; TSOUMAKAS;

VLAHAVAS, 2008a). Por sua parte Usenet2 é mais moderado nas mudanças de conceito (2 de 3 classes mudam de conceito todo o tempo). Ambas bases de dados apresentam mudanças de conceitos abruptas e recorrentes.

6.1.2.9 *Wine White*

Wine White é uma das duas bases de dados apresentada em (CORTEZ et al., 2009), baseadas nos dados de vinhos (vermelho e branco). Devido a problemas de privacidade e logística na base unicamente contém as variáveis físico-químicas (entradas) e sensoriais (a saída), não tem dados sobre tipos de uvas, marca de vinho, preço de venda de vinho, etc. A base possui 4.898 instâncias e 11 atributos mais a saída que é baseada em dados de sensoriais (mediana de pelo menos 3 avaliações feitas por especialistas em vinhos). Onde cada especialista classificou a qualidade do vinho entre 0 (muito ruim) e 10 (excelente). O objetivo é modelar a qualidade do vinho com base em testes físico-químicos.

6.2 Configuração da experimentação

Na pesquisa, a seleção dos componentes adequados para a realização dos experimentos levou mais de 35% do tempo planificado para o desenvolvimento da mesma. Nas seções seguintes serão apresentados todos os componentes que foram levados em conta na hora de fazer a experimentação. Procura-se ter uma experimentação a mais rigorosa e imparcial na hora de definir os melhores métodos de detecção de mudanças de conceitos. Este capítulo não é unicamente baseado em avaliar o rendimento do método (ANOVA_C), que é resultado da nova metodologia de construção de detectores de mudanças de conceitos proposta nesta pesquisa. Além do ANOVA_C, foi decidido avaliar também o comportamento dos demais detectores (SADD, BFDD, OBDD) propostos na investigação, pelo seu possível uso em cenários que não demande balanceamento nas métricas (sub-seção 6.2.2).

6.2.1 Descrição da experimentação

A seguir serão descritas todas as informações importantes na configuração dos experimentos para os testes e avaliação dos métodos.

Na investigação decidiu-se que todos os métodos de detecção de mudança de conceito do estado da arte, ao comparar com as novas propostas, conservaram os parâmetros padrões fornecidos pelos autores e mantidos nas implementações incluídas no MOA. Torna-se assim mais justa a experimentação. Os classificadores utilizados foram o HT e NB, escolhidos pelas suas simplicidades, além de estarem disponível no framework MOA e serem amplamente usados pelos pesquisadores.

Com o objetivo de construir bases de dados artificiais de três tamanhos diferentes que tenham presença de mudanças de conceitos abruptas e graduais, foram selecionados

quatro geradores (seção 6.1), obtendo-se no final um total de vinte e quatro combinações. Nas bases de dados geradas há quatro mudanças de conceitos distribuídos em intervalos regulares. Assim, o tamanho dos conceitos em cada versão de conjunto de dados do mesmo gerador (secção 6.1) é diferente, cobrindo quatro cenários diferentes. As mudanças de conceitos abruptas foram simuladas juntando conceitos diferentes, enquanto as mudanças graduais duram 500 instâncias antes de serem declaradas e foram geradas usando uma função de probabilidade para aumentar a chance de selecionar das instâncias do novo conceito em vez do antigo.

Devido ao grande tamanho das bases de dados artificias escolhidas, os experimentos para todas são executados somente dez (exemplos no anexo B) vezes para calcular a acurácia dos métodos e a média resultante foi calculada com um intervalo de confiança de 95%. Também nove bases de dados reais foram selecionadas para complementar a experimentação (secção 6.1.2).

6.2.2 Critérios de avaliação

Nos cenários com presença de mudanças de conceitos a não fácil definição e detecção das mesmas leva a pensar no emprego de diferentes critérios de avaliação para diferentes tipos de mudanças de conceitos (DU et al., 2014). Critérios de avaliação tradicional são comumente usados nos ambientes de mudanças abruptas dado por sua mais fácil definição e detecção que as graduais. Assim, podem ser definidos como critérios as quantidades de:

- **Verdadeiros Positivos** (conhecido como **TP**, por seu acrônimo em inglês) acontece quando o método declara mudança de conceito no fluxo de dados e a mesma ocorre em realidade;
- **Verdadeiros Negativos** (nomeado **TN**, por seu acrônimo em inglês) acontece quando o método não declara a ocorrência de mudança de conceito no fluxo de dados e a mesma em realidade não acontece;
- **Falsos Positivos** (com acrônimo em inglês, **FP**), é declarada a mudança de conceito mas a mesma não acontece no fluxo de dados;
- **Falsos Negativos** (tem como acrônimo em inglês, **FN**), neste caso de fato a mudança de conceito ocorre no fluxo de dados, mas o método não consegue detectá-lo.

Na atualidade, outros critérios que juntam as métricas anteriores são a precisão (do inglês,*Precision*), sensibilidade (do inglês, *Recall*) apresentadas em (PERRY; KENT; BERRY, 1955), assim como a taxa de falsos positivos, taxa de falsos negativos e Matthews Correlation Coefficient (MCC) (MATTHEWS, 1975), estão sendo muito usados para avaliar os detectores de mudança de conceito. O uso do MCC é destacado na investigação devido a que as outras métricas podem ser severamente influenciadas pela desigual quantidade

de FP e FN (LIU et al., 2016). O MCC retorna valores no intervalo $[-1, 1]$, sendo baseado nos quatro valores da matriz de confusão TP, TN, FP, e FN.

Outra métrica que está tendo aceitação pelos pesquisadores na hora de avaliar os métodos de detecção de mudanças de conceitos é o F_1-score (F_1, F-score ou F-measure) que é a média harmônica de precision e recall. Portanto, essa pontuação leva em conta falsos positivos e falsos negativos. F_1 geralmente é útil quando existe uma distribuição desigual de FN e FP. Uma pontuação F_1 atinge seu melhor valor em 1 (*Precision* e *Recall* perfeitos) e pior em 0.

Para um melhor entendimento das métricas ver o anexo B, onde se apresenta a formulação matemática para o cálculo das mesmas. Nesta investigação somente serão usados os valores da quantidade de FN, FP, *Precision*, *Recall*, F_1 e MCC para avaliar o comportamento dos métodos de detecção de mudanças de conceitos. Em cada um destes critérios de avaliação valores maiores identificam os melhores métodos.

6.3 Apresentação dos resultados

Na presente seção são exibidos os resultados obtidos de forma tabelada dos métodos propostos, assim como de outros métodos que formam parte do estado da arte, escolhidos principalmente por serem comumente utilizados em investigações anteriores. Também pelo fato que utilizam diferentes estratégias para detectar mudanças de conceitos, enfatizando nos métodos que usam duas janelas como as propostas desta investigação. Os métodos com melhores desempenhos são estabelecidos mediante o uso de testes estatísticos.

6.3.1 Acurácia

Como mencionado anteriormente, os detectores de mudanças de conceitos contribuem à não degradação do desempenho do modelo de aprendizagem. A métrica de medição de desempenho usada nesta pesquisa é a *acurácia prequential* (DAWID; VOVK et al., 1999), que representa o cálculo em tempo real de uma precisão média obtida pela predição de cada exemplo antes de ser aprendida (anexo B). Nas tabelas 2 e 3 são apresentados os resultados de acurácia de 10 métodos de detecção de mudança de conceitos, com os classificadores HT e NB respetivamente, dos quais 4 são resultados desta pesquisa.

Além disso, para o melhor entendimento do comportamento dos métodos se decidiu incluir nas tabelas os ranks resultantes determinados pelo teste estatístico de *Friedman* (DEMSAR, 2006). O cálculo dos ranks dos métodos foi realizado para cada uma das mudanças em todas as bases de dados artificias, assim como nas reais. Também foram calculados ranks de totalização. A seguir são apresentadas as abreviaturas presentes nas tabelas 2 e 3. A forma em que os ranks são calculados para a comparação de acurácia, assim como sua representação gráfica apresenta-se na sub-seção 6.3.2.

- Rank_Ab (fila que contém os rank dos métodos nas bases de dados artificias com presença de mudanças abruptas);
- Rank_G (fila que contém os rank dos métodos nas bases de dados artificias em cenários de mudanças graduais);
- Rank_Ar (fila que contém a rank total dos métodos nas bases de dados artificias);
- Rank_R (fila que contém a rank dos métodos nas bases de dados reais);
- Rank_T (fila que contém o rank total dos métodos).

Nota-se que os melhores valores dos rank e acurácia encontra-se em **negrito (bold)** nas tabelas. Os tamanhos escolhidos para produzir as bases de dados são três, contendo 500k, 1M (milhão) e 2M de instâncias para os quatro geradores usados (sub-seção 6.1.1) na pesquisa. No caso de Agrawal, unicamente foram usadas as cinco primeiras funções (F1 a F5).

Na tabela 2 identifica-se o ANOVA_C como o método melhor posicionado nos ranks nas diferentes mudanças em todas as bases de dados, tanto artificias como reais. O posicionamento anterior está dado pelo fato que a acurácia obtida pelo classificador HT com a utilização do comitê de métodos estatísticos tem resultados relevantes nas bases de dados construídas pelo gerador Agrawal (principalmente nos cenários de mudanças abruptas). Além disso, mantém um comportamento estável nas demais. Nota-se também que o método BFDD, apoiado em ter o melhor desempenho nos fluxos pertencentes ao gerador LED, é o segundo com melhor rank nas mudanças graduais. Já o OBDD, de forma geral, é o detector com a segunda melhor colocação nos ranks, ajudado pela apresentação de ótimos rendimentos nas mudanças graduais nas bases de dados criadas pelo gerador Agrawal. Também é possível observar na tabela 2 o inferior desempenho do detector SADD em ralação à quase a totalidade dos demais métodos em ambos tipos de mudanças.

Já com o classificador NB, o comitê de métodos estatístico ANOVA_C apresenta os melhores resultados nos cenários de mudanças abruptas, como mostra a tabela 3. Correspondendo com que nas bases criadas pelo gerador Waveform, se podem perceber os melhores resultados. Embora o ANOVA_C, nas mudanças graduais, não seja o método melhor posicionado, este se comporta semelhante aos demais métodos melhores posicionados. Em geral, quando é realizada a análise nas bases artificias e reais, o comitê de métodos estatísticos apresenta um melhor posicionamento comparado aos demais métodos. É válido destacar o comportamento estável mantido pelo detector OBDD para ambos tipos de mudanças, apresentando os melhores resultados nas bases de dados criadas pelo gerador LED. De forma geral, o OBDD obtém o segundo melhor rank, superado unicamente pelo ANOVA_C. Também é possível perceber que o comportamento dos detectores

SADD e BFDD são inferiores, não apresentando resultados relevantes em nenhum dos tipos das bases utilizadas (artificias e reais). Assim, esse baixo desempenho se reflete em suas respectivas posições no rank.

Tabela 2 – Médias de acurácias dos detectores em (%) utilizando o classificador HT, com 95% de Intervalo de Confiança nas bases de dados artificias.

$\frac{DT}{\#}$	DATASET	ADWIN	DDM	ECDD	STEPD	FHDDM	WSTD	SADD	BFDD	OBDD	ANOVA_C
	Agrawal	66.86	76.88	66.81	68.36	75.90	76.81	66.67	78.05	78.16	**78.25**
Abrupt	LED	65.09	70.95	69.15	69.51	72.94	73.25	68.41	**73.60**	73.58	73.59
500K	Mixed	92.15	94.80	89.88	91.59	**94.89**	94.86	91.58	94.68	94.84	94.88
	Waveform	79.78	**81.69**	79.18	80.20	80.51	80.75	80.25	81.12	80.99	81.44
	Agrawal	66.98	76.26	66.95	68.30	76.91	78.23	66.67	79.24	78.67	**79.76**
Abrupt	LED	65.45	71.31	69.23	69.85	72.99	73.37	67.67	**73.66**	73.64	73.65
1M	Mixed	92.23	95.91	89.92	91.53	**95.99**	95.93	91.55	95.76	95.92	95.89
	Waveform	79.84	**82.40**	79.19	80.23	81.22	81.20	80.30	81.53	81.45	82.29
	Agrawal	67.02	80.24	67.01	68.52	78.16	80.58	66.61	79.70	80.11	**81.24**
Abrupt	LED	65.60	70.00	69.30	69.99	73.07	73.46	67.70	**73.71**	73.70	73.69
2M	Mixed	92.24	96.85	89.91	91.47	**96.92**	96.91	91.52	96.61	96.79	96.84
	Waveform	79.93	**83.16**	79.20	80.27	82.06	81.63	80.34	81.78	81.66	82.67
	Rank_Ab	8.50	3.83	9.25	7.67	4.08	3.92	8.58	3.42	3.50	**2.25**
	Agrawal	66.91	77.15	66.74	68.48	75.45	77.66	66.70	76.75	**78.45**	78.39
Gradual	LED	65.48	71.62	69.12	69.43	72.88	73.19	68.42	73.34	73.48	**73.53**
500K	Mixed	92.06	94.76	89.76	91.46	**94.78**	94.72	91.44	94.52	94.59	94.71
	Waveform	79.86	**81.58**	79.17	80.19	80.46	80.62	80.23	80.97	80.81	81.53
	Agrawal	66.95	76.14	66.92	68.33	76.34	79.34	66.64	78.84	79.23	**79.78**
Gradual	LED	65.73	70.80	69.21	69.84	72.96	73.30	67.57	73.39	**73.58**	73.55
1 M	Mixed	92.16	**95.93**	89.86	91.45	95.92	95.86	91.47	95.64	95.70	95.82
	Waveform	79.84	**82.48**	79.18	80.23	81.21	81.28	80.29	81.70	81.57	82.26
	Agrawal	67.01	80.72	66.99	68.59	78.08	**81.02**	66.59	78.74	80.38	80.11
Gradual	LED	65.81	70.31	69.29	69.98	73.05	73.43	67.69	73.64	**73.67**	73.63
2 M	Mixed	92.20	**96.91**	89.87	91.43	96.89	96.85	91.49	96.64	96.70	96.81
	Waveform	79.89	**83.17**	79.20	80.27	82.16	81.45	80.33	81.64	81.75	82.58
	Rank_G	8.50	3.08	9.25	7.67	4.33	3.58	8.58	4.25	3.17	**2.58**
	Rank_Ar	8.50	3.46	9.25	7.67	4.21	3.75	8.58	3.83	3.33	**2.42**
	Airlines	65.02	65.30	63.82	65.37	65.37	65.15	**65.84**	65.04	65.62	65.63
	Cars	87.94	89.22	**89.41**	88.87	88.01	88.30	88.64	88.23	88.23	88.44
	Connect_4	74.46	74.12	74.99	75.25	**75.28**	75.27	75.22	74.84	74.66	74.70
	covertypeSorted	71.41	**75.64**	70.05	70.75	70.84	71.08	70.69	70.97	70.98	70.97
Real	LetterRecognition	39.32	**62.04**	56.23	25.79	57.88	54.95	28.75	60.01	38.06	56.30
	pokerHand1M	50.53	**51.85**	51.00	49.75	49.49	50.00	49.78	50.85	48.19	47.88
	rialto	45.39	36.88	31.32	40.72	42.73	37.38	**52.02**	46.82	48.68	42.05
	Usenet2	**69.10**	68.29	68.70	68.86	68.48	68.48	68.56	68.86	68.86	69.05
	WineWhite	43.33	43.33	43.33	**47.62**	46.09	45.27	45.47	43.88	46.45	46.14
	Rank_R	6.11	5.44	6.44	5.28	5.56	5.72	5.11	5.11	5.28	**4.94**
	Rank_T	7.85	4.00	8.48	7.02	4.58	4.29	7.64	4.18	3.86	**3.11**

Tabela 3 – Médias de acurácias dos detectores em (%) utilizando o classificador NB, com 95% de Intervalo de Confiança nas bases de dados artificias.

$\frac{DT}{\#}$	DATASET	ADWIN	DDM	ECDD	STEPD	FHDDM	WSTD	SADD	BFDD	OBDD	ANOVA_C
	Agrawal	**66.39**	64.72	62.96	65.58	65.35	66.23	65.38	66.05	66.38	66.24
Abrupt	LED	67.61	72.63	69.18	70.03	72.93	73.45	68.46	73.58	**73.64**	73.52
500K	Mixed	92.04	91.21	89.94	91.64	92.06	**92.07**	91.68	92.05	92.06	92.06
	Waveform	80.39	79.80	79.19	80.26	80.39	80.38	80.35	80.32	80.37	**80.40**
	Agrawal	**66.49**	64.28	62.98	65.67	65.39	66.30	65.46	66.26	66.46	66.42
Abrupt	LED	68.49	72.95	69.25	70.16	72.99	73.52	68.01	73.63	**73.74**	73.61
1M	Mixed	92.09	90.11	89.97	91.67	**92.10**	**92.10**	91.75	92.09	**92.10**	**92.10**
	Waveform	80.43	79.85	79.20	80.27	80.42	80.40	80.38	80.40	80.41	**80.44**
	Agrawal	**66.55**	64.12	62.98	65.66	65.40	66.31	65.47	66.35	66.43	66.43
Abrupt	LED	70.54	72.78	69.31	70.24	73.07	73.64	68.09	73.82	73.80	**73.83**
2M	Mixed	92.06	89.91	89.95	91.64	**92.07**	**92.07**	91.70	92.05	92.06	92.06
	Waveform	80.46	79.55	79.21	80.31	80.45	80.46	80.42	80.41	80.45	**80.47**
	Rank_Ab	4.29	8.33	9.50	7.33	4.67	3.50	7.50	4.67	2.88	**2.33**
	Agrawal	66.38	63.93	62.94	65.55	65.32	66.20	65.27	64.92	**66.39**	66.22
Gradual	LED	67.26	72.69	69.15	69.98	72.86	73.35	68.51	73.04	**73.54**	73.24
500K	Mixed	91.88	91.76	89.83	91.51	**91.94**	91.93	91.55	91.89	91.91	91.92
	Waveform	**80.37**	79.81	79.18	80.24	80.36	80.33	80.33	79.50	79.95	80.08
	Agrawal	66.47	64.57	62.96	65.64	65.36	66.29	65.42	65.95	**66.48**	66.40
Gradual	LED	68.94	72.21	69.23	70.12	72.96	73.46	67.90	73.14	**73.56**	72.47
1 M	Mixed	92.00	91.89	89.92	91.60	**92.04**	92.03	91.68	92.01	92.02	92.03
	Waveform	**80.40**	79.89	79.19	80.27	**80.40**	80.38	80.36	80.07	80.19	80.30
	Agrawal	**66.54**	64.06	62.98	65.65	65.39	66.30	65.45	65.92	66.51	66.45
Gradual	LED	70.62	71.68	69.31	70.23	73.06	**73.60**	68.08	73.13	73.54	72.77
2 M	Mixed	92.02	88.51	89.92	91.60	**92.03**	**92.03**	91.67	92.01	**92.03**	**92.03**
	Waveform	**80.45**	79.53	79.21	80.31	80.44	**80.45**	80.42	80.25	80.38	80.44
	Rank_G	4.33	7.92	9.50	6.92	3.88	**2.67**	7.13	5.75	3.21	3.71
	Rank_Ar	4.31	8.13	9.50	7.13	4.27	3.08	7.31	5.21	3.04	**3.02**
	Airlines	66.96	65.35	63.66	65.73	65.82	66.68	66.29	66.94	66.98	**67.14**
	Cars	87.94	89.22	**89.41**	88.87	88.01	88.30	88.64	88.23	88.23	88.44
	Connect_4	74.32	74.47	75.05	75.14	**75.23**	75.17	75.08	74.45	74.54	74.63
	covertypeSorted	67.51	67.14	67.39	67.62	67.94	**68.15**	66.68	67.35	67.47	67.82
Real	LetterRecognition	39.70	**62.04**	56.25	30.66	57.89	50.50	28.93	60.03	54.39	56.32
	pokerHand1M	**50.11**	**50.11**	**50.11**	48.82	48.24	49.51	48.88	50.07	46.93	46.74
	rialto	45.51	36.63	21.59	40.55	41.24	30.48	**50.85**	43.56	46.29	42.48
	Usenet2	70.15	69.87	69.27	68.87	69.38	67.75	**70.43**	66.97	67.07	67.88
	WineWhite	42.82	42.82	42.82	**48.13**	46.53	46.39	46.60	43.37	46.55	47.31
	Rank_R	5.78	5.67	6.00	5.33	5.00	5.44	5.00	6.17	5.83	**4.78**
	Rank_T	4.71	7.45	8.55	6.64	4.47	3.73	6.68	5.47	3.80	**3.50**

6.3.2 Avaliação estatística

Como exposto anteriormente, o cálculo dos ranks foi utilizado para comparar os resultados da acurácia. A partir desses ranks, um teste estatístico conhecido como F_F, baseado no teste não paramétrico de *Friedman* (DEMSAR, 2006), foi aplicado. Seu objetivo é, unicamente, informar se existe diferença estatística entre os métodos comparados, mas sem especificar quais (BARROS, 2017). Portanto, com o intuito de identificar quais métodos são estatisticamente diferentes, este trabalho utilizou o pós-teste de *Nemenyi* (DEMSAR, 2006). Uma explicação mais detalhada desses testes estatísticos é apresentada no anexo B.

Para ter uma visão mais apropriada dos ranks apresentados na seção anterior, os ranks das tabelas 2 e 3 foram agrupados nas tabelas 4 e 5, respectivamente. Os resultados, novamente apresentados, são mostrados graficamente nas figuras 6 e 7, onde a Diferença Crítica (**CD**) é representada por barras que conectam aqueles métodos que não são estatisticamente diferentes. Desta maneira, quando vários métodos são comparados, os resultados dos pós-testes são representados através de um diagrama simples, o qual possui uma linha horizontal incluindo os ranks com todos os métodos analisados, de forma que os melhores detectores se encontram localizados à direita do gráfico *Nemenyi* (DEMSAR, 2006; CABRAL, 2017).

Tabela 4 – Ranks dos métodos usando como classificador base HT.

Ranks	ADWIN	DDM	ECDD	STEPD	FHDDM	WSTD	SADD	BFDD	OBDD	ANOVA_C	
Rank_Ab	8.50	3.83	9.25	7.67	4.08	3.92	8.58	3.42	3.50	**2.25**	CD = 3.42752
Rank_G	8.5	3.08	9.25	7.67	4.33	3.58	8.58	4.25	3.17	**2.58**	CD = 3.42752
Rank_Ar	8.5	3.46	9.25	7.67	4.21	3.75	8.58	3.83	3.33	**2.42**	CD = 2.42362
Rank_R	6.11	5.44	6.44	5.28	5.56	5.72	5.11	5.11	5.28	**4.94**	CD = 4.51581
Rank_T	7.85	4.00	8.48	7.02	4.58	4.29	7.64	4.18	3.86	**3.11**	CD = 2.35831

Continuando com a análise, note-se que na figura 6, correspondente aos dados da tabela 4, os resultados ilustrados nos gráficos podem ser resumidos da seguinte maneira:

- No esboço 6(a) pertencente aos cenários de mudanças abruptas, os métodos propostos pela pesquisa, ANOVA_C, BFDD e OBDD, nesta ordem, encontram-se melhores posicionados nos ranks, ao contrário do SADD. A proposta Anova_C apresenta superioridade estatística em relação aos detectores ECDD, ADWIN, STEPD e SADD. Também é possível perceber que os métodos BFDD e OBDD alcançaram superioridade estatística sobre ECDD, ADWIN e SADD.

- Já a ilustração 6(b), referente ao comportamento dos detectores nas bases de dados artificias com presença de mudanças graduais, mostra o ANOVA_C como o método melhor colocado nos ranks. O ANOVA_C e o OBDD são estatisticamente superiores ao ECDD, ADWIN, STEPD e SADD. No caso do método BFDD, apresenta

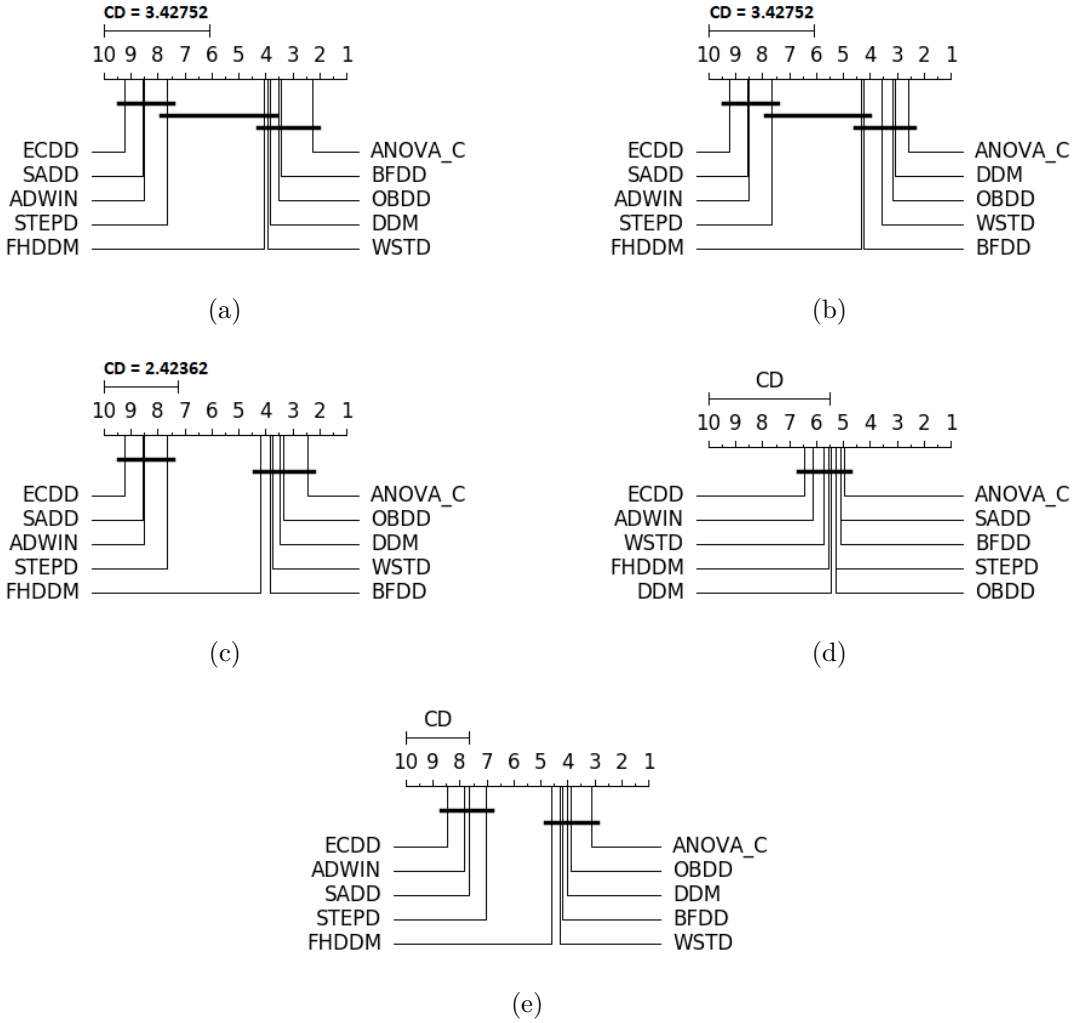


Figura 6 – Comparação estatística da acurácia de ANOVA_C e os outros métodos nos diferentes cenários de mudança (a) abruptas e (b) graduais, assim como por tipo de base de dados (c) artificial e (d) reais, e uma (e) totalização das mesmas, através do Teste F_F e o Pós-Teste *Nemenyi*, com 95% intervalo de confiança, tendo como classificador base HT.

diferença estatística em relação aos mesmos métodos que ANOVA_C unicamente excetuando STEPD.

- O gráfico 6(c) foi realizado com o objetivo de sintetizar os resultados dos detectores para ambas mudanças (abruptas e graduais). O mesmo nos permite perceber que os métodos ANOVA_C e OBDD se encontram posicionados como primeiro e segundo, respectivamente, nos ranks. Enquanto o BFDD alcança uma quinta posição, os três métodos têm superioridade estatística em relação aos detectores ECDD, ADWIN, STEPD e SADD.

- Já no diagrama 6(d), que mostra o comportamento nas bases de dados reais, o método ANOVA_C continua sendo o melhor posicionado. Além disso, nota-se que

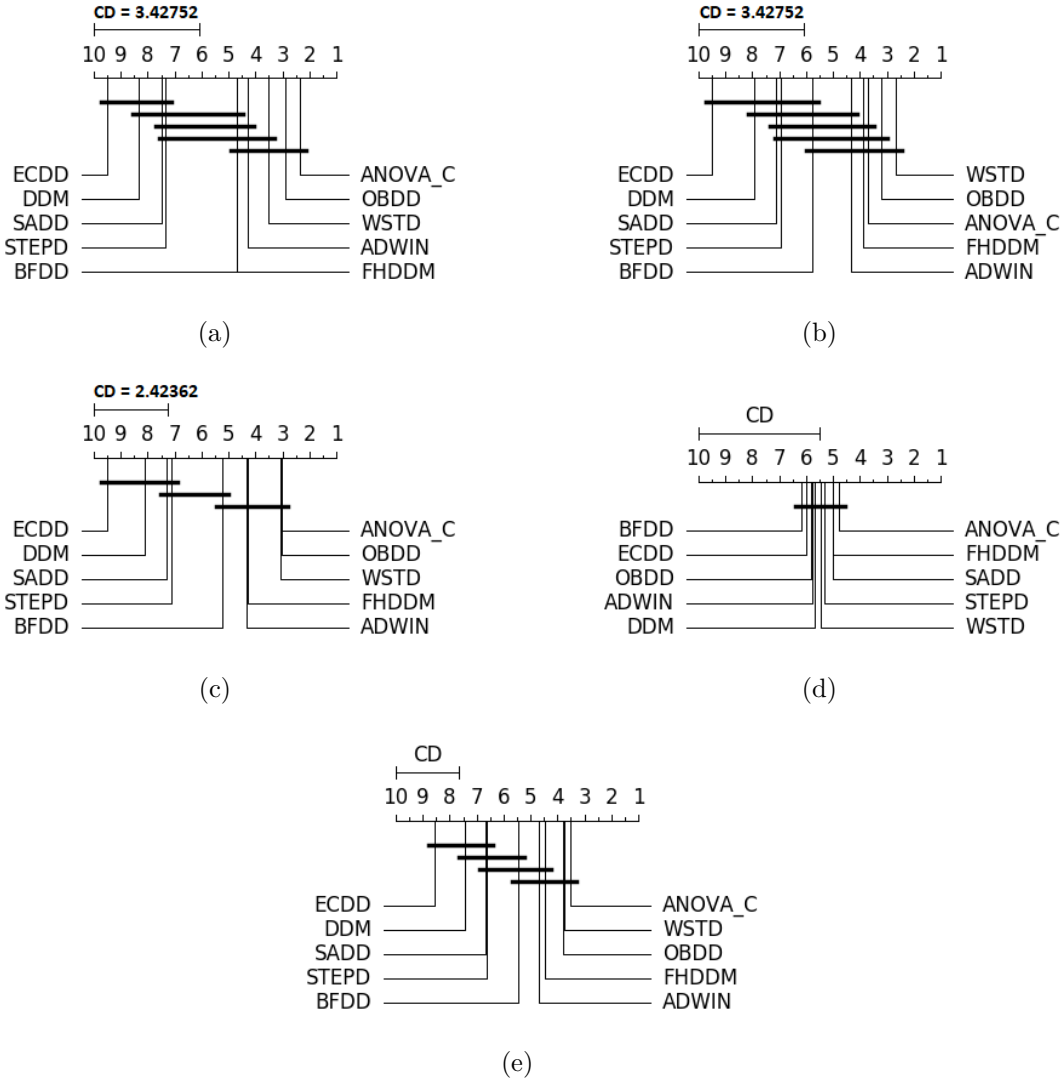


Figura 7 – Comparação estatística das acurácias de ANOVA_C e os outros métodos nos diferentes cenários de mudança (a) abruptas e (b) graduais, assim como por tipo de base de dados (c) artificial e (d) reais, e uma (e) totalização das mesmas, através do Teste F_F e o Pós-Teste *Nemenyi*, com 95% intervalo de confiança, tendo como classificador base NB.

não existe superioridade estatística entre os métodos comparados.

- Finalmente, na ilustração 6(e) que agrupa o comportamento em geral de todos os detectores em ambos tipos de bases de dados (artificiais e reais), pode-se perceber que o método ANOVA_C é o melhor situado, seguido pelo OBDD. Ambos exibem superioridade estatística em relação os ECDD, ADWIN, STEPD e SADD. Embora o BFDD não se encontra entre os três métodos melhores posicionados, ele é superior aos mesmo métodos que o detector ANOVA_C.

Igualmente, são mostrados os gráficos com os resultados (tabela 5) do comportamento dos métodos quando o classificador base é NB (figura 7). De acordo com os resultados

apontados pelo teste de F_F, a hipótese nula de que todos os métodos possuem acurácias equivalente é rejeitada para todos os cenários avaliados. Seguindo com o Pós-Teste de *Nemenyi*, os resultados encontrados foram:

Tabela 5 – Ranks dos métodos usando como classificador base NB.

Ranks	ADWIN	DDM	ECDD	STEPD	FHDDM	WSTD	SADD	BFDD	OBDD	ANOVA_C	
Rank_Ab	4.29	8.33	9.50	7.33	4.67	3.50	7.50	4.67	2.88	**2.33**	CD = 3.42752
Rank_G	4.33	7.92	9.50	6.92	3.88	**2.67**	7.13	5.75	3.21	3.71	CD = 3.42752
Rank_Ar	4.31	8.13	9.50	7.13	4.27	3.08	7.31	5.21	3.04	**3.02**	CD = 2.42362
Rank_R	5.78	5.67	6.00	5.33	5.00	5.44	5.00	6.17	5.83	**4.78**	CD = 4.51581
Rank_T	4.71	7.45	8.55	6.64	4.47	3.73	6.68	5.47	3.80	**3.50**	CD = 2.35831

- Na ilustração 7(a), que mostra o comportamento nas bases de dados artificias com presença de mudanças abruptas, os métodos ANOVA_C e OBDD são os melhores colocados, nessa ordem, e exibem superioridade estatística em comparação ao ECDD, DDM, STEPD e SADD. Neste cenário, o BFDD apresenta superioridade na diferença estatística frente ao ECDD.

- Já nos cenários de mudanças graduais, como ilustrado no esboço 7(b), o método WSTD é o melhor posicionado, alcançando superioridade estatística em relação ao ECDD, DDM, STEPD e SADD, influenciado pelo alto rendimento obtido pelo classificador nas bases de dados de tamanho 2M. Já o OBDD e o ANOVA_C alcançaram a segunda e a terceira posição, respectivamente. Seus desempenhos renderam diferenças estatísticas em relação ao ECDD e ao DDM. Já o BFFD não foi superior estatisticamente ao método pior colocado.

- O desempenho geral das bases artificiais é apresentado no gráfico 7(c). É possível perceber o método ANOVA_C como melhor posicionado, mostrando superioridade estatística em relação ao ECDD, DDM, STEPD e SADD, o anterior está dado por ser um método com resultados de acurácia estável em ambos tipos de mudanças (abruptas e graduais). Constata-se também que o OBBD é o segundo colocado, sendo superior aos mesmos métodos que o ANOVA_C. Já o BFDD é superior unicamente a ECDD e DDM.

- Por outro lado, o gráfico 7(d), que exibe o comportamento nas bases de dados reais, mostra que o método ANOVA_C é o melhor posicionado, mas não manifesta superioridade estatística entre os métodos comparados.

- Numa análise mais profunda da ilustração 7(e), responsável por agrupar o comportamento geral do comitê de teste estatísticos em ambos tipos de bases de dados

(artificiais e reais), se percebe que o método ANOVA_C é o melhor situado, exibindo superioridade estatística em relação aos métodos ECDD, DDM, STEPD e SADD.

Nas análises anteriores, não é enfatizado sobre o método SADD pelos seus resultados inferiores, apenas superior no posicionamento dos ranks que o ECDD em quase a totalidade dos cenários considerados com a utilização do classificador HT. Dos métodos do estado da arte usados podem-se destacar o comportamento do WSTD e do DDM com o uso do HT. É importante destacar que nenhum desses métodos obteve desempenho superior ao da proposta principal desta investigação (ANOCA_C), em nenhum dos ambientes explorados.

Já com a utilização do classificador NB, o SADD continua com resultados inferiores, somente sendo melhor colocado nos ranks que o ECDD e DDM quase que na totalidade dos contextos avaliados. Com o uso do NB os métodos tomados das bibliografias a ressaltar são: WSTD, FHDDM e ADWIN. Embora encontra-se inferiormente situados de forma geral.

Finalizando, nota-se que o comportamento do método SADD melhora nos cenários de bases de dados reais. Já os métodos BFDD e OBDD pioram seus resultados. Por outro lado, o ANOVA_C permanece estável, sendo o melhor posicionado.

6.3.3 Identificação das mudanças de conceitos

Na atualidade um ponto de vista diferente de avaliação do desempenho dos detectores pode ser obtida através da análise das suas detecções de mudanças de conceitos. Nesta pesquisa, as análises unicamente foram realizadas para as bases de dados com presença de mudanças abruptas, porque cada posição exata da mudança é conhecida.

Nas tabelas 6, 7, 8, e 9 são apresentados os valores da distância média entre os reais pontos de mudanças de conceitos e as detecções (μD). Além disso, também são incluídos os valores da quantidade de FN e FP, assim como de *Precision* (Prec), *Recall*, F_1 e MCC (subseção 6.2.2 e anexo B). As duas primeiras tabelas correspondem aos resultados obtidos com o uso do classificador HT. Já as duas últimas correspondem aos resultados do NB. Nas tabelas também são colocadas as médias totais dos desempenhos por tamanho das bases, assim como uma média geral. Nota-se que, como nas demais tabelas apresentadas nesta investigação, os valores em **negrito** representam os melhores resultados.

É valido ressaltar que μD calcula o número médio de instâncias referentes ao atraso dos métodos em detectarem modificações nas distribuições dos dados analisados (PESARANGHADER; VIKTOR, 2016; CABRAL, 2017). Ademais, para categorizar as identificações de mudança de conceito, as detecções foram consideradas TP se ocorressem dentro de 2% do tamanho do conceito após a ponto de mudança correto. Por exemplo: nos conjuntos de dados de 500.000 casos, os conceitos duram 100.000 instâncias e, assim, as mudanças

Tabela 6 – Identificação das mudanças de conceitos abruptas nas bases de dados artificias utilizando o classificador HT (Parte 1)

		Resultados usando HT como classificador base						
DATASET	**DETEC.**	**μD**	**FN**	**FP**	**Prec**	**Recall**	**F1**	**MCC**
Agrawal 500K	**ADWIN**	249.46	3	1440	0.0251	0.92500	0.0488	0.1522
	DDM	593.46	14	**25**	**0.5098**	0.65000	**0.5714**	**0.5756**
	ECDD	201.79	1	7880	0.0049	0.97500	0.0098	0.0692
	STEPD	175.14	3	1722	0.0210	0.92500	0.0411	0.1395
	FHDDM	71.00	**0**	1173	0.0330	**1.00000**	0.0638	0.1816
	WSTD	**56.92**	1	248	0.1359	0.97500	0.2385	0.3640
	SADD	170.81	3	2310	0.0158	0.92500	0.0310	0.1207
	BFDD	117.63	2	83	0.3140	0.95000	0.4720	0.5462
	OBDD	154.36	1	115	0.2532	0.97500	0.4021	0.4969
	ANOVA_C	94.32	3	92	0.2868	0.92500	0.4379	0.5151
LED 500K	**ADWIN**	397.75	**0**	10405	0.0038	**1.00000**	0.0076	0.0618
	DDM	774.62	27	**45**	**0.2241**	0.32500	**0.2653**	0.2699
	ECDD	327.63	2	3821	0.0098	0.95000	0.0195	0.0967
	STEPD	161.58	2	5593	0.0067	0.95000	0.0134	0.0800
	FHDDM	72.89	2	710	0.0508	0.95000	0.0964	0.2197
	WSTD	108.65	3	667	0.0526	0.92500	0.0995	0.2205
	SADD	148.38	3	6397	0.0058	0.92500	0.0114	0.0729
	BFDD	87.78	4	196	0.1552	0.90000	0.2647	**0.3737**
	OBDD	100.29	5	207	0.1446	0.87500	0.2482	0.3557
	ANOVA_C	**65.81**	9	183	0.1449	0.77500	0.2441	0.3351
Mixed 500K	**ADWIN**	40.00	**0**	1183	0.0327	**1.00000**	0.0633	0.1808
	DDM	158.50	**0**	20	0.6667	**1.00000**	0.8000	0.8165
	ECDD	**10.00**	**0**	6418	0.0062	**1.00000**	0.0123	0.0787
	STEPD	13.75	**0**	1542	0.0253	**1.00000**	0.0493	0.1590
	FHDDM	20.00	**0**	**0**	**1.0000**	**1.00000**	**1.0000**	**1.0000**
	WSTD	16.25	**0**	8	0.8333	**1.00000**	0.9091	0.9129
	SADD	28.00	**0**	2018	0.0194	**1.00000**	0.0381	0.1394
	BFDD	27.00	**0**	32	0.5556	**1.00000**	0.7143	0.7454
	OBDD	28.50	**0**	5	0.8889	**1.00000**	0.9412	0.9428
	ANOVA_C	21.50	**0**	3	0.9302	**1.00000**	0.9639	0.9645
Waveform 500K	**ADWIN**	325.24	19	306	0.0642	0.52500	0.1144	0.1836
	DDM	1553.33	37	**31**	0.0882	0.07500	0.0811	0.0813
	ECDD	387.00	**0**	8351	0.0048	**1.00000**	0.0095	0.0690
	STEPD	275.71	12	1581	0.0174	0.70000	0.0340	0.1103
	FHDDM	126.09	17	172	0.1179	0.57500	0.1957	0.2604
	WSTD	142.50	20	145	0.1212	0.50000	0.1951	0.2462
	SADD	275.20	15	1747	0.0141	0.62500	0.0276	0.0939
	BFDD	113.33	22	54	0.2500	0.45000	**0.3214**	**0.3354**
	OBDD	104.67	25	66	0.1852	0.37500	0.2479	0.2635
	ANOVA_C	**80.00**	27	33	**0.2826**	0.32500	0.3023	0.3031
MEAN 500K	**ADWIN**	253.11	5.5	3333.5	0.0315	0.86250	0.0585	0.1446
	DDM	769.98	19.5	**30.25**	0.3722	0.51250	0.4295	0.4358
	ECDD	231.61	**0.75**	6617.5	0.0064	**0.98125**	0.0128	0.0784
	STEPD	156.55	4.25	2609.5	0.0176	0.89375	0.0345	0.1222
	FHDDM	72.50	4.75	513.75	0.3004	0.88125	0.3390	0.4154
	WSTD	81.08	6	267	0.2857	0.85000	0.3606	0.4359
	SADD	155.60	5.25	3118	0.0138	0.86875	0.0270	0.1067
	BFDD	86.44	7	91.25	0.3187	0.82500	0.4431	0.5002
	OBDD	96.96	7.75	98.25	0.3680	0.80625	0.4598	0.5147
	ANOVA_C	**65.41**	9.75	77.75	**0.4111**	0.75625	**0.4870**	**0.5294**
Agrawal 1M	**ADWIN**	522.25	**0**	2807	0.0140	**1.00000**	0.0277	0.1185
	DDM	890.45	18	**23**	**0.4889**	0.55000	**0.5176**	**0.5185**
	ECDD	250.25	**0**	15762	0.0025	**1.00000**	0.0050	0.0503
	STEPD	308.50	**0**	3419	0.0116	**1.00000**	0.0229	0.1075
	FHDDM	113.33	1	2178	0.0176	0.97500	0.0346	0.1309
	WSTD	**43.50**	**0**	398	0.0913	**1.00000**	0.1674	0.3022
	SADD	420.00	4	4578	0.0078	0.90000	0.0155	0.0838
	BFDD	72.89	2	110	0.2086	0.93548	0.3412	0.4418
	OBDD	185.95	3	136	0.2139	0.92500	0.3474	0.4448
	ANOVA_C	90.77	1	113	0.2566	0.97500	0.4063	0.5002
LED 1M	**ADWIN**	301.58	2	38	0.0019	0.95000	0.0038	0.0423
	DDM	1678.00	25	**15**	**0.1648**	0.37500	**0.2290**	0.2486
	ECDD	575.25	**0**	40	0.0052	**1.00000**	0.0104	0.0724
	STEPD	223.24	3	37	0.0035	0.92500	0.0071	0.0572
	FHDDM	**72.97**	3	37	0.0245	0.92500	0.0477	0.1504
	WSTD	144.47	2	38	0.0311	0.95000	0.0603	0.1720
	SADD	173.42	2	38	0.0026	0.95000	0.0051	0.0494
	BFDD	88.06	9	31	0.0690	0.77500	0.1268	0.2313
	OBDD	102.81	8	32	0.0674	0.80000	0.1243	0.2321
	ANOVA_C	78.29	5	35	0.0816	0.87500	0.1493	**0.2672**
Mixed 1M	**ADWIN**	41.50	**0**	2329	0.0169	**1.00000**	0.0332	0.1299
	DDM	194.75	**0**	36	0.5263	**1.00000**	0.6897	0.7255
	ECDD	**10.00**	**0**	12770	0.0031	**1.00000**	0.0062	0.0558
	STEPD	12.00	**0**	3301	0.0120	**1.00000**	0.0237	0.1094
	FHDDM	20.00	**0**	**0**	**1.0000**	**1.00000**	**1.0000**	**1.0000**
	WSTD	16.25	**0**	11	0.7843	**1.00000**	0.8791	0.8856
	SADD	20.00	**0**	4088	0.0097	**1.00000**	0.0192	0.0984
	BFDD	25.50	**0**	27	0.5970	**1.00000**	0.7477	0.7727
	OBDD	34.25	**0**	9	0.8163	**1.00000**	0.8989	0.9035
	ANOVA_C	23.25	**0**	8	0.8333	**1.00000**	0.9091	0.9129

Tabela 7 – Identificação das mudanças de conceitos abruptas nas bases de dados artificias utilizando o classificador HT (Parte 2)

		Resultados usando HT como classificador base						
DATASET	**DETEC.**	**μD**	**FN**	**FP**	**Prec**	**Recall**	**F1**	**MCC**
Waveform 1M	**ADWIN**	484.44	13	574	0.0449	0.67500	0.0842	0.1741
	DDM	2678.00	30	**25**	0.2857	0.25000	0.2667	0.2673
	ECDD	279.00	**0**	16706	0.0024	**1.00000**	0.0048	0.0488
	STEPD	976.06	7	3047	0.0107	0.82500	0.0212	0.0940
	FHDDM	264.09	18	316	0.0651	0.55000	0.1164	0.1892
	WSTD	257.69	14	244	0.0963	0.65000	0.1677	0.2502
	SADD	866.18	6	3531	0.0095	0.85000	0.0189	0.0900
	BFDD	120.00	25	112	0.1642	0.46809	0.2431	0.2772
	OBDD	127.22	22	141	0.1132	0.45000	0.1809	0.2257
	ANOVA_C	**75.00**	22	37	**0.3273**	0.45000	**0.3789**	**0.3838**
MEAN 1M	**ADWIN**	337.44	3.75	6448.75	0.0194	0.90625	0.0372	0.1162
	DDM	1360.30	18.25	**40**	0.3664	0.54375	0.4257	0.4400
	ECDD	278.63	**0**	13207.8	0.0033	**1.00000**	0.0066	0.0568
	STEPD	379.95	2.5	5045.5	0.0094	0.93750	0.0187	0.0920
	FHDDM	117.60	5.5	992.25	0.2768	0.86250	0.2997	0.3676
	WSTD	115.48	4	458.75	0.2508	0.90000	0.3186	0.4025
	SADD	369.90	3	6726	0.0074	0.92500	0.0147	0.0804
	BFDD	76.61	9	166.75	0.2597	0.79464	0.3647	0.4307
	OBDD	112.56	8.25	182.25	0.3027	0.79375	0.3879	0.4515
	ANOVA_C	**66.83**	7	138	**0.3747**	0.82500	**0.4609**	**0.5160**
Agrawal 2M	**ADWIN**	488.97	1	5609	0.0069	0.97500	0.0137	0.0820
	DDM	1399.00	10	**22**	**0.5769**	0.75000	**0.6522**	**0.6578**
	ECDD	181.00	**0**	31721	0.0013	**1.00000**	0.0025	0.0355
	STEPD	226.67	1	6761	0.0057	0.97500	0.0114	0.0748
	FHDDM	173.85	1	4083	0.0095	0.97500	0.0187	0.0960
	WSTD	**32.50**	**0**	644	0.0585	**1.00000**	0.1105	0.2418
	SADD	978.21	1	9235	0.0042	0.97500	0.0084	0.0640
	BFDD	213.33	1	209	0.1144	0.96429	0.2045	0.3321
	OBDD	354.05	3	263	0.1233	0.92500	0.2176	0.3378
	ANOVA_C	518.50	0	181	0.1810	**1.00000**	0.3065	0.4254
LED 2M	**ADWIN**	441.54	1	39755	0.0010	0.97500	0.0020	0.0309
	DDM	4038.57	33	**69**	**0.0921**	0.17500	**0.1207**	0.1270
	ECDD	1032.25	**0**	15088	0.0026	**1.00000**	0.0053	0.0514
	STEPD	**243.50**	**0**	20386	0.0020	**1.00000**	0.0039	0.0442
	FHDDM	278.65	3	2886	0.0127	0.92500	0.0250	0.1082
	WSTD	245.26	2	2134	0.0175	0.95000	0.0344	0.1289
	SADD	372.00	0	29610	0.0013	**1.00000**	0.0027	0.0367
	BFDD	243.75	8	855	0.0361	0.80000	0.0690	0.1699
	OBDD	377.88	7	904	0.0352	0.82500	0.0676	**0.1705**
	ANOVA_C	307.50	8	873	0.0354	0.80000	0.0677	0.1682
Mixed 2M	**ADWIN**	39.00	**0**	4578	0.0087	**1.00000**	0.0172	0.0931
	DDM	271.50	**0**	34	0.5405	**1.00000**	0.7018	0.7352
	ECDD	**9.00**	**0**	25471	0.0016	**1.00000**	0.0031	0.0396
	STEPD	12.50	**0**	6705	0.0059	**1.00000**	0.0118	0.0770
	FHDDM	20.00	0	0	**1.0000**	**1.00000**	**1.0000**	**1.0000**
	WSTD	11.00	**0**	7	0.8511	**1.00000**	0.9195	0.9225
	SADD	31.25	0	8299	0.0048	**1.00000**	0.0095	0.0692
	BFDD	20.75	**0**	33	0.5479	**1.00000**	0.7080	0.7402
	OBDD	74.50	**0**	21	0.6557	**1.00000**	0.7921	0.8098
	ANOVA_C	20.00	0	8	0.8333	**1.00000**	0.9091	0.9129
Waveform 2M	**ADWIN**	835.93	13	1208	0.0219	0.67500	0.0424	0.1215
	DDM	4012.22	22	**50**	**0.2647**	0.45000	**0.3333**	**0.3451**
	ECDD	282.00	**0**	33699	0.0012	**1.00000**	0.0024	0.0344
	STEPD	1677.63	2	6209	0.0061	0.95000	0.0121	0.0760
	FHDDM	**274.64**	12	383	0.0681	0.70000	0.1242	0.2184
	WSTD	376.45	9	349	0.0816	0.77500	0.1476	0.2514
	SADD	1189.49	1	6776	0.0057	0.97500	0.0114	0.0747
	BFDD	582.38	19	228	0.1257	0.00000	0.0000	0.2569
	OBDD	426.52	17	266	0.0796	0.57500	0.1398	0.2139
	ANOVA_C	400.00	22	70	0.2045	0.45000	0.2813	0.3034
MEAN 2M	**ADWIN**	451.36	3.75	12787.5	0.0096	0.90625	0.0188	0.0819
	DDM	2430.32	16.25	**43.75**	**0.3686**	0.59375	**0.4520**	**0.4663**
	ECDD	376.06	**0**	26494.8	0.0017	**1.00000**	0.0033	0.0402
	STEPD	540.08	0.75	10015.3	0.0049	0.98125	0.0098	0.0680
	FHDDM	186.79	4	1838	0.2726	0.90000	0.2920	0.3557
	WSTD	**166.30**	2.75	783.5	0.2522	0.93125	0.3030	0.3862
	SADD	642.74	0.5	13480	0.0040	0.98750	0.0080	0.0612
	BFDD	265.05	7	331.25	0.2060	0.69107	0.2454	0.3748
	OBDD	308.24	6.75	363.5	0.2235	0.83125	0.3043	0.3830
	ANOVA_C	311.50	7.5	283	0.3136	0.81250	0.3911	0.4525
MEAN	**ADWIN**	347.31	4.33333	7523.25	0.0202	0.89167	0.0382	0.1142
	DDM	1520.20	18	**38**	**0.3691**	0.55000	0.4357	0.4474
	ECDD	295.43	**0.25**	15440	0.0038	**0.99375**	0.0076	0.0585
	STEPD	358.86	2.5	5890.08	0.0107	0.93750	0.0210	0.0941
	FHDDM	125.63	4.75	1114.67	0.2833	0.88125	0.3102	0.3796
	WSTD	**120.95**	4.25	503.083	0.2629	0.89375	0.3274	0.4082
	SADD	389.41	2.91667	7774.67	0.0084	0.92708	0.0166	0.0828
	BFDD	142.70	7.66667	196.417	0.2615	0.77024	0.3511	0.4352
	OBDD	172.58	7.58333	214.667	0.2980	0.81042	0.3840	0.4497
	ANOVA_C	147.91	8.08333	166.25	0.3665	0.79792	**0.4464**	**0.4993**

detectadas até 2.000 instâncias após os pontos corretos foram calculados como TP (BARROS, 2017). Para cada método são consideradas 10 repetições, sendo omitido na tabela o cálculo de TP e TN (sub-seção 6.2.2) pela fácil obtenção dos mesmos como mostram as equações 6.1, 6.2 .

$$TP = 40 - FN \tag{6.1}$$

$$TN = tamanho \times 10 - 40 - FP \tag{6.2}$$

Como se pode perceber nas tabelas 6 e 7, usando o classificador HT, o detector WSTD, na média geral, tem melhor resultado na métrica μD, favorecido pelos valores obtidos nas bases de tamanho 2M. Assim como, nas criadas pelo gerador Agrawal nos demais tamanhos. Realizando análise mais detalhada se pode perceber que os métodos propostos nesta investigação, com exceção do SADD, apresentam os melhores resultados nos tamanhos 500K e 1M.

Já na avaliação do comportamento relacionado a *Precision*, a melhor média geral é apresentada pelo DDM, tendo em conta que foi o método que em quase a totalidade das bases de dados apresentou o menor número de FP (principalmente no tamanho 2M). Adicionalmente, é válido ressaltar que o DDM também é o pior método considerando o número de FN (erra muito na hora de detectar a mudança existente). Aprofundando no estudo se identifica que o ANOVA_C tem os melhores valores da média geral nesta métrica nos tamanhos 500K e 1M. Este resultado é apoiado nos valores baixos e balanceados de FN e FP, principalmente na sequências criadas pelo gerador Wavefrom. Já os métodos BFDD e OBDD apresentam resultados aceitáveis de *Precision*, não sendo assim o caso do SADD.

Com relação à métrica *Recall*, temos o método ECDD como o melhor posicionado, influenciado por ter baixo número de FN no geral. Não obstante, temos que acentuar que o ECDD apresenta o maior número de FP (detecta muitas mudanças que não existem). Nota-se que dos métodos aportados pela pesquisa, o de melhor resultado nos três tamanhos avaliados foi SADD. O desempenho dos outros métodos (ANOVA_C, BFDD e OBDD), por outro lado, pode ser considerado aceitável.

As duas últimas métricas que foram tomadas em conta para avaliar os resultados são F_1 e MCC, usadas na pesquisa para ter um critério final mais justo, dado que as métricas anteriores podem ser influenciadas pelo alto número de FN e FP, assim como pelo seu desbalanceamento. Nestas métricas, o comitê de métodos estatísticos ANOVA_C apresenta os melhores resultados com o uso do classificador HT, como está sendo analisado. Desta forma, é catalogado como um método estável para todas as bases de dados usadas na pesquisa. Contudo, é válido mencionar que o segundo com os melhores resultados foi o DDM, apoiado nos ótimos resultados obtidos nas bases de tamanho 2M em geral.

Os resultados dos detectores com respeito às métricas com o uso do classificador NB, de forma geral mantém o mesmo comportamento que com o classificador HT, unicamente mudando que neste caso o detector que apresenta melhores resultados na μD é o BFDD, e com relação à métrica F_1 o melhor posicionado foi o DDM. Ressaltando que o ANOVA_C continua com o melhor resultado de MCC.

Com o objetivo de descriminar ainda mais o comportamento dos detectores e localizar o que levou às mudanças nas métricas μD e F_1 com o uso de um detector ou outro, foi realizada uma análise desagregada por tamanhos, dando como resultado que: O desempenho dos detectores propostos nesta investigação, nas bases de dados com tamanho 1M, são inferiores ao método DDM em relação as métricas Precision, F_1 e MCC. Também muda a métrica μD, neste caso o melhor valor é ostentado pelo detector BFDD. Estas variações são as que principalmente afetaram o resultado final. Outra variação nos resultados foi para o tamanho 500K de forma geral o comportamento mudou com respeito à métrica μD, neste caso o WSTD foi o método que tem o melhor resultado.

Note-se também que o comitê de métodos estatístico ANOVA_C é superior de forma geral aos métodos de detecção de mudanças de conceitos baseados nos teste estatístico que o conformam, tendo melhor resultados de F_1 e MCC usando ambos classificadores bases (HT e NB).

6.3.4 Memória e tempo de execução

Nesta pesquisa foi computada também a memória empregada e o tempo de execução dos métodos usados na comparação com ambos classificadores (HT e NB). O comitê de métodos estatísticos ANOVA_C tende a consumir um pouco mais de memória e tempo de execução do que os outros métodos, especialmente com o classificador base HT. No entanto, os números absolutos são ainda insignificantes para computadores modernos e, por esse motivo, esses resultados são omitidos.

Tabela 8 – Identificação das mudanças de conceitos abruptas nas bases de dados artificias utilizando o classificador NB (Parte 1)

	Resultados usando NB como classificador base							
DATASET	**DETEC.**	**μD**	**FN**	**FP**	**Prec**	**Recall**	**F1**	**MCC**
Agrawal 500K	**ADWIN**	216.00	**0**	92	0.3030	**1.00000**	0.4651	0.5505
	DDM	1311.90	19	**19**	**0.5250**	0.52500	0.5250	0.5250
	ECDD	290.26	1	7782	0.0050	0.97500	0.0099	0.0697
	STEPD	138.38	3	1453	0.0248	0.92500	0.0484	0.1515
	FHDDM	164.74	2	1628	0.0228	0.95000	0.0445	0.1472
	WSTD	148.46	1	321	0.1083	0.97500	0.1950	0.3250
	SADD	272.70	3	1927	0.0188	0.92500	0.0369	0.1320
	BFDD	**111.79**	12	69	0.2887	0.70000	0.4088	0.4495
	OBDD	215.64	1	68	0.3645	0.97500	**0.5306**	**0.5961**
	ANOVA__C	185.14	3	185	0.1667	0.92500	0.2824	0.3926
LED 500K	**ADWIN**	251.54	**1**	7492	0.0052	**0.97500**	0.0103	0.0710
	DDM	687.92	16	**24**	**0.5000**	0.60000	0.5455	0.5477
	ECDD	276.92	**1**	3844	0.0100	**0.97500**	0.0199	0.0989
	STEPD	209.21	2	4888	0.0077	0.95000	0.0153	0.0856
	FHDDM	70.79	2	725	0.0498	0.95000	0.0946	0.2175
	WSTD	95.14	3	401	0.0845	0.92500	0.1548	0.2795
	SADD	133.78	3	6412	0.0057	0.92500	0.0114	0.0728
	BFDD	87.35	6	74	0.3148	0.85000	0.4595	0.5173
	OBDD	103.43	5	78	0.3097	0.87500	0.4575	0.5206
	ANOVA__C	**69.39**	7	40	0.4521	0.82500	**0.5841**	**0.6107**
Mixed 500K	**ADWIN**	40.00	**0**	57	0.4124	**1.00000**	0.5839	0.6422
	DDM	300.26	1	29	0.5735	0.97500	0.7222	0.7478
	ECDD	**10.00**	**0**	6425	0.0062	**1.00000**	0.0123	0.0786
	STEPD	13.25	**0**	1118	0.0345	**1.00000**	0.0668	0.1858
	FHDDM	20.00	0	0	**1.0000**	**1.00000**	**1.0000**	**1.0000**
	WSTD	18.75	**0**	**0**	**1.0000**	**1.00000**	**1.0000**	**1.0000**
	SADD	20.25	0	1511	0.0258	**1.00000**	0.0503	0.1606
	BFDD	36.00	**0**	25	0.6154	**1.00000**	0.7619	0.7845
	OBDD	36.00	**0**	5	0.8889	**1.00000**	0.9412	0.9428
	ANOVA__C	30.00	0	0	**1.0000**	**1.00000**	**1.0000**	**1.0000**
Waveform 500K	**ADWIN**	188.33	10	77	0.2804	0.75000	0.4082	0.4586
	DDM	1161.25	32	17	0.3200	0.20000	0.2462	0.2530
	ECDD	387.00	**0**	8406	0.0047	**1.00000**	0.0094	0.0688
	STEPD	331.07	12	1470	0.0187	0.70000	0.0364	0.1144
	FHDDM	203.70	13	85	0.2411	0.67500	0.3553	0.4034
	WSTD	**144.62**	14	89	0.2261	0.65000	0.3355	0.3833
	SADD	318.85	14	1679	0.0152	0.65000	0.0298	0.0995
	BFDD	177.62	19	39	0.3500	0.52500	0.4200	0.4287
	OBDD	167.39	17	52	0.3067	0.57500	0.4000	0.4199
	ANOVA__C	208.95	21	11	**0.6333**	0.47500	**0.5429**	**0.5485**
MEAN 500K	**ADWIN**	173.97	2.75	1929.5	0.2502	0.93125	0.3669	0.4305
	DDM	865.33	17	**22.25**	0.4796	0.57500	0.5097	0.5184
	ECDD	241.05	**0.5**	6614.25	0.0065	**0.98750**	0.0129	0.0790
	STEPD	172.98	4.25	2232.25	0.0214	0.89375	0.0417	0.1343
	FHDDM	114.81	4.25	609.5	0.3284	0.89375	0.3736	0.4420
	WSTD	**101.74**	4.5	202.75	0.3547	0.88750	0.4213	0.4970
	SADD	186.40	5	2882.25	0.0164	0.87500	0.0321	0.1162
	BFDD	103.19	9.25	51.75	0.3922	0.76875	0.5125	0.5450
	OBDD	130.62	5.75	50.75	0.4674	0.85625	0.5823	0.6199
	ANOVA__C	123.37	7.75	59	**0.5630**	0.80625	**0.6023**	**0.6380**
Agrawal 1M	**ADWIN**	218.50	**0**	94	0.2985	**1.00000**	0.4598	**0.5464**
	DDM	1979.52	19	**17**	**0.5526**	0.52500	**0.5385**	0.5386
	ECDD	345.50	**0**	15587	0.0026	**1.00000**	0.0051	0.0506
	STEPD	135.68	3	2772	0.0132	0.92500	0.0260	0.1104
	FHDDM	199.19	3	3303	0.0111	0.92500	0.0219	0.1012
	WSTD	248.25	**0**	628	0.0599	**1.00000**	0.1130	0.2447
	SADD	437.84	3	3690	0.0099	0.92500	0.0196	0.0958
	BFDD	**116.90**	11	127	0.1859	0.72500	0.2959	0.3671
	OBDD	335.13	1	142	0.2155	0.97500	0.3529	0.4583
	ANOVA__C	187.11	2	319	0.1064	0.95000	0.1914	0.3180
LED 1M	**ADWIN**	393.33	1	13016	0.0030	0.97500	0.0060	0.0539
	DDM	1316.40	15	**20**	**0.5556**	0.62500	**0.5882**	**0.5893**
	ECDD	575.25	**0**	7653	0.0052	**1.00000**	0.0103	0.0721
	STEPD	188.38	3	9567	0.0039	0.92500	0.0077	0.0597
	FHDDM	72.97	3	1480	0.0244	0.92500	0.0475	0.1502
	WSTD	**58.38**	3	827	0.0428	0.92500	0.0819	0.1990
	SADD	163.24	3	14093	0.0026	0.92500	0.0052	0.0492
	BFDD	89.68	9	130	0.1925	0.77500	0.3085	0.3863
	OBDD	129.69	8	185	0.1475	0.80000	0.2490	0.3435
	ANOVA__C	72.73	7	70	0.3204	0.82500	0.4615	0.5141
Mixed 1M	**ADWIN**	40.00	**0**	60	0.4000	**1.00000**	0.5714	0.6325
	DDM	500.00	3	18	0.6727	0.92500	0.7789	0.7888
	ECDD	**10.00**	**0**	12850	0.0031	**1.00000**	0.0062	0.0557
	STEPD	**10.00**	**0**	2267	0.0173	**1.00000**	0.0341	0.1317
	FHDDM	20.00	0	0	**1.0000**	**1.00000**	**1.0000**	**1.0000**
	WSTD	15.50	**0**	**0**	**1.0000**	**1.00000**	**1.0000**	**1.0000**
	SADD	20.75	0	2762	0.0143	**1.00000**	0.0281	0.1195
	BFDD	31.75	**0**	30	0.5714	**1.00000**	0.7273	0.7559
	OBDD	31.75	**0**	5	0.8889	**1.00000**	0.9412	0.9428
	ANOVA__C	29.00	0	5	0.8889	**1.00000**	0.9412	0.9428

Tabela 9 – Identificação das mudanças de conceitos abruptas nas bases de dados artificias utilizando o classificador NB (Parte 2)

	Resultados usando NB como classificador base							
DATASET	**DETEC.**	**μD**	**FN**	**FP**	**Prec**	**Recall**	**F1**	**MCC**
Waveform 1M	**ADWIN**	391.47	6	85	0.2857	0.85000	0.4277	**0.4928**
	DDM	1810.91	29	**23**	0.3235	0.27500	0.2973	0.2983
	ECDD	279.00	**0**	16776	0.0024	**1.00000**	0.0047	0.0487
	STEPD	931.21	7	2871	0.0114	0.82500	0.0224	0.0968
	FHDDM	291.48	13	193	0.1227	0.67500	0.2077	0.2878
	WSTD	225.71	19	169	0.1105	0.52500	0.1826	0.2409
	SADD	841.76	6	3358	0.0100	0.85000	0.0198	0.0923
	BFDD	208.18	18	74	0.2292	0.55000	0.3235	0.3550
	OBDD	**206.96**	17	95	0.1949	0.57500	0.2911	0.3348
	ANOVA_C	310.50	20	32	**0.3846**	0.50000	**0.4348**	0.4385
MEAN 1M	**ADWIN**	260.83	1.75	3313.75	0.2468	0.95625	0.3662	0.4314
	DDM	1401.71	16.5	**19.5**	**0.5261**	0.58750	**0.5507**	**0.5538**
	ECDD	302.44	**0**	13216.5	0.0033	**1.00000**	0.0066	0.0568
	STEPD	316.32	3.25	4369.25	0.0114	0.91875	0.0225	0.0996
	FHDDM	145.91	4.75	1244	0.2895	0.88125	0.3193	0.3848
	WSTD	136.96	5.5	406	0.3033	0.86250	0.3444	0.4211
	SADD	365.90	3	5975.75	0.0092	0.92500	0.0182	0.0892
	BFDD	**111.63**	9.5	90.25	0.2948	0.76250	0.4138	0.4661
	OBDD	175.88	6.5	106.75	0.3617	0.83750	0.4586	0.5198
	ANOVA_C	149.84	7.25	106.5	0.4251	0.81875	0.5072	0.5534
Agrawal 2M	**ADWIN**	209.75	**0**	78	0.3390	**1.00000**	0.5063	**0.5822**
	DDM	3246.84	21	**16**	**0.5429**	0.47500	**0.5067**	0.5078
	ECDD	328.00	**0**	31384	0.0013	**1.00000**	0.0025	0.0356
	STEPD	514.00	**0**	5595	0.0071	**1.00000**	0.0141	0.0842
	FHDDM	538.25	**0**	6611	0.0060	**1.00000**	0.0120	0.0775
	WSTD	**116.22**	3	1324	0.0272	0.92500	0.0528	0.1586
	SADD	732.05	1	7330	0.0053	0.97500	0.0105	0.0718
	BFDD	381.48	13	252	0.0968	0.67500	0.1693	0.2556
	OBDD	372.22	4	297	0.1081	0.90000	0.1930	0.3119
	ANOVA_C	765.50	**0**	623	0.0603	**1.00000**	0.1138	0.2456
LED 2M	**ADWIN**	225.75	**0**	16580	0.0024	**1.00000**	0.0048	0.0490
	DDM	2368.33	10	**23**	**0.5660**	0.75000	**0.6452**	**0.6516**
	ECDD	917.25	**0**	15214	0.0026	**1.00000**	0.0052	0.0512
	STEPD	286.25	**0**	19043	0.0021	**1.00000**	0.0042	0.0458
	FHDDM	278.65	3	2891	0.0126	0.92500	0.0249	0.1081
	WSTD	266.58	2	1355	0.0273	0.95000	0.0530	0.1610
	SADD	310.50	**0**	28052	0.0014	**1.00000**	0.0028	0.0377
	BFDD	111.56	8	274	0.1046	0.80000	0.1850	0.2892
	OBDD	161.61	9	300	0.0937	0.77500	0.1671	0.2694
	ANOVA_C	**97.35**	6	107	0.2411	0.85000	0.3757	0.4527
Mixed 2M	**ADWIN**	40.00	**0**	58	0.4082	**1.00000**	0.5797	0.6389
	DDM	748.46	1	29	0.5735	0.97500	0.7222	0.7478
	ECDD	**9.00**	**0**	25762	0.0016	**1.00000**	0.0031	0.0393
	STEPD	12.00	**0**	4596	0.0086	**1.00000**	0.0171	0.0929
	FHDDM	19.50	**0**	**0**	**1.0000**	**1.00000**	**1.0000**	**1.0000**
	WSTD	17.00	**0**	**0**	**1.0000**	**1.00000**	**1.0000**	**1.0000**
	SADD	26.75	**0**	5893	0.0067	**1.00000**	0.0134	0.0821
	BFDD	30.00	**0**	101	0.2837	**1.00000**	0.4420	0.5326
	OBDD	31.00	**0**	8	0.8333	**1.00000**	0.9091	0.9129
	ANOVA_C	29.00	**0**	45	0.4706	**1.00000**	0.6400	0.6860
Waveform 2M	**ADWIN**	552.97	3	109	0.2534	0.92500	**0.3978**	**0.4842**
	DDM	2553.33	31	**19**	**0.3214**	0.22500	0.2647	0.2689
	ECDD	**271.25**	**0**	33800	0.0012	**1.00000**	0.0024	0.0344
	STEPD	1664.59	3	5768	0.0064	0.92500	0.0127	0.0768
	FHDDM	544.52	9	312	0.0904	0.77500	0.1619	0.2647
	WSTD	360.67	10	244	0.1095	0.75000	0.1911	0.2866
	SADD	892.70	3	6482	0.0057	0.92500	0.0113	0.0724
	BFDD	288.57	19	146	0.1257	0.00000	0.0000	0.2569
	OBDD	280.43	17	169	0.1198	0.57500	0.1983	0.2624
	ANOVA_C	555.24	19	64	0.2471	0.52500	0.3360	0.3601
MEAN 2M	**ADWIN**	257.12	0.75	4206.25	0.2507	0.98125	0.3722	0.4386
	DDM	2229.24	15.75	**21.75**	**0.5010**	0.60625	**0.5347**	**0.5440**
	ECDD	381.38	**0**	26540	0.0017	**1.00000**	0.0033	0.0401
	STEPD	619.21	0.75	8750.5	0.0060	0.98125	0.0120	0.0749
	FHDDM	345.23	3	2453.5	0.2773	0.92500	0.2997	0.3626
	WSTD	**190.12**	3.75	730.75	0.2910	0.90625	0.3242	0.4015
	SADD	490.50	1	11939.3	0.0048	0.97500	0.0095	0.0660
	BFDD	202.90	10	193.25	0.1527	0.61875	0.1991	0.3336
	OBDD	211.32	7.5	193.5	0.2887	0.81250	0.3669	0.4392
	ANOVA_C	361.77	6.25	209.75	0.2548	0.84375	0.3664	0.4361
MEAN	**ADWIN**	230.64	1.75	3149.83	0.2493	0.95625	0.3684	0.4335
	DDM	1498.76	16.42	**21.17**	**0.5022**	0.58958	**0.5317**	0.5387
	ECDD	308.29	**0.17**	15456.92	0.0038	**0.99583**	0.0076	0.0586
	STEPD	369.50	2.75	5117.33	0.0130	0.93125	0.0254	0.1030
	FHDDM	201.98	4.00	1435.67	0.2984	0.90000	0.3309	0.3965
	WSTD	142.94	4.58	446.50	0.3163	0.88542	0.3633	0.4399
	SADD	347.60	3.00	6932.42	0.0101	0.92500	0.0199	0.0905
	BFDD	**139.24**	9.58	111.75	0.2799	0.71667	0.3751	0.4482
	OBDD	172.60	6.58	117.00	0.3726	0.83542	0.4693	0.5263
	ANOVA_C	211.66	7.08	125.08	0.4143	0.82292	0.4920	**0.5425**

6.4 Considerações Finais

No decorrer do capítulo foram efetuadas comparações empíricas e estatísticas dos métodos de detecção de mudanças de conceitos tendo em conta as métricas de avaliação: Acurácia, μD, quantidade de FN e FP, assim como, *Precision*, *Recall*, F_1, MCC. Depois de uma exaustiva análise, é demostrado que os detectores BFDD e OBDD são competitivos, dado por ter aceitáveis resultados em relação às métricas em comparação com os métodos do estado da arte escolhidos. Entretanto, são superados de forma geral pelo comitê de métodos estatísticos Anova_C, que alcança melhores valores na Acurácia, F_1 e MCC, principalmente se é usado com o classificador base HT. Outro método resultante desta pesquisa (SADD), alcançou resultados inferiores em quase todas as métricas. No entanto, é notório que ele apresenta ótimos valores de FN.

7 CONCLUSÕES

Esta investigação propõe uma nova metodologia para a construção de detectores de mudanças de conceitos em fluxos de dados, baseada nos detectores de mudanças de conceitos STEPD e WSTD o qual propõe no capítulo 4 três métodos: O SADD, BFDD e OBDD que implementam os testes estatísticos ANOVA padrão, Brown-Forsythe e O'brien, respectivamente. Estes foram construídos com o objetivo de demonstrar que a implementação da nova metodologia que se apresenta no capítulo 5, e chamada como ANOVA_C, que combina os métodos estatísticos já mencionados, obtém melhores resultados em relação aos métodos do estado da arte ADWIN, DDM, ECDD, FHDDM, STEPD, WSTD apresentados no capítulo 2.

No capítulo 6 foram realizados os experimentos comparativos, a fim de avaliar o desempenho dos detectores de mudanças de conceito propostos com respeito a oito métricas, com o uso total de nove bases reais e vinte e quatro artificiais construídas para três tamanhos diferentes (500K, 1M, 2M instâncias).

Avaliando o desempenho em relação à métrica Acurácia se percebe que o método ANOVA_C tem os melhores resultados. Portanto, o método na análise geral é o melhor posicionado nos ranks resultante da comparação estatística realizada pelo teste de *Friedman*, sendo superior estatisticamente aos métodos STEPD, ECDD e SADD. O comportamento anterior somente é diferenciado com que o ANOVA_C também é estatisticamente superior ao método ADWIN com a utilização do classificador HT, assim como em relação ao DDM se é usado o classificador NB. Os ranks foram apresentados em gráficos de Diferenças Críticas (**CD**) utilizando o Pós-Teste de *Nemenyi*.

Para a avaliação do desempenho dos métodos testados em relação as detecções de mudanças de conceitos, foram avaliadas as métricas a distância média entre os reais pontos de mudanças de conceitos e as detecções (μD), Falsos Positivos (FP), Falsos Negativos (FN), *Precision*, *Recall*, F_1 e Matthews Correlation Coefficient (MCC). A MCC mostrou-se a mais balanceada sendo utilizada como referência para identificar os métodos melhores avaliados quanto à capacidade de detecção de mudanças de conceitos. Nesse contexto, o Anova_C obteve as melhores médias de MCC. Além disso, tem o melhor resultado da métrica F_1 com o uso do classificador HT, enquanto com classificador NB está posicionado como o segundo melhor método. Nas demais métricas o ANOVA_C tem um comportamento balanceado. É válido destacar que ANOVA_ C encontra-se colocado na segunda posição com respeito aos valores da métrica *Precision*.

Já em relação a métrica *Recall* o ANOVA_C não se encontra como o melhor posicionado a respeito a essa métrica. É válido destacar os altos valores de *Recall* do método SADD por reportar baixas quantidades de FN. Entretanto, apresenta resultados inferiores

na quase totalidade das demais métricas.

Finalizando, com base nas análises e testes estatísticos comparativos realizados na pesquisa, é demostrado que o desenvolvimento de um detector baseado na combinação das respostas de vários testes estatísticos é uma boa alternativa para tratar com mudanças de conceitos abruptas e graduais, aumentando a precisão dos modelos de classificação ou predição. O uso do método ANOVA_C é aconselhado, pelo fato de ser competitivo e possuir um comportamento estável, em cenários onde o objetivo é maximizar as acurácias dos classificadores assim como manter a precisão nas detecções das mudanças de conceitos. A utilização dos outros métodos (SADD, BFDD e OBDD), resultantes desta pesquisa, é recomentado quando se conta com recursos computacionais muito limitados. Já que o ANOVA_C tende a consumir um pouco mais de memória e tempo de execução do que os outros métodos, especialmente com o classificador base HT.

⋆−⋆

7.1 Contribuições

As principais contribuições deste investigação são:

- O fornecimento de uma nova metodologia de construção de detectores de mudanças de conceitos.
- O desenvolvimento de quatro métodos (SADD, BFDD, OBDD e ANOVA_C) para a detecção de mudanças de conceitos.

O método de detecção de mudanças de conceitos OBDD apresentado nesta investigação foi base do artigo a ser submetido:

- "*O'Brien Drift Detection Method*". **José L. M. Perez, Bruno I. F. Maciel, Roberto S. M. Barros**.

7.2 Trabalhos Futuros

- Modificar o sistema de votação proposto no comitê de métodos estatístico ANOVA_C, usando votação ponderada entre outros.
- Construir um comitê de métodos estatístico adaptativo.
- Implementar um sistema usando ANOVA não padrão para o trabalho em presença de diferentes condições (fluxos desbalanceados, dados semi-supervisados).
- Desenvolver um pacote que contenham conjuntos de funções a serem combinadas na construção de comitês de estatísticas no MOA.

- Implementar comitês usando os métodos estatísticos nos que são baseados os seguintes detectores: o STEPD, FPDD e WSTD.

REFERÊNCIAS

ABDI, H. O'brien test for homogeneity of variance. *Encyclopedia of Measurement and Statistics*, SAGE: Thousand Oaks, CA, USA, v. 2, p. 701–704, 2007.

AGGARWAL, C. C.; PHILIP, S. Y. A survey of synopsis construction in data streams. In: *Data Streams*. [S.l.]: Springer, 2007. p. 169–207.

AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. N. Database mining: a performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 5, n. 6, p. 914–925, 1993.

AGRESTI, A. A survey of exact inference for contingency tables. *Statistical Science*, v. 7, n. 1, p. 131–153, 1992.

ALIPPI, C.; BORACCHI, G.; ROVERI, M. A hierarchical, nonparametric, sequential change-detection test. In: IEEE. *Neural Networks (IJCNN), The 2011 International Joint Conference on*. [S.l.], 2011. p. 2889–2896.

ALIPPI, C.; BORACCHI, G.; ROVERI, M. Hierarchical change-detection tests. *IEEE transactions on neural networks and learning systems*, IEEE, v. 28, n. 2, p. 246–258, 2017.

ALLUA, S.; THOMPSON, C. B. Hypothesis testing. *Air Medical Journal*, Elsevier, v. 28, p. 108–110, 2009.

ASUNCION, A.; NEWMAN, D. *UCI machine learning repository*. 2007.

ATTAR, V.; CHAUDHARY, P.; RAHAGUDE, S.; CHAUDHARI, G.; SINHA, P. An instance-window based classification algorithm for handling gradual concept drifts. In: . Springer Berlin Heidelberg, 2012. p. 156–172. Disponível em: <http://dx.doi.org/10-.1007/978-3-642-27609-5_11>.

BABCOCK, B.; BABU, S.; DATAR, M.; MOTWANI, R.; WIDOM, J. Models and issues in data stream systems. In: ACM. *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. [S.l.], 2002. p. 1–16.

BACH, S. H.; MALOOF, M. A. Paired learners for concept drift. In: IEEE. *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. [S.l.], 2008. p. 23–32.

BAENA-GARCIA, M.; CAMPO-ÁVILA, J. D.; FIDALGO, R.; BIFET, A.; GAVALDÀ, R.; MORALES-BUENO, R. Early drift detection method. In: *Proceedings of the Fourth International Workshop on Knowledge Discovery from Data Streams*. [S.l.: s.n.], 2006. p. 77–86.

BARROS, R. S.; CABRAL, D. R.; JR, P. M. G.; SANTOS, S. G. RDDM: Reactive drift detection method. *Expert Systems with Applications*, Elsevier, v. 90, p. 344–355, 2017.

BARROS, R. S. M. *Advances in Data Stream Mining with Concept Drift*. 2017. Professorship (Full) Thesis. Centro de Informática, Universidade Federal de Pernambuco, Brazil.

BARROS, R. S. M. de; HIDALGO, J. I. G.; CABRAL, D. R. de L. Wilcoxon rank sum test drift detector. *Neurocomputing*, Elsevier, v. 275, p. 1954–1963, 2018.

BIFET, A. Adaptive learning and mining for data streams and frequent patterns. *ACM SIGKDD Explorations Newsletter*, ACM, v. 11, n. 1, p. 55–56, 2009.

BIFET, A.; GAVALDÀ, R. Learning from time-changing data with adaptive windowing. In: *Proceedings of the 7th SIAM International Conference on Data Mining (SDM'07)*. Minneapolis, MN, USA: [s.n.], 2007. p. 443–448.

BIFET, A.; HOLMES, G.; KIRKBY, R.; PFAHRINGER, B. MOA: Massive online analysis. *Journal of Machine Learning Research*, MIT Press, v. 11, p. 1601–1604, 2010.

BIFET, A.; HOLMES, G.; PFAHRINGER, B. Leveraging bagging for evolving data streams. In: SPRINGER. *Joint European conference on machine learning and knowledge discovery in databases*. [S.l.], 2010. p. 135–150.

BIFET, A.; HOLMES, G.; PFAHRINGER, B.; KIRKBY, R.; GAVALDÀ, R. New ensemble methods for evolving data streams. In: ACM. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2009. p. 139–148.

BIFET, A.; READ, J.; ŽLIOBAITĖ, I.; PFAHRINGER, B.; HOLMES, G. Pitfalls in benchmarking data stream classification and how to avoid them. In: *Machine Learning and Knowledge Discovery in Databases*. [S.l.]: Springer, 2013, (LNCS, v. 8188). p. 465–479.

BLANCO, F.; INOCENCIO, I. *Nuevos métodos para el aprendizaje en flujos de datos no estacionarios*. [S.l.]: Universidad de Granada, 2014.

BLUMAN, A. *Elementary Statistics: A Step by Step Approach*. New York, USA: McGraw-Hill, 2014.

BOQUÉ, R.; MAROTO, A. El análisis de la varianza (ANOVA). *Comparación de múltiples poblaciones, Universitat Rovira i Virgili, Italia*, 2004.

BOUCKAERT, R. R.; FRANK, E.; HALL, M. A.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. WEKA–experiences with a java open-source project. *Journal of Machine Learning Research*, v. 11, n. Sep, p. 2533–2541, 2010.

BREIMAN, L.; FRIEDMAN, J. H.; OLSHEN, R. A.; STONE, C. J. *Classification and Regression Trees*. Belmont, California: Wadsworth International Group, 1984. (Wadsworth Statistics / Probability series).

BROWN, M. B.; FORSYTHE, A. B. Robust tests for the equality of variances. *Journal of the American Statistical Association*, Taylor & Francis Group, v. 69, n. 346, p. 364–367, 1974.

BRZEZINSKI, D.; STEFANOWSKI, J. Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, v. 25, n. 1, p. 81–94, 2014.

BRZEZINSKI, D.; STEFANOWSKI, J. Stream classification. In: *Encyclopedia of Machine Learning*. [S.l.]: Springer, 2016.

BUSSAB, W. d. O. *Estatística Básica/Wilton de O. Bussab, Pedro A. Morettin–*. [S.l.]: São Paulo: Saraiva, 2004.

CABRAL, D. R. de L.; BARROS, R. S. M. de. Concept drift detection based on fisher's exact test. *Information Sciences*, Elsevier, p. 220–234, 2018.

CABRAL, D. R. L. *Testes estatísticos e detecções de Mudanças de conceitos em fluxos de dados*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, Brazil, 2017.

CAZZOLATO, M. T. *Classificação de data streams utilizando árvore de decisão estatística e a teoria dos fractais na análise evolutiva dos dados*. Dissertação (Mestrado) — Universidade Federal de São Carlos, 2014.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995.

CORTEZ, P.; CERDEIRA, A.; ALMEIDA, F.; MATOS, T.; REIS, J. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, Elsevier, v. 47, n. 4, p. 547–553, 2009.

DAVIS, R. B.; MUKAMAL, K. J. Hypothesis testing: Means. *Circulation*, v. 114, n. 10, p. 1078–1082, 2006.

DAWID, A. P.; VOVK, V. G. et al. Prequential probability: Principles and properties. *Bernoulli*, v. 5, n. 1, p. 125–162, 1999.

DELANY, S. J.; CUNNINGHAM, P.; TSYMBAL, A.; COYLE, L. A case-based technique for tracking concept drift in spam filtering. *Knowledge-Based Systems*, Elsevier, v. 18, n. 4-5, p. 187–195, 2005.

DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, MIT Press, v. 7, p. 1–30, 2006.

DÍAZ, A. A. O. *Algoritmo multiclasificador con aprendizaje incremental al que manipula cambios de conceptos*. [S.l.]: Universidad de Granada, 2014.

DIETTERICH, T. Nature encyclopedia of cognitive science. *Machine learning. Macmillan*, 2003.

DU, L.; SONG, Q.; JIA, X. Detecting concept drift: An information entropy based method using an adaptive sliding window. *Intelligent Data Analysis*, IOS Press, v. 18, n. 3, p. 337–364, 2014.

DU, L.; SONG, Q.; ZHU, L.; ZHU, X. A selective detector ensemble for concept drift detection. *The Computer Journal*, Oxford University Press, v. 58, n. 3, p. 457–471, 2014.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern classification (Second Edition)*. New York, NY, USA: JOHN WlLEY & SONS, INC., 2001.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. L. F. *Inteligência Artificial: Uma Abordagem de Aprendizagem de Máquina*. Rio de Janeiro, RJ, Brasil: LTC, 2011.

FAN, W.; HUANG, Y.-a.; WANG, H.; YU, P. S. Active mining of data streams. In: SIAM. *Proceedings of the 2004 SIAM International Conference on Data Mining*. [S.l.], 2004. p. 457–461.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P.; UTHURUSAMY, R. *Advances in knowledge discovery and data mining*. [S.l.]: AAAI press Menlo Park, 1996. v. 21.

FISHER, R. A. *Statistical methods for research workers*. [S.l.]: Genesis Publishing Pvt Ltd, 1925.

FRANK, A.; ASUNCION, A. UCI machine learning repository [http://archive. ics. uci. edu/ml]. irvine, ca: University of california. *School of information and computer science*, v. 213, 2010.

FRÍAS-BLANCO, I.; CAMPO-ÁVILA, J. del; RAMOS-JIMÉNEZ, G.; CARVALHO, A. C.; ORTIZ-DÍAZ, A.; MORALES-BUENO, R. Online adaptive decision trees based on concentration inequalities. *Knowledge-Based Systems*, v. 104, p. 179 – 194, 2016. Disponível em: <http://www.sciencedirect.com/science/article/pii-/S0950705116300715>.

FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937.

FUKUNAGA, K.; NARENDRA, P. M. A branch and bound algorithm for computing k-nearest neighbors. *IEEE transactions on computers*, IEEE, v. 100, n. 7, p. 750–753, 1975.

GAMA, J.; GABER, M. *Learning from data streams*. [S.l.]: Springer, 2007.

GAMA, J.; MEDAS, P.; CASTILLO, G.; RODRIGUES, P. Learning with drift detection. In: *Advances in Artificial Intelligence: SBIA 2004*. [S.l.]: Springer, 2004, (LNCS, v. 3171). p. 286–295.

GAMA, J.; ŽLIOBAITĖ, I.; BIFET, A.; PECHENIZKIY, M.; BOUCHACHIA, A. A survey on concept drift adaptation. *ACM Computing Surveys*, v. 46, n. 4, p. 44:1–37, 2014.

GAMALLO, P.; GARCIA, M.; FERNÁNDEZ-LANZA, S. TASS: A naive-bayes strategy for sentiment analysis on spanish tweets. In: *Workshop on Sentiment Analysis at SEPLN (TASS2013)*. [S.l.: s.n.], 2013. p. 126–132.

GODASE, A.; ATTAR, V. Classification of data streams with skewed distribution. In: *Proceedings of the IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*. [S.l.: s.n.], 2012. p. 151–156.

GOLDSCHMIDT, R.; PASSOS, E. Data mining: um guia prático, conceitos, técnicas, ferramentas, orientações e aplicações. *Rio de Janeiro: Campus*, v. 1, 2005.

GONÇALVES, P. M.; SANTOS, S. G. T. C.; BARROS, R. S. M.; VIEIRA, D. C. L. A comparative study on concept drift detectors. *Expert Systems with Applications*, Elsevier, v. 41, n. 18, p. 8144–8156, 2014.

GONÇALVES JR., P. M.; BARROS, R. S. M. RCD: A recurring concept drift framework. *Pattern Recognition Letters*, Elsevier, v. 34, n. 9, p. 1018–1025, 2013.

GRAYBILL, F. A.; IYER, H. K.; BURDICK, R. K. *Applied statistics: A first course in inference*. [S.l.]: Prentice Hall, 1998.

HATCHAVANICH, D. A comparison of type i error and power of bartlett's test, levene's test and o'brien's test for homogeneity of variance tests. *Southeast Asian Journal of Sciences*, v. 3, n. 2, 2014.

HIDALGO, J. I. G. *Experiências com Variações Prequential para Avaliação da Aprendizagem em Fluxo de Dados*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, Brazil, 2017.

HOEFFDING, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, Taylor & Francis Group, v. 58, n. 301, p. 13–30, 1963.

HULTEN, G.; SPENCER, L.; DOMINGOS, P. Mining time-changing data streams. In: *Proceedings of the Seventh ACM SIGKDD Intern. Conference on Knowledge Discovery and Data Mining*. New York, USA: [s.n.], 2001. (KDD '01), p. 97–106.

IENCO, D.; BIFET, A.; ŽLIOBAITĖ, I.; PFAHRINGER, B. Clustering based active learning for evolving data streams. In: SPRINGER. *International Conference on Discovery Science*. [S.l.], 2013. p. 79–93.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in bayesian classifiers. In: *Eleventh Conference on Uncertainty in Artificial Intelligence*. San Mateo: Morgan Kaufmann, 1995. p. 338–345.

KATAKIS, I.; TSOUMAKAS, G.; VLAHAVAS, I. *Incremental clustering for the classification of concept-drifting data streams*. [S.l.]: Citeseer, 2008.

KATAKIS, I.; TSOUMAKAS, G.; VLAHAVAS, I. P. An ensemble of classifiers for coping with recurring contexts in data streams. In: *ECAI*. [S.l.: s.n.], 2008. p. 763–764.

KHAMASSI, I.; SAYED-MOUCHAWEH, M.; HAMMAMI, M.; GHÉDIRA, K. Self-adaptive windowing approach for handling complex concept drift. *Cognitive Computation*, Springer, v. 7, n. 6, p. 772–790, 2015.

KOHAIL, S. N. *Learning Concept Drift Using Adaptive Training Set Formation Strategy*. Tese (Doutorado) — The Islamic University of Gaza, 2011.

KRAWCZYK, B.; MINKU, L. L.; GAMA, J.; STEFANOWSKI, J.; WOŹNIAK, M. Ensemble learning for data stream analysis: a survey. *Information Fusion*, Elsevier, v. 37, p. 132–156, 2017.

KUNCHEVA, L. I. *Combining pattern classifiers: methods and algorithms*. [S.l.]: John Wiley & Sons, 2004.

LARSON, R.; FARBER, B. *Elementary Statistics: Picturing the World*. Fourth. [S.l.]: Pearson, 2010.

LEMAIRE, V.; SALPERWYCK, C.; BONDU, A. A survey on supervised classification on data streams. In: *Business Intelligence*. [S.l.]: Springer, 2015. p. 88–125.

LEVENE, H. Robust tests for equality of variances. *Contributions to probability and statistics*, v. 1, p. 278–292, 1960.

LIM, T.-S.; LOH, W.-Y. A comparison of tests of equality of variances. *Computational Statistics & Data Analysis*, Elsevier, v. 22, n. 3, p. 287–301, 1996.

LIU, J.; MIAO, Q.; SUN, Y.; SONG, J.; QUAN, Y. Fast structural ensemble for one-class classification. *Pattern Recognition Letters*, Elsevier, v. 80, p. 179–187, 2016.

LOSING, V.; HAMMER, B.; WERSING, H. KNN classifier with self adjusting memory for heterogeneous concept drift. In: *Proceedings of IEEE International Conference on Data Mining (ICDM)*. Barcelona, Spain: [s.n.], 2016. p. 291–300.

MACIEL, B. I. F.; SANTOS, S. G. T. C.; BARROS, R. S. M. A lightweight concept drift detection ensemble. In: *Proceedings of 27th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. Vietri sul Mare, Italy: [s.n.], 2015. p. 1061–1068.

MARON, O.; MOORE, A. W. Hoeffding races: Accelerating model selection search for classification and function approximation. *Robotics Institute*, p. 263, 1993.

MASSART, D. L.; VANDEGINSTE, B. G.; BUYDENS, L.; LEWI, P.; SMEYERS-VERBEKE, J.; JONG, S. d. *Handbook of chemometrics and qualimetrics: Part A*. [S.l.]: Elsevier Science Inc., 1997.

MATTHEWS, B. W. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, Elsevier, v. 405, n. 2, p. 442–451, 1975.

MEHTA, C. R.; PATEL, N. R. *SPSS Exact Tests 7.0 ^{TM} for Windows ®*. Chicago, USA: SPSS Inc, 1996.

MINKU, L. L.; WHITE, A. P.; YAO, X. The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Transactions on Knowledge and Data Engineering*, v. 22, n. 5, p. 730–742, May 2010. ISSN 1041-4347.

MINKU, L. L.; YAO, X. DDD: A new ensemble approach for dealing with concept drift. *IEEE transactions on knowledge and data engineering*, IEEE, v. 24, n. 4, p. 619–633, 2012.

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

NEMENYI, P. Distribution-free multiple comparisons. *Princeton University*, 1963.

NISHIDA, K.; YAMAUCHI, K. Detecting concept drift using statistical testing. In: *Proceedings of the 10th International Conference on Discovery Science (DS'07)*. [S.l.]: Springer, 2007. (LNCS, v. 4755), p. 264–269.

NORDSTOKKE, D. W.; ZUMBO, B. D. A new nonparametric levene test for equal variances. *Psicológica*, Universitat de València, v. 31, n. 2, 2010.

O'BRIEN, R. G. A general ANOVA method for robust tests of additive models for variances. *Journal of the American Statistical Association*, Taylor & Francis, v. 74, n. 368, p. 877–880, 1979.

O'BRIEN, R. G. A simple test for variance effects in experimental designs. *Psychological Bulletin*, v. 89, n. 3, p. 570–574, 1981.

OGURI, P. *Aprendizado de máquina para o problema de sentiment classification.* Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro, Brasil, 2006.

ORALLO, H.; RAMIREZ, J.; QUINTANA, C. R.; ORALLO, M. J. H.; QUINTANA, M. J. R.; RAMÍREZ, C. F. *Introducción a la Minería de Datos.* [S.l.]: Pearson Prentice Hall,, 2004.

PERRY, J.; KENT, A.; BERRY, M. Machine literature searching X. machine language; factors underlying its design and development. *American Documentation*, v. 6, n. 4, p. 242–254, 1955.

PESARANGHADER, A.; VIKTOR, H. Fast hoeffding drift detection method for evolving data streams. In: *Machine Learning and Knowledge Discovery in Databases.* [S.l.]: Springer, 2016. (LNCS, v. 9852), p. 96–111.

RAFTER, J. A.; ABELL, M. L.; BRASELTON, J. P. Multiple comparison methods for means. *Siam Review*, SIAM, v. 44, n. 2, p. 259–278, 2002.

READ, J.; BIFET, A.; PFAHRINGER, B.; HOLMES, G. Batch-incremental versus instance-incremental learning in dynamic and evolving data. *Advances in Intelligent Data Analysis XI*, Springer, p. 313–323, 2012.

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

ROGAN, J. C.; KESELMAN, H. Is the anova f-test robust to variance heterogeneity when sample sizes are equal?: An investigation via a coefficient of variation. *American Educational Research Journal*, Sage Publications Sage CA: Los Angeles, CA, v. 14, n. 4, p. 493–498, 1977.

ROSS, G. J.; ADAMS, N. M.; TASOULIS, D. K.; HAND, D. J. Exponentially weighted moving average charts for detecting concept drift. *Pattern Recognition Letters*, Elsevier, v. 33, n. 2, p. 191–198, 2012.

ROSS, G. J.; TASOULIS, D. K.; ADAMS, N. M. Nonparametric monitoring of data streams for changes in location and scale. *Technometrics*, Taylor & Francis, v. 53, n. 4, p. 379–389, 2011.

RUTKOWSKI, L.; JAWORSKI, M.; PIETRUCZUK, L.; DUDA, P. A new method for data stream mining based on the misclassification error. *IEEE transactions on neural networks and learning systems*, IEEE, v. 26, n. 5, p. 1048–1059, 2015.

SALPERWYCK, C.; BOULLÉ, M.; LEMAIRE, V. Concept drift detection using supervised bivariate grids. In: *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN).* Killarney, Ireland: [s.n.], 2015. p. 1–9.

SANTOS, S. G. T. C.; BARROS, R. S. M.; GONÇALVES JR., P. M. Optimizing the parameters of drift detection methods using a genetic algorithm. In: *Proceedings of 27th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'15)*. Vietri sul Mare, Italy: [s.n.], 2015. p. 1077–1084.

SIEGEL, S.; JR, N. J. C. *Estatística não-paramétrica para ciências do comportamento*. [S.l.]: Artmed Editora, 1975.

SPARKS, R. S. Cusum charts for signalling varying location shifts. *Journal of Quality Technology*, American Society for Quality, v. 32, n. 2, p. 157, 2000.

STANLEY, K. O. Learning concept drift with a committee of decision trees. *Informe técnico: UT-AI-TR-03-302, Department of Computer Sciences, University of Texas at Austin, USA*, 2003.

SUN, Y.; WANG, Z.; LIU, H.; DU, C.; YUAN, J. Online ensemble using adaptive windowing for data streams with concept drift. *International Journal of Distributed Sensor Networks*, SAGE Publications Sage UK: London, England, v. 12, n. 5, p. 4218973, 2016.

TROYANO, F. J. F.; RUIZ, J. S. A.; SANTOS, J. C. R. Incremental rule learning and border examples selection from numerical data streams. *Journal of Universal Computer Science*, Graz University of Technology, v. 11, n. 8, p. 1426–1439, 2005.

VERDIER, G.; HILGERT, N.; VILA, J.-P. Adaptive threshold computation for cusum-type procedures in change detection and isolation problems. *Computational Statistics & Data Analysis*, Elsevier, v. 52, n. 9, p. 4161–4174, 2008.

WANG, H.; FAN, W.; YU, P. S.; HAN, J. Mining concept-drifting data streams using ensemble classifiers. In: ACM. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2003. p. 226–235.

WANKHADE, K.; DONGRE, S.; THOOL, R. New evolving ensemble classifier for handling concept drifting data streams. In: *Proceedings of the 2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC)*. [S.l.: s.n.], 2012. p. 657–662.

WITTEN, I. H.; FRANK, E.; HALL, M. A.; PAL, C. J. *Data Mining: Practical machine learning tools and techniques*. [S.l.]: Morgan Kaufmann, 2016.

WOŹNIAK, M.; KSIENIEWICZ, P.; CYGANEK, B.; WALKOWIAK, K. Ensembles of heterogeneous concept drift detectors-experimental study. In: SPRINGER. *IFIP International Conference on Computer Information Systems and Industrial Management*. [S.l.], 2016. p. 538–549.

ZHOU, Z.-H.; WU, J.; TANG, W. Ensembling neural networks: many could be better than all. *Artificial intelligence*, Elsevier, v. 137, n. 1-2, p. 239–263, 2002.

ŽLIOBAITĖ, I. Learning under concept drift: an overview. *arXiv preprint arXiv:1010.4784*, 2010.

ŽLIOBAITĖ, I.; BIFET, A.; READ, J.; PFAHRINGER, B.; HOLMES, G. Evaluation methods and decision theory for classification of streaming data with temporal dependence. *Machine Learning*, Springer, v. 98, n. 3, p. 455–482, 2015.

APÊNDICE A – HOEFFDING

Neste anexo, com o objetivo de esclarecer da melhor maneira terminologias usadas na investigação, decidiu-se apresentar o teorema da inequação de Hoeffding e o pseudo-código do classificador base Hoeffding Tree.

A.1 Teorema da inequação de Hoeffding

Sejam $X_1, X_2, ..., X_n$, variáveis aleatórias independentes tais que para cada $X_i \in [a_i, b_i]$ onde $0 \leq i \leq n$, e com probabilidade máxima δ. Seja a média empírica $\bar{X} = \sum_{i=1}^{n} \frac{X_i}{n}$ uma variável aleatória cujo valor esperado é $E\left[\bar{X}\right]$. Então, para qualquer $\varepsilon_H > 0$ se pode aplicar equação A.1.

$$P_r(\left|\bar{X} - E\left[\bar{X}\right]\right| \geq \varepsilon) \leq 2e^{-2n^2\varepsilon^2 / \sum_{i=1}^{n}(b_i - a_i)_2} \tag{A.1}$$

O erro pode ser estimado, $\varepsilon_\delta = \sqrt{\frac{1}{2n} ln \frac{1}{\delta}}$, sendo conhecido ademais que a desigualdade de Hoeffding assume unicamente variáveis aleatórias independentes, mas não é assumida nenhuma função de probabilidade. O estatístico em questão ($\bar{X}$) e a taxa de erro (ε_δ) podem ser calculados em O (1) de complexidade temporal e espacial, o que faz o teorema aplicável a aprendizagem em fluxos de dados (BLANCO; INOCENCIO, 2014).

A.2 Pseudo-código do classificador HT

A continuação neste anexo será colocado o pseudo-código do Hoeffding Tree, um dos classificadores mais usados na área de aprendizagem de máquina em fluxos de dados não estacionários, em cenários com presença de mudança de conceito.
Já exposto no começo da pesquisa, o classificador HT é conhecido também como VFDT (Very Fast Decision Trees for Mining High-Speed Data Streams). O mesmo permite utilizar o Ganho de Informação ou Gini Index como medida de avaliação (HULTEN; SPENCER; DOMINGOS, 2001; CAZZOLATO, 2014).
Como se pode perceber na implementação, na linha **1** o algoritmo começa com um nó folha, a raiz da árvore. Quando uma nova instância chega, ele é ordenado até sua folha correspondente, são coletadas as estatísticas suficientes do dado e é incrementado n_l, que é o número de instâncias observadas no nó l (linhas **3-5**).
Já na linha **6** é verificado se foram observadas instâncias suficientes no nó em questão para tentar realizar a divisão de mesmo. Isto é realizado utilizando o n_{min} (número mínimo de instâncias que devem ser lidos para a tentativa de divisão) e também

Algoritmo 8: Hoeffding tree induction algorithm

Input: n_{min} (minimum number of examples of the tolerance period), τ (a tie threshold)
Output: HT decision tree

```
1  Let HT be a tree with a single leaft (the root)
2  for all training examples do
3      Sort example into leaft l using HT
4      Update sufficient statistics in l
5      Increment n_l the number of examples seen at l
6      if n_l mod n_min = 0 and examples seen at l not all of same class then
7          Compute Ḡ_l(X_i) for each attribute
8          Let X_a be attribute with highest Ḡ_l
9          Let X_b be attribute with second-highest Ḡ_l
10         Compute Hoeffding bound ε = sqrt(R²(ln 1/δ) / 2n)
11         if X_a ≠ X_∅ and (Ḡ_l(X_a) − Ḡ_l(X_b)) > ε or ε < τ then
12             Replace l with an internal node that splits on X_a
13             for all branches of the split do
14                 Add a new leaft with initialized sufficient statistics
```

verificando se todos os dados do nó até aquele momento pertencem à mesma classe. Se sim, não há a necessidade de divisão.
Nas linhas **7-9** é utilizada a heurística Ganho de Informação e o Gini Index para a escolha dos dois melhores atributos a serem utilizados na divisão do nó. Para solucionar o problema de decidir exatamente quantas instâncias são necessárias para cada nó, na linha **10** é utilizado o Hoeffding bound (equação A.1).
Na linha **11** são verificadas as condições a seguir:

- $X_a \neq X_\emptyset$: se ao menos um atributo foi escolhido, ou seja, se o melhor atributo é diferente de nulo;
- $(\bar{G}_l(X_a) - \bar{G}_l(X_b)) > \varepsilon$: se a diferença entre os dois melhores atributos é maior que ε. Essa condição é testada para evitar realizar a divisão de um nó quando dois ou mais atributos têm valores muitos próximos, pois um atributo poderia se tornar o melhor nas próximas iterações;
- $\varepsilon < \tau$: conforme n (número de instâncias observadas no nó) aumenta, ϵ tende a diminuir. Caso os dois melhores atributos tenham valores muito próximos em diversas iterações, ϵ seria tão pequeno quando τ, que é um critério de desempate.

Caso a condição da linha **11** seja satisfeita, na linha **12** a então folha l torna-se um nó interno que divide utilizando o atributo X_a. Na linha **14** são iniciadas as estatísticas suficientes para cada folha que resulta da divisão do nó.

APÊNDICE B – DETALHES DA IMPLEMENTAÇÃO

Neste apêndice é realizada uma descrição detalhada das funcionalidades do MOA. Assim como explica-se seu processo de instalação e requerimentos para o uso do mesmo. Além disso, é apresentada a descrição das tarefas (Script or task) especificamente de classificação usados na experimentação.

B.1 Descrição de instalação do MOA

O framework MOA foi implementado na linguagem de programação java. Para sua execução e desenvolvimento é precisado pelo menos da versão 5.0 da Java Runtime Environment (JRE) e Kit de desenvolvimento de software (SDK) respectivamente. O framework MOA é multiplataforma, pode ser executado nos sistemas operacionais Windows, Unix/Linux e Macintosh.
Os ambientes de desenvolvimento integrados (do inglês, IDE) Eclipse e NetBeans são os mais usados para o trabalho com o framework MOA. Precisando-se dos pacotes moa, weka, zizeofag com extensão de arquivo *.jar. Os mesmo podem encontra-se nos link a seguir:

1. `https://sourceforge.net/projects/moa-datastream/`;
2. `https://sourceforge.net/projects/weka/`;
3. `http://www.jroller.com/resources/m/maxim/sizeofag.jar`.

Na figura 8 apresenta-se a tela principal do framework MOA, sendo executado o detector OBDD junto com o classificador base HT. Observa-se que o MOA pode ser usado para realizar tarefas de classificação, regressão, agrupamentos (clustering), tratamentos de outliers e detecção de mudanças de conceitos (concept drift). Una versão estendida do MOA versão 2014, contendo os detectores de mudanças de conceito SADD, BFDD, OBDD e ANOVA_C é fornecida junto com esta teses facilitando posteriores comparação e otimização dos teste por parte de outros investigadores. Como o sistema operacional usado foi Windows podemos usar a seguinte linhas de comando para executar uma tarefa:

- ***java.exe -cp*** *<moaFolder>\moa.jar -javaagent:<sizeofagFolder>\sizeofag.jar moa.DoTask<taskName><taskParameters>*

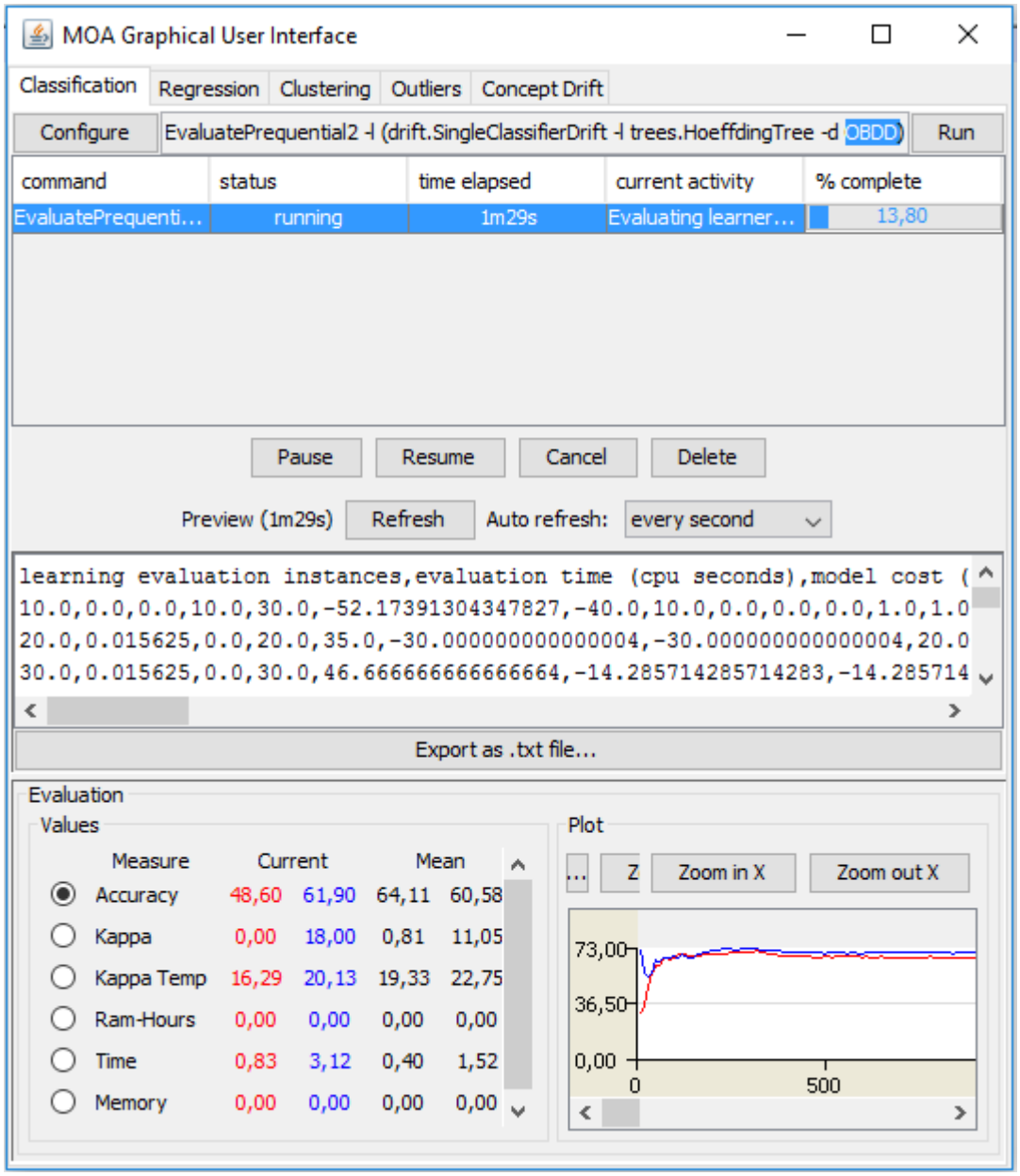


Figura 8 – Tela de execução do framework MOA

- ***java -cp*** *<moaFolder>\moa.jar -javaagent:<sizeofagFolder>\sizeofag.jar moa.gui.TaskLauncher*

  A última opção executa a interface gráfica do framework.

Para exibir mais claramente o processo, apresentamos um exemplo de um experimento das dois variantes, onde tudo o que é necessário encontra-se armazenado no diretório experimentos_tese. O exemplo da primeira variante que a continuação é colocado, se pode observar que é usado o detector OBDD junto com o classificador base HT para a base dados real airlines.arff.

- ***java.exe -cp*** *"F:\experimentos_tese\moa2014.jar" -javaagent:"F:\experimentos_tese\sizeofag.jar"*

moa.DoTask EvaluatePrequential
-l (drift.SingleClassifierDrift -l trees.HoeffdingTree -d OBDD) -s (ArffFileStream -f (F:\experimentos_tese\bases_reais\airlines.arff))

- ***java -cp*** *"F:\experimentos_tese\moa2014.jar"*
-javaagent:"F:\experimentos_tese\sizeofag.jar"moa.gui.TaskLauncher

B.2 MOAManager

A ferramenta MOAManager atualmente se encontra em desenvolvimento e teste. Implementada pelos autores Bruno I. F. Maciel, Silas G. T. Carvalho Santos, e Roberto S. M. Barros. A mesma forma parte das contribuições do doutorado de Maciel, permitindo facilitar a execução, coleta e análise de dados dos experimentos executados no MOA Framework. Possui código implementando, usuários ativos, código comentado e documentação em andamento. Uma tela de execução da ferramenta é mostrada observada na ilustração 9.

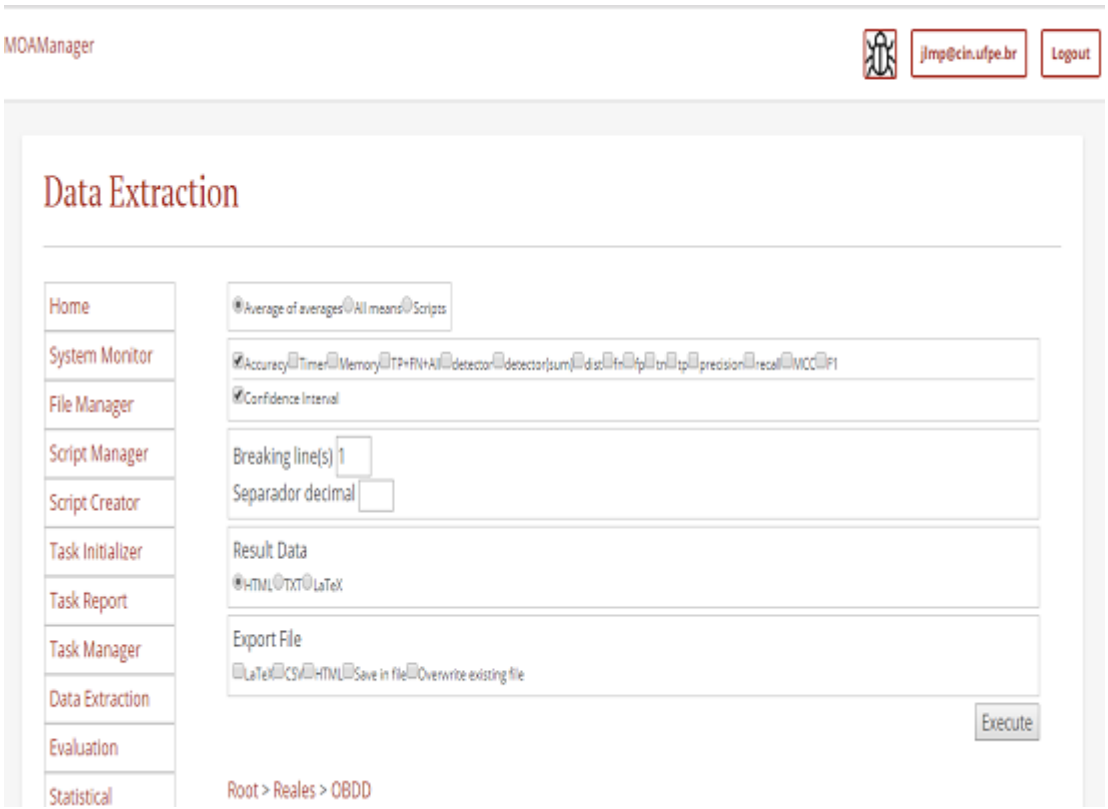


Figura 9 – Tela de execução do MOAManager

B.3 Descrição de parâmetros

Os algoritmos SADD, BFDD, OBDD, ANOVA_C foram implementados de acordo com os pseudo-códigos 1, 2, 3 (capítulo, 4), e 4 (capítulo 5), se faz válido ressaltar que a versão do MOA usada foi a do ano 2014. Onde os detectores desenvolvidos estendem da classe AbstractChangeDetector e foram armazenado no pacote:

$$moa2014/src/moa/classifiers/core/driftdetection \begin{cases} \textbf{Classes}(Detectores) \\ \quad SADD.java \\ \quad BFDD.java \\ \quad OBDD.java \\ ANOVA\_C.java \end{cases}$$

Os detectores estão disponível a partir da caixa de diálogo de seleção da interface gráfica, podendo ser ajustados seus seguintes parâmetros:

- -d nome do detector (-d OBDD),
- -r tamanho da janela recente (-r 100),
- -o nível a ser atingido para notificar a mudança (-o 0.001),
- -w nível a ser atingido para notificar a alerta (-w 0.05),
- -m tamanho da janela antiga (-m 200).

  Além do associado ao classificador base usado e o fluxo de dados.

- -l classificador base a treinar (-l trees.HoeffdingTree),
- -s fluxo de dados para aprendizagem (-s generators.AgrawalGenerator).

Estes não são os únicos parâmetros que se podem encontrar dentro das tarefas a executar, mas são os mais usados na construção de detectores. A seguir apresentamos um exemplo de como é indicada a execução das tarefas no MOA.

1. EvaluatePrequential -l (drift.SingleClassifierDrift -l trees.HoeffdingTree -d ANOVA_S) -s (ConceptDriftStream -s (ConceptDriftStream -s (ConceptDriftStream -s (ConceptDriftStream -s (generators.AgrawalGenerator -f 1 -p 1.0) -d (generators.AgrawalGenerator -f 2) -p 2000 -w 1) -d (generators.AgrawalGenerator -f 3) -p 4000 -w 1) -d (generators.AgrawalGenerator -f 4) -p 6000 -w 1) -d (generators.AgrawalGenerator -f 5) -p 8000 -w 1) -r 30 -c -i 10000 -f 10 -q 10 -b 40

B.4 Bases de dados

O framework MOA contém um número amplo de geradores de bases de dados. Bem como também permite a integração de bases reais no formato (extensão) de arquivo *.arff. No anexo apresenta-se a tabela 10 mostrando os geradores de fluxos de dados, que permitem introduzir mudanças de conceito mais usado nas investigações na atualidade. Também se exibe uma quantidade razonável de bases reais (tabela 11) disponível na internet e a descrição detalhada das características dos atributos (tabela 12) presentes nas bases de dados criadas pelo gerador Agrawal.

Geradores	Acrônimos	Caracteristicas	Classes
LED	LED	24	10
Sine	Sine	2	2
Mixed drift	Mixed	4	2
Wavefrom	Wave21	21	3
Wavefrom	Wave40	40	3
SEA Concept	SEA	3	2
Random RBF Simples	RBFS	10	2
Random RBF Complexa	RBFC	50	2
Random Tree Simples	RTS	20	2
Random Tree Complexa	RTC	100	2
Function Generator (Agrawal)	Agrawal	9	2
STAGGER Concept	STAGGER	3	2
Rotating Hyperplane	Hyperplane	10	2

Tabela 10 – Características dos gerados de base de dados artificias mais usadas na área.

Base de dados	Acrônimos	Instâncias	Nominal	Numérico	Valores Faltantes	Classes
Airlines	AIR	539.382	4	3	no	2
Adult	ADU	32.561	8	6	sim	2
Bank marketing	BAN	41.188	9	7	no	2
Connect-4	COM	67.557	21	0	no	3
Forest Covert	COV	581.012	44	10	no	7
Cars	CAR	1.728	6	0	no	4
EEG Eye State	EYE	14.980	0	14	no	2
Electricity	ELE	45.312	1	7	yes	2
Letter Recognition	LET	20.000	0	16	no	26
Mushroom	MUS	8.124	22	0	yes	2
NSL-KDD 99 joined	NSLK	148.561	7	34	no	2
Nursey	NUR	12.960	8	0	no	5
Outdoor	OUT	4.000	0	21	no	40
Poker Hand	POK	1.000.000	10	0	no	10
Rialto	RIA	82.250	0	27	no	10
Spam coprus 2	SPA	9.323	500	0	no	2
Segment	SEG	2.310	0	19	no	7
Usenet1	USE1	1.500	100	0	no	2
Usenet2	USE2	1.500	100	0	no	2
Usenet3	USE3	3.000	100	0	no	2
WineWhite	WINW	4.898	0	11	no	1
WineRed	WINR	1.599	0	11	no	1
Weather	WEA	18.159	0	8	no	2

Tabela 11 – Características das base de dados reais mais usadas na área

Atributo	**Descripção**	**Valores**
salary	salário	de 20000 a 150000
commision	Comissão	Si salary $\geq$ 75000 $\Rightarrow$ comission = 0, en otro caso, entre 10000 e 75000
age	idade	Entre 20 e 80
elevel	nível de escolaridade	Entre 0 e 4 (categórico)
car	marca do carro	Entre 1 e 20 (categórico)
zipcode	código postal (CEP)	Entre 1 e 9 (categórico)
hvalue	valor do imóvel	Entre 0.5k100000 e 1.5k100000 onde k depende de zipcode
hyears	idade do imóvel	Entre 1 e 30
loan	valor do empréstimo	Entre 0 e 500000

Tabela 12 – Descrição detalhada dos atributos das base de dados criadas pelo gerador Agrawal.

B.5 Critérios de avaliação

O anexo pretende deixar expostas as formulações matemáticas dos critérios de avaliação usadas na pesquisa. Os critérios de avaliação tradicional como precisão, sensibilidade (recall), F_1 e MCC , que são usadas na pesquisa no capítulo 6, sendo calculadas as mesmas como apresentamos abaixo:

$$Precision = \frac{TP}{TP + FP} \tag{B.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{B.2}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{B.3}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (FN + TN) \times (FP + TN) \times (TP + FN)}} \tag{B.4}$$

Embora não sendo usadas na pesquisa como critérios de avaliação se menciona as taxa de falsos positivos (também chamada de taxas de alarme e com acrônimo em inglês, FPR) e falsos negativos (que denota a taxa de detecção de falta ou a taxa de detecção de falha, de acrônimo em inglês FNR) formuladas como a continuação se apresenta:

$$FPR = \frac{FP}{TP + FP} \tag{B.5}$$

$$FNR = \frac{FN}{TP + FN} \tag{B.6}$$

As taxas com pontuações menores representam os métodos com melhores desempenhos.

B.6 Teste de Friedman e Pós-Teste Nemenyi

B.6.1 Teste de Friedman

Na área de aprendizagem de máquina usualmente é utilizado o estatístico de *Friedman* (F_F) para comparar o desempenho dos preditores. F_F compara dados amostrais de forma emparelhados, ou seja, quando o mesmo indivíduo é avaliado mais de uma vez (HIDALGO, 2017). Este método estatístico não utiliza os dados numéricos diretamente, mas sim os postos ocupados por eles após a ordenação feita para cada grupo separadamente. Após a ordenação é testada a hipótese de igualdade da soma dos postos de cada grupo (FRIEDMAN, 1937; HIDALGO, 2017).

Com o objetivo de compreender mais claramente o funcionamento do estatístico F_F, e sua utilização com os resultados da investigação, se apresenta uma explicação detalhada de como foram comparados os métodos propostos nesta dissertação junto com os métodos selecionados do estado da arte.
O desempenho dos métodos adaptativos (classificador + detector) obtidos da experimentação pode ser organizado pela matriz B.7,

$$\begin{array}{cc} & \text{Adaptive Methods} \\ \text{Datasets} & \begin{bmatrix} X_{11} & X_{12} & ... & X_{1n} \\ X_{21} & X_{21} & ... & X_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ X_{m1} & X_{m2} & ... & X_{mn} \end{bmatrix} \end{array} \tag{B.7}$$

onde X_{ij} denota o desempenho do i-ésimo conjunto de dados na j-ésimo método (para $i = 1, ..., m$ e $j = 1, ..., n$).
No teste é assumido que as observações (diferenças percentuais entre as médias) nas diferentes colunas são independentes. O primeiro passo é ordenar as entradas X_{ij} da matriz B.7 de forma crescente por filas. Depois, cada entrada é substituída por seu relativo posto (posição) em relação às outras observações na j-ésima coluna como pode-se apreciar na expressão matricial B.8, onde R_{ij} é o posto do conjunto de dados i na j-ésima variação (HIDALGO, 2017).

$$R = \begin{bmatrix} R_{11} & R_{12} & ... & R_{1n} \\ R_{21} & R_{22} & ... & R_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ R_{m1} & R_{m2} & ... & R_{mn} \end{bmatrix} \tag{B.8}$$

Consequentemente, é realizada a soma dos postos resultantes por coluna para obter os respectivos ranks. Não entanto, vale ressaltar que a soma da j-ésima coluna $R_j = \sum_{i=1}^{m} Rij$, $\forall\ j = 1, ..., n$, depende de como o j-ésimo conjunto de dados se comporta em relação aos outros conjuntos de dados $(n - 1)$. Desse modo, assumindo a hipótese nula do método de *Friedman* (não há diferença entre as variações), é realizado o cálculo da estatística do teste como ilustra-se na equação B.9, onde n e m referem-se ao número de filas e colunas da matriz B.8 respectivamente, e R_j é o rank corresponde a cada coluna.

$$S = \frac{12}{nm\,(m+1)} \sum_{j=1}^{n} \left[R_j - \frac{n\,(m+1)}{2} \right]^2 \tag{B.9}$$

Finalmente procura-se o valor crítico associado ao resultado do valor S na tabela de distribuição de qui-quadrado com $m - 1$ graus de liberdade para determinar se a

hipótese nula é rejeitada, uma vez que o valor crítico seja menor que S. No caso contrário, a hipótese nula não será rejeitada (HIDALGO, 2017).

B.6.2 Pós-Teste de Nemenyi

Quando o teste de *Friedman* rejeita a hipótese nula, é preciso estabelecer quais são as diferenças significativas entre as métodos adaptativos. Para esse fim, procede-se com a execução de um Pós-Teste (teste realizado depois que outro teste global foi executado). Várias são as alternativas a escolher quando se tem que usar um Pós-Teste, mais primeiro se deve escolher que tipo de comparação se ajusta ao contexto de investigação. As comparações são agrupadas como se apresenta a seguir:

- Non-parametric multiple groups One vs All (Se realiza a comparação de um método em relação aos demais). O Bonferroni-Dunn (DEMSAR, 2006) é utilizado na grande maioria das pesquisas da área quando se precisa realizar este tipo de comparação.
- Non-parametric multiple groups All vs All (Todos a grupos são comparados entre eles). Para este caso, o Pós-Teste que é usual encontrar nas investigações, é o de *Nemenyi* (NEMENYI, 1963).

Nesta dissertação foi usado o Pós-Teste de *Nemenyi*. Usado para fazer múltiplas comparações entre os algoritmos (Adaptive Methods). Portanto, o desempenho de dois métodos é significativamente diferente, se os ranks das diferenças percentuais entre as médias correspondentes diferem ao menos da Diferença Crítica (CD) definida pela equação B.10

$$CD = q_\alpha \sqrt{\frac{m\,(m+1)}{6n}}\,, \tag{B.10}$$

onde os valores críticos q_α são baseados na estatística do intervalo estudado (Studentized Range Statistic) (RAFTER; ABELL; BRASELTON, 2002) dividida por $\sqrt{2}$.
Por último, a estatística do teste para comparar o i-ésimo conjunto de dados e a j-ésim método usando este método é determinado pela equação B.11

$$z = (R_i - R_j) \Big/ \sqrt{\frac{m\,(m+1)}{6n}} \tag{B.11}$$

O valor calculado de z é utilizado para encontrar a probabilidade pertencente na da tabela de distribuição normal, que é então comparado com um α apropriado (HIDALGO, 2017). Outros testes divergem na forma como eles ajustam o valor de α para compensar comparações múltiplas (DEMSAR, 2006; HIDALGO, 2017). Nota-se que os estatísticos para Pós-Teste são não paramétricos.

B.7 Acurácia Prequential

O anexo na equação B.12 mostra o cômputo da acurácia no tempo determinado t (BAENA-GARCIA et al., 2006; DU; SONG; JIA, 2014), onde a acc_{ex} é 1 se o exemplo atual for corretamente classificado e 0 caso contrário; f é a primeira fixação do tempo de cada cálculo, ou seja, a primeira fixação do tempo para cada mudança de conceito detectada. Evidentemente, na acurácia Prequential, o maior valor é o melhor, sugerindo um melhor desempenho (MINKU; YAO, 2012; HIDALGO, 2017).

$$acc(t) = \begin{cases} acc_{ex}(t), \ \ se \ \ t = f \\ acc(t-1) + \frac{acc_{ex}(t) - acc(t-1)}{t-f+1}, \ \text{em caso contrário.} \end{cases} \tag{B.12}$$

No framework MOA utilizado na investigação, o cálculo desta métrica tem particularidades diferentes após as atualizações efetuadas aos erros de estimação, em conformidade com a variação escolhida para avaliar o desempenho dos modelos de decisão evolutivos.
Na janela básica o cálculo da acurácia prequential é realizado em cada instância utilizando a quantidade de predições corretas e as observações das instâncias passadas de cada critério. A acurácia final é determinada pela média aritmética dos valores das acurácias prequential encontrados nas instâncias, observando unicamente o valor resultante da medição em cada frequência utilizada, que é definida como parâmetro antes de iniciar o processo de avaliação (HIDALGO, 2017).

B.8 Janela básica

No capítulo 6 são mencionadas as variações de prequential, sendo escolhida a janela básica (Basic windows ou Interleaved Test-Then-Train), onde como também já foi exposto todas as instâncias processadas do fluxo de dados são utilizadas no cálculo que confecciona o modelo de decisão, fornecendo a taxa média de acertos do modelo utilizado (HIDALGO, 2017). Esta característica pode ser visualizada como uma desvantagem do método porque dificulta a análise do potencial de classificação real do classificador num instante de tempo, influenciando assim os erros obtidos nas instâncias processadas no fluxo de dados para atualizar o modelo de decisão.
Como foi apresentado em (GAMA et al., 2014; HIDALGO, 2017), o erro Prequential no instante de tempo i é baseado na soma acumulada de uma função de perda L entre as previsões y_k e os valores observados $\hat{y}_k$ (ver figura 10).

Na equação B.13 apresenta-se o cálculo detalhado, onde o limite do erro prequential é o erro de Bayes $\lim_{i\to\infty} P_e(i) = B$ para algoritmos de aprendizagem consistentes. De fato, se a distribuição dos exemplos é estacionaria e os exemplos são independentes, a taxa de erro diminuirá à medida que o número de exemplos de treinamento aumentar.

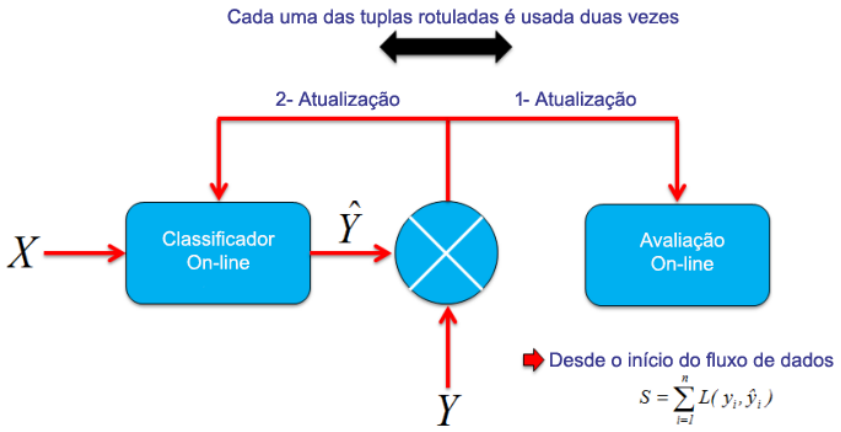


Figura 10 – Avaliação Prequential.

$$P_e(i) = \frac{1}{i}\sum_{k=1}^{i} L(y_k, \hat{y}_k) = \frac{1}{i}\sum_{k=1}^{i} e_k \tag{B.13}$$

O pseudo-código para a regra de atualização na estimativa do erro Prequential da variação janela básica, usando todos os dados é apresentado no algoritmo 9. Como pode-se apreciar nas linhas **1** e **2**, é necessário ter a perda na instância e_i para garantir o estimador do erro P_e no início da aprendizagem. Uma vez que o estimador do erro é inicializado na linha **3**, o algoritmo continua com a parte fundamental que consiste no apresentado nas linhas **4-6** onde é realizada a atualização da estimativa do erro cada vez que uma instância chega.

Algoritmo 9: Atualização da estimativa do error Prequential

1 **Require:** e_i {/∗ Loss at example i ∗/}
2 **Ensure:** Error estimator $P_e(i)$
3 $P_e(0) \leftarrow 0$ {/∗ Initialize the error estimate ∗/}
4 ...
5 $P_e(i) \leftarrow \frac{e_i+(i-1)*P_e(i-1)}{i}$ {/∗ Update the error estimate ∗/}
6 ...