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

José Luis Martínez Pérez

Estratégias para Aprimorar Técnicas Supervisionadas de Classificação para Contextos Semi-Supervisionados

Recife

2025

José Luis Martínez Pérez

Estratégias para Aprimorar Técnicas Supervisionadas de Classificação para Contextos Semi-Supervisionados

Tese de Doutorado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Orientador: Roberto Souto Maior de Barros

Coorientador: Silas Garrido Teixeira de Carvalho Santos

Recife

2025

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Pérez, José Luis Martínez.
Estratégias para aprimorar técnicas supervisionadas de classificação para contextos semi-supervisionados / José Luis Martínez Pérez. - Recife, 2025.
132f.: il.

Tese (Doutorado) - Universidade Federal de Pernambuco, Centro de Informática, Programa de Pós-graduação em Ciência da Computação.
Orientação: Roberto Souto Maior de Barros.
Coorientação: Silas Garrido Teixeira de Carvalho Santos.
Inclui referências e apêndices.

1. Inteligência computacional; 2. Aprendizado semi-supervisionado; 3. Detectores de mudanças de conceito; 4. Autoaprendizado; 5. Comitê de classificadores; 6. Fluxo de dados. I. Barros, Roberto Souto Maior de. II. Santos, Silas Garrido Teixeira de Carvalho. III. Título.

UFPE-Biblioteca Central

José Luis Martínez Pérez

“Estratégias para Aprimorar Técnicas Supervisionadas de Classificação para Contextos Semi-Supervisionados”

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação. Área de Concentração: Inteligência Computacional.

Aprovada em: 25/02/2025.

_____

Orientador: Prof. Dr. Roberto Souto Maior de Barros

BANCA EXAMINADORA

_____

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

_____

Prof. Dr. Cleber Zanchettin
Centro de Informática / UFPE

_____

Prof. Dr. João Roberto Bertini
Faculdade de Tecnologia / UNICAMP

_____

Prof. Dr. Rodolfo Carneiro Cavalcante
Escola Politécnica de Pernambuco/UPE

_____

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

Para minha avó Norma, cuja sabedoria me inspirou a alcançar meus objetivos.

AGRADECIMENTOS

A meu orientador, Professor Roberto Souto Maior de Barros, agradeço por me aceitar como orientando e pelos ensinamentos nestes anos de doutorado.

A meu coorientador, Silas Garrido Teixeira de Carvalho Santos, seu conhecimento foi fundamental para a realização desta tese.

A minha avó Norma, por a senhora ser minha maior fonte de inspiração; a meus pais Luis e Nuvia, que incentivaram a converter a saudade imensa deles em força para não desistir; a minha namorada Ivis, por ensinar-me que com Deus no coração, paciência e dedicação tudo é possível.

Aos professores e funcionários do programa de pós-graduação em ciências da computação do Centro de Informática (CIN-UFPE).

"A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original."

(Oliver Wendell Holmes Sr.)

RESUMO

Os algoritmos de aprendizado de máquina estão se tornando cruciais, e quando expostos a uma quantidade maior e mais relevante de dados de treinamento, tendem a apresentar melhor desempenho. No entanto, a disponibilidade de dados rotulados sem a intervenção de humanos é uma tarefa desafiadora, especialmente no aprendizado em fluxo de dados com mudanças de conceito, em que os dados são gerados rapidamente, em tempo real e com a possibilidade de alterações na distribuição de probabilidade. As mudanças de conceito ocorrem em ambientes de aprendizado supervisionado, semi-supervisionado e não supervisionado. Atualmente, o uso de mecanismos de detecção de mudanças em aprendizado semi-supervisionado é incomum, e a adição desses mecanismos aumenta o custo computacional. Além disso, a classificação em ambientes semi-supervisionados pode levar a problemas relacionados à rotulagem de dados para treinamento. Um erro nesse processo pode impactar negativamente o desempenho do modelo. Esta tese explora os seguintes pontos: 1) o uso de detectores de mudanças de conceito supervisionados em problemas de aprendizado semi-supervisionado; 2) a influência da diversidade nos comitês de classificadores em cenários com mudanças de conceito; 3) introduz uma abordagem de *self-training* (auto-treinamento) para otimizar o aprendizado; e, por fim, 4) detalha as modificações realizadas no *framework Massive Online Analysis* (MOA) para a simulação de cenários semi-supervisionados. Os experimentos realizados utilizaram os classificadores *Hoeffding Tree* (HT) e *Naïve Bayes* (NB), individualmente ou como membros de comitê, sempre combinados com detectores e testados em 84 bases de dados artificiais e 11 reais. Os experimentos foram conduzidos com 15% e 30% de dados rotulados. Os resultados indicam que detectores desenvolvidos para aprendizado supervisionado podem ser utilizados de forma eficaz em ambientes semi-supervisionados. Além disso, os testes com a nova abordagem de *self-training* demonstram que a inclusão de rótulos adicionais melhora significativamente o desempenho dos classificadores. Essas descobertas podem levar a uma mudança de paradigma em pesquisas futuras, uma vez que muitos pesquisadores não consideram os detectores de mudanças de conceito como uma alternativa viável devido à disponibilidade limitada de rótulos na maioria dos fluxos de dados do mundo real.

Palavras-chaves: Aprendizado semi-supervisionado, detectores de mudanças de conceito, *self-training*, comitê de classificadores, fluxo de dados.

ABSTRACT

Machine learning algorithms are becoming crucial, and when exposed to a larger and more relevant amount of training data, they tend to perform better. However, the availability of labeled data without human intervention is a challenging task, especially in data stream learning with concept drifts, where data is generated rapidly, in real-time and with the possibility of changes in the probability distribution. Concept drift occurs in supervised, semi-supervised, and unsupervised learning environments. Currently, the use of drift detectors with base classifiers in semi-supervised learning is uncommon, and the addition of a detection mechanism increases the computational cost. Furthermore, classification in semi-supervised environments can lead to problems related to labeling data to training. An error in this process can negatively impact model performance. This thesis explores and contributes to the following points: 1) the use of supervised concept drift detectors in semi-supervised learning problems; 2) the influence of diversity on classifier ensembles in concept drift scenarios; 3) it introduces a self-training approach to optimize learning; and, finally, 4) it details the modifications made to the *Massive Online Analysis* (MOA) framework to simulation in semi-supervised scenarios. The experiments employed *Hoeffding Tree* (HT) and *Naïve Bayes* (NB) classifiers, either individually or as members of the ensembles, always combined with drift detectors and evaluated on 84 synthetic and 11 real datasets. The experiments were conducted with 15% and 30% labeled data. The results indicate that detectors developed for supervised learning can be effectively used in semi-supervised environments. Additionally, the tests with the new self-training approach demonstrate that the inclusion of additional labels significantly improves classifier performance. These findings may lead to a paradigm shift in future research, as many researchers do not consider concept drift detectors a viable alternative due to the limited availability of labels in most real-world data streams.

Keywords: Semi-supervised learning, concept drift detectors, self-training, classifier ensemble, data stream.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABC	*Artificial Bee Colony*
ADOB	*Adaptable Diversity-based Online Boosting*
ADWIN	*Adaptive Windowing*
AELME	*Advanced ELM Ensemble*
AGNES	*Agglomerative Nesting*
AM	Aprendizagem de Máquina
ASSEMBLE	*Adaptive Semi-Supervised Ensemble*
BDF	*Batch training using distance-based confidence score and fixed confidence threshold*
BOLE	*Boosting-like Online Learning Ensemble*
CD	Diferença Crítica
CW	*Chinese Whispers*
DBSCAN	*Density-Based Spatial Clustering of Applications with Noise*
DCL-LA	*Dynamic Classifier Selection with Local Accuracy*
DDE	*Drift Detection Ensemble*
DDM	*Drift Detection Method*
DIANA	*Divisive Analysis Clustering*
DMDDM-S	*Fast Reaction to Sudden Concept Drift in the Absence of Class Labels*
DSDD	*A drift detection method based on dynamic classifier selection*
DS-MCB	*Dynamic Classifier Selection based on Multiple Classifier Behavior*
ECDD	EWMA *for Concept Drift Detection*
ECHO	*Efficient Semi-Supervised Adaptive Classification and Novel Class Detection over Data Stream*
EDDM	*Early Drift Detection Method*

e-Detector	*A Selective Detector Ensemble for Concept Drift Detection*
EHCD^2	*Ensembles of Heterogeneous Concept Drift Detectors - Experimental Study*
EM	*Expectation Maximization*
FASE	*Fast Adaptive Stacking of Ensembles*
FHDDM	*Fast Hoeffding Drift Detection Method for Evolving Data Stream*
GMM	*Gaussian Mixture Models*
HDBSCAN	*Hierarchical DBSCAN*
HDDM	*Hoeffding-based Drift Detection Method*
HDDM$_A$	*Hoeffding-based Drift Detection Method A-Test*
HT	*Hoeffding Tree*
IA	Inteligência artificial
JRE	*Java Runtime Environment*
KDD	*Knowledge Discovery in Databases*
KNN	*K-nearest neighbors*
LDA	*Latent Dirichlet Allocation*
LevBag	*Leveraging Bagging*
MCL	*Markov Cluster*
MOA	*Massive Online Analysis*
MONA	*Monothetic Clustering*
MOPSO	*Multi-objective particle swarm optimization*
MST	*Minimum Spanning Tree*
NB	*Naïve Bayes*
OABM1	*Online AdaBoost-based M1*
OABM2	*Online AdaBoost-based M2*
OACE$_{BA}$	*Online Adaptive Classifier Ensemble based on bagging*

OACE$_{BO}$	*Online Adaptive Classifier Ensemble based on boosting*
OOD	*Out-of-Distribution*
OPTICS	*Ordering Points To Identify the Clustering Structure*
PAC	*Probably Approximately Correct*
RDDM	*Reactive Drift Detection Method*
SAND	*Semi-Supervised Adaptive Novel Class Detection and Classification over Data Stream*
SDK	Kit de desenvolvimento de software
SemiBoost	*Semi-supervised On-Line Boosting for Robust Tracking*
SSEL	*Semi-supervised Ensemble Learning of Data Streams in the Presence of Concept Drift*
STED	*Statistical Tests Ensemble Drift Detector*
STEPD	*Statistical Test of Equal Proportions*
SVM	*Support Vector Machine*
WEA	*Weight Estimation Algorithm*
WMA	*Weighted Majority Algorithm*
WSTD	*Wilcoxon Rank Sum Test Drift Detector*

SUMÁRIO

1 INTRODUÇÃO

O ramo da inteligência artificial que investiga como programar computadores para que aprendam com os dados (ORALLO; QUINTANA; RAMÍREZ, 2004), melhorando seu desempenho para realizar tarefas automaticamente, é o Aprendizado de Máquina (do inglês, *Machine Learning*). O aprendizado indutivo é utilizado para a criação de modelos baseados em observações anteriores, podendo produzir decisões e resultados confiáveis. O mesmo pode ser dividido em aprendizado supervisionado, não supervisionado e semi-supervisionado. No aprendizado supervisionado, a totalidade dos exemplos contém rótulos (classes), diferente do aprendizado não supervisionado, onde não existem exemplos rotulados. Por sua vez, o aprendizado semi-supervisionado posiciona-se entre os aprendizados anteriores e tem exemplos onde os rótulos são conhecidos e exemplos que não possuem rótulos, sendo este último o foco desta proposta de pesquisa.

As técnicas utilizadas no Aprendizado de Máquina podem ser divididas em dois grandes grupos: descritivas, para o aprendizado não supervisionado; e preditivas, relacionadas 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.

No aprendizado de máquina, a descoberta de conhecimento (*Knowledge Discovery in Databases (KDD)*) em fluxos de dados (*data streams*) representa uma tarefa desafiadora, dado que os fluxos de dados são ambientes que, frequentemente, contêm uma grande quantidade de dados fluindo rapidamente e continuamente. Nesse cenário, os métodos utilizados para análise dos dados de forma online devem ser atualizados constantemente, adaptando-se de forma ágil aos dados (novas instâncias) que podem chegar a uma velocidade muito rápida (DU; SONG; JIA, 2014). Além disso, a distribuição dos dados pode mudar ao longo do tempo, gerando um fenômeno conhecido como mudança de conceito (*Concept Drift*), afetando o rendimento do modelo de aprendizagem (GONÇALVES JR. et al., 2014; PESARANGHADER; VIKTOR, 2016; BRZEZINSKI; STEAFNOWSKI, 2016).

Os sistemas de classificação supervisionada dependem de uma amostra de treinamento suficientemente representativa para o problema que pretende-se resolver. O conjunto de treinamento deve ser preparado com antecedência por um especialista humano, que seleciona um conjunto de classes representativas e os atributos que conseguem reconhecê-las.

Por outro lado, se houver uma mudança de conceito no cenário onde o classificador tem sido treinado, ele deve ser treinado novamente, sendo necessária a intervenção de um especialista para a reconstrução da amostra. Geralmente, esse é um processo complicado e muito custoso e nem sempre é possível obter um conjunto de treinamento suficientemente bom. Na prática, é relativamente comum a obtenção de amostras não rotuladas. Dessa forma, faz-se necessário projetar métodos de aprendizagem que permitam utilizar tanto amostras rotuladas quanto não rotuladas no processo de construção de um conjunto de treinamento, além de levar em consideração as mudanças de conceito.

A classificação semi-supervisionada, por outro lado, é uma extensão da classificação supervisionada, onde o conjunto de treinamento contém informações disponíveis pelos dados rotulados, e também utiliza-se informações presentes nos dados não rotulados. O total da quantidade de dados de treinamento é a soma dos dados rotulados e não rotulados. Neste cenário, é incomum na literatura encontrar métodos de detecção de mudanças de conceito que sejam utilizados em conjunto com um classificador.

Os algoritmos de aprendizagem semi-supervisionada são, em sua maioria, variações de algoritmos de aprendizagem de máquina tradicional. Algoritmos como o *COP-k* (WAGSTAFF et al., 2001) são baseados na proposta do algoritmo não supervisionado *k-means* (MACQUEEN, 1967), com algumas modificações. Outras propostas são *SEEDED-k-means* (BASU; BANERJEE; MOONEY, 2002), *CONSTRAINED-k-means* (BASU; BANERJEE; MOONEY, 2002), $k\text{-}means_{ki}$ (SANCHES, 2003) e REDLLA (LI; WU; HU, 2010) também baseadas em *k-means*.

Outras abordagens utilizam técnicas de *Self-training*, como no caso do algoritmo *CO-Training* (BLUM; MITCHELL, 1998), a partir do qual outras variações foram propostas (GOLDMAN; ZHOU, 2000; MATSUBARA, 2004; ZHOU; GOLDMAN, 2004; HADY; SCHWENKER, 2008; WANG et al., 2014; MONTEIRO; SOARES; BARROS, 2021). Por outro lado, o algoritmo *TriTraining* e sua melhoria *Co-forest* foram propostos pelos autores (ZHOU; LI, 2005) e (LI; ZHOU, 2007) respectivamente. O primeiro faz uso de três classificadores baseados no mesmo algoritmo de aprendizado, mas utilizando diferentes conjuntos de treinamento, sendo que a sua ideia básica é que a maioria ensine à minoria. O segundo algoritmo utiliza a teoria de comitês para adicionar mais de três classificadores e fazer a predição do rótulo final.

Uma abordagem não menos interessante é o uso de comitês de classificadores, que podem ser implementados em combinação com outras abordagens. Um exemplo notável

na literatura é o ASSEMBLE (*Adaptive Semi-Supervised Ensemble*) (BENNETT; DEMIRIZ; MACLIN, 2002) baseado em algumas alterações na implementação do *Boosting*[1] (FREUND, 1995) e usa árvores de decisão como classificador base. A partir do ASSEMBLE surge o SemiBoost (*Semi-supervised On-Line Boosting for Robust Tracking*) (GRABNER; LEISTNER; BISCHOF, 2008) que tem um mecanismo interno que lhe permite lidar com as mudanças de conceito. Outros comitês têm sido propostos para o aprendizado semi-supervisionado na presença de mudanças de conceito, como o algoritmo SSEL (*Semi-supervised Ensemble Learning of Data Streams in the Presence of Concept Drift*) (AHMADI; BEIGY, 2012) e o WEA (*Weight Estimation Algorithm*) (DITZLER; POLIKAR, 2011).

Também existem propostas de algoritmos semi-supervisionados usando aprendizado por reforço e cortes em grafos (COLLINS; SINGER, 1999; BLUM; CHAWLA, 2001; IVANOV; BLUMBERG; PENTLAND, 2001). Além desses, existem algoritmos que utilizam *Support Vector Machine* (SVM) (BOSER; GUYON; VAPNIK, 1992; CORTES; VAPNIK, 1995) com variações (BENNETT; DEMIRIZ, 1999; DEMIRIZ; BENNETT, 2001; NEGRI; SANT'ANNA; DUTRA, 2013). É importante destacar que o primeiro algoritmo semi-supervisionado para fluxo de dados é baseado em SVM e foi apresentado em (KLINKENBERG, 2001), sendo o uso de janelas ajustáveis uma de suas principais características (AHMADI; BEIGY, 2012). Posteriormente, foi introduzido RK-TS^3VM (ZHANG; ZHU; GUO, 2009), proposta que relaciona *K-means* com SVM, alcançando resultados satisfatórios de acordo com os autores e sendo superior a propostas como ReaSC (WOOLAM; MASUD; KHAN, 2009).

O aprendizado semi-supervisionado envolve outra família de algoritmos que consistem em variações de *Expectation Maximization* (EM) (DEMPSTER; LAIRD; RUBIN, 1977), como o algoritmo apresentado em (BRUCE, 2001). Nele, os passos *Expectation* e *Maximization* são incrementados com uma variável chamada pseudo-contador (GUTIÉRREZ, 2010), o que permite estimar o rótulo de dados não rotulados. Outra abordagem proposta em (NIGAM et al., 2000) apresenta uma extensão do algoritmo EM com *Naïve Bayes* (MITCHELL, 1997) e utiliza máxima probabilidade ou estimação máxima posterior de problemas com dados incompletos. Nesse caso, os dados não rotulados são tratados como se fossem dados incompletos. A indução do classificador pelo algoritmo *Naïve Bayes* é realizada só com os dados rotulados. Como EM também é um algoritmo bayesiano, no passo *Expectation*

[1] Boosting is a well-known and general method for improving the accuracy of other algorithms ("weak" learners): it trains several classifiers using different distributions over the training data and combines them in an ensemble.

é inserido o classificador *Naïve Bayes*, calculado anteriormente, para estimar algumas probabilidades. Já no passo *Maximization*, os cálculos são realizados como no algoritmo EM básico (SANCHES, 2003; MATSUBARA, 2004).

As mudanças de conceito estão presentes no aprendizado supervisionado, não supervisionado e semi-supervisionado, sendo abordadas por diferentes pontos de vista, a exemplo de: estatística, processamento de sinais e aprendizagem automática, cada uma trazendo soluções com diferentes graus de garantias matemáticas (FRÍAS-BLANCO, 2014). As estratégias para detectar mudanças de conceito são inúmeras, podendo ser usadas para incorporar a habilidade de aprender em diferentes algoritmos, mesmo quando os conceitos mudam ao longo do tempo.

Os detectores de mudanças de conceito podem ser categorizados de diversas maneiras. Uma das possíveis categorizações divide-se em duas estratégias baseadas na adaptação dos classificadores às mudanças de conceito, como mostra a taxonomia a seguir (GAMA et al., 2004):

1. Adaptação implícita: o aprendizado é adaptado em intervalos regulares de tempo sem considerar se aconteceu uma mudança. Por exemplo, o classificador poderia ser treinado novamente a cada semana ou mês com dados mais recentes.

2. Adaptação explícita: primeiro a mudança é detectada e depois executam algum processo de aprendizagem para adaptar o modelo, e incluem as técnicas de seleção e ponderação de instâncias (ORTIZ-DÍAZ, 2014). Neste caso, só acontece novo treinamento após uma mudança de conceito ser detectada.

A maioria das propostas que fazem parte do estado da arte dos detectores de mudanças de conceito foi construída para cenários de aprendizado supervisionado, onde todas as instâncias são rotuladas. Como principais exemplos podemos citar: *Drift Detection Method* (DDM) (GAMA et al., 2004), *Early Drift Detection Method* (EDDM) (BAENA-GARCIA et al., 2006), *Adaptive Windowing* (ADWIN) (BIFET; GAVALDÀ, 2007), *Statistical Test of Equal Proportions* (STEPD) (NISHIDA; YAMAUCHI, 2007), EWMA *for Concept Drift Detection* (ECDD) (ROSS et al., 2012), *Hoeffding-based Drift Detection Method* (HDDM) (FRÍAS-BLANCO et al., 2015), *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; HI-

DALGO; CABRAL, 2018), etc. Além dos detectores únicos, existem também os comitês de detectores e métodos estatísticos, como, por exemplo, *A Selective Detector Ensemble for Concept Drift Detection* (e-Detector) (DU et al., 2014), *Drift Detection Ensemble* (DDE) (MACIEL; SANTOS; BARROS, 2015), *Statistical Tests Ensemble Drift Detector* (STED) (PÉREZ; BARROS; SANTOS, 2020) e *Ensembles of Heterogeneous Concept Drift Detectors - Experimental Study* (EHCD^2) (WOŹNIAK et al., 2016).

Como mencionado anteriormente, na atualidade é difícil encontrar modelos de detecção de mudanças de conceito para cenários de aprendizado semi-supervisionado (TRAT; BENDER; OVTCHAROVA, 2022). Entretanto, é válido ressaltar que nos últimos anos o interesse dos pesquisadores na área vem crescendo. O reduzido estado da arte conta com métodos como: *Semi-Supervised Adaptive Novel Class Detection and Classification over Data Stream* (SAND) (HAQUE; KHAN; BARON, 2016), *Efficient Semi-Supervised Adaptive Classification and Novel Class Detection over Data Stream* (ECHO) (HAQUE et al., 2016), *Fast Reaction to Sudden Concept Drift in the Absence of Class Labels* (DMDDM-S) (MAHDI et al., 2020), e *A drift detection method based on dynamic classifier selection* (DSDD) (PINAGÉ; SANTOS; GAMA, 2020), ao qual foi atribuída a sigla no artigo intitulado *"An overview of unsupervised drift detection methods"* (GEMAQUE et al., 2020). Outro método que podemos adicionar ao grupo acima mencionado é o *Batch training using distance-based confidence score and fixed confidence threshold* (BDF) (NGUYEN; GOMES; BIFET, 2019). Embora não seja um método de detecção de mudanças de conceitos, ele é mencionado pela facilidade com que pode ser combinado com todos os métodos de detecção em cenários supervisionados anteriormente mencionados, gerando-se assim novos métodos para cenários semi-supervisionados. Um aspecto que se destaca na revisão da literatura é o frequente uso de comitês de classificadores no contexto do aprendizado semi-supervisionado. Observa-se, contudo, que na maioria dos casos os comitês são compostos por métodos de classificação do mesmo tipo, sendo a diversidade introduzida por meio de alterações no acesso às instâncias ou nos parâmetros de configuração. Entretanto, o impacto da diversidade nos comitês introduzida pela combinação de vários tipos de métodos de classificação, ainda não é amplamente investigado, apesar de evidências em aprendizado supervisionado já demonstrarem que a diversidade pode favorecer o desempenho dos modelos.

Os métodos de classificação semi-supervisionados, na sua maioria, são algoritmos de alto consumo de memória e tempo (LE et al., 2016; SHARMA; JONES, 2023; SHI et al., 2024), especialmente aqueles baseados em estrutura típica de *frameworks* que integram

dois módulos principais: um para o tratamento de dados rotulados e outro para dados não rotulados. Esses *frameworks* frequentemente possuem alta complexidade computacional (ZHU, 2005; TANHA, 2013; DUARTE; BERTON, 2023; SHI et al., 2024; MVULA et al., 2024; GARRIDO-LABRADOR et al., 2024), devido à necessidade de incorporar e combinar algoritmos de classificação e agrupamento. Nesse contexto, a adição de mecanismos de detecção torna-se inviável devido às limitações de recursos computacionais. Além disso, a tarefa de classificação no aprendizado semi-supervisionado pode apresentar problemas associados à rotulagem e à inserção de instâncias no treinamento. Um erro no processo pode influenciar negativamente o desempenho do modelo subsequente. Por exemplo, em (LU, 2009) é defendido que, mesmo satisfazendo a maioria das condições necessárias, o aprendizado considerando o uso de agrupamento pode ser perigoso, levando a um desempenho de previsão muito pior do que simplesmente ignorar os dados não rotulados e fazer o aprendizado supervisionado. Macário (MACÁRIO, 2009) também indica o uso da parte semi-supervisionada dos métodos apenas quando não há dados rotulados suficientes, mas não recomenda nenhuma metodologia ou parâmetros empiricamente testados para definir este valor limite para cada situação em que o modelo deve ser criado.

Os desafios apresentados até o momento na pesquisa, relacionados aos métodos semi-supervisionados e à necessidade real de dispor de algoritmos de alta performance para esses cenários, motivaram a formulação do seguinte problema de investigação: Combinar métodos de detecção de mudanças de conceito, diversidade nos comitês de classificadores e *self-training* para lidar com a escassez de rótulos e as mudanças na distribuição dos dados, melhora o desempenho dos métodos de classificação supervisionados em cenários de aprendizado semi-supervisionado?

1.1 OBJETIVOS

O objetivo geral desta investigação é explorar a viabilidade do uso de detectores de mudanças de conceito, de técnicas para promover a diversidade em comitês de classificadores e da abordagem de *self-training*, bem como de sua combinação, para aumentar a eficácia e a adaptabilidade de métodos de classificação supervisionada em ambientes semi-supervisionados.

Portanto, com o intuito de alcançar o objetivo geral estabelecido, os seguintes objetivos específicos são propostos:

- Analisar o desempenho com base na métrica acurácia dos algoritmos clássicos desenvolvidos sobre a plataforma MOA para aprendizagem supervisionada e semi-supervisionada, em presença de mudanças de conceito;
- Demonstrar o impacto positivo da inclusão de diversidade em comitês de classificadores em cenários de aprendizado supervisionado e semi-supervisionado;
- Apresentar uma nova abordagem de *self-training*, que contribui com o aumento dos dados rotulados para treinamento e, consequentemente, que favorece os métodos de classificação e detecção de mudanças de conceitos.

Os meios para alcançar os objetivos são listados a seguir:

- Analisar em detalhe a literatura especializada e estabelecer o estado da arte das investigações relacionadas à pesquisa;
- Modificar o *framework* MOA, adicionando funcionalidades para a construção de ambientes de teste para o aprendizado semi-supervisionado, uma atividade trabalhosa mas que era necessária;
- Criar *script* de geração de bases de dados artificiais semi-supervisionadas utilizando os geradores disponíveis no *framework* MOA, e de bases de dados reais selecionadas do *UCI Machine Learning Repository*.

Na tese, com base no problema da investigação e alinhado com o objetivo geral traçado, foram definidas as hipóteses (ideias a serem investigadas) descritas a seguir:

1. A utilização de métodos de detecção de mudanças de conceito permite a adaptação dos métodos de classificação às alterações na distribuição dos dados, melhorando o desempenho em cenários de aprendizado semi-supervisionado.
2. Promover diversidade nos comitês de classificadores com a combinação de vários tipos de métodos resulta em maior robustez e capacidade de generalização, contribuindo para um desempenho superior em cenários de escassez de rótulos e presença de mudanças de conceitos.
3. O uso de *self-training* para pseudo-rotular dados não rotulados aumenta a quantidade de informações disponíveis para o treinamento, mitigando os efeitos da escassez de rótulos e melhorando a acurácia dos métodos de classificação.

4. O desempenho dos métodos de classificação supervisionados em cenários semi-supervisionados pode ser significativamente melhorado pela integração das três abordagens (detecção de mudanças de conceito, uso de diversidade nos comitês de classificadores, e *self-training*), em termos de métricas como acurácia.

1.2 CONTRIBUIÇÕES

No decorrer da tese, para alcançar o objetivo geral e responder às hipóteses formuladas, foi realizada uma análise do estado da arte e foram estabelecidos conceitos teóricos e empíricos relacionados a esta investigação. Mais especificamente, foram definidas as desvantagens e necessidades das abordagens existentes para a detecção de mudanças de conceitos em cenários de aprendizado semi-supervisionado. Essa definição contou com um estudo aprofundado de três estratégias (introduzir detectores de mudanças de conceito, aplicar técnica para promover a diversidade nos comitês de classificadores e utilizar a abordagem de *self-training*) e de métodos a serem utilizados para confrontar possíveis deficiências em cenários de aprendizado semi-supervisionado. As abordagens propostas são validadas experimentalmente utilizando a métrica acurácia e levando em consideração um conjunto abrangente de dados.

Para facilitar a compreensão das contribuições desta tese, mencionadas no parágrafo anterior, apresenta-se, a seguir, uma lista detalhada das principais:

1. Fornece a implementação de novas funcionalidades no *framework* MOA, incluindo novos módulos que possibilitam a geração de cenários semi-supervisionados para experimentação, gerando a ferramenta *MOAManagerSS*.

2. Demonstra empiricamente que os métodos de detecção de mudanças de conceito, originalmente desenvolvidos para ambientes de aprendizado supervisionado, podem ser aplicados de maneira satisfatória em cenários semi-supervisionados;

3. Demonstra o impacto positivo da inclusão de diversidade (variando o tipo e a quantidade de classificadores base) nos comitês de classificadores. A pesquisa estabelece parâmetros para a quantidade de classificadores NB e HT, bem como o percentual de classificadores necessários para a votação, visando formar um comitê de desempenho geral satisfatório em cenários supervisionados e semi-supervisionados;

4. Propõe uma nova abordagem de *self-training* a ser incorporada em métodos de classificação, com o objetivo de aumentar o número de dados de treinamento e, consequentemente, fornecer mais informações ao detector de mudanças de conceito;

5. Demonstra que a combinação da diversidade nos comitês de classificadores com a nova proposta de *self-training* pode resultar em métodos de desempenho superior em ambientes semi-supervisionados, superando o uso isolado dessas técnicas;

É importante ressaltar que os resultados dessa tese incluem informações publicadas nos artigos listados a seguir:

1. PÉREZ, J. L. M.; BARROS, R. S. M.; SANTOS, S. G. T. C. **Experimenting with Supervised Drift Detectors in Semi-supervised Learning**. In: 2023 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2023. p. 730–735.

2. PÉREZ, J. L. M.; MARIÑO, L. M. P.; BARROS, R. S. M. **Improving Diversity in Concept Drift Ensembles**. In: 2021 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2021. p. 1–8.

3. PÉREZ, J. L. M.; BARROS, R. S. M.; SANTOS, S. G. T. C. **Enhancing Semi-Supervised Learning with Concept Drift Detection and Self-Training: A Study on Classifier Diversity and Performance**. In: IEEE Access, 2025. DOI *https://doi.org/10.1109/ACCESS.2025.3538710*.

1.3 ORGANIZAÇÃO DO TRABALHO

O restante deste documento está organizado em sete capítulos, descritos a seguir:

- O Capítulo 2 apresenta os conceitos associados às áreas de pesquisa que sustentam a presente tese. O capítulo faz uma breve revisão dos principais conceitos de aprendizagem de máquina em fluxos de dados não estacionários com a possibilidade de mudanças de conceito, com foco no aprendizado semi-supervisionado.

- O Capítulo 3 apresenta uma recompilação de ferramentas utilizadas no aprendizado de máquina. Além disso, é exposta a implementação de novas funcionalidades no *framework* MOA para geração de bases de dados e a aprendizagem em cenários semi-supervisionados.

- O Capítulo 4 descreve a configuração dos experimentos realizados nos cenários de aprendizado supervisionado e semi-supervisionado. Também é realizada a apresentação das bases de dados reais e artificiais utilizadas na experimentação, assim como, dos critérios de avaliação.

- O Capítulo 5 realiza experimentos com o objetivo de demonstrar que os métodos de detecção de mudanças de conceito, originalmente criados para cenários de aprendizado supervisionado, podem ser utilizados de forma satisfatória em cenários semi-supervisionados.

- O Capítulo 6 apresenta os experimentos realizados a fim de demonstrar o impacto positivo da inclusão de diversidade nos comitês de classificadores. Também são fornecidos parâmetros visando construir comitês com desempenho satisfatório em cenários supervisionados e semi-supervisionados.

- O Capítulo 7 propõe uma metodologia para treinar algoritmos de aprendizado supervisionado, adaptando-os para cenários semi-supervisionados. Essa abordagem potencializa o uso de detectores de mudanças de conceito para melhorar a performance. A metodologia é apresentada através de uma descrição geral e pseudo-código correspondente. Para concluir, apresenta os experimentos realizados, a fim de comparar alguns dos principais métodos da atualidade e os métodos resultantes da implementação da metodologia proposta. Os algoritmos são comparados através de suas acurácias.

- Finalmente, o Capítulo 8 apresenta as conclusões desse trabalho de doutorado. Esta última parte contém o detalhamento das considerações finais a respeito desta pesquisa, além de descrever alguns possíveis trabalhos futuros.

2 REVISÃO DA LITERATURA

O presente capítulo apresenta um levantamento da literatura, bem como os métodos mais utilizados na área de pesquisa abordada. Além disso, destaca-se a dificuldade de encontrar métodos especificamente desenvolvidos para lidar com mudanças de conceito em cenários do aprendizado semi-supervisionado.

2.1 CONTEXTO

A Inteligência artificial (IA) é formada por várias subáreas e, nos últimos anos, tem sido aplicada em diversos setores como política, economia, saúde, esporte, educação, etc. A utilização da subárea Aprendizagem de Máquina (AM) para a criação de modelos analíticos com o objetivo de obter uma melhor performance é uma tarefa desafiadora. Na AM, técnicas preditivas ou descritivas (FRÍAS-BLANCO, 2014) podem ser utilizadas, permitindo executar tarefas de classificação ou regressão para o aprendizado supervisionado e tarefas de agrupamento no aprendizado não supervisionado (FACELI et al., 2011).

Na atualidade, podemos afirmar que quatro são os tipos de aprendizagem mais usados:

1. O aprendizado supervisionado, no qual os dados são exibidos como pares ordenados (entrada, saída pretendida). Entende-se por entrada os dados das variáveis de entrada do algoritmo para uma situação específica (GOLDSSCHMIDT; PASSOS, 2005). A saída pretendida é o valor aguardado que o algoritmo possa fornecer, sempre que receber os dados especificados na entrada;

2. O aprendizado não supervisionado, no qual a saída pretendida é inexistente e o objetivo dos métodos é estabelecer algum padrão/relação a partir dos dados;

3. O aprendizado por reforço, que é baseado na avaliação de um reforço ou recompensa para o conjunto de ações realizadas (ORTIZ-DÍAZ, 2014) e é principalmente utilizado na subárea da robótica; e

4. O aprendizado semi-supervisionado, que é uma subárea de pesquisa relativamente nova em AM, com potencial de reduzir a necessidade de dados rotulados quando somente um pequeno conjunto de exemplos rotulados está disponível (PÉREZ, 2018). Este último é o foco central desta pesquisa.

2.2 FLUXOS DE DADOS

Segundo (SANTOS; BARROS; GONÇALVES JR., 2015), na Aprendizagem de Máquina, um fluxo contínuo de dados é uma sequência de dados muito longa (possivelmente com tamanho ilimitado) que flui em alta velocidade e não se tem controle da ordem que os exemplos chegam. A afirmação anterior pode ser formalizada considerando que as instâncias (x_i, y_i) pertencem a um conjunto de dados (exemplos) que chegam ao longo do tempo, e cada x_i tem a forma de vetor $(x_{i1}, x_{i2}, \ldots, x_{in}) \in \mathbb{R}^n$, onde cada membro x_{in} é conhecido como atributo e contém um valor discreto, com n sendo o número de atributos (dimensão). O valor de y_i é o rótulo (classe) correspondente ao x_i, obtido de um conjunto limitado Y de possibilidades (BIFET et al., 2009; SUN et al., 2016; KRAWCZYK et al., 2017). Logo, podemos apresentar um fluxo como $M = \{(x_1, y_1), (x_2, y_2), \ldots\}$ para o aprendizado supervisionado e $M = \{(x_1, y_1), (x_2, ?), \ldots\}$ para o aprendizado semi-supervisionado, sendo o último caracterizado pelo fato de possuir instâncias sem rótulo, da forma $(x_i, ?)$ (MITCHELL, 1999).

As restrições que podem ser apresentadas pelo fluxo contínuo de dados são muitas, tornando seu processamento uma tarefa desafiadora. Uma das principais é ter que lidar com a quantidade ilimitada de instâncias que serão processadas (BIFET, 2009). Essa restrição inviabiliza o armazenamento contínuo na memória, obrigando o modelo a descartar a maior parte dos dados (PÉREZ, 2018). Além disso, dados chegando em altas velocidades também são um problema, uma vez que isso irá limitar o tempo de processamento (DU; SONG; JIA, 2014). Por fim, mas não menos relevante, é a restrição determinada pela possibilidade da variação no tempo da distribuição de probabilidade dos dados (RUTKOWSKI et al., 2015). Essa variação pode provocar um fenômeno conhecido como mudança de conceito (*concept drift*) (GAMA et al., 2004).

2.3 MUDANÇAS DE CONCEITO

Nos ambientes de fluxos contínuos de dados, quando a relação entre os atributos e rótulos das instâncias muda com o tempo, é considerado que acontece o fenômeno conhecido como mudanças de conceito (READ et al., 2012; SANTOS; BARROS; GONÇALVES JR., 2015), onde o modelo aprendido anteriormente não representa mais o cenário atual. Como resultado, o modelo erra mais, devido a estar treinado com um conceito antigo.

As mudanças são categorizadas na bibliografia de duas formas. A primeira, conhecida como mudança real, tem relação com a mudança na distribuição das classes que pode ser diferente em determinados intervalos de tempo (DELANY et al., 2005; ŽLIOBAITĖ, 2010; GAMA et al., 2014). Já a segunda é principalmente identificada quando só a distribuição dos atributos muda. Ela é conhecida na literatura como mudança virtual (KHAMASSI et al., 2015; KRAWCZYK et al., 2017; PÉREZ; BARROS; SANTOS, 2020).

Devido à degradação de desempenho do modelo com respeito às instâncias atuais, faz-se necessário atualizá-lo constantemente. Autores como Woźniak et al. (2016), Pérez, Barros e Santos (2020) consideram a mudança de conceito real mais relevante, principalmente pelo impacto na tarefa de classificação.

Na bibliografia, outra taxonomia para categorizar as mudanças de conceito em fluxos de dados diz respeito à velocidade com que acontecem. Essas mudanças podem ser divididas em abruptas, quando a distribuição é modificada num único intervalo de tempo, ou graduais, onde as modificações nos conceitos são mais sutis e necessitam de número elevado de instâncias para acontecerem (MINKU; WHITE; YAO, 2010). Adicionalmente, na mudança gradual, é comum que a velocidade seja dividida em duas subcategorias: moderada e lenta (STANLEY, 2003). Outra categoria que pode ser encontrada na literatura são as mudanças recorrentes, definidas como uma mudança que acontece temporalmente e que retorna ao seu estado normal após um intervalo de tempo.

As mudanças de conceito podem impactar consideravelmente no desempenho do modelo de aprendizagem, tornando-o inválido. Consequentemente, é recomendável que os algoritmos de aprendizagem incremental incorporem mecanismos adicionais para se adaptarem às mudanças de conceito (PÉREZ, 2018). O algoritmo de aprendizagem deve ser capaz de manter-se atualizado com respeito aos possíveis padrões transitórios subjacentes no fluxo contínuo de dados (WANG et al., 2003). Portanto, o modelo de aprendizagem deve ser constantemente atualizado, aprendendo com os dados mais atuais e esquecendo experiências que representam conceitos passados (FRÍAS-BLANCO, 2014).

A abordagem das mudanças de conceito nos ambientes de fluxos contínuos de dados é um problema comum, que tem levado os pesquisadores a criar e usar diferentes metodologias para o tratamento das mesmas (GONÇALVES JR.; BARROS, 2013; MACIEL; SANTOS; BARROS, 2015; PÉREZ, 2018; PÉREZ; BARROS; SANTOS, 2020), etc. Como exemplos de metodologias, é possível citar a adaptação de classificadores originalmente propostos para serem usados no modo lote (*batch*), e a detecção de mudanças de conceito e criação de

um novo classificador a fim de representar o novo contexto. Também são muito usadas a combinação de classificadores, bem como de detectores.

Os algoritmos para o tratamento das mudanças de conceito devem adaptar-se rapidamente às mudanças, e ser robustos na diferenciação entre uma verdadeira mudança de conceito e ruído[1]. Alguns algoritmos podem ser muito suscetíveis ao ruído, realizando uma interpretação errônea do mesmo como uma mudança de conceito. Já outros podem ser muito robustos em termos de falsos positivos, mas serem mais lentos na detecção (PÉREZ, 2018).

2.4 CLASSIFICAÇÃO E AGRUPAMENTO

As tarefas de classificação e agrupamento (ou clusterização, do inglês *clustering*) são parte da mineração de dados muito usadas na atualidade. A primeira destas tarefas na aprendizagem de máquina está diretamente relacionada ao aprendizado supervisionado, assim como o agrupamento ao aprendizado não supervisionado. Esta seção realiza uma análise detalhada das características vinculadas a ambas as tarefas. Também serão apresentados os principais métodos que fazem parte do estado da arte.

Identificar a qual categoria pertence uma determinada amostra do problema é o objetivo da classificação. Geralmente, na bibliografia, é assumida a existência de uma função objetivo $f(\vec{x}_i) = y_i$. Assim, a tarefa de aprendizagem é obter um modelo $\hat{f}$ que aproxime f, de modo que $\hat{f}$ maximize a precisão na previsão (FERRER-TROYANO; AGUILAR-RUIZ; RIQUELME-SANTOS, 2005; FRÍAS-BLANCO et al., 2016; PÉREZ, 2018).

As técnicas de classificação mais usadas são:

- Algoritmo de *Naïve Bayes*: classificador que usa as probabilidades de ocorrência de cada classe para cada valor de atributo, supondo que as variáveis são independentes;
- Árvores de Decisão: os nós internos representam uma característica de entrada, e os nós folhas são rotulados com a classe;
- Máquina de Vetores de Suporte (SVM): o objetivo principal é encontrar o hiperplano ótimo que separa os dados em diferentes classes com a maior margem possível.

[1] No contexto de aprendizado de máquina, ruído refere-se a qualquer informação irrelevante, incorreta ou imprevisível presente no conjunto de dados que obscurece a verdadeira relação entre os atributos e os rótulos de classe (ANGLUIN; LAIRD, 1988; HICKEY, 1996; ZHU; WU, 2004; GUPTA; GUPTA, 2019).

- k-Vizinhos Mais Próximos (KNN): os dados são classificados com base nas classes dos vizinhos mais próximos. Sensível à escala dos dados e ao número de vizinhos escolhidos.

- Comitê de classificadores: múltiplos classificadores-base são combinados para melhorar o desempenho. Inclui métodos como *Bagging*, *Boosting* e *Stacking*.

- Redes Neurais: usa redes de neurônios artificiais interconectados - os perceptrons, onde os pesos das camadas ocultas são ajustados.

- Modelos de Aprendizado Profundo: baseia-se em redes neurais artificiais profundas, contendo várias camadas de nós interconectados, cada uma delas baseada na camada anterior, refinando e otimizando a previsão ou a classificação.

O processo de classificação geralmente é dividido nas fases conhecidas como treinamento e teste. Na fase de treinamento, o modelo é criado a partir da análise de uma base de dados. Na segunda fase, a de teste, o objetivo é estimar a precisão do modelo criado na fase de treinamento. Muitas pesquisas incluem outra fase intermediária entre as duas anteriores, que é conhecida como fase de validação, onde o modelo é submetido a provas para ajustar parâmetros e para detectar *overfitting* — o caso em que o modelo aprende tão bem os dados de treino que perde a capacidade de generalização.

O agrupamento (ou clusterização) é uma técnica que permite segmentar automaticamente um conjunto de dados em análise, em grupos de acordo com métricas de similaridade ou de distância. Várias são as fórmulas usadas para obter métricas de similaridade, dentre elas podemos destacar as distâncias Euclidiana, Manhattan, Minkowski, a correlação de Pearson e o método do cosseno.

Os métodos de agrupamento podem ser classificados como (CRUZ, 2015):

- Métodos Particionais: geram k grupos (partições de dados ou aglomerados) a partir de um conjunto de n objetos onde $k < n$, cada k_{Grupo} deve conter ao menos um objeto, e cada objeto deve fazer parte de um único k_{Grupo}. Os métodos particionais utilizam técnicas para reposicionar iterativamente os objetos entre as k partições iniciais baseando-se em parâmetros que medem a qualidade dos grupos. Exemplos de métodos particionais podemos mencionar o *k-means* (MACQUEEN, 1967) e o *K-medoids* (SOARES et al., 2019);

- Métodos de Densidade: são adequados para entender distribuição de dados e detectar grupos de formato complexo e irregular. Os grupos são criados tendo em consideração que um objeto pertence a um aglomerado (região densa), se numa vizinhança de raio (α) pequeno existe ao menos M números de objetos informados pelo usuário. Estes métodos têm robustez frente aos ruídos e *outliers* identificados como regiões de baixa densidade. Entre os algoritmos de agrupamento baseado em densidade temos o DBSCAN (ESTER et al., 1996), OPTICS (ANKERST et al., 1999) e HDBSCAN (CAMPELLO; MOULAVI; SANDER, 2013);

- Métodos hierárquico: tais métodos requerem uma matriz de similaridades entre agrupamentos. No agrupamento hierárquico, os dados são particionados criando uma representação hierárquica (árvore) conhecida como dendrograma. É também comum encontrar na literatura os métodos hierárquicos divididos em dois grupos de algoritmos: os aglomerativos (mais usados nas pesquisas) e os divisórios. Três métodos hierárquico amplamente conhecidos são o AGNES, DIANA e MONA, pode-se encontrar todos apresentados formalmente em (KAUFMAN; ROUSSEEUW, 1990);

- Métodos baseado Grafo: representam os dados como um grafo de proximidade onde cada nó é um ponto de dado, e as arestas representam similaridades ou distâncias. O agrupamento é feito dividindo o grafo em subgrafos (grupos) com conexões fortes internas e fracas externas. Exemplos de métodos baseado Grafo podemos mencionar o MST (ZAHN, 1971), CW (BIEMANN, 2006) e MCL (DONGEN, 2008);

- Métodos Probabilísticos: esta abordagem pressupõe que os dados foram gerados por um modelo probabilístico desconhecido, e o objetivo é descobrir esse modelo. Na abordagem cada grupo é visto como uma distribuição de probabilidade e procura-se uma distribuição bem definida a cada dimensão. As médias do agrupamento são as médias da distribuição gaussiana em cada dimensão. Exemplos dos métodos probabilísticos temos o GMM (DEMPSTER; LAIRD; RUBIN, 1977) e LDA (BLEI; NG; JORDAN, 2003), entre os mais conhecidos;

- Métodos de Otimização e Meta-Heurísticas: formulam o problema de agrupamento como uma tarefa de otimização. Esses métodos utilizam estratégias heurísticas ou meta-heurísticas, pelo geral inspiradas em fenômenos naturais, sociais ou físicos para encontrar soluções boas (não necessariamente ótimas). Alguns exemplos de

métodos que usam técnicas de otimização e meta-heurísticas são: *GA-clustering* (MAULIK; BANDYOPADHYAY, 2000), *KGA-clustering* (BANDYOPADHYAY; MAULIK, 2002), MOPSO (BOQIANG; CHUANWEN, 2009) e ABC (KARABOGA; OZTURK, 2011).

A abordagem de agrupamentos pode ser utilizada em combinação com métodos de classificação para cenários supervisionados (PINAGÉ; SANTOS; GAMA, 2020), principalmente para abordar problemas do aprendizado semi-supervisionado. Nas seções 2.4.1 e 2.4.2 serão apresentados os classificadores e algoritmos de agrupamentos mais usados na literatura.

Embora os comitês de classificadores sejam técnicas de classificação, sua discussão será apresentada na Seção 2.6. A criação de uma seção independente justifica-se pela versatilidade dos comitês, que permitem a integração com diversas outras abordagens. Devido a essa flexibilidade, a distinção clara entre os comitês e outras técnicas pode ser difícil em algumas situações, o que torna mais apropriado abordá-los em uma seção posterior.

2.4.1 Classificadores

O universo dos classificadores tem propostas variadas e eficientes, entre as quais podemos citar *Hoeffding Tree* (HT), *Naïve Bayes* (NB), *K-nearest neighbors* (KNN) (EVELYN; JOSEPH, 1951), *Support Vector Machine* (SVM) (CORTES; VAPNIK, 1995), etc. Nos experimentos, decidimos utilizar HT e NB como classificadores base (modelos de aprendizado ou preditores) dentre outras propostas acima mencionadas, por serem classificadores rápidos, disponíveis e amplamente usados em cenários de fluxo contínuo de dados.

2.4.1.1 NB

O baixo custo computacional e a simplicidade de implementação tornam o algoritmo de classificação NB (JOHN; LANGLEY, 1995) um método comumente utilizado por pesquisadores no aprendizado de máquina (OGURI, 2006), reportando bom desempenho em várias tarefas de classificação. O classificador NB utiliza o Teorema de Bayes e assume que a presença de um determinado atributo (característica) é condicionalmente independente dos demais atributos, dada a classe (GAMALLO; GARCIA; FERNÁNDEZ-LANZA, 2013).

Na Equação 2.1 aplicada no algoritmo glsnb podemos perceber como, para cada instância não rotulada, o NB prediz uma classe $\hat{Y}$, com base na probabilidade a posteriori da classe y_j, dada a instância x_i (PÉREZ, 2018).

$$\hat{Y} = \arg\max_{y_j \in Y} P(y_j) \prod_{i=1}^{n} P(x_i|y_j) \tag{2.1}$$

onde $P(y_j)$ é a probabilidade a priori da classe y_j, e $P(x_i \mid y_j)$ é a probabilidade condicional do atributo x_i dado a classe y_j (MARIÑO, 2019).

2.4.1.2 HT

O classificador HT (HULTEN; SPENCER; DOMINGOS, 2001) constrói uma árvore de decisão indutiva para aprender a partir de um fluxo de dados. Este algoritmo incremental é baseado no teorema matemático do limite de *Hoeffding* (HOEFFDING, 1963), o qual quantifica o número de exemplos (instâncias) para obter um certo nível de confiança. Assumindo ter n observações independentes da variável aleatória r cujo intervalo é R, o limite de *Hoeffding* afirma que, com probabilidade $1 - \delta$, a média real da variável é pelo menos $\bar{r} - \varepsilon$, onde $\bar{r}$ é o valor médio calculado a partir de n observações independentes (HIDALGO; MACIEL; BARROS, 2019) e ε é obtido pela Equação 2.2.

$$\varepsilon = \sqrt{\frac{R^2(\ln 1/\delta)}{2n}} \tag{2.2}$$

O limite de *Hoeffding* é atraente pela capacidade de obter resultados consistentes, independentemente da distribuição de probabilidade que gera os exemplos. Contudo, o número de exemplos necessários para alcançar certos valores de δ e ϵ é diferente entre as distribuições de probabilidade. A heurística utilizada para escolher os atributos de teste é o ganho de informação (PÉREZ, 2018).

2.4.1.3 KNN

O algoritmo *K-nearest neighbors* (KNN) (EVELYN; JOSEPH, 1951) é uma técnica de aprendizado supervisionado utilizada em tarefas de classificação e regressão, operando sob a suposição de que pontos de dados semelhantes estão em proximidade no espaço de características (COVER; HART, 1967). Como um aprendiz preguiçoso baseado em instâncias,

ele memoriza os dados de treinamento e realiza a classificação apenas no momento da predição, identificando os k vizinhos mais próximos utilizando uma métrica de distância (Euclidiana, Hamming, Manhattan, Mahalanobis, etc.) (ATALLAH; BADAWY; EL-SAYED, 2019), sendo a distância Euclidiana a mais comum. A classificação baseia-se em uma votação majoritária entre os vizinhos, onde a classe mais frequente é atribuída ao ponto de consulta. O hiperparâmetro k desempenha um papel crucial, com valores baixos podendo levar ao *overfitting* (sobreajuste) e valores altos resultando em subajuste, exigindo, portanto, uma seleção cuidadosa, frequentemente realizada por meio de validação cruzada. Embora o KNN seja simples de implementar e capaz de lidar com limites de decisão lineares e não lineares, pode ser computacionalmente caro, especialmente em conjuntos de dados grandes, pois requer o cálculo das distâncias para todos os pontos de treinamento. Otimizações como *KD-Trees* (BENTLEY, 1975) ou *Ball Trees* (FUKUNAGA; NARENDRA, 1975) podem ajudar a mitigar esse problema. Apesar de ser eficaz em muitas aplicações, como classificação de imagens e sistemas de recomendação, o KNN é sensível a características irrelevantes, tornando etapas de pré-processamento, como normalização, importantes para um desempenho ideal (CUNNINGHAM; DELANY, 2021).

2.4.2 Agrupamentos

A dificuldade de encontrar cenários com dados rotulados disponíveis em tempo hábil tem convertido o uso dos métodos de agrupamento numa opção amplamente utilizada nos últimos anos. Algoritmos como o *k-means* apresentado por (MACQUEEN, 1967) são um dos métodos mais populares e usados na atualidade, e é tomado como base para vários outros métodos, como o *K-medoids* (LAAN; POLLARD; BRYAN, 2003; PARK; JUN, 2009), *K-modes* (HUANG, 1997), *K-prototype* (HUANG, 1998) e *Fuzzy C-Means* (DUNN, 1973). A seguir, apresentamos uma breve descrição dos métodos acima mencionados, mesmo que os métodos de agrupamento para o aprendizado não supervisionado não sejam o foco desta pesquisa.

2.4.2.1 *K-means*

O *k-means* (MACQUEEN, 1967) divide os dados em k (número informado pelo especialista/usuário) partições mutuamente exclusivas. Este algoritmo é iterativo e minimiza a

soma das distâncias de cada objeto ao centroide de cada agrupamento, enquanto maximiza a variabilidade entre os grupos. A cada iteração, o *k-means* recalcula os centroides como a média dos pontos atribuídos, e os objetos são trocados entre os diferentes grupos até a função objetivo alterar pouco ou parar. Também se o número de iterações prefixado é alcançado. Apesar de sua simplicidade e eficiência, o *k-means* é sensível à inicialização dos centroides podendo convergir para mínimos locais, e funciona melhor com grupos esféricos e de tamanhos similares.

2.4.2.2 *K-medoids*

Considerado em várias literaturas como a versão relacional do *k-means* (SOARES et al., 2019), o algoritmo *k-medoids* difere do *k-means* por possuir só uma matriz de dissimilaridade entre os objetos do grupo de dados. Consequentemente, impede-se determinar a dissimilaridade para os objetos que não estão no grupo. A definição do medóide (objeto mais central do agrupamento) é baseada em fixar um objeto e, a partir do mesmo, minimizar o somatório das distâncias entre os objetos do mesmo agrupamento. O *k-medoids* é mais robusto a objetos atípicos (conhecidos como observações incomuns ou *outliers*), e dados ruidosos em comparação com o *k-means*, considerando que o medóide é sempre um objeto real, o que torna o algoritmo menos sensível a extremos. No entanto, isso também significa que o *k-medoids* pode ser mais computacionalmente custoso, especialmente para grandes conjuntos de dados, devido à necessidade de calcular repetidamente a soma das distâncias para todos os objetos.

2.4.2.3 *K-modes*

O *k-modes* (HUANG, 1997) é uma variação do algoritmo *k-means*, mas projetado para lidar com dados categóricos. A principal diferença entre o algoritmo *k-means* é que o *k-modes* utiliza a moda como centroide de cada grupo, e mede a similaridade entre os dados com base em uma métrica de dissimilaridade (distância de Hamming), que conta as diferenças entre as categorias. O algoritmo agrupa os dados minimizando a dissimilaridade total dentro de cada grupo. O processo de agrupamento continua até que o processo convirja e não haja nenhuma alteração nos grupos em duas iterações consecutivas. O *k-modes* é considerado um algoritmo eficiente para tarefas como segmentação e análise de

dados qualitativos.

2.4.2.4 *K-prototype*

O método *k-prototype* (HUANG, 1998) surge com o objetivo de eliminar as limitações do *k-means* em tratar com diferentes tipos de dados, sejam eles numéricos ou categóricos. O *k-prototype* integra características dos algoritmos *k-means* (para dados numéricos) e *k-modes* (para dados categóricos) para lidar com exemplos contendo misturas destes tipos de dados, para o qual utiliza uma métrica de distância híbrida que combina a distância Euclidiana para variáveis numéricas e a dissimilaridade para variáveis categóricas. O algoritmo encontra protótipos que representam cada grupo, permitindo agrupar dados mistos de forma eficiente. O algoritmo *k-prototype* é mais útil na prática, pois os dados coletados no mundo real são geralmente de tipos mistos (MAHDI et al., 2020).

2.4.2.5 *Fuzzy C-Means*

O *Fuzzy C-Means* forma parte dos métodos particionais não-exclusivos que estendem do *k-means*. O método permite agrupar objetos que pertencem a um espaço multidimensional em um número específico de diferentes grupos (DUNN, 1973). Neste algoritmo, se parte de assinalar um grau de pertinência de cada um dos objetos em relação a cada grupo, assim como de uma suposição inicial dos centroides de cada grupo. Centroides e graus de pertinência são atualizados iterativamente usando uma média ponderada dos pontos de dados e uma função de pertinência baseada na distância dos pontos aos centroides, com o objetivo de minimizar a função objetivo.

2.4.3 Métodos para o aprendizado semi-supervisionado

Os métodos desenhados para serem utilizados no aprendizado semi-supervisionado combinam dados rotulados e não rotulados para treinar modelos de aprendizagem de máquina. A utilização e exploração de ambos os conjuntos de dados pelos métodos semi-supervisionados possibilita melhorar a generalização e o desempenho dos modelos. Os métodos semi-supervisionados são úteis em cenários onde a rotulagem de dados é cara ou demorada, mas os dados não rotulados estão prontamente disponíveis.

O estudo bibliográfico revela que a maioria dos métodos projetados para o aprendizado semi-supervisionado são variações de algoritmos originalmente desenvolvidos para os cenários supervisionado ou não supervisionado, embora sejam comuns estratégias que combinam elementos de ambos os tipos de aprendizado para aproveitar ao máximo os dados disponíveis. Outras diversas estratégias foram desenvolvidas para lidar com a falta de rótulos no aprendizado semi-supervisionado, destacando-se entre elas os modelos de *Self-training* (auto-treinamento) (NGUYEN; GOMES; BIFET, 2019), *Co-training* (co-treinamento) (BLUM; MITCHELL, 1998), técnicas baseadas em entropia e abordagens generativas. Cada uma dessas metodologias oferece diferentes mecanismos para integrar informações dos dados não rotulados.

A seguir, exploramos algumas das abordagens analisando suas características, vantagens e limitações, com o objetivo de fornecer uma visão abrangente sobre as técnicas utilizadas no aprendizado semi-supervisionado. Neste trabalho, entre as abordagens apresentadas, destaca-se o *Self-training*, sendo proposto um novo método que explora sua aplicação.

2.4.3.1 Self-Training

O *Self-training* (NGUYEN; GOMES; BIFET, 2019) é um processo iterativo onde um classificador aprende e tenta melhorar a si mesmo. O *Self-training* usa primeiramente um grupo de objetos rotulados para treinar, e após faz a predição dos objetos não rotulados que serão incorporados ao conjunto de treino da próxima iteração, se a predição tiver um alto grau de confiança. Com o propósito da adaptação do *Self-training* tradicional para o trabalho em fluxos de dados, dois estilos de treino foram propostos: um por um (ou incremental) e lote incremental. Apesar de sua simplicidade, o *Self-training* pode ser muito eficaz, especialmente quando usado com classificadores que têm uma boa capacidade de generalização. No entanto, é importante monitorar cuidadosamente a confiança nas previsões e garantir que os dados rotulados adicionais sejam de alta qualidade para evitar a propagação de erros.

2.4.3.2 Co-Training

O *Co-training* (BLUM; MITCHELL, 1998; GOLDMAN; ZHOU, 2000) usa dois métodos (classificadores) do aprendizado supervisionado, com o objetivo de incrementar a precisão na classificação. No *Co-training*, um pequeno grupo de objetos rotulados é utilizado na rotulação dos objetos sem rótulo. Sua fundamentação é baseada em que, após a fase de treinamento dos classificadores, as instâncias de teste classificadas por um classificador serão adicionadas ao conjunto de treino do outro, se forem classificadas com alto grau de certeza (SANCHES, 2003). A principal vantagem do *Co-training* é sua capacidade de alavancar grandes quantidades de dados não rotulados para melhorar a performance dos classificadores, mesmo quando os dados rotulados são escassos. Ao explorar diferentes visões dos dados, o *Co-training* permite que os classificadores aprendam diferentes aspectos dos dados, levando a uma melhor generalização.

2.4.3.3 *SEEDED-K-Means e CONSTRAINED-K-Means*

Os métodos *SEEDED-K-Means* e *CONSTRAINED-K-Means* foram apresentados em (BASU; BANERJEE; MOONEY, 2002), e melhor detalhados na tese de doutorado de Basu (2005). Ambos os métodos têm como base o algoritmo *k-means* e utilizam objetos inicialmente rotulados como os centroides iniciais dos grupos, sendo esta a principal contribuição e diferença do *SEEDED-K-Means* para o método *K-Means*. No *SEEDED-K-Means* os centroides iniciais, assim como o método de geração, não são usados nos próximos passos do algoritmo. Neste algoritmo, é necessária a intervenção do usuário na inicialização dos centroides iniciais. Já o *CONSTRAINED-K-Means* é considerado uma melhoria do *SEEDED-K-Means*, e a diferença entre eles foi inserida nos passos posteriores à inicialização dos centroides, nos quais os exemplos que conformam o conjunto das sementes, e que foram inicialmente associados a um dado grupo pelo usuário, não poderão ser associados a um outro grupo (SILVA, 2017). Dessa forma, só os exemplos não selecionados como sementes serão reagrupados, ao contrário do *SEEDED-K-means* em que as sementes podem acabar sendo alocadas a grupos diferentes aos quais foram originalmente associados.

2.4.3.4 *K-Means*$_{ki}$

O objetivo do algoritmo *K-Means*$_{ki}$ (SANCHES, 2003) é rotular objetos a partir de um pequeno grupo de objetos rotulados, para melhorar o processo de classificação do método *SEEDED-K-Means* (BASU; BANERJEE; MOONEY, 2002), no qual se baseia. O *K-Means*$_{ki}$ diferencia-se do *k-Means* na fase de seleção dos centroides, onde são colocados como centroides iniciais cada um dos objetos rotulados disponíveis e não de forma aleatória como no método base. O uso de um parâmetro t pré-definido pelo usuário para associação de um objeto a um grupo, caso o objeto esteja a uma distância menor ou igual ao parâmetro do centroide do grupo, é outra diferença a ser sinalizada. O valor de t não é um valor absoluto, mas sim relativo.

No levantamento bibliográfico sobre métodos desenvolvidos para cenários semi-supervisionados, destaca-se que os pesquisadores têm dado pouca atenção à criação de métodos que incorporem mecanismos de detecção ou adaptação a mudanças de conceito. Essa percepção também foi apontada por Mahdi et al. (2020). Na próxima seção, são apresentados métodos que são parte da bibliografia que fornece algoritmos de detecção de mudanças de conceito.

2.5 MANIPULAÇÃO DE MUDANÇAS DE CONCEITO

O método adaptativo, ou detector de mudanças de conceito como também é conhecido, é um método criado com o objetivo de detectar mudanças na distribuição dos exemplos que estão sendo processados. Os métodos detectores geralmente estão caracterizados pela sua execução em paralelo com um classificador base, onde o classificador, para cada instância recebida, gera a predição 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 (PÉREZ, 2018).

Os métodos detectores são capazes de sinalizar se aconteceram mudanças de conceito, geralmente baseados nos erros sequencialmente cometidos pelo classificador base (BRZEZINSKI; STEAFNOWSKI, 2016). A maioria dos detectores criados usa dois níveis de alarmes: *warning* (alerta) e *drift* (mudança) (ATTAR et al., 2012), onde o *drift* é considerado o nível maior de mudança na distribuição analisada e sinaliza que, de fato, aconteceu uma mudança 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 classificador antigo é excluído e o novo é mantido. Por outro lado, caso o sinal de *warning* passe a ser considerado um alarme falso, a nova instância do classificador é excluída (PÉREZ, 2018).

A seguir, apresentamos quatro métodos detectores que são parte importante dos enfoques providos para o tratamento de mudanças de conceito em ambientes supervisionados. A implementação destes métodos encontra-se disponível no *framework* para a mineração de fluxos de dados *Massive Online Analysis* (MOA).

2.5.1 DDM

Drift Detection Method (DDM) (GAMA et al., 2004) utiliza os erros sequenciais na 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 esta aproxima-se da 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 exemplo (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. Em contrapartida, um incremento significativo na quantidade de erros do algoritmo sugere que a distribuição das classes está mudando e, portanto, o modelo de decisão atual é inapropriado. O DDM armazena os valores de (p_{min}) e (s_{min}) no treinamento do algoritmo de aprendizagem, que são atualizados quando uma nova instância causa que $p_i + s_i < p_{min} + s_{min}$.

DDM é caracterizado por apresentar um bom desempenho quando as mudanças não são muito lentas (BAENA-GARCIA et al., 2006). Os valores dos parâmetros do DDM sugeridos pelos autores e fixados como padrão no MOA para os níveis *warning* (w) e *drift* (d) são: 2.0 e 3.0, respectivamente. Também foi introduzido o número mínimo de instâncias (n) a considerar antes que a detecção das mudanças seja permitida, inicializado com valor de 30 instâncias.

Os três estados usados no DDM são listados a seguir:

- *in-control* (em-controle) é considerado no caso que $p_i + s_i < p_{min} + w \times s_{min}$, sendo assumido que os exemplos pertencem à mesma distribuição pelo que o comportamento

do sistema é estável.

- *warning* (alerta) é sinalizado no momento que $p_i + s_i \geq p_{min} + w \times s_{min}$, o nível avisa que o erro está aumentando, mas ainda não chegou ao nível considerado significativamente alto para declarar a mudança.

- *drift* (mudança) indica a detecção da mudança, baseado em que a condição $p_i + s_i \geq p_{min} + d \times s_{min}$ é satisfeita, consequentemente os valores de p_{min} e s_{min} são reiniciados.

2.5.2 FHDDM

Fast Hoeffding Drift Detection Method for Evolving Data Stream (FHDDM) (PESARANGHADER; VIKTOR, 2016) implementa o uso de uma janela deslizante de tamanho n contendo os valores correspondentes erro (0) ou acerto (1) do classificador. Para a detecção das mudanças de conceito, é usada a inequação de *Hoeffding* (HOEFFDING, 1963). 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 apresentada na Equação 2.3 para t.

$$if p_{max}^1 < p_t^1 \Rightarrow p_t^1 \rightarrow p_{max}^1 \tag{2.3}$$

O modelo de aprendizado *Probably Approximately Correct* (PAC) (MITCHELL, 1997) é usado no detector FHDDM sendo demonstrado que a possibilidade de acontecer uma mudança de conceito aumenta caso p_{max}^1 não mude e p_t^1 diminua ao longo do tempo. Uma diferença significativa entre p_{max}^1 e p_t^1 indica a ocorrência da mudança de conceito no fluxo de dados, como apresenta-se na Equação 2.4.

$$\Delta p = p_{max}^1 - p_t^1 \geq \varepsilon_d \Rightarrow Drift := True \tag{2.4}$$

Onde o valor de ε_d é calculado utilizando a probabilidade do erro δ (padrão 10^{-7}) fornecida pelo conceito de *Hoeffding bound* (HOEFFDING, 1963; MARON; MOORE, 1993). O valor de n é igual a 200 na implementação do MOA.

2.5.3 HDDM$_A$

O método *Hoeffding-based Drift Detection Method A-Test* (HDDM$_A$) proposto em (FRÍAS-BLANCO et al., 2015) monitora uma média estimada de desempenho baseada nos acertos e erros do classificador, e utiliza a comparação de médias móveis para detectar mudanças de conceitos. Segundo os autores, o método realiza as detecções em um único (com complexidade $O(1)$) passo e oferece garantias de desempenho quanto às taxas de falsos positivos e falsos negativos. Este método de detecção usa a inequação de *Hoeffding* (HOEFFDING, 1963) para definir um limite superior para o nível de diferença entre as médias.

Na implementação de HDDM$_A$ dois níveis (α_W *e* α_D) de confiança são utilizados para aceitar ou rejeitar a hipótese nula (H_0). O estado de *warning* (alerta) é ativado quando H_0 é rejeitada com tamanho α_W, caso a rejeição de H_0 seja com tamanho α_D é informado o estado de *drift* (mudança). Os valores padrões do HDDM$_A$ sugeridos por Frías-Blanco et al. (2015) e presentes na implementação na ferramenta MOA são $\alpha_W = 0.005$ e $\alpha_D = 0.001$.

O HDDM$_A$ é considerado um método competitivo e de alto desempenho na detecção da maioria dos tipos de mudanças de conceito, mas é mais apropriado para detectar mudanças abruptas, de acordo com observações empíricas realizadas pelos autores. Embora outros autores como Barros e Santos (2018) não tenham confirmado isto nas suas experimentações.

2.5.4 RDDM

No detector *Reactive Drift Detection Method* (RDDM) (BARROS et al., 2017) os autores apresentam uma modificação do método DDM para melhorar um conhecido problema de perda de desempenho em conceitos longos. O algoritmo sinaliza um novo tipo de *drift* denominado RDDM *drift* quando o número de instâncias do conceito atual atinge o número máximo predefinido de instâncias e atualiza as estatísticas do DDM usando apenas as instâncias mais recentes. Faz-se válido ressaltar que o RDDM também sinaliza mudança de conceito quando o número de instâncias do nível de aviso atinge um certo limite.

Quando tratamos da detecção de mudanças de conceito em ambientes semi-supervisionados, o estado da arte apresenta uma quantidade limitada de abordagens (MAHDI

et al., 2020). O fenômeno de não utilizar mecanismos de detecção em métodos semi-supervisionados decorre do fato de que muitos autores consideram que a ausência de rótulos impacta negativamente os detectores (TAN; LEE; SALEHI, 2019), os quais geralmente dependem de rótulos (Seção 2.5) para calcular as predições usadas na identificação de mudanças. Assumir que, em cenários de fluxos de dados, todos os rótulos estarão disponíveis imediatamente após a chegada dos dados é uma expectativa pouco realista; com base nisso, autores sugerem a abordagem de restringir o uso desses mecanismos de detecção apenas aos dados rotulados (TAN; LEE; SALEHI, 2019; PINAGÉ; SANTOS; GAMA, 2020). Contudo, não foram encontrados registros na bibliografia de estudos empíricos que avaliem o impacto da ausência de rótulos nos detectores e, consequentemente, no modelo de aprendizado.

A seguir, são apresentados cinco métodos desenvolvidos para cenários de aprendizado semi-supervisionado. Nesses métodos, os autores incorporam mecanismos de detecção de mudanças de conceito, com exceção da proposta *Batch training using distance-based confidence score and fixed confidence threshold* (BDF), que não foi apresentada com um mecanismo de detecção específico. No entanto, sua estrutura e a plataforma (MOA), onde foi implementada, permitem a integração de métodos adaptativos, como DDM, FHDDM, HDDM$_A$, RDDM entre outros.

2.5.5 SAND

O algoritmo *Semi-Supervised Adaptive Novel Class Detection and Classification over Data Stream* (SAND) *framework* (HAQUE; KHAN; BARON, 2016) é um método semi-supervisionado dividido nos módulos de detecção de dado discrepante em relação ao conjunto de dados (*outlier*) e detecção de mudanças de conceito, usando como método de classificação base um comitê de classificadores KNN. SAND utiliza uma janela W para monitorar estimativas de confiança do classificador nas instâncias recentes e introduz de forma implícita uma técnica de detecção de mudanças de conceito, que ao identificar um decrescimento significativo na confiança do classificador emite um sinal de mudança de conceito. Além das mudanças de conceito, o método detecta *outliers* com forte coesão entre si.

2.5.6 ECHO

Efficient Semi-Supervised Adaptive Classification and Novel Class Detection over Data Stream (ECHO) (HAQUE et al., 2016) é um método baseado em SAND e utiliza programação dinâmica para reduzir a complexidade de tempo do módulo de detecção de mudanças de conceito presente em SAND. Além disso, ECHO tem um tamanho máximo permitido para a janela deslizante: se nenhuma mudança de conceito for detectada dentro deste limite, ECHO atualiza os classificadores e reinicia a janela deslizante. Os resultados dos experimentos realizados pelos autores mostram que ECHO atinge um aumento significativo de velocidade sobre SAND enquanto mantém uma precisão semelhante.

2.5.7 DMDDM-S

O *Fast Reaction to Sudden Concept Drift in the Absence of Class Labels* (DMDDM-S) (MAHDI et al., 2020) é um *framework* que propõe utilizar dois classificadores base em paralelo, HT e *Perceptron*, para detectar mudanças de conceito mediante o monitoramento do nível de discordância das predições de ambos. Assim, esse método concentra-se na quantificação da discrepância entre as previsões de pares de classificadores, sem considerar os rótulos reais das classes. Ao focar na discordância, o DMDDM-S pode detectar mudanças sem depender exclusivamente de métricas de desempenho, como acurácia, que podem ser difíceis de calcular em cenários semi-supervisionados. Em DMDDM-S, os rótulos das classes de entrada nem sempre são necessários, mas os autores usam o algoritmo de agrupamento *k-Prototype* (HUANG, 1998) para rotular os dados não rotulados, que são usados posteriormente para treinar o modelo. De acordo com seus autores, mesmo com apenas 50% dos dados rotulados, identifica *drifts* de maneira mais rápida e eficiente, tanto em termos de tempo de execução quanto de uso de memória, quando comparado a métodos tradicionais totalmente supervisionados. Já em termos de resultados de acurácia, observa-se em (MAHDI et al., 2020; PÉREZ; BARROS; SANTOS, 2025) que DMDDM-S *framework* não obtém resultados superiores a métodos supervisionados como o RDDM.

2.5.8 BDF

O BDF (NGUYEN; GOMES; BIFET, 2019) é um método semi-supervisionado baseado em uma abordagem de *Self-training*. A etapa de treinamento no algoritmo BDF é realizada utilizando todas as instâncias rotuladas, enquanto obtém as previsões das não rotuladas e, com base em um parâmetro de confiança, decide se utiliza as instâncias com previsões mais confiáveis no treinamento. Também vale ressaltar que o BDF foi implementado no *framework* MOA e aproveita que, neste *framework* cada classificador implementado fornece uma função para estimar a confiança da previsão da instância informada. Além disso, usa uma pontuação de confiança baseada na distância e um limite de confiança fixo, assim como o algoritmo treinado é o HT.

2.5.9 DSDD

O DSDD (PINAGÉ; SANTOS; GAMA, 2020) é uma versão modificada do método *Online Bagging* e usa uma abordagem de *self-training* no aprendizado online (GEMAQUE et al., 2020) para rotular instâncias. Este algoritmo tem três módulos: criação do comitê, seleção dinâmica de classificadores e detecção de mudanças de conceito. O primeiro módulo constrói o comitê de classificadores, usando a distribuição de Poisson para controlar a quantidade de instâncias apresentadas a cada classificador membro do comitê. O segundo módulo tem como premissa que cada membro do comitê é um especialista em uma região de competência. Portanto, quando chega uma instância não rotulada, o classificador membro mais acurado é considerado especialista na região atual e é o responsável pela predição do rótulo. O DSDD assume que o rótulo atribuído à instância foi o correto e incorpora a mesma ao conjunto de treinamento. O terceiro módulo é usado para detecção de mudanças de conceito: nele, para cada classificador membro, um detector é aplicado. Os autores Pinagé, Santos e Gama (2020) sugerem o uso dos detectores DDM ou *Early Drift Detection Method* (EDDM) (BAENA-GARCIA et al., 2006).

2.6 COMITÊ DE CLASSIFICADORES

Os comitês de classificadores combinam a saída de múltiplos modelos de aprendizado de máquina com o objetivo de melhorar o desempenho preditivo em relação a um único

classificador. A motivação central baseia-se no princípio de que a combinação de vários classificadores, preferencialmente diversos e complementares, pode reduzir erros associados ao viés, à variância e ao ruído. Além disso, o uso de comitês favorece maior robustez e generalização, especialmente em problemas complexos. O desenvolvimento de comitês está apresentando resultados relevantes, mesmo que associados a problemas como alto consumo de tempo e memória. Escolher os classificadores base e combiná-los estruturalmente, bem como suas saídas para formar um comitê de boa performance, é uma tarefa desafiadora e custosa (SANTOS; BARROS, 2020).

Incluir mecanismo de detecção de mudanças de conceito nos comitês, como uma forma de identificar e adaptar-se às mudanças que podem ocorrer na distribuição dos dados impactando a performance, tem se tornado uma opção viável. De fato, usar métodos detectores de mudanças de conceito de forma individual por cada classificador ou um detector vinculado ao comitê são algumas das abordagens encontradas no estado da arte. Os primeiros comitês não contavam com sistemas de detecção de mudanças de conceito e, na sua maioria, se adaptavam de forma implícita com atualizações dos modelos em períodos de tempo determinados.

Identificar um conjunto ótimo de classificadores para formar o comitê é influenciado pelo contexto do problema específico que se pretende resolver, assim como pelos dados disponíveis para treinar os métodos base. A inclusão de diversidade entre os classificadores do conjunto é necessária para o sucesso e a robustez dos comitês. A diversidade pode ser introduzida com o uso de diferentes técnicas e nem sempre exige que os modelos base sejam de alta performance.

As estratégias mais populares de combinação de modelos utilizadas na criação de comitês para problemas de regressão e classificação (AGGARWAL, 2014; MARIÑO, 2019) são o *Stacking* (WOLPERT, 1992), *Boosting* (FREUND, 1995) e *Bagging* (BREIMAN, 1996). Os mecanismos de detecção de mudanças de conceitos geralmente substituem um membro do comitê. A maioria dos comitês que usam detectores utiliza *Weighted Majority Algorithm* (WMA) (LITTLESTONE; WARMUTH, 1994), segundo afirmações (MAHDI et al., 2024).

Baseado nas abordagens acima mencionadas, com o passar dos anos, vários métodos surgem na busca por melhorar o desempenho em fluxo de dados na presença de mudanças de conceito no aprendizado supervisionado, entre os quais podemos mencionar *Boosting-like Online Learning Ensemble* (BOLE) (BARROS; SANTOS; GONÇALVES JR., 2016), *Fast Adaptive Stacking of Ensembles* (FASE) (FRÍAS-BLANCO et al., 2016), *Advanced ELM*

Ensemble (AELME) (ABUASSBA et al., 2017), e *Online AdaBoost-based methods for Multiclass problems* (SANTOS; BARROS, 2020) com suas versões *Online AdaBoost-based M1* (OABM1) e *Online AdaBoost-based M2* (OABM2). Outro método disponível em duas versões é o *Online Adaptive Classifier Ensemble*: *Online Adaptive Classifier Ensemble based on bagging* (OACE_{BA}) e *Online Adaptive Classifier Ensemble based on boosting* (OACE_{BO}).

Já nos cenários de aprendizado semi-supervisionado, o algoritmo *CoAdabost* (HADY; SCHWENKER, 2008) encontra-se entre os mais conhecidos e serve como base para outras propostas como as apresentadas em (WANG et al., 2014). Outras propostas bem conhecidas são o ECHO (HAQUE et al., 2016), que é baseado no SAND (HAQUE; KHAN; BARON, 2016), DMDDM-S (MAHDI et al., 2020) e DSDD (PINAGÉ; SANTOS; GAMA, 2020). A explicação detalhada desses métodos pode ser encontrada na seção 2.5, onde foram adicionados por fazerem parte do estado da arte de métodos criados para serem utilizados no aprendizado em cenários semi-supervisionados e contarem com mecanismo de detecção de mudanças de conceito.

A seguir, serão apresentados em detalhe alguns dos métodos previamente mencionados, que integram o estado da arte dos comitês de classificadores. Esses comitês, originalmente desenvolvidos para ambientes supervisionados, foram selecionados para um detalhamento neste capítulo devido à sua ampla utilização em vários estudos.

2.6.1 FASE

O FASE é um método que incorpora um mecanismo de detecção de mudanças de conceito e tem como inspiração a versão online do algoritmo *Bagging* (OZA; RUSSELL, 2001). O algoritmo FASE utiliza votação ponderada para combinar as previsões dos modelos. Outra inovação introduzida no método é o uso de um meta-classificador para combinar as predições dos classificadores. O meta-classificador é treinado com meta-instâncias criadas, onde os atributos de cada meta-instância são as predições dos classificadores-base do comitê e o rótulo utilizado é o da instância original usada no treinamento (FRÍAS-BLANCO et al., 2016). Considerado um método adaptativo, o FASE em sua versão publicada usa HDDM_A como detector de mudanças de conceito, mas sua estrutura modular permite alteração de métodos de detecção de mudança de conceito e dos classificadores base.

2.6.2 BOLE

BOLE (BARROS; SANTOS; GONÇALVES JR., 2016) surge de modificações simples na heurística de ADOB (SANTOS et al., 2014). As duas principais modificações foram: (1) diminuir os requisitos para permitir a votação dos classificadores e (2) alterar o método de detecção de mudanças de conceito utilizado, melhorando a precisão do modelo na maioria das situações, especialmente quando as mudanças de conceito são frequentes e/ou abruptas. O (*BOLE4*) é a versão do método que usa como detector de mudança de conceitos o DDM (GAMA et al., 2004), sendo a recomendada pelos autores com base em (GONÇALVES JR. et al., 2014). Porém, atualmente, os autores do BOLE recomendam o uso do RDDM como seu detector (BARROS; SANTOS, 2019).

2.6.3 OABM1

O método *Online AdaBoost-based M1* (OABM1), apresentado por (SANTOS; BARROS, 2020), é baseado no *AdaBoost.M1* (FREUND; SCHAPIRE, 1997). Nesta versão, as principais características do método original são mantidas, como a utilização de um mecanismo que ajusta os pesos dos classificadores, aumentando-os ou diminuindo-os, dependendo se eles acertam ou erram na classificação das instâncias, respectivamente. Uma diferença é que, no método OABM1, a entrada é uma instância e não um lote, como no algoritmo base, devido ao fato de que o OABM1 foi desenvolvido para cenários online de fluxos de dados. No OABM1, assim como no *AdaBoost.M1*, cada classificador é treinado de acordo com os pesos atuais representados por um vetor, que são utilizados em uma distribuição de Poisson (OZA; RUSSELL, 2001) com parâmetro (λ), para controle de diversidade. O valor de λ é variável, com um valor inicial padrão de 1, mas OABM1 introduz o valor inicial de λ como um parâmetro formal, permitindo alteração do parâmetro. O desempenho experimental relacionado aos resultados de acurácia e à sua demonstração de convergência obtidos pelo método OABM1 é considerado bom (HIDALGO; SANTOS; BARROS, 2021; SANTOS; BARROS, 2020).

2.6.4 OACE

O OACE (VERDECIA-CABRERA; FRÍAS-BLANCO; CARVALHO, 2018) utiliza um mecanismo de detecção de mudanças em cada classificador base para lidar com possíveis mudanças na função alvo subjacente. Cada classificador base no comitê pode alternar entre três estágios diferentes durante o processo de aprendizado: estável, alarme e mudança. O OACE utiliza *Bagging online* ou *Boosting online* para treinar os classificadores base de um comitê. Para realizar a classificação, o OACE$_{BA}$ utiliza esquemas de votação por maioria simples, enquanto o *Online Adaptive Classifier Ensemble based on boosting* (OACE$_{BO}$) utiliza esquemas de votação por maioria ponderada.

2.7 CONSIDERAÇÕES FINAIS

Este capítulo fez uma introdução à base conceitual das áreas a serem abordadas na pesquisa. Além disso, vários algoritmos propostos para a aprendizagem de máquina e detecção de mudanças de conceito em cenários supervisionados e semi-supervisionados foram apresentados. Estes métodos serão opções de escolha para serem utilizados nas experimentações a realizar em próximos capítulos.

3 FERRAMENTAS PARA CENÁRIOS SEMI-SUPERVISIONADOS

Este capítulo aborda a necessidade de ferramentas apropriadas para a experimentação em cenários do aprendizado semi-supervisionado. Além disso, são destacadas as funcionalidades existentes no *framework* MOA, assim como a implementação de novas funções que o habilitam para o trabalho no aprendizado semi-supervisionado.

3.1 MOA *FRAMEWORK* PARA CENÁRIOS SEMI-SUPERVISIONADOS

No começo desta pesquisa, foram encontrados dois problemas principais. O primeiro era não contar com uma ferramenta para gerar fluxos de dados para cenários do aprendizado semi-supervisionado. O segundo problema é dado pela existência de poucos métodos semi-supervisionados que possuíssem mecanismos de detecção de mudanças de conceito.

A resolução do primeiro problema foi abordada com a implementação de novas funcionalidades no *framework Massive Online Analysis* (MOA) (BIFET et al., 2010). A implementação do MOA utiliza a linguagem de programação *Java*, possibilitando que o *framework* seja multiplataforma (compatível com *Windows*, *Unix/Linux* e *Macintosh*). No desenvolvimento das novas funcionalidades foi usado o ambiente de desenvolvimento integrado *Intellij*-2019.1 com a Kit de desenvolvimento de software (SDK) versão 8. Com relação ao *Java Runtime Environment* (JRE), é aconselhável usar a mesma versão do SDK para a execução.

Além das informações acima mencionadas, é importante destacar que foi utilizado o pacote da versão 2018.1 do MOA[1] para incorporar as novas funcionalidades. Assim como, as bibliotecas *weka*[2], *sizeofag*[3] com extensão de arquivo *.jar.

O MOA é usado em tarefas de classificação, regressão, agrupamentos (*clustering*), tratamento de *outliers* e detecção de mudanças de conceito (*concept drift*). Cada uma dessas tarefas possui uma aba no *framework*. Na versão 2018.1, duas novas abas foram incorporadas: *Other tasks* e *Semi-Supervised Learning* (versão apresentada (NGUYEN; GOMES; BIFET, 2019)). A primeira, além de permitir avaliações adicionais/alternativas para os modelos, permite também a geração de bases de dados no formato *arff*. Já a segunda,

[1] <https://sourceforge.net/projects/moa-datastream/>
[2] <https://sourceforge.net/projects/weka/>
[3] <https://jar-download.com/artifact-search/sizeofag/>

apresentada em (NGUYEN; GOMES; BIFET, 2019), traz diversas funcionalidades voltadas à aprendizagem semi-supervisionada, mas ainda não foi incorporada ao repositório oficial. A tela principal do *framework* MOA apresenta-se na Figura 1.

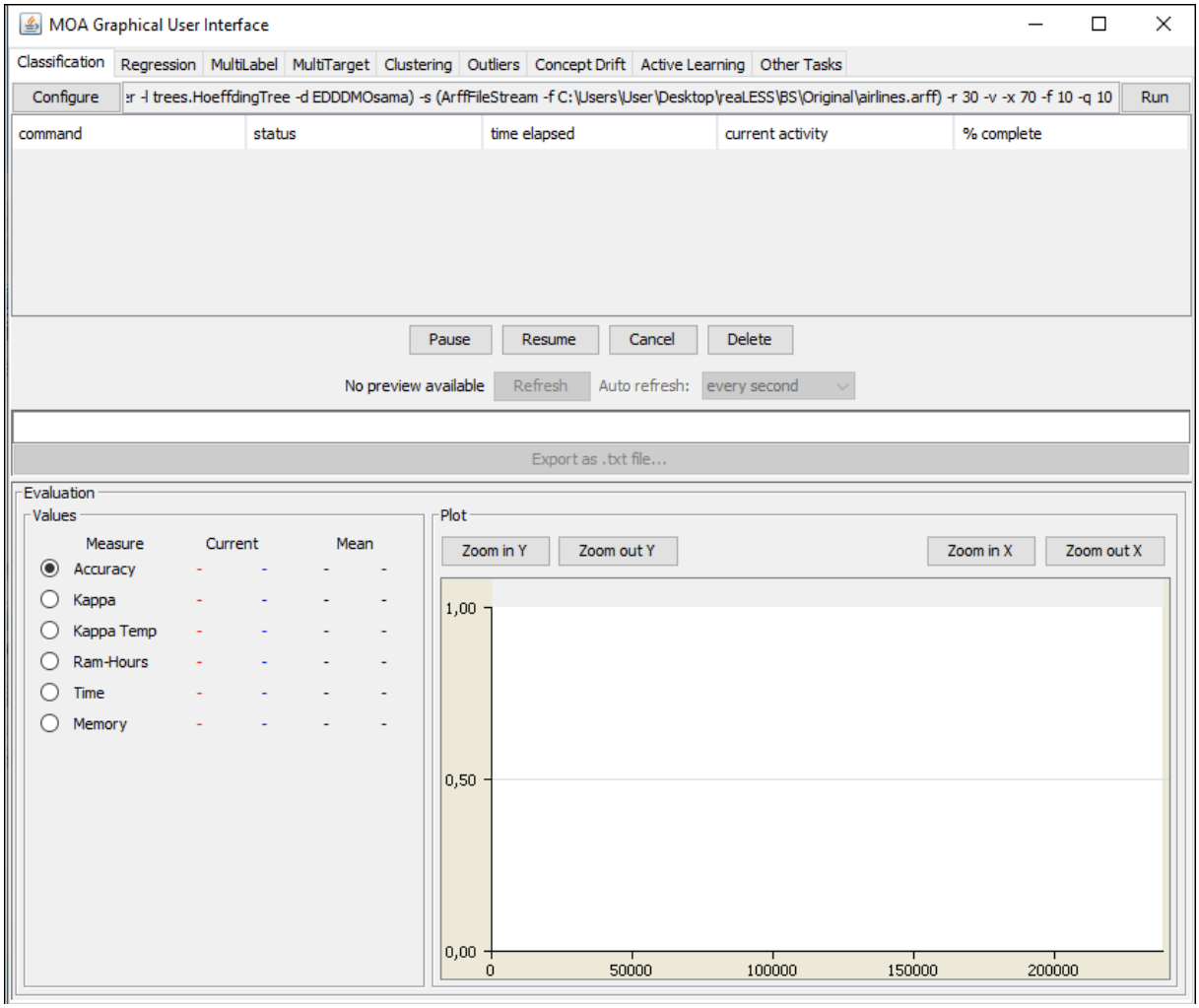


Figura 1 – Tela de execução do *framework* MOA, versão 2018.1

A execução dos experimentos ou tarefas no MOA podem ser realizadas basicamente de duas formas: por linhas de comando, rodando *scripts* contendo a configuração dos experimentos, ou pela interface gráfica. A primeira opção permite rodar mais rapidamente grande quantidade de experimentos, mas ainda pode ser considerada uma alternativa trabalhosa. Com o objetivo de facilitar a experimentação e transformá-la em um processo menos cansativo, os autores Bruno I. F. Maciel, Silas G. T. C. Santos, e Roberto S. M. Barros criaram a ferramenta *MOAManager* (MACIEL; SANTOS; BARROS, 2020). Além de facilitar a execução, a ferramenta permite a coleta e análise dos dados dos experimentos executados no MOA. Na Figura 2 é apresentada a tela de execução da ferramenta.

Os *scripts* de configuração das tarefas rodados no MOA são 100% compatíveis com a

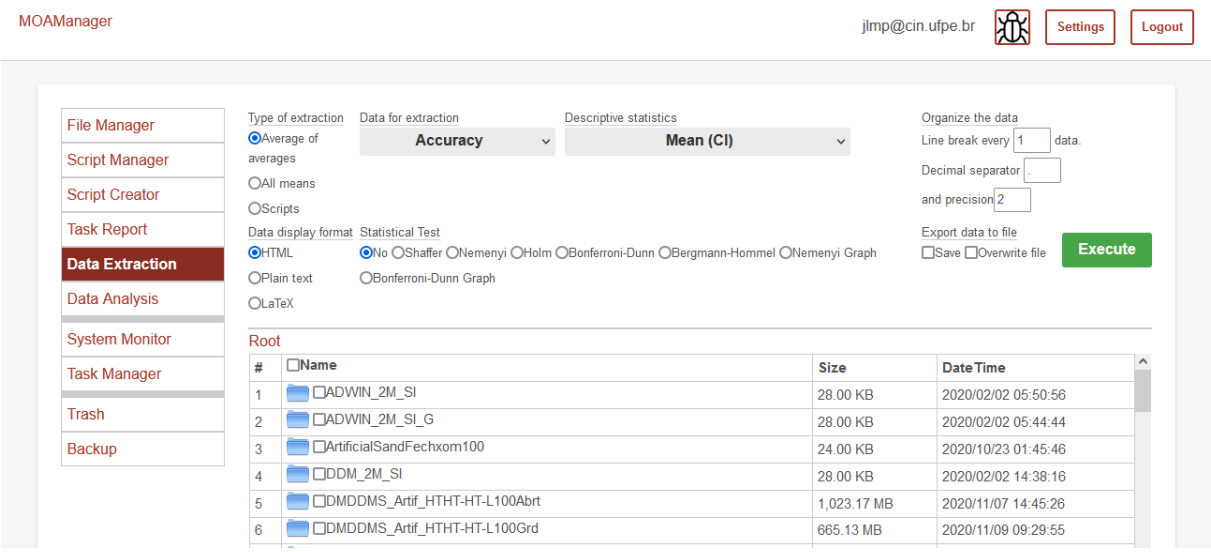


Figura 2 – Tela de execução da Ferramenta *MOAManager*

ferramenta *MOAManager*, desde que utilize a mesma versão do MOA. A seguir, apresenta-se um exemplo de *script* onde cada parâmetro é definido por um hífen (**-**) seguido pela letra que o identifica e o valor atribuído. Quando o valor atribuído não é apresentado, significa que o método utilizará o valor padrão da ferramenta.

- *EvaluatePrequential* **-l** (*drift.DriftDetectionMethodClassifier* **-l** *trees.HoeffdingTree* **-d** DMDDMS) **-s** (*ArffFileStream* **-f** C:\computador\reais\baseDados*airlines.arff*) **-r** 30 **-v** **-x** 70 **-f** 10 **-q** 10

Ao longo do trabalho que vem sendo desenvolvido, várias têm sido as limitações. Para enfrentar algumas, como já mencionado anteriormente, foram implementadas novas funcionalidades que serão descritas nos próximos parágrafos.

Inicialmente, na aba classificação (*classification*), como apresenta a Figura 3, foi introduzido um seletor (*semiSupervised*) para o tipo de aprendizado semi-supervisionado. Quando marcado, o modelo treina somente com os dados que possuem rótulos e testa com todos. Para fins experimentais, foi assumido que todos os rótulos estarão disponíveis para o teste, só que sua apresentação ao detector de mudança de conceito vai depender do seletor e porcentagem escolhida. Por outro lado, caso esteja desmarcado, o aprendizado será 100% supervisionado e o MOA manterá o seu comportamento original. Um outro campo introduzido (*percentage_W/L*) controla a porcentagem de instâncias que devem ser consideradas sem rótulos.

Outra alteração implementada (campo *numberInst_I/T*) é utilizada para controlar

a quantidade de instâncias rotuladas necessárias para o início do treinamento (antes de começar a considerar os dados não rotulados). As alterações mencionadas podem ser observadas na Figura 3, dentro do quadro verde. Já no quadro laranja na mesma figura, é apresentado o campo de repetição introduzido no MOA e fixado como padrão no *MOA-Manager tool* (MACIEL; SANTOS; BARROS, 2020). O campo de repetição permite repetir os experimentos com o objetivo de obter os resultados com um intervalo de confiança. O campo de repetição da versão agora modificada do MOA foi adaptado para ser usado também nas bases de dados reais. A diferença é que, nas bases artificiais, o r_i da repetição atual é usado como semente para os geradores de bases de dados, enquanto que nas reais o r_i é usado para definir as posições onde serão retirados os rótulos das instâncias.

Na aba *other tasks* foram adicionados os mesmos três campos que na aba classificação, além do campo repetição. Neste caso, os campos serão usados para a geração das bases de dados para o aprendizado semi-supervisionado. As bases resultantes serão armazenadas no formato *arff*, e podem ser criadas a partir de geradores artificiais ou bases reais já existentes. A geração das bases de dados em arquivos *arff* vai permitir a execução dos experimentos com outros algoritmos criados em plataformas diferentes do MOA.

A seguir serão listados os parâmetros acima mencionados e incorporados no MOA:

- **-v** (*semiSupervised*): a presença desse parâmetro no *script* indica que o aprendizado é semi-supervisionado;
- **-x** (*percentage_W/L*): sinaliza a porcentagem de rótulos a serem retirados das instâncias das bases. O valor padrão no MOA foi definido como 75%;
- **-n** (*numberInst_I/T*): com o valor padrão de 100, o parâmetro fixa a quantidade de instâncias a serem aguardadas antes de começar a retirar os rótulos dos dados.
- **-r** (*repetition*): único dos parâmetros que já existia no MOA e foi adaptado para ser usado com as novas funcionalidades. O valor definido como padrão é 1.

As linguagens de programação utilizadas para a aprendizagem de máquina são diversas e têm evoluído com o passar dos anos para se ajustar às necessidades. Na atualidade, linguagens mais antigas como *C++*, *C#* e *Java* encontram-se em constante atualização, o que permite facilidade de implementação de novos algoritmos para aprendizagem de máquina. Embora seja válido ressaltar que nos últimos 5 anos linguagens como *Matlab*, *R*, *Go* e *Python*, assim como bibliotecas baseadas nelas, estão sendo cada vez mais utilizadas

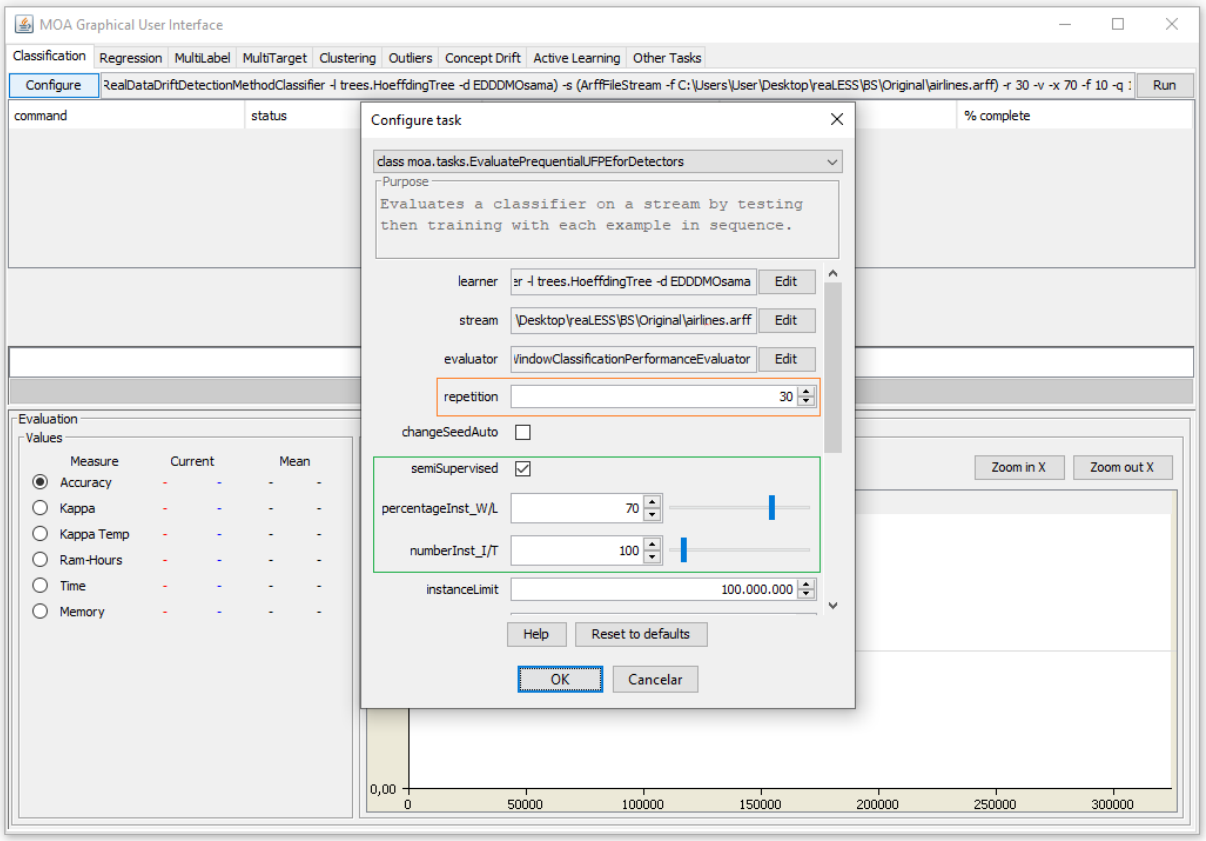


Figura 3 – Alterações no *framework* MOA para o aprendizado semi-supervisionados. (ABA Classificação)

pelos pesquisadores e no mercado para aprendizagem de máquina. A escolha do MOA baseou-se na ampla disponibilidade de algoritmos nesse *framework*, bem como em sua ampla utilização em pesquisas de aprendizado de máquina na presença de mudanças de conceito. Além disso, o MOA inclui geradores de bases de dados que permitem simular essas mudanças.

3.2 OUTRAS FERRAMENTAS PARA CENÁRIOS SEMI-SUPERVISIONADOS

Nos últimos anos, tem ocorrido um aumento significativo no número de bibliotecas, *frameworks* e plataformas desenvolvidos para treinamento, teste e aplicação de algoritmos de aprendizado de máquina. As ferramentas mais comumente utilizadas são:

- *Shogun* [4]: é considerada uma das bibliotecas mais antigas utilizadas para aprendizagem de máquina. Escrita na linguagem *C++*, *Shogun* usa a biblioteca *SWIG* para integrar-se de forma transparente com diversas linguagens e ambientes, a exemplo de *Java*, *Python*, *C#*, *Ruby*, *R*, *Lua*, *Octave* e *Matlab*;

[4] <https://shogun-toolbox.org/>

- *Weka* [5]: desenvolvida na Universidade de *Waikato*, Nova Zelândia, contém um amplo grupo de algoritmos de aprendizagem de máquina e mineração de dados implementados em *Java*. Além disso, conta com várias bibliotecas que permitem estender suas funcionalidade para criar outras ferramentas, como MOA. O *Weka* conta com numerosos materiais de ajuda, dentre os quais destaca-se um livro (WITTEN; FRANK, 2002) desenvolvido para explicar o *software* e as técnicas utilizadas;

- *Scikit-learn* [6]: é um software totalmente aberto e reutilizável, e foi criado baseando-se em vários pacotes existentes de *Python*: *NumPy*, *SciPy* e *matplotlib*. O *Scikit-learn* pode ser utilizado para aplicativos interativos ou incorporadas em outros softwares;

- *Orange3* [7]: com uma excelente interface de usuário, é utilizado sobretudo no pré-processamento e na visualização de dados. Além disso, seu acesso pode ser realizado via Anaconda ou através de chamada por linha de comando no *Python*. Sua acessibilidade é uma das razões do aumento da popularidade do *framework*.

- *TensorFlow* [8]: é um *framework* de código aberto útil para a aprendizagem de máquina em grande escala. Seus modelos são implementados em *Python* e podem ser executados tanto em *CPU* quanto em *GPU*. O processamento de linguagem natural e a classificação de imagens são suas principais funcionalidades.

- *PyTorch* [9]: é escrito em *Python* para uso na criação de modelos de *Deep Learning* e considerado um *framework* de fácil aprendizado. O *framework* é usado principalmente para o reconhecimento de imagens e o processamento do linguagens.

3.3 CONSIDERAÇÕES FINAIS

Na atualidade, a necessidade de ferramentas que se ajustem às demandas da pesquisa é perceptível. A não existência desta ferramenta tornou indispensável a implementação da mesma ou adaptação de alguma já existente para trabalhar nos cenários de teste a serem executados. Levando-se em consideração que o MOA é frequentemente utilizado nas diversas pesquisas que tratam de mudanças de conceito, além de reunir uma grande

[5] <https://cs.waikato.ac.nz/ml/weka/>
[6] <https://scikit-learn.org/stable/>
[7] <https://orangedatamining.com/>
[8] <https://tensorflow.org/>
[9] <https://pytorch.org/>

quantidade de algoritmos que podem ser usados nas experimentações, ele foi escolhido para ser modificado. Como apresentado neste capítulo, duas abas foram alteradas, sendo estas *Classification* e *Other tasks*. As novas funcionalidades, em resumo, habilitam o MOA para um funcionamento em cenários de experimentação semi-supervisionados e também para criar os cenários semi-supervisionados para serem testados em outros *frameworks*, obtendo-se desta forma um *framework* mais completo e compatível com a ferramenta de extração de dados *MOAManager*. A versão contendo as alterações inseridas como resultado desta tese foi nomeada *MOAManagerSS*.

4 CONFIGURAÇÃO E METODOLOGIA EXPERIMENTAL

No atual capítulo são apresentadas as configurações utilizadas na experimentação, tendo uma cronologia definida da seguinte maneira: 1) selecionar as bases de dados artificiais e reais; 2) informar as ferramentas e características do computador utilizado nos testes; 3) definir os métodos de detecção de conceitos e classificadores utilizados na experimentação; e, por fim, 4) determinar os critérios de avaliação a serem usados na comparação dos algoritmos escolhidos.

É pertinente esclarecer que 100% dos experimentos realizados em ambientes semi-supervisionados utilizam a mesma configuração. Contudo, este capítulo também apresenta as configurações utilizadas para testes em cenários supervisionados. A experimentação nesses cenários tem como objetivo demonstrar o impacto da diversidade nos comitês de classificadores e definir parâmetros adequados de configuração. Para manter a concordância com o artigo (PÉREZ; MARIÑO; BARROS, 2021), incluímos as configurações e informações publicadas nele. Embora existam poucas diferenças em relação à experimentação restante, essas diferenças serão destacadas nas seções correspondentes, e os motivos das alterações em relação aos experimentos para cenários semi-supervisionados serão explicados.

4.1 BASES DE DADOS ARTIFICIAIS

Para o desenvolvimento da pesquisa é importante que se tenha fluxos de dados com presença de mudanças de conceito controladas, na qual a informação da quantidade de mudanças, o tempo em que as mudanças vão sendo construídas e o momento exato em que ocorrem, são conhecidos. Consequentemente, na investigação, para ter o controle necessário da mudança foi utilizada a função sigmoide implementada no MOA. Essa função induz as mudanças nos fluxos e determina a mudança de uma origem de dados a outra.

Como visto anteriormente, é possível gerar fluxos de dados artificiais usando os geradores oferecidos pelo MOA. Nesta pesquisa, foram selecionados para experimentação um total de sete geradores de bases artificiais, que serão detalhados nas próximas subseções (4.1.1 a 4.1.7). Neste momento, é válido informar que o gerador *SEA* foi utilizado apenas na experimentação supervisionada que resultou no artigo (PÉREZ; MARIÑO; BARROS, 2021). Nos experimentos realizados em cenários semi-supervisionados, ele foi substituído

pelo gerador *LED*, com o objetivo de introduzir diversidade no conjunto de bases selecionadas para testes, considerando a existência de vários bancos de dados artificiais voltados para classificação binária.

4.1.1 *Agrawal*

O gerador *Agrawal* (AGRAWAL; IMIELINSKI; SWAMI, 1993) utiliza dez funções diferentes com nove atributos para construir fluxos de dados. Dos atributos, três (nível de escolaridade, marca do carro e código postal) são categóricos e os restantes (salário, comissão, idade, valor do imóvel, idade do imóvel e o valor do empréstimo) são numéricos. A classificação das instâncias é binária, simulando a hipotética concessão de um empréstimo bancário, sendo as classes A e B a interpretação da concessão positiva ou negativa, respectivamente. *Agrawal* é considerada uma ótima opção para gerar mudanças de conceitos, pelo fato de que a classificação pode ser realizada com até dez funções distintas, e o rótulo de cada instância poderá variar de acordo com a função utilizada.

4.1.2 *LED*

LED (BREIMAN et al., 1984) é um gerador que constrói um fluxo de dados com um total de vinte e quatro atributos binários categóricos, dos quais sete são relevantes. A tarefa acima do fluxo de dados é prever o dígito mostrado em um visor *LED* de sete segmentos onde cada atributo tem 10% de possibilidades de estar invertido. Uma mudança de conceito pode ser inserida indicando o número d de atributos com mudanças. Este conjunto de dados foi proposto no livro *CART* e uma implementação em linguagem de programação *C* foi doada para o UCI (ASUNCION; NEWMAN, 2007) *Machine Learning Repository* por David Aha.

4.1.3 *Mixed*

Mixed (GAMA et al., 2004) gera um fluxo de dados onde as instâncias contêm quatro atributos, dos quais dois são booleanos (v, w) e dois numéricos (x, y). As instâncias são classificadas como positivas quando ao menos duas das três seguintes condições forem cumpridas: os valores de v e w devem ser verdadeiros, e $y < 0.5 + 0.3 \times \sin(3\pi x)$; caso

contrário, são consideradas negativas. As mudanças de conceitos podem ser introduzidas com a inversão dos rótulos, onde, para que uma instância seja apontada como positiva, pelo menos duas das seguintes condições têm que ser cumpridas: os valores de v e w deverão ser falsos e $y \geq 0,5 + 0,3 \times \sin(3\pi x)$.

4.1.4 *Sine*

O gerador *Sine* (GAMA et al., 2004; SANTOS et al., 2014; HIDALGO; MACIEL; BARROS, 2019) constrói fluxos de dados que contêm unicamente dois atributos relevantes (x e y). Todos os atributos têm valores uniformemente distribuídos no intervalo $[0, 1]$. *Sine1* e *Sine2* são os dois possíveis conceitos existentes nos fluxos de dados criados pelo gerador. No primeiro contexto, todos os pontos abaixo da curva $y = \sin(x)$ são classificados como positivos. Já no segundo contexto, a inequação $y < 0,5 + 0,3\sin(3\pi x)$ deve ser verdadeira para que uma determinada instância seja considerada como positiva. Para ambos os contextos acima mencionados, as mudanças de conceitos podem ser introduzidas com a inversão das condições ou pela alternância entre os contextos *Sine1* e *Sine2*.

4.1.5 *Waveform*

O *Waveform* (BIFET et al., 2010) é um gerador de fluxos de dados apresentado por Breiman et al. (1984). Existem duas versões disponíveis: a primeira, conhecida como *Wave21*, dispõe de vinte e um atributos numéricos com ruído, e a segunda, chamada de *Wave40*, traz dezenove atributos a mais que a primeira versão, mas os mesmos são irrelevantes e introduzidos para incluir ruído. O objetivo nestes fluxos gerados é diferenciar entre três classes diferentes de ondas que são resultado de uma combinação de duas ou três ondas. As mudanças de conceito podem ser introduzidas alterando a posição de d atributos. Todas as instâncias são geradas com ruído (média 0, variância 1). A taxa de erro de *Bayes* pode ser utilizada para derivar uma expressão analítica da sequência, sabendo que o erro *Bayes* ideal é 14% para uma amostra de teste de tamanho 5000.

4.1.6 *RandomRBF*

RandomRBF (BIFET et al., 2009) gera um fluxo de dados, onde os m atributos de cada exemplo criado são numéricos, levando em consideração : 1) a posição do centro (previamente criado) que, quanto maior for seu peso, maior será a probabilidade de ser selecionado; e 2) um número aleatório no intervalo de $(-1, +1)$ multiplicado por um valor baseado na distribuição Gaussiana e no desvio padrão (SANTOS; BARROS, 2020). Os diferentes centroides (classes) são criados de forma aleatória, que são usados para definir funções de base radial. As instâncias serão criadas controladas por hiperesferas, e vão cair perto de qualquer um dos centroides escolhidos. A distância ao centroide é alcançada de forma aleatória de uma distribuição normal cujo desvio padrão é definido pelo centroide. As mudanças de conceitos podem ser introduzidas no fluxo de dados com o deslocamento de k centroides no espaço a uma velocidade (s).

4.1.7 *SEA*

SEA foi proposto por Street e Kim (2001). Os conjuntos de dados criados por este gerador são formados com a possibilidade de as instâncias serem classificadas em duas classes, com base em três atributos numéricos. Desses, dois contêm informações relevantes, enquanto o terceiro é utilizado apenas como ruído. Os valores de cada atributo são gerados aleatoriamente, variando de 0 a 10.

As bases de dados geradas por *SEA* utilizam um limiar (θ) com 4 possíveis valores: 7, 8, 9 e 9,5. Uma determinada instância irá pertencer à classe 1, caso $f_1 + f_2 \leq \theta$, onde: f_1 e f_2 são o primeiro e o segundo atributo de cada exemplo. O gerador *SEA* introduz as mudanças de conceito nas bases através da atribuição de diferentes valores à θ.

4.2 BASE DE DADOS REAIS

Nesta pesquisa foram utilizadas onze bases de dados reais, que podem ser encontradas, em sua maioria, no *UCI Machine Learning Repository*[1] (FRANK; ASUNCION, 2010). A escolha das mesmas foi feita levando em consideração a diversidade do tamanho assim como a quantidade de atributos e classes. Já informações sobre quais os tipos de mu-

[1] <http://archive.ics.uci.edu/ml/index.php>

danças de conceito ocorrem (abruptas, graduais, recorrentes), ou quantas ocorrem, ou a quantidade de instâncias pertencentes a cada conceito nas bases reais não são descritas na documentação oficial das bases, nem são investigadas nesta pesquisa.

4.2.1 *Airlines*

Airlines (SANTOS; BARROS; GONÇALVES JR., 2015) é uma base de dados binária que possui um total de 539.383 instâncias. O propósito nesta base é predizer se um determinado voo sofrerá atraso ou não, baseado em sete atributos com informação do voo: nome da companhia aérea, número do voo, aeroportos de saída e de chegada, dia da semana, hora de partida e duração do voo.

4.2.2 *Connect_4*

Connect_4 (BARROS; HIDALGO; CABRAL, 2018) é uma base de dados constituída por 42 atributos e 67.557 instâncias. A base de dados contém todas as combinações possíveis permitidas no jogo *Connect_4* onde nenhum dos jogadores ganhou ainda e nas quais o próximo movimento não é forçado. A base de dados tem três classes (*win*, *loss*, *draw*) e não tem valores faltantes.

4.2.3 *Covertype*

Covertype (SANTOS et al., 2014) é uma base de dados que contém as células de dados de 30×30 metros da Região 2 do Serviço Florestal dos Estados Unidos. A base de dados contém 581.012 instâncias e 54 atributos, tanto numéricos como categóricos, e o objetivo é prever o tipo de cobertura da floresta. Na pesquisa foi utilizada uma versão (*CovertypeSorted*) apresentada por Ienco et al. (2013) onde a base é ordenada pelo atributo de elevação, o que induz mudanças de conceitos graduais na distribuição da classe: dependendo da elevação, alguns tipos de vegetação desaparecem, enquanto outros começam a aparecer.

4.2.4 *Spam_data*

Spam_data [2] é uma base de dados binária construída por Katakis et al. (KATAKIS; TSOUMAKAS; VLAHAVAS, 2010) com base nas mensagens da coleção de *e-mails Spam Assassin*. Ela consiste em 9.324 instâncias e 500 atributos, uma lista de palavras derivadas após a seleção de atributos.

4.2.5 *Weather*

Weather (SANTOS; BARROS, 2020) contém 18.154 leituras meteorológicas diárias do site da Administração Atmosférica e Oceânica Nacional (NOAA), obtidas ao longo de 50 anos pela Base Aérea Offutt em Bellevue, Nebraska, Estados Unidos. A base contém oito atributos, usados para determinar o clima de cada dia: temperatura, nível do mar, visibilidade, velocidade média e máxima do vento, temperatura máxima e mínima, etc. O objetivo é prever se chove ou não.

4.2.6 *Electricity*

Electricity (GAMA et al., 2004) contém dados que foram coletados do mercado de eletricidade de *New South Wales*, na Austrália. Nesse mercado, os preços não são fixos e são afetados pela demanda e oferta do mercado, sendo definidos a cada cinco minutos. O conjunto de dados *Electricity* contém 45.312 instâncias. O rótulo da classe identifica a variação do preço em relação à média móvel das últimas 24 horas.

4.2.7 *KDD99Joined*

A *KDD99Joined* (FRANK; ASUNCION, 2010) encontra-se disponível em *UCI Machine Learning Repository*. Esta é uma versão da base de dados original *KDD Cup 99*, usada na 3^{rd} *International Knowledge Discovery and Data Mining Tools Competition*. As 148.561 instâncias em *KDD99Joined* representam uma conexão de rede, que possui 41 atributos de entrada cujos valores podem ser discretos ou contínuos. O valor possível das classes são 23, das quais 22 significam ataque e a restante que tudo está normal (PERVEZ; FARID,

[2] <http://mlkd.csd.auth.gr/concept_drift.html>

2014), pelo que na pesquisa foi usada uma versão da base de dados que só contém duas classes (ataque e normal).

4.2.8 *Sick*

A base de dados *Sick* (QUINLAN, 1986) foi doada para *UCI Machine Learning Repository* em 1986 pelo *Garavan Institute e J. Ross Quinlan*, do *New South Wales Institute* em Sydney, Austrália. *Sick* é formada por dados provenientes da triagem de doenças da tireoide, e contém 3.772 registros com 29 atributos descrevendo os dados médicos dos pacientes. Esses registros estão divididos em duas classes: 231 pacientes são diagnosticados como doentes (Classe C1 – positiva), enquanto 3.541 são classificados como saudáveis (Classe C2 – negativa). O objetivo é classificar o paciente como doente ou saudável.

4.2.9 *Usenet1*

O *Usenet1* (ASUNCION; NEWMAN, 2007) é baseado em uma coleção de 20 grupos de notícias, formando um fluxo de 1500 instâncias, dividido em cinco períodos de tempo, cada um contendo 300 instâncias. Após a conclusão de cada período, ocorre uma mudança de conceito. O fluxo simula mensagens de diferentes grupos de notícias apresentadas sequencialmente a um usuário, que as classifica como interessantes (+) ou lixo (-), conforme seus interesses pessoais. O *Usenet1* é um conjunto de dados diversificado, no qual todas as categorias alteram-se de classe em cada período. Além disso, a base de dados apresenta mudanças de conceito abruptas e recorrentes.

4.2.10 *WineWhite / WineRed*

O conjunto de dados *Wine Quality*[3] (CORTEZ et al., 2009) é dividido em duas partes: *WineRed*, com 1.599 amostras de vinho tinto (vermelho), e *WineWhite*, com 4.898 amostras de vinho branco, ambos provenientes da região norte de Portugal. Em ambas as bases de dados, cada instância é descrita por 11 atributos físico-químicos, que influenciam diretamente as características sensoriais dos vinhos. O *WineWhite* inclui um décimo segundo atributo que identifica a cor. A variável-alvo em ambas as bases de dados é a qualidade do

[3] <http://www3.dsi.uminho.pt/pcortez/wine/>

vinho, avaliada em uma escala de 0 (muito ruim) a 10 (excelente), com base na mediana de pelo menos três avaliações realizadas por especialistas em vinhos.

As informações de todas as bases de dados utilizadas na pesquisa foram resumidas para uma análise mais eficiente na Tabela 1. As bases de dados marcadas com * na tabela correspondem às que foram usadas exclusivamente na experimentação em cenários de aprendizado supervisionado e descritas no artigo (PÉREZ; MARIÑO; BARROS, 2021).

Tabela 1 – Principais características das bases de dados artificiais e reais

Tipo	**Bases**	**#Instâncias**	**#Atributos**	**#Classes**
Artificial	*Agrawal*	20K, 50K, 100K	9	2
	LED	20K, 50K, 100K	24	10
	Mixed	20K, 50K, 100K	4	2
	Sine	20K, 50K, 100K	2	2
	Waveform	20K, 50K, 100K	40	3
	RandomRBF	20K, 50K, 100K	40	6
	SEA*	10K, 50K, 100K	3	2
Real	*Airlines*	539383	7	2
	Connect4	67557	42	3
	Covertype	581012	54	7
	Spam_data	4601	57	2
	Weather	18159	8	2
	*Electricity**	45312	8	2
	*Kdd99Joined**	148517	41	2
	*Sick**	3772	29	2
	*Usenet1**	1500	99	2
	*WineRed**	1599	11	9
	*WineWhite**	4898	12	9

4.3 CONFIGURAÇÃO DA EXPERIMENTAÇÃO

Esta seção apresenta os detalhes das configurações experimentais, com ênfase nas configurações gerais e comuns aplicadas a todos os testes. Destaca-se que a pesquisa concentra-se em experimentos em cenários semi-supervisionados com a presença de mudanças de conceito. No entanto, experimentos em ambientes supervisionados também foram realizados com o objetivo de investigar parâmetros de diversidade em comitês de classificadores, cujos resultados são apresentados no Capítulo 6 desta tese.

4.3.1 Descrição da experimentação

Os experimentos foram rodados na ferramenta *MOAManager*, que está integrada tanto ao MOA-SS (que disponibiliza o método BDF) quanto com o *MOAManager-SS*, versão modificada do MOA para ambientes semi-supervisionados. As características do computador *desktop* onde foi realizada a experimentação são: processador I9-9900k; 32GB de memória RAM DDR4; SSD de 256GB; e sistema operacional Ubuntu 18.10 cosmic LTS 64*bits*.

Os geradores de bases de dados artificiais apresentados na Seção 4.1 foram utilizados na experimentação com bases contendo 20 mil, 50 mil e 100 mil instâncias. A única exceção foi na experimentação realizada em ambientes supervisionados, onde foram utilizadas 10 mil instâncias em vez de 20 mil. Essa alteração no tamanho das bases entre cenários ocorreu devido a sugestões de revisores após a publicação do artigo (PÉREZ; MARIÑO; BARROS, 2021). Quando os resultados das experimentações semi-supervisionadas foram apresentados no artigo publicado em (PÉREZ; BARROS; SANTOS, 2023), alguns revisores recomendaram o aumento do tamanho da menor base, argumentando que, por se tratar de fluxos de dados, bases com 10 mil instâncias eram consideradas pequenas. Os demais tamanhos foram mantidos para não aumentar o custo de recursos computacionais e de tempo, considerando que os experimentos para cada gerador são realizados 30 vezes sobre uma configuração, alterando só um parâmetro aleatório conhecido como semente. Os resultados médios da acurácia foram computados para cada configuração e apresentados como resultado.

A escolha dos geradores utilizados na pesquisa baseou-se em sua ampla utilização por diversos autores em investigações relacionadas à aprendizagem em fluxos de dados com mudanças de conceito (FRÍAS-BLANCO et al., 2016; BARROS et al., 2017; HIDALGO; MACIEL; BARROS, 2019). Além disso, foram empregadas as onze bases de dados reais descritas na Seção 4.2. Em ambos os cenários experimentais, as bases *Spam_data* e *Weather* foram utilizadas em comum. Em cenários de aprendizado supervisionado, onde todos os dados contêm rótulos, o resultado de um modelo de classificação é o mesmo em cada iteração acima dos exemplos da base e, portanto, o resultado da acurácia é baseado em uma única iteração. Uma observação a ser realizada é que, nos experimentos com bases reais em cenários semi-supervisionados, o resultado da acurácia é baseado em 30 iterações, e a cada iteração a posição das instâncias cujos rótulos são retirados é diferente.

A exclusão de diversas bases inicialmente utilizadas nos experimentos supervisionados ao conduzir os experimentos em cenários semi-supervisionados, foco desta pesquisa, deveu-se principalmente ao seu tamanho reduzido. Adicionalmente, essas bases apresentavam limitações, como a ausência de variação no número de classes e uma baixa quantidade de atributos, características que impactariam negativamente na análise e nos resultados.

As bases usadas na experimentação para cenários semi-supervisionados foram geradas versões com 15%, 25% e 30% de rótulos nas instâncias, com o objetivo de ter um alto número de cenários disponíveis para teste. Os resultados com 25% e com 100% (cenário supervisionado) dos rótulos são apresentados nos anexos A e B, respectivamente.

4.3.2 Critérios de avaliação

Na pesquisa, a metodologia *Prequential* foi usada para avaliar a precisão dos modelos de aprendizagem. Nesta metodologia, o modelo é avaliado com cada nova instância de entrada, que depois é usada para treinamento (HIDALGO; MACIEL; BARROS, 2019). É importante destacar que a metodologia *Prequential* possui três variações principais: o fator de desvanecimento (*Fading factors*), a janela básica (*Basic window ou Interleaved test-then-train*) e a janela deslizante (*Sliding window*) (GAMA et al., 2014; LEMAIRE; SALPERWYCK; BONDU, 2015; HIDALGO; MACIEL; BARROS, 2019), que foi utilizada nesta pesquisa.

A acurácia *Prequential* (DAWID, 1984; DAWID; VOVK, 1999) é a métrica usada pelas variações, e o cálculo da mesma no tempo t é definido pela equação 4.1 (BAENA-GARCIA et al., 2006). Onde acc_{ex} é 1 se a instância atual for corretamente classificada 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 (HIDALGO; MACIEL; BARROS, 2019).

$$acc(t) = \begin{cases} acc_{ex}(t), & \text{if } t = f \\ acc(t-1) + \frac{acc_{ex}(t) - acc(t-1)}{t-f+1}, & otherwise. \end{cases} \tag{4.1}$$

Observa-se que no início da avaliação, os métodos têm grande probabilidade de ter resultados ruins, já que o modelo ainda não está bem treinado. Também se faz válido mencionar que a metodologia *Prequential* pode ser usada independentemente das sequências de dados apresentarem mudanças de conceito ou não.

Os resultados de acurácia obtidos ao longo da experimentação, tanto em bases de dados artificiais quanto reais, são apresentados em forma de tabelas na tese. Essa abordagem destaca de forma clara as informações sobre os métodos de classificação utilizados, bem como os tipos de mudanças de conceito avaliados (abruptas e graduais), no caso das bases artificiais. Para uma melhor compreensão do comportamento dos algoritmos, em todas as tabelas apresentadas, além da acurácia, são definidos também os *ranks* médios, i.e. suas posições comparativas na avaliação. É importante mencionar que os melhores valores do *rank* e da acurácia aparecem escritos em **negrito** nas tabelas.

Para avaliar estatisticamente os resultados dos experimentos, foi utilizada uma variação do teste não paramétrico de *Friedman*, o teste F_F (DEMSAR, 2006). O teste *Friedman* é usado para determinar se a diferença nos *ranks* média observada das métricas fornecidas pelos algoritmos é estatisticamente significativa. Além disso, a rejeição da hipótese nula do teste *Friedman* indica que as diferenças observadas entre os métodos são globalmente estatisticamente significativas, mas não define quais métodos são estatisticamente diferentes. Além disso, para definir explicitamente as diferenças estatísticas encontradas, foi utilizado o pós-teste de *Nemenyi* (NEMENYI, 1963; DEMSAR, 2006). O pós-teste foi aplicado para realizar comparações múltiplas utilizando os valores das porcentagens das médias de cada abordagem avaliada em todas as versões dos conjuntos de dados executados em relação aos seus respectivos valores de acurácia. Os resultados dos testes de *Nemenyi* são apresentados graficamente usando diagramas simples e a Diferença Crítica (CD) é representada por barras conectando os métodos que são estatisticamente semelhantes. Além disso, cada diagrama possui uma régua horizontal que marca os *ranks* de todos os métodos analisados. Os testes foram realizados com um nível de significância de 5%.

4.4 CONSIDERAÇÕES FINAIS

O capítulo apresenta uma descrição detalhada das bases de dados utilizadas na experimentação, englobando tanto bases artificiais quanto reais. As bases artificiais foram criadas para explorar cenários controlados e destacar comportamentos específicos dos métodos, enquanto as bases reais representam desafios encontrados em aplicações do mundo real. Essa combinação foi fundamental para garantir uma avaliação abrangente e diversificada dos métodos investigados.

Adicionalmente, foram descritas as configurações experimentais adotadas. O objetivo dessas configurações foi assegurar reprodutibilidade e permitir análises consistentes e comparativas entre os métodos. Para avaliar o desempenho dos algoritmos, foi empregada a acurácia.

Por fim, a análise estatística dos resultados foi conduzida utilizando os testes de *Friedman* e *Nemenyi*. O teste de *Friedman* permitiu verificar se existiam diferenças estatisticamente significativas entre os algoritmos em relação à métrica avaliada, enquanto o teste de *Nemenyi* foi aplicado para identificar quais métodos apresentaram desempenhos significativamente distintos. Esses testes estatísticos foram essenciais para garantir que as conclusões apresentadas fossem robustas e respaldadas por análises matemáticas confiáveis.

5 DETECTORES DE MUDANÇAS DE CONCEITO EM CENÁRIOS SEMI-SUPERVISIONADOS

O presente capítulo busca demonstrar a primeira hipótese levantada nesta tese, a qual sustenta que a integração de detectores de mudanças de conceito com métodos de classificação favorece a adaptação às alterações na distribuição dos dados, aprimorando o desempenho em cenários de aprendizado semi-supervisionado.

5.1 CONFIGURAÇÕES

Para fins experimentais, foram selecionados quatro detectores supervisionados: DDM, HDDM$_A$, FHDDM e RDDM. Foram utilizados também três métodos criados especificamente para o aprendizado semi-supervisionado: DMDDM-S, BDF e DSDD. Adicionalmente, com o objetivo de diversificar ainda mais a experimentação, uma versão extra do BDF configurada com o detector de mudanças de conceito RDDM e chamada nesta tese de (BDF$_{RDDM}$) foi testada. A seleção dos detectores utilizados levou em consideração a popularidade e eficiência dos mesmos. O objetivo dos detectores de mudanças de conceito é contribuir para a não degradação do desempenho do modelo de aprendizagem.

Os parâmetros de configuração usados na experimentação para todos os métodos foram os fornecidos por seus respectivos autores. Além disso, todos os detectores foram testados em conjunto com as versões dos classificadores base HT e NB disponíveis no MOA. Neste ponto, é válido mencionar que o DMDDM-S usava um classificador *Perceptron* o qual foi substituído pelo NB, para dar melhor coerência aos experimentos e depois de demonstrar melhores resultados frente à versão original.

O desempenho dos métodos selecionados avaliados em termos de acurácia nos experimentos usando o classificador HT nas bases de dados artificiais são mostrados nas Tabelas 2 e 3, para as mudanças abruptas e graduais, respectivamente. Os resultados correspondentes com a utilização do classificador NB são apresentados nas Tabelas 4 e 5. Finalmente, o desempenho dos métodos nas bases de dados reais com os classificadores HT e NB é exibido nessa ordem, nas Tabelas 6 e 7.

Um ponto a ressaltar é a utilização do gerador *Agrawal*, onde as dez funções que ele contém foram separadas em dois conjuntos de cinco funções na geração das bases de dados: a primeira base é formada unicamente com as cinco primeiras funções (F1 a F5) e

a segunda base utiliza as funções F6 a F10. Uma explicação detalhada do gerador *Agrawal* pode ser encontrada na Seção 4.1.1.

5.2 ANÁLISE GERAL DA ACURÁCIA

A análise dos resultados nas Tabelas 2 e 3 mostra que os detectores desenvolvidos para o aprendizado supervisionado são efetivos, mesmo em cenários onde somente 15% dos rótulos são conhecidos. A afirmação anterior pode ser constatada com a comparação dos resultados dos quatro métodos voltados para a aprendizagem supervisionada (DDM, HDDM$_A$, FHDDM e RDDM) com aqueles três métodos específicos para a aprendizagem semi-supervisionada (DMDDM-S, BDF e DSDD): na maioria das situações, os métodos supervisionados tiveram melhor desempenho. O comportamento de desempenho superior dos métodos supervisionados frente aos semi-supervisionados em ambientes com ausência parcial de rótulos é observado anteriormente em (MAHDI et al., 2020), mesmo que não abordado diretamente pelos autores.

Outro ponto a ser ressaltado é que, nos experimentos realizados com mais rótulos (30% e 100%), a acurácia tem um incremento, como já era esperado. Este comportamento pode ser associado a dois motivos: o primeiro é ter mais treinamento, pois o classificador receberá mais instâncias com rótulos, e o segundo é que as detecções de mudanças de conceito ocorrerão mais cedo, sendo, portanto, mais rápidas e precisas.

Desta forma, uma conclusão natural que pode ser obtida levando-se em consideração esses resultados é que ter menos rótulos acessíveis atrasa as detecções. Isso ocorre pois, com menos instâncias rotuladas, menos retorno os detectores terão para acusar uma mudança. No entanto, apesar de ocorrerem mais lentamente, mudanças de conceito são eventualmente detectadas e auxiliam na recuperação do classificador, influenciando positivamente na acurácia final. A relevância dos detectores fica mais clara analisando as duas versões do BDF: no segundo (BDF$_{RDDM}$), a utilização do RDDM melhorou os resultados em quase todos os cenários. É importante ressaltar que a acurácia pôde ser calculada porque os rótulos reais de todas as instâncias eram conhecidos, mesmo para aquelas consideradas semi-supervisionadas, cujos rótulos foram removidos durante os experimentos para simular um cenário semi-supervisionado.

Em relação aos detectores usados com HT, o RDDM se destacou: sua precisão foi a melhor ou próxima da melhor na maioria dos conjuntos de dados. Uma característica

que provavelmente foi benéfica para o RDDM é sua maior sensibilidade na detecção de mudanças de conceito. Embora esse aspecto possa causar mais falsos positivos, em um contexto onde poucos rótulos estão disponíveis, as mudanças de conceito provavelmente serão identificadas mais rapidamente.

Para completar a análise dos resultados obtidos usando HT como classificador base, a Tabela 6 apresenta as acurácias dos métodos nas bases de dados reais. Em geral, além de mostrar os mesmos padrões presentes nos conjuntos de dados artificiais, o ganho de desempenho do BDF_{RDDM} chama a atenção e sugere que a junção de métodos semi-supervisionados com detectores de mudanças de conceito pode render bons resultados.

Por fim, vale a pena mencionar que os resultados correspondentes dos testes usando NB como classificador base, apresentados nas Tabelas 4, 5 e 7, foram bastante semelhantes aos obtidos utilizando HT. Por este motivo, as observações feitas para os resultados com HT também são válidas para os resultados com NB.

5.2.1 Avaliação estatística

A forma como os *ranks* são definidos para a comparação de acurácia, assim como sua representação gráfica, é detalhada na Seção 4.3.2.

Para realizar a análise estatística, inicialmente as classificações gerais foram calculadas para toda a configuração do experimento. Em termos absolutos, é possível observar que, nos experimentos usando os classificadores base e os conjuntos de dados artificiais, RDDM, HDDM_A e FHDDM foram os três melhores métodos, enquanto BDF_{RDDM} teve o melhor desempenho dentre os algoritmos semi-supervisionados.

Por outro lado, a análise dos resultados das classificações nos testes usando HT com os conjuntos de dados do mundo real mostra FHDDM, BDF_{RDDM} e RDDM como os três melhores métodos, nesta ordem. Vale a pena acrescentar que, nos testes usando NB como classificador base, os *ranks* dos métodos FHDDM e RDDM foram iguais e melhores que o *rank* de BDF_{RDDM}.

Os resultados dos testes de *Nemenyi* são apresentados graficamente nas Figuras 4 e 5, que representam os resultados dos métodos testados com HT como classificador base nos conjuntos de dados com mudanças abruptas e graduais, respectivamente. Nos conjuntos de dados abruptos, RDDM foi o melhor método, embora sem diferença estatística para HDDM_A e FHDDM, e com apenas algumas diferenças estatísticas para outros mé-

Tabela 2 – Médias de acurácias em (%) usando o classificador HT, com 95% de intervalo de confiança nas bases de dados artificiais com mudanças abruptas.

%RÓT.	TAM.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$	DSDD
15%	20K	Agrawal (F1-F5)	59.84±0.46	60.73±0.34	61.02±0.36	61.07±0.39	53.64±0.22	57.97±0.43	**62.32±0.39**	51.03±1.10
		Agrawal (F6-F10)	71.44±1.85	78.40±0.84	**78.89±1.05**	77.59±1.93	70.52±0.16	61.34±0.50	68.50±1.28	66.57±2.61
		LED	64.55±0.50	64.34±0.47	63.75±0.50	**64.95±0.33**	38.27±0.73	46.85±0.36	61.71±0.41	35.39±3.56
		Mixed	87.44±0.21	**88.20±0.22**	87.45±0.21	87.88±0.23	60.99±0.21	63.45±0.37	87.69±0.26	57.00±0.74
		Sine	84.55±0.59	85.43±0.34	85.00±0.27	**85.48±0.32**	62.94±0.23	61.79±0.55	84.81±0.40	60.20 ±0.86
		Waveform	77.59±0.45	78.03±0.39	78.34±0.38	**78.66±0.29**	58.01±0.94	76.28±0.39	77.72±0.42	58.37 ±2.46
		RandomRBF	31.42±0.49	31.40±0.52	30.38±0.52	31.07±0.46	21.82±0.26	28.90±1.20	**33.92±0.52**	26.52±1.38
	50K	Agrawal (F1-F5)	61.50±0.73	63.32±0.40	63.58±0.34	63.67±0.37	53.19±0.12	59.85±0.63	**64.07±0.34**	54.24±1.05
		Agrawal (F6-F10)	77.56±1.96	82.39±0.59	**83.63±0.52**	82.77±0.97	71.37±0.15	64.36±0.37	71.51±0.58	70.37±0.80
		LED	69.63±0.18	69.68±0.18	69.43±0.28	**69.81±0.16**	38.81±0.68	50.01±0.91	67.98±0.36	36.81±5.22
		Mixed	89.95±0.59	**90.74±0.22**	90.62±0.19	90.69±0.20	61.23±0.16	65.00±0.54	89.76±0.21	63.50±0.91
		Sine	86.92±0.55	**88.33±0.15**	88.23±0.14	87.92±0.17	62.97±0.12	62.65±0.43	86.50±0.29	66.91±0.97
		Waveform	78.99±0.25	79.17±0.21	79.25±0.26	**79.58±0.22**	57.84±0.62	77.02±0.27	79.09±0.25	62.43±1.57
		RandomRBF	31.11±0.51	31.59±0.39	30.79±0.48	31.38±0.38	21.67±0.14	29.85±1.85	**35.88±0.50**	25.00±1.36
	100K	Agrawal (F1-F5)	65.25±0.86	66.66±0.45	66.09±0.66	**66.89±0.40**	53.52±0.17	61.60±0.44	66.36±0.31	55.55±0.66
		Agrawal (F6-F10)	82.62±0.51	84.62±0.30	**84.98±0.24**	84.51±0.37	71.62±0.09	65.79±0.32	73.09±0.30	72.78±1.20
		LED	71.09±0.20	71.18±0.15	71.06±0.30	**71.57±0.14**	38.37±0.36	50.62±0.83	70.59±0.20	27.41±4.39
		Mixed	90.47±0.21	91.06±0.10	**91.06±0.09**	90.96±0.11	61.20±0.11	65.98±0.65	90.76±0.17	71.13±1.08
		Sine	88.71±0.19	**89.50±0.13**	89.49±0.11	89.04±0.16	63.04±0.11	63.77±0.54	88.16±0.21	75.33±0.64
		Waveform	79.07±0.18	79.66±0.18	79.66±0.15	**79.71±0.12**	58.40±0.44	77.42±0.18	79.52±0.16	61.34±1.41
		RandomRBF	31.86±0.37	32.02±0.34	31.05±0.23	31.86±0.32	21.81±0.10	30.89±1.54	**36.91±0.39**	23.12±0.74
		Rank 15%	4.12	2.24	2.95	**2.07**	7.33	6.52	3.67	7.10
30%	20K	Agrawal (F1-F5)	61.05±0.46	62.44±0.47	62.76±0.37	62.81±0.38	59.99±0.26	59.18±0.25	**63.94±0.29**	56.22±1.05
		Agrawal (F6-F10)	72.66±1.80	80.05±0.43	**81.54±0.51**	80.64±1.04	78.55±0.54	63.19±0.31	69.56±0.98	72.58±1.19
		LED	68.25±0.42	68.27±0.27	67.98±0.35	**68.47±0.23**	55.79±0.28	50.59±0.44	61.85±0.49	51.16±5.63
		Mixed	89.22±0.22	89.17±1.17	89.51±0.21	**89.65±0.22**	58.97±0.86	65.32±0.53	87.80±0.27	73.83±1.13
		Sine	86.19±0.54	**87.22±0.18**	87.07±0.16	87.08±0.19	68.60±1.12	64.13±0.37	85.22±0.32	77.80±0.83
		Waveform	78.48±0.34	78.71±0.30	78.58±0.44	**79.28±0.29**	76.74±0.22	76.96±0.28	78.83±0.32	74.73±0.77
		RandomRBF	31.14±0.44	**31.58±0.45**	31.13±0.46	31.37±0.40	30.02±0.58	28.95±1.34	31.23±0.51	27.07±1.05
	50K	Agrawal (F1-F5)	64.36±1.11	66.83±0.42	66.08±0.48	**66.87±0.40**	61.90±0.31	61.19±0.21	66.28±0.27	59.78±0.69
		Agrawal (F6-F10)	82.23±0.89	84.43±0.21	**84.50±0.29**	84.24±0.30	79.87±0.30	66.39±0.40	72.61±0.37	81.07±0.87
		LED	71.16±0.16	71.26±0.18	70.90±0.27	**71.46±0.15**	57.47±0.73	53.00±1.15	63.26±0.30	60.08±5.49
		Mixed	90.31±0.36	**90.99±0.13**	90.93±0.11	90.87±0.12	67.36±1.27	65.56±0.42	88.55±0.21	85.29±0.58
		Sine	88.74±0.23	**89.41±0.13**	89.37±0.12	88.88±0.16	82.30±0.60	64.03±0.45	86.06±0.16	85.61±0.33
		Waveform	79.03±0.25	79.53±0.22	79.58±0.22	**79.76±0.20**	78.11±0.14	77.44±0.24	79.49±0.28	76.57±0.38
		RandomRBF	31.57±0.38	**32.17±0.28**	31.13±0.30	31.88±0.27	30.80±0.49	30.71±1.45	31.45±0.29	26.05±1.08
	100K	Agrawal (F1-F5)	67.16±1.49	**70.27±0.34**	69.23±0.55	70.26±0.27	62.21±0.16	59.55±0.15	68.30±0.27	60.40±0.66
		Agrawal (F6-F10)	83.20±0.92	85.30±0.37	**86.08±0.16**	85.39±0.38	81.29±0.17	66.08±0.59	73.41±0.39	83.19±0.28
		LED	72.01±0.40	72.54±0.14	71.91±0.20	**72.66±0.13**	63.65±0.19	54.71±0.87	63.48±0.25	64.66±4.90
		Mixed	90.38±0.41	91.25±0.09	**91.26±0.08**	91.08±0.09	80.95±0.35	67.74±0.94	88.84±0.14	88.63±0.38
		Sine	90.12±0.08	90.64±0.09	**90.62±0.08**	90.25±0.15	83.25±0.36	64.52±0.45	86.76±0.14	87.65±0.19
		Waveform	79.29±0.20	79.69±0.17	79.81±0.15	**79.88±0.12**	79.05±0.10	78.39±0.19	79.83±0.15	77.00±0.20
		RandomRBF	32.16±0.39	**32.46±0.22**	31.13±0.16	32.35±0.21	32.27±0.28	30.66±1.32	31.69±0.24	25.09±1.04
		Rank 30%	4.10	2.10	2.95	**1.86**	6.19	7.52	4.62	6.67
		Rank	4.11	2.17	2.95	**1.96**	6.76	7.02	4.14	6.88

todos, dependendo da porcentagem de rótulos (Figuras 4(a)-4(c)). Também vale a pena apontar que, apenas nos conjuntos de dados totalmente rotulados (100%) (Figura 20(a)), BDF$_{RDDM}$ está classificado à frente de DDM. Além disso, os resultados nos conjuntos de dados graduais (Figuras 5(a)-5(c)) foram muito semelhantes aos dos conjuntos de dados abruptos.

Os resultados com a utilização do classificador NB são muito semelhantes aos apresentados usando HT na maioria dos cenários testados, o que pode ser observado nas Figuras 6 e 7, referentes aos testes com bases de dados artificiais com mudanças de conceito abruptas e graduais, respectivamente. Nos cenários testados com mudanças abruptas, mais uma vez os métodos RDDM, HDDM$_A$ e FHDDM são os três melhores, nessa ordem, em todos os

Tabela 3 – Médias de acurácias em (%) usando o classificador HT, com 95% de intervalo de confiança nas bases de dados artificiais com mudanças graduais.

%RÓT.	TAM.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$	DSDD
15%	20K	Agrawal (F1-F5)	59.31±0.30	60.13±0.30	60.43±0.35	60.62±0.37	53.40±0.20	57.95±0.41	**61.92±0.31**	51.82±1.14
		Agrawal (F6-F10)	70.76±1.69	74.98±1.32	**77.26±1.18**	76.15±2.01	69.98±0.18	61.33±0.47	67.47±1.04	66.60±2.63
		LED	63.77±0.58	63.29±0.41	63.22±0.45	**64.13±0.38**	37.21±0.47	47.26±0.52	60.88±0.38	32.98±3.61
		Mixed	85.98±0.16	86.14±0.16	85.99±0.15	**86.23±0.16**	60.14±0.19	63.80±0.31	85.65±0.30	56.81±0.70
		Sine	83.45±0.47	83.88±0.24	83.68±0.25	**83.99±0.25**	62.03±0.17	62.08±0.56	83.31±0.33	59.07±0.79
		Waveform	77.15±0.39	77.48±0.45	77.54±0.39	**78.15±0.33**	57.79±0.87	75.94±0.36	77.22±0.38	58.49±2.52
		RandomRBF	31.18±0.51	31.24±0.56	30.28±0.62	31.14±0.49	21.66±0.16	28.86±1.31	**33.87±0.69**	26.47±1.38
	50K	Agrawal (F1-F5)	61.20±0.72	62.86±0.46	63.33±0.36	63.66±0.40	53.21±0.11	59.73±0.60	**64.04±0.32**	54.10±0.87
		Agrawal (F6-F10)	76.30±1.86	81.63±0.50	**82.32±0.56**	81.62±1.37	71.17±0.12	64.40±0.31	70.98±0.66	70.33±0.98
		LED	69.50±0.19	69.34±0.21	69.12±0.21	**69.66±0.19**	38.08±0.43	49.68±0.69	67.36±0.36	32.00±4.81
		Mixed	89.39±0.16	**89.52±0.13**	89.49±0.13	89.46±0.14	60.77±0.14	64.86±0.47	88.77±0.21	62.55±1.04
		Sine	87.06±0.20	**87.39±0.15**	87.29±0.14	87.24±0.16	62.57±0.13	62.89±0.37	86.05±0.20	66.81±1.13
		Waveform	78.83±0.24	78.86±0.15	78.96±0.27	**79.38±0.22**	58.32±0.67	77.13±0.27	78.93±0.27	61.27±1.98
		RandomRBF	31.63±0.42	31.81±0.38	31.14±0.43	31.61±0.36	21.70±0.15	29.11±1.80	**36.09±0.44**	25.62±1.05
	100K	Agrawal (F1-F5)	64.29±1.03	66.62±0.48	65.78±0.54	**66.80±0.42**	53.48±0.15	61.60±0.48	66.15±0.35	55.89±0.69
		Agrawal (F6-F10)	82.69±0.39	**84.50±0.25**	84.70±0.24	84.52±0.25	71.52±0.07	65.94±0.37	73.02±0.58	72.24±0.89
		LED	71.29±0.15	71.25±0.14	70.88±0.18	**71.63±0.14**	38.43±0.32	50.72±0.92	70.37±0.16	25.38±3.81
		Mixed	90.46±0.15	90.71±0.07	**90.72±0.07**	90.69±0.07	61.00±0.10	65.10±0.31	90.44±0.15	71.44±1.07
		Sine	88.72±0.11	**89.12±0.09**	89.11±0.08	88.93±0.11	62.83±0.09	63.70±0.56	87.71±0.21	75.74±0.77
		Waveform	79.04±0.21	79.52±0.20	79.50±0.17	**79.83±0.14**	58.33±0.58	77.43±0.18	79.49±0.15	61.74±1.15
		RandomRBF	31.50±0.47	31.99±0.29	31.10±0.23	31.80±0.31	21.67±0.09	30.65±1.49	**36.73±0.34**	23.48±0.92
		Rank 15%	3.90	2.43	2.86	**2.05**	7.38	6.43	3.86	7.10
30%	20K	Agrawal (F1-F5)	60.57±0.36	61.52±0.36	62.30±0.34	62.27±0.41	59.90±0.27	58.96±0.27	**63.41±0.30**	56.22±1.15
		Agrawal (F6-F10)	71.69±1.81	78.94±0.72	**79.23±0.75**	79.12±1.40	78.42±0.47	63.03±0.32	68.69±1.01	71.90±1.30
		LED	67.81±0.26	67.30±0.30	66.88±0.36	**67.85±0.22**	55.92±0.26	50.54±0.39	60.92±0.41	53.56±5.16
		Mixed	86.93±0.22	87.12±0.19	**87.15±0.18**	86.94±0.20	59.28±0.85	65.05±0.44	85.85±0.29	70.70±0.99
		Sine	84.86±0.26	85.18±0.21	**85.19±0.20**	85.00±0.20	68.12±1.43	63.87±0.39	83.45±0.29	77.17±0.78
		Waveform	78.11±0.30	78.10±0.33	78.27±0.30	**78.60±0.26**	76.74±0.23	76.73±0.31	78.10±0.34	74.53±0.97
		RandomRBF	31.35±0.47	**31.73±0.43**	30.63±0.40	31.31±0.39	30.22±0.52	29.01±1.35	31.28±0.49	27.06±1.09
	50K	Agrawal (F1-F5)	64.22±1.06	**66.48±0.37**	65.99±0.47	66.39±0.35	61.64±0.30	61.12±0.18	65.82±0.23	59.61±0.42
		Agrawal (F6-F10)	81.94±0.98	84.04±0.34	**84.07±0.27**	83.98±0.31	79.93±0.29	66.19±0.50	72.52±0.39	81.05±0.80
		LED	71.02±0.17	70.91±0.17	70.51±0.24	**71.27±0.15**	56.87±0.46	53.01±0.96	63.06±0.32	58.62±5.98
		Mixed	89.75±0.14	**89.92±0.10**	**89.92±0.09**	89.83±0.12	66.97±0.95	65.38±0.33	87.74±0.17	84.84±0.52
		Sine	88.24±0.21	88.55±0.12	**88.57±0.11**	88.39±0.14	81.69±0.49	63.97±0.42	85.57±0.22	85.26±0.29
		Waveform	78.90±0.21	79.27±0.22	79.18±0.17	**79.55±0.19**	78.09±0.12	77.47±0.22	79.37±0.24	76.49±0.39
		RandomRBF	31.59±0.44	**32.03±0.28**	31.10±0.28	31.95±0.32	30.67±0.56	31.43±1.31	31.42±0.26	25.95±0.95
	100K	Agrawal (F1-F5)	67.35±1.55	69.91±0.34	68.93±0.58	**70.12±0.32**	62.28±0.19	59.43±0.14	68.19±0.27	61.16±0.85
		Agrawal (F6-F10)	83.39±0.79	85.00±0.31	**85.86±0.13**	85.21±0.40	81.11±0.17	66.00±0.54	73.31±0.49	82.91±0.24
		LED	72.13±0.25	72.28±0.15	71.75±0.17	**72.60±0.13**	63.71±0.18	55.08±0.95	63.23±0.22	61.64±5.46
		Mixed	90.40±0.25	90.72±0.07	**90.73±0.07**	90.67±0.07	81.05±0.37	66.91±0.58	88.56±0.15	88.09±0.34
		Sine	90.09±0.10	90.20±0.09	**90.23±0.09**	90.10±0.12	82.98±0.36	64.61±0.54	86.50±0.16	87.50±0.27
		Waveform	79.46±0.18	79.68±0.18	79.76±0.16	**79.90±0.12**	79.05±0.10	78.39±0.15	79.85±0.13	76.89±0.25
		RandomRBF	31.97±0.32	**32.27±0.22**	31.05±0.17	32.12±0.20	32.18±0.31	30.76±1.36	31.73±0.27	24.88 ±1.15
		Rank 30%	3.90	2.33	2.64	**2.14**	6.10	7.43	4.74	6.71
		Rank	3.90	2.38	2.75	**2.10**	6.74	6.93	4.30	6.90

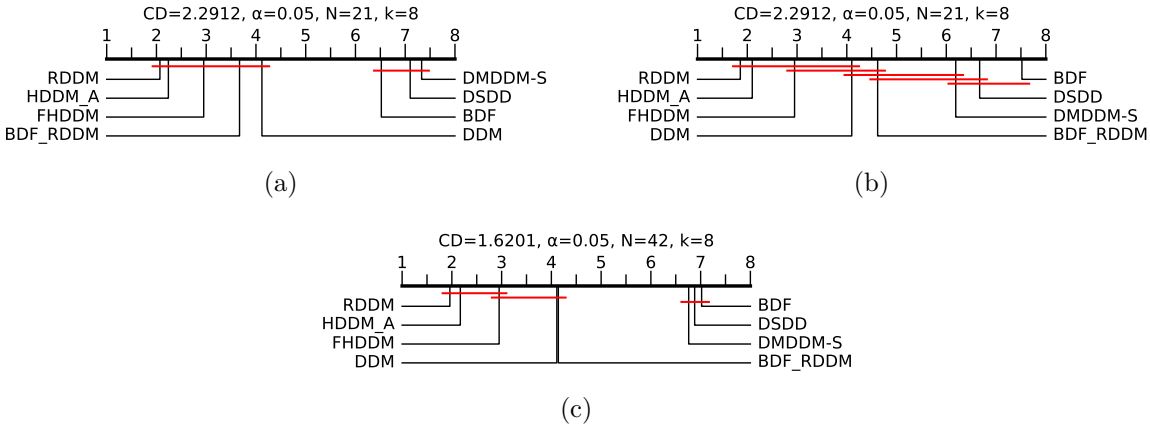


Figura 4 – Comparação estatística das acurácias de todos os métodos em todos os cenários do número de rótulos: (a) 15%, (b) 30% e (c) todos, através do Teste F_F e do Pós-Teste *Nemenyi*, com 95% de confiança, usando o classificador HT e bases de dados artificiais com mudanças abruptas.

Tabela 4 – Médias de acurácias em (%) usando o classificador NB, com 95% de intervalo de confiança nas bases de dados artificiais com mudanças abruptas.

%RÓT.	TAM.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$	DSDD
15%	20K	Agrawal (F1-F5)	58.30±0.51	58.96±0.54	59.76±0.49	**60.06±0.53**	53.64±0.22	56.40±0.55	58.51±0.60	53.55±1.09
		Agrawal (F6-F10)	68.75±0.87	77.00±0.63	**77.87±1.13**	74.50±2.01	70.52±0.16	60.88±0.50	66.74±1.01	63.27±2.19
		LED	64.53±0.52	64.36±0.47	63.80±0.49	**64.96±0.33**	38.27±0.73	46.75±0.38	61.75±0.43	41.30±0.74
		Mixed	87.52±0.21	**88.29±0.20**	87.53±1.19	87.98±0.21	60.99±0.21	63.29±0.34	87.75±0.28	58.56±0.47
		Sine	84.21±0.49	**85.08±0.27**	84.66±0.27	**85.08±0.32**	62.94±0.23	61.81±0.34	84.90±0.40	60.49±0.83
		Waveform	77.65±0.46	78.00±0.41	78.36±0.38	**78.64±0.32**	58.01±0.94	76.33±0.38	77.78±0.41	61.09±1.70
		RandomRBF	30.15±0.49	30.04±0.48	29.24±0.51	29.72±0.44	21.82±0.26	29.06±1.19	**34.05±0.55**	27.17±1.21
	50K	Agrawal (F1-F5)	60.88±0.61	62.52±0.44	62.63±0.37	**62.91±0.35**	53.19±0.12	57.67±0.30	61.80±0.40	53.21±1.11
		Agrawal (F6-F10)	73.94±1.36	**81.30±0.48**	83.17±0.40	80.25±1.15	71.37±0.15	61.76±0.29	68.84±0.85	65.75±2.08
		LED	69.63±0.18	69.69±0.19	69.46±0.28	**69.81±0.16**	38.81±0.68	49.12±0.39	68.02±0.35	45.96±2.00
		Mixed	89.74±0.92	**90.81±0.19**	90.67±0.18	90.74±0.18	61.23±0.16	63.95±0.34	89.84±0.19	58.13±0.49
		Sine	84.53±1.05	86.71±0.19	**86.82±0.16**	86.72±0.18	62.97±0.12	62.14±0.18	86.05±0.26	61.91±1.84
		Waveform	79.00±0.25	79.18±0.20	79.26±0.26	**79.59±0.22**	57.84±0.62	77.05±0.26	79.13±0.26	58.52±1.39
		RandomRBF	30.00±0.51	29.77±0.55	29.10±0.51	29.48±0.47	21.67±0.14	29.80±1.92	**35.70±0.45**	25.70±1.19
	100K	Agrawal (F1-F5)	62.44±0.62	64.65±0.18	64.08±0.23	**64.69±0.14**	53.52±0.17	58.70±0.27	63.93±0.23	52.19±1.33
		Agrawal (F6-F10)	78.72±1.14	82.94±0.56	**84.30±0.29**	83.07±0.66	71.62±0.09	62.71±0.21	71.89±0.51	66.96±1.48
		LED	71.09±0.20	71.22±0.15	71.09±0.30	**71.58±0.14**	38.37±0.36	50.42±0.24	70.61±0.20	53.37±1.50
		Mixed	90.51±0.20	91.10±0.10	**91.12±0.07**	91.02±0.08	61.20±0.11	63.65±0.51	90.74±0.15	57.73±0.62
		Sine	84.69±1.02	86.88±0.17	**87.01±0.15**	86.87±0.15	63.04±0.11	62.40±0.15	86.77±0.15	62.44±1.85
		Waveform	79.14±0.17	79.81±0.18	79.72±0.17	**79.84±0.12**	58.40±0.44	77.24±0.16	79.54±0.16	61.62±1.57
		RandomRBF	30.51±0.49	30.19±0.45	29.26±0.37	30.10±0.41	21.81±0.10	29.06±1.86	**36.71±0.40**	26.28±1.09
		Rank 15%	4.17	2.40	2.79	**2.07**	7.05	6.38	3.86	7.29
30%	20K	Agrawal (F1-F5)	60.20±0.63	61.70±0.43	62.10±0.43	**62.16±0.36**	59.99±0.26	57.42±0.27	61.60±0.34	55.93±0.89
		Agrawal (F6-F10)	71.13±1.09	79.73±0.59	**81.02±0.65**	78.06±1.49	78.55±0.54	61.85±0.33	66.86±1.24	67.28±2.07
		LED	68.26±0.43	68.30±0.27	68.03±0.35	**68.50±0.24**	55.79±0.28	50.35±0.34	61.88±0.48	60.38±0.75
		Mixed	89.02±0.63	89.27±1.15	89.59±0.17	**89.73±0.20**	58.97±0.86	64.13±0.17	87.91±0.28	74.64±0.75
		Sine	84.20±1.10	**86.31±0.30**	86.16±0.27	86.17±0.29	68.60±1.12	62.47±0.20	85.02±0.24	76.84±0.58
		Waveform	78.53±0.33	78.73±0.30	78.59±0.44	79.26±0.30	76.74±0.22	**80.00±0.28**	78.86±0.31	74.81±1.00
		RandomRBF	30.14±0.45	30.16±0.49	29.66±0.49	29.72±0.44	30.02±0.58	29.70±1.33	**31.54±0.46**	26.93±1.18
	50K	Agrawal (F1-F5)	62.37±0.65	64.53±0.31	64.26±0.24	**64.59±0.21**	61.90±0.31	58.69±0.24	64.05±0.26	54.79±1.20
		Agrawal (F6-F10)	77.27±1.69	83.04±0.58	**84.14±0.33**	83.01±0.60	79.87±0.30	62.76±0.20	70.58±0.68	71.38±1.69
		LED	71.16±0.16	71.28±0.18	70.92±0.27	**71.46±0.15**	57.47±0.73	51.64±0.19	63.28±0.29	68.92±0.34
		Mixed	90.36±0.24	**91.01±0.13**	90.96±0.11	90.89±0.12	67.36±1.27	64.21±0.34	88.60±0.20	80.10±0.69
		Sine	84.39±1.43	86.95±0.18	**87.06±0.17**	86.93±0.18	82.30±0.60	62.29±0.14	85.37±0.16	81.24±0.40
		Waveform	79.07±0.26	79.59±0.26	79.66±0.23	**79.87±0.22**	78.11±0.14	77.22±0.24	79.54±0.28	76.63±0.56
		RandomRBF	30.60±0.37	30.51±0.40	29.61±0.42	30.22±0.32	30.80±0.49	30.49±1.59	**31.72±0.28**	25.61±1.10
	100K	Agrawal (F1-F5)	63.29±0.70	65.30±0.17	64.79±0.18	**65.31±0.18**	62.21±0.16	59.12±0.18	64.82±0.22	55.23±1.10
		Agrawal (F6-F10)	81.99±0.59	84.67±0.38	**85.20±0.22**	85.03±0.27	81.29±0.17	63.40±0.15	72.85±0.38	73.50±1.38
		LED	72.02±0.38	72.53±0.14	71.93±0.23	**72.66±0.13**	63.65±0.19	52.34±0.22	63.55±0.27	70.81±0.23
		Mixed	90.32±0.83	91.45±0.08	**91.54±0.07**	91.43±0.07	80.95±0.35	64.72±0.25	88.70±0.13	79.79±0.88
		Sine	83.63±q.71	87.10±0.10	**87.22±0.10**	87.11±0.10	83.25±0.36	62.45±0.09	85.49±0.11	82.00±0.30
		Waveform	79.65±0.22	80.14±0.15	80.13±0.15	**80.16±0.11**	79.05±0.10	77.52±0.14	79.90±0.15	73.23±0.23
		RandomRBF	31.21±0.45	30.82±0.37	29.43±0.27	30.53±0.34	32.27±0.28	28.51±1.74	**31.96±0.22**	25.40±1.07
		Rank 30%	4.29	2.43	3.05	**2.33**	5.52	7.10	4.38	6.90
		Rank	4.23	2.42	2.92	**2.20**	6.29	6.74	4.12	7.10

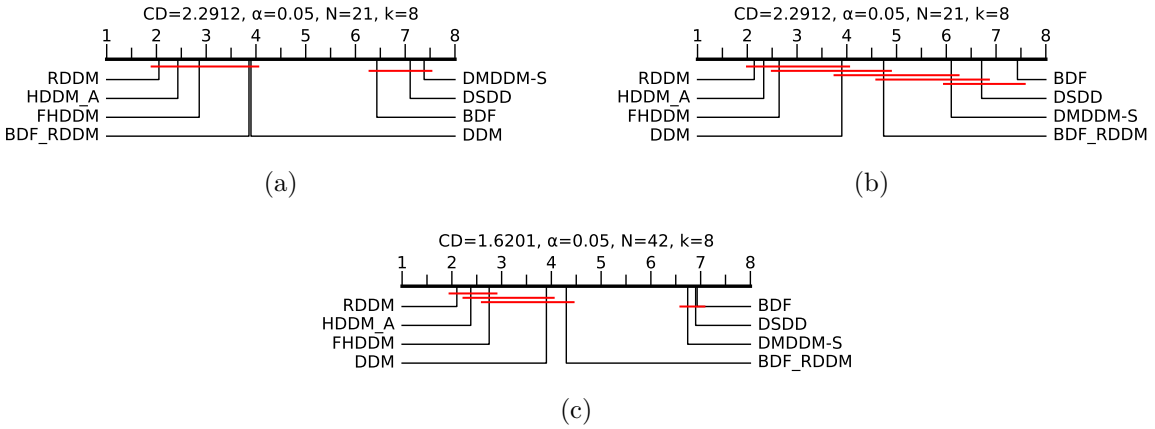


Figura 5 – Comparação estatística das acurácias de todos os métodos em todos os cenários do número de rótulos: (a) 15%, (b) 30% e (c) todos, através do Teste F_F e do Pós-Teste *Nemenyi*, com 95% de confiança, usando o classificador HT e bases de dados artificiais com mudanças graduais.

Tabela 5 – Médias de acurácias em (%) usando o classificador NB, com 95% de intervalo de confiança nas bases de dados artificiais com mudanças graduais.

%RÓT.	TAM.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$	DSDD
15%	20K	Agrawal (F1-F5)	57.60±0.44	58.28±0.50	59.05±0.39	**59.20±0.46**	53.40±0.20	56.34±0.54	58.23±0.47	52.95±1.21
		Agrawal (F6-F10)	69.07±1.22	74.07±1.16	**75.13±1.30**	71.96±1.65	69.98±0.18	60.86±0.57	64.83±1.24	63.22±2.05
		LED	63.65±0.63	63.27±0.47	63.28±0.45	**64.20±0.35**	37.21±0.47	47.20±0.36	61.03±0.42	41.19±0.80
		Mixed	86.08±0.16	86.25±0.15	86.06±0.14	**86.35±0.15**	60.14±0.19	63.67±0.30	85.78±0.29	58.08±0.38
		Sine	83.23±0.26	83.49±0.21	83.38±0.20	**83.63±0.25**	62.03±0.17	62.37±0.35	83.39±0.34	60.08±1.02
		Waveform	77.27±0.38	77.41±0.45	77.56±0.39	**78.11±0.34**	57.79±0.87	75.98±0.36	77.29±0.56	60.33±1.43
		RandomRBF	29.85±0.55	29.82±0.61	29.14±0.60	29.60±0.51	21.66±0.16	29.02±1.30	**33.86±0.65**	27.16±1.20
	50K	Agrawal (F1-F5)	60.49±0.61	62.38±0.47	62.73±0.33	**63.01±0.32**	53.21±0.11	57.65±0.31	61.64±0.33	54.05±1.13
		Agrawal (F6-F10)	71.13±1.61	80.05±0.47	**81.71±0.61**	77.87±1.42	71.17±0.12	61.88±0.27	68.44±1.03	65.79±1.95
		LED	69.50±0.20	69.40±0.19	69.16±0.21	**69.68±0.19**	38.08±0.43	49.21±0.35	67.40±0.35	46.24±1.56
		Mixed	89.54±0.13	**89.63±0.12**	89.60±0.11	89.59±0.13	60.77±0.14	63.88±0.29	88.92±0.20	58.85±0.88
		Sine	85.38±0.64	86.11±0.17	86.14±0.18	**86.22±0.19**	62.57±0.13	62.34±0.22	85.69±0.23	60.76±1.22
		Waveform	78.85±0.24	78.85±0.27	78.97±0.27	**79.39±0.22**	58.32±0.67	77.16±0.27	78.96±0.27	59.95±1.73
		RandomRBF	30.09±0.40	30.09±0.41	29.45±0.43	29.87±0.33	21.70±0.15	29.93±1.92	**35.87±0.44**	26.03±0.97
	100K	Agrawal (F1-F5)	62.61±0.59	64.28±0.23	63.94±0.29	**64.42±0.21**	53.48±0.15	58.79±0.25	63.72±0.27	52.71±1.23
		Agrawal (F6-F10)	76.52±1.50	81.93±0.90	**83.49±0.52**	81.66±0.95	71.52±0.07	62.89±0.20	71.32±0.54	62.25±2.38
		LED	71.26±0.18	71.27±0.13	70.90±0.18	**71.63±0.14**	38.43±0.32	50.46±0.25	70.39±0.17	52.86±1.33
		Mixed	90.49±0.15	**90.75±0.07**	**90.75±0.07**	90.72±0.07	61.00±0.10	63.59±0.54	90.49±0.15	57.55±0.66
		Sine	85.59±0.45	86.51±0.14	86.62±0.12	**86.67±0.13**	62.83±0.09	62.36±0.14	86.49±0.17	60.99±1.48
		Waveform	79.13±0.20	79.62±0.21	79.47±0.20	**79.94±0.13**	58.33±0.58	77.26±0.18	79.52±0.14	60.38±1.60
		RandomRBF	30.34±0.45	30.18±0.36	29.50±0.30	30.01±0.29	21.67±0.09	29.04±1.88	**36.61±0.35**	25.85±1.08
		Rank 15%	3.98	2.55	2.88	**1.95**	6.95	6.38	3.98	7.33
30%	20K	Agrawal (F1-F5)	59.64±0.51	60.48±0.34	61.49±0.32	**61.69±0.38**	59.90±0.27	57.39±0.34	61.01±0.40	54.82±1.16
		Agrawal (F6-F10)	69.84±1.38	75.50±1.72	77.79±0.99	75.64±1.51	**78.42±0.47**	61.74±0.34	65.94±1.19	67.17±1.98
		LED	67.82±0.26	67.33±0.30	66.92±0.36	**67.87±0.22**	55.92±0.26	50.42±0.31	61.01±0.40	59.65±0.82
		Mixed	87.11±0.21	87.25±0.18	**87.26±0.17**	87.08±0.19	59.28±0.85	64.13±0.18	85.93±0.31	73.07±0.80
		Sine	83.47±0.72	84.35±0.21	**84.36±0.20**	84.31±0.23	68.12±1.43	62.62±0.23	83.14±0.27	76.28±0.51
		Waveform	78.08±0.31	78.06±0.35	78.29±0.30	**78.60±0.26**	76.74±0.23	76.76±0.31	78.12±0.34	74.34±1.38
		RandomRBF	30.32±0.55	30.21±0.47	29.39±0.51	29.88±0.43	30.22±0.52	29.12±1.34	**31.49±0.45**	26.77±1.22
	50K	Agrawal (F1-F5)	62.54±0.63	64.22±0.17	63.81±0.20	**64.33±0.14**	61.64±0.30	58.70±0.18	63.75±0.19	55.26±1.14
		Agra (F6-F10)	76.28±1.84	81.30±0.59	**83.21±0.37**	80.88±0.90	79.93±0.29	62.85±0.21	69.28±0.69	70.07±2.15
		LED	71.02±0.17	70.94±0.17	70.54±0.24	**71.26±0.14**	56.87±0.46	51.68±0.21	63.12±0.32	68.54±0.29
		Mixed	89.81±0.14	89.98±0.10	**90.01±0.09**	89.88±0.11	66.97±0.95	64.35±0.33	87.73±0.19	78.70±0.80
		Sine	85.24±0.97	86.34±0.17	86.36±0.16	**86.41±0.15**	81.69±0.49	62.33±0.13	84.81±0.21	80.99±0.41
		Waveform	78.99±0.23	79.29±0.24	79.16±0.16	**79.67±0.19**	78.09±0.12	77.80±0.22	79.39±0.24	76.47±0.46
		RandomRBF	30.66±0.40	30.52±0.37	29.36±0.32	30.39±0.33	30.67±0.56	31.38±1.38	**31.68±0.24**	26.14±1.07
	100K	Agrawal (F1-F5)	63.05±0.72	65.14±0.18	64.65±0.18	**65.21±0.17**	62.28±0.19	59.08±0.18	64.70±0.20	54.02±1.47
		Agrawal (F6-F10)	78.89±1.46	83.90±0.99	**84.63±0.31**	83.68±0.82	81.11±0.17	63.43±0.17	72.70±0.44	72.96±1.67
		LED	72.13±0.25	72.28±0.15	71.77±0.17	**72.60±0.13**	63.71±0.18	52.27±0.22	63.27±0.25	70.67±0.20
		Mixed	90.92±0.11	91.04±0.08	**91.06±0.07**	91.00±0.08	81.05±0.37	64.65±0.26	88.44±0.16	79.08±0.73
		Sine	85.69±1.08	86.94±0.13	86.97±0.12	**87.02±0.10**	82.98±0.36	62.43±0.11	85.16±0.11	81.74±0.31
		Waveform	79.63±0.25	80.05±0.16	79.90±0.18	**80.14±0.10**	79.05±0.10	77.58±0.14	79.92±0.14	77.15±0.26
		RandomRBF	30.74±0.32	30.52±0.28	29.25±0.19	30.29±0.22	**32.18±0.31**	28.48±1.72	32.00±0.24	25.48±1.04
		Rank 30%	4.00	2.90	2.95	**2.33**	5.33	7.24	4.33	6.90
		Rank	3.99	2.73	2.92	**2.14**	6.14	6.81	4.15	7.12

Tabela 6 – Acurácias (%) usando o classificador HT nas bases de dados reais.

%RÓT.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$	DSDD
15%	Airlines	63.18	63.87	65.01	64.77	65.13	64.10	**66.35**	58.74
	Connect4	70.09	70.52	70.43	70.77	69.81	70.57	**73.55**	65.87
	Covertype	75.50	**78.50**	78.13	78.42	73.72	73.21	78.01	66.26
	Spam_data	**90.27**	89.64	89.23	89.57	89.85	84.54	89.99	76.67
	Weather	68.69	68.72	68.81	68.88	69.09	**69.94**	68.94	67.92
	Rank 15%	5.20	4.20	4.40	3.40	4.00	4.60	**2.20**	8.00
30%	Airlines	63.69	64.59	65.38	64.82	65.18	67.56	**68.24**	63.17
	Connect4	71.64	72.29	72.38	72.46	70.08	70.08	**73.58**	69.25
	Covertype	78.50	79.98	**80.41**	80.24	74.19	59.74	68.07	78.76
	Spam_data	**91.01**	90.80	90.53	90.47	90.04	83.49	88.93	88.36
	Weather	70.29	69.63	70.05	69.99	69.23	69.94	**70.35**	69.76
	Rank 30%	4.00	4.40	**2.60**	3.40	5.90	5.90	3.20	6.60
	Rank	4.60	4.30	3.50	3.40	4.95	5.25	**2.70**	7.30

Tabela 7 – Acurácias (%) usando o classificador NB nas bases de dados reais.

%RÓT.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$	DSDD
15%	Airlines	64.97	65.39	65.55	65.71	64.41	**66.39**	64.78	56.23
	Connect4	69.84	70.20	70.06	70.55	68.99	70.36	**73.17**	63.58
	Covertype	74.37	**78.34**	77.77	78.04	60.52	66.46	78.24	65.08
	Spam_data	87.23	**89.13**	87.53	87.96	87.46	83.63	87.60	73.26
	Weather	67.51	68.24	68.53	68.84	67.30	69.07	**69.13**	65.49
	Rank 15%	5.60	3.00	4.00	**2.40**	6.80	3.80	2.60	7.80
30%	Airlines	65.29	66.24	65.95	66.55	65.18	**67.83**	67.25	58.91
	Connect4	71.24	72.00	72.13	72.30	70.08	69.15	**72.97**	68.39
	Covertype	76.48	80.06	**80.15**	79.90	74.19	59.86	68.58	77.76
	Spam_data	88.09	90.00	89.39	88.74	**90.04**	82.68	85.37	89.38
	Weather	68.10	69.40	70.24	69.35	69.23	69.30	**70.37**	64.95
	Rank 30%	5.80	3.00	**2.80**	3.40	5.20	5.80	3.60	6.40
	Rank	5.80	3.10	3.50	**3.00**	6.10	4.20	3.20	7.10

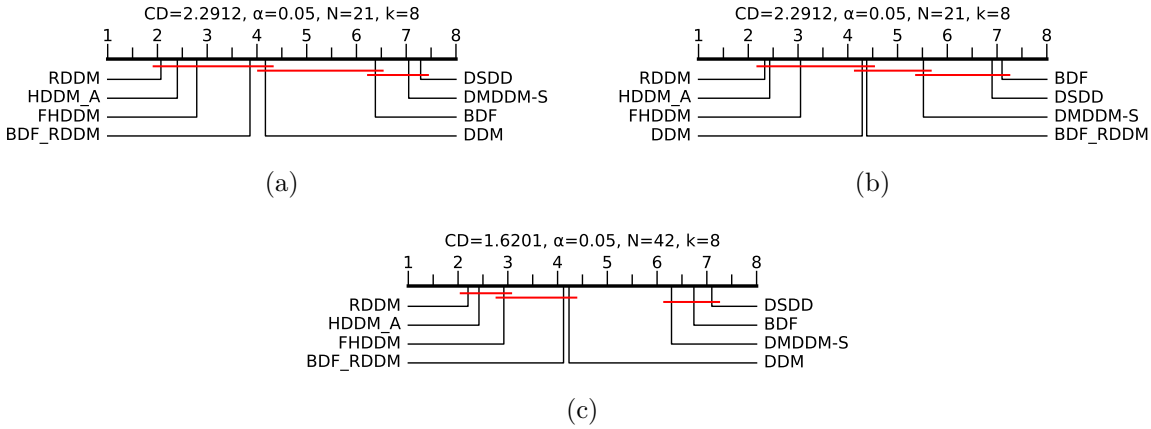


Figura 6 – Comparação estatística das acurácias de todos os métodos em todos os cenários do número de rótulos: (a) 15%, (b) 30% e (c) todos, através do Teste F_F e do Pós-Teste *Nemenyi*, com 95% de confiança, usando o classificador NB e bases de dados artificiais com mudanças abruptas.

casos. Além disso, na classificação geral, mostrada na Figura 6(c), esses três métodos são superiores estatisticamente a todos os outros métodos testados, sem apresentar diferença significativa entre eles.

Já nos cenários de experimentação com o classificador NB e mudanças graduais, os resultados são semelhantes aos das bases com mudanças abruptas, mantendo-se os mesmos três melhores métodos na maioria dos casos, com exceção do cenário com 100% dos rótulos, onde BDF$_{RDDM}$ aparece como o segundo melhor colocado, deslocando HDDM$_A$ para a próxima posição, como ilustrado na Figura 21(b). Nestes cenários, o resultado geral dos métodos, apresentado na Figura 7(c), mostra que não existe diferença significativa entre os três melhores métodos, mas que somente o RDDM é estatisticamente superior a todos os outros métodos testados.

Finalmente, a Figura 8 ilustra o desempenho dos métodos nas bases de dados reais usando HT e NB. Nos testes usando HT, capturados na Figura 8(a), os três métodos mais

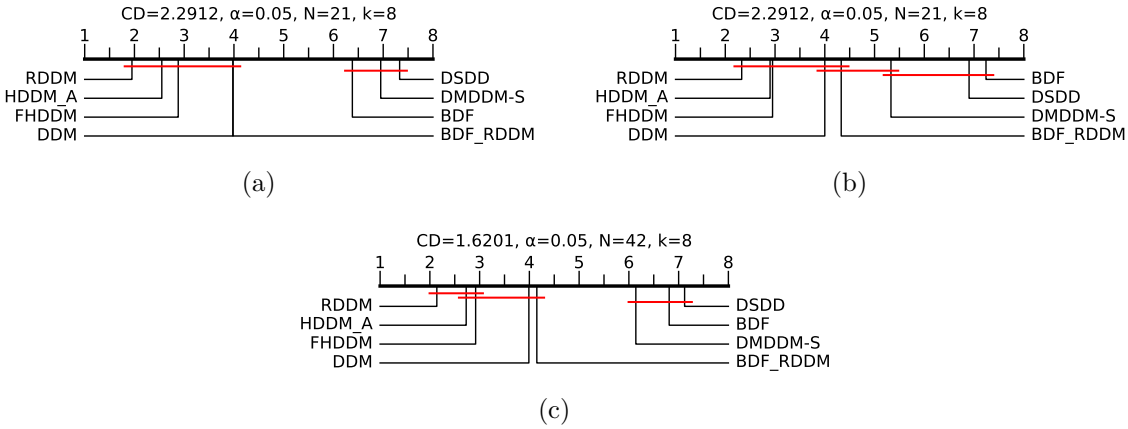


Figura 7 – Comparação estatística das acurácias de todos os métodos em todos os cenários do número de rótulos: (a) 15%, (b) 30% e (c) todos, através do Teste F_F e do Pós-Teste *Nemenyi*, com 95% de confiança, usando o classificador NB e bases de dados artificiais com mudanças graduais.

bem posicionados são: BDF_{RDDM}, RDDM e FHDDM, apresentando apenas diferença estatística significativa com o método DSDD. Já nas bases de dados reais usando NB, Figura 8(b), os resultados foram diferentes com RDDM, HDDM_A e BDF_{RDDM}, sendo os melhores métodos colocados, mas a única diferença estatística observada é que eles foram significativamente superiores a DSDD.

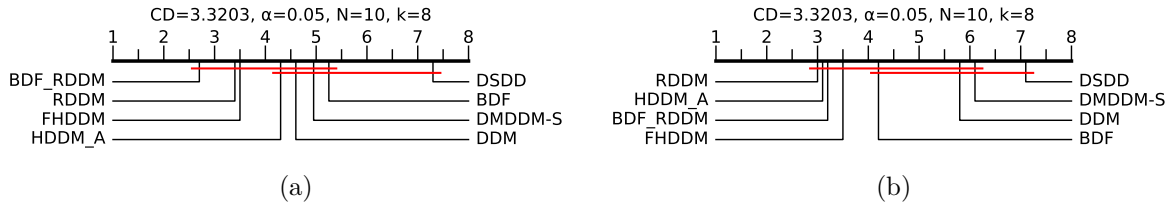


Figura 8 – Comparação estatística das acurácias de todos os métodos usando o Teste F_F e o Pós-Teste *Nemenyi*, com 95% de confiança, nas bases de dados reais utilizando os classificadores (a) HT e (b) NB.

Durante a pesquisa, a variação nos percentuais de rótulos utilizados nos testes foi ampliada. Enquanto na qualificação os resultados foram apresentados usando 25%, 30% e 100% dos rótulos, no artigo publicado (PÉREZ; BARROS; SANTOS, 2023) e nesta tese o valor 25% foi substituído por 15%, mas os resultados com 25% e com 100% dos rótulos são apresentados nos anexos A e B, respectivamente. Os resultados de performance dos métodos com a presença de 15% e 25% foram semelhantes, e confirmaram as posições dos melhores métodos nos *rankings*.

5.3 CONSIDERAÇÕES FINAIS

Neste capítulo, foram realizadas comparações empíricas e avaliações estatísticas dos métodos de detecção de mudanças de conceito utilizando a métrica de avaliação acurácia. Após uma ampla análise, foi demonstrado que os detectores criados para cenários de aprendizado supervisionado são competitivos também em situações com pouca disponibilidade de instâncias rotuladas (cenário semi-supervisionado). Essas conclusões são apoiadas nos resultados de acurácia apresentados pelos métodos de detecção de mudanças de conceito supervisionados em comparação com os métodos especificamente criados para aprendizado semi-supervisionado. É importante ressaltar que os resultados apresentados neste capítulo apareceram anteriormente em artigo (PÉREZ; BARROS; SANTOS, 2023) publicado na conferência SSCI da IEEE, realizada na Cidade do México.

6 DIVERSIDADE E DETECTORES DE MUDANÇAS DE CONCEITO EM COMITÊS DE CLASSIFICADORES

Neste capítulo, é realizada uma experimentação com comitês de classificadores em ambientes semi-supervisionados para responder à segunda hipótese formulada na pesquisa. Consequentemente, tem o objetivo de demonstrar que promover diversidade nos comitês de classificadores com a combinação de vários tipos de métodos resulta em maior robustez e capacidade de generalização, contribuindo para um desempenho superior em cenários de escassez de rótulos. Visa ainda validar se o uso de diferentes detectores de mudança de conceito pode influenciar positivamente na performance dos comitês nos cenários semi-supervisionados.

6.1 FUNDAMENTAÇÃO INICIAL

Os comitês de classificadores surgiram como uma técnica poderosa para melhorar a precisão dos modelos base (AGGARWAL, 2014). Existem dois componentes principais quando se aborda o aprendizado de comitês: 1) treinar os modelos base e 2) combinar os modelos, com a esperança de que eles atinjam um desempenho melhor (AGGARWAL, 2014; PÉREZ; MARIÑO; BARROS, 2021).

Os comitês de classificadores têm recebido grande interesse, pois demonstraram uma melhor capacidade de generalização do que os classificadores individuais. A seleção dos classificadores base envolve a identificação de um conjunto ideal de modelos (ONAN; KORUKOĞLU; BULUT, 2016), e alguns estudos revelaram que a chave para o sucesso é que os classificadores base apresentem diversidade (MINKU; YAO, 2011). Especificamente, as vantagens dos comitês para lidar com as mudanças de conceito dependem da diversidade e da adaptabilidade (AGGARWAL, 2014).

Aumentar a diversidade de um comitê envolve, entre outros aspectos, elevar a probabilidade de que seus membros recebam mais instâncias para treinamento ao longo do fluxo de dados. Esse ajuste de parâmetro pode ser particularmente benéfico em cenários com mudanças de conceito, uma vez que o aumento do treinamento auxilia o classificador a se recuperar mais rapidamente após um evento de mudança de conceito. No entanto, é importante destacar que o aumento indiscriminado do parâmetro de diversidade pode levar ao *overfitting* e, consequentemente, à redução do desempenho global (PÉREZ; BARROS;

SANTOS, 2025).

Comitês frequentemente adotam a estratégia de adaptação implícita às mudanças de conceito, onde o aprendizado é adaptado em intervalos regulares de tempo sem considerar se aconteceu uma mudança, pois possuem mecanismos que permitem sua evolução sem a necessidade de detectar diretamente os pontos de troca. No entanto, atualmente existem muitas pesquisas que inserem mecanismos para a detecção direta de mudanças de conceito nos comitês (BARROS; SANTOS, 2019), o que é considerado uma segunda estratégia. O benefício de incorporar detectores de mudanças está em aproveitar a capacidade dos comitês de se adaptarem a mudanças graduais, combinada com o trabalho do detector de mudanças para lidar com mudanças abruptas (ORTIZ-DÍAZ, 2014).

Também é conhecido que os classificadores base não precisam ser altamente precisos: enquanto houver uma quantidade boa de classificadores base, classificadores fracos podem ser fortalecidos para um classificador forte por meio da combinação. No entanto, quando cada classificador estiver extremamente errado, a combinação deles retornará resultados ainda piores (ONAN; KORUKOĞLU; BULUT, 2016). A independência entre os classificadores é outra questão importante a ser observada no conjunto de classificadores base. Se os classificadores forem altamente correlacionados e fizerem previsões muito semelhantes, a combinação deles não trará melhorias adicionais. Em contraste, quando os classificadores base são independentes e fazem previsões diversas, os erros independentes têm maiores chances de serem cancelados.

Pode-se observar que os comitês têm-se tornado cada vez mais populares no tratamento de mudanças de conceito em fluxos de dados, devido à sua eficácia e flexibilidade (HAN et al., 2023). No entanto, o uso de comitês com detectores de mudanças de conceito em ambientes semi-supervisionados ainda é pouco explorado e apresenta desafios significativos, bem como oportunidades de aprimoramento. Com base nas informações mencionadas anteriormente, este capítulo tem como objetivo demonstrar o impacto na performance do uso de diferentes detectores de mudanças de conceito e da introdução de diversidade nos comitês em cenários de aprendizado semi-supervisionado.

6.2 EXPLORAÇÃO DE PARÂMETROS DE DIVERSIDADE

Nesta seção é investigado como a diversidade e o comportamento dos classificadores base influenciam a precisão dos comitês nos ambientes supervisionados, com base

na parametrização. Embora a pesquisa seja focada em cenários semi-supervisionados, foi realizado o estudo inicial em ambientes supervisionados, baseado em dois fundamentos: 1) a maioria dos algoritmos de aprendizado para ambientes semi-supervisionados é baseada em métodos criados para aprendizado supervisionado, como pode ser observado no levantamento realizado no Capítulo 1; 2) o objetivo central da pesquisa é demonstrar a viabilidade de usar algoritmos supervisionados em ambientes semi-supervisionados. Portanto, encontrar parâmetros explorando ambientes supervisionados e aplicá-los em cenários semi-supervisionados pode ser uma opção viável.

Na exploração, propõe-se combinar diferentes tipos de classificadores base, por exemplo, 3 HT, 3 KNN e 6 NB, e selecionar os melhores membros para votar com base em seu desempenho de classificação. A votação é realizada com base em uma porcentagem parametrizada entre 25% e 100% dos classificadores com melhor precisão.

Estudos anteriores já abordaram questões de diversidade, com o objetivo de melhorar os comitês de classificadores (MINKU; YAO, 2011; LUONG et al., 2021). No entanto, esta pesquisa também se concentra em como o método proposto (Seção 6.2.1) influencia diferentes esquemas de votação e adaptação de comitês: é investigado experimentalmente o desempenho alcançado por três comitês modificados com o método proposto, a saber, FASE (FRÍAS-BLANCO et al., 2016), OACE$_{BA}$ e OACE$_{BO}$ (VERDECIA-CABRERA; FRÍAS-BLANCO; CARVALHO, 2018). Esses métodos foram escolhidos porque utilizam diferentes esquemas de votação (*stacking*, maioria simples e maioria ponderada) para realizar a classificação. Uma breve explicação desses comitês foi apresentada na Seção 2.6.

6.2.1 Metodologia de exploração de diversidade proposta

A presente tese propõe um método para aumentar a diversidade no comitê de classificadores $H(\mathbf{x})$, introduzindo diferentes tipos de classificadores base e ajustando o procedimento de votação, com base em inúmeros estudos que demonstram a força de $H(\mathbf{x})$, fundamentada em sua diversidade e adaptabilidade.

Seja $\mathbf{m} = (\mathbf{x}, y)$ uma instância e $H(\mathbf{x})$ um comitê de classificadores treinado e construído a partir de um conjunto de classificadores $h_1(\mathbf{x}), \ldots, h_j(\mathbf{x}), \ldots, h_k(\mathbf{x})$, onde h_j, $(1 \leq j \leq k)$, representa um classificador. Neste contexto, durante o processo de treinamento, o comitê recebe instâncias rotuladas ($\mathbf{m}$) como entrada e constrói o modelo $H(\mathbf{x})$ com o objetivo de prever a classe correta $\hat{y}$ de cada instância no conjunto de teste não ro-

tulado. Para realizar essa tarefa, existem diferentes esquemas de combinação de comitês, como: *Stacking*, votação por maioria simples e votação por maioria ponderada.

O presente trabalho propõe adaptar as estratégias adotadas pelos comitês estudados, com o objetivo de aumentar a diversidade do comitê, combinando diferentes tipos de classificadores. Isso introduz a possibilidade de utilizar classificadores base diferentes no mesmo modelo, como, por exemplo, os classificadores NB e HT juntos.

O Algoritmo 1 apresenta o procedimento geral de treinamento dos comitês MFASE, MOACE$_{BA}$ e MOACE$_{BO}$ obtidos após a aplicação do método proposto aos algoritmos de comitê originais.

Algoritmo 1: Procedimento geral para treinar os comitês modificados

Input: tamanho de cada lote $N(n_1, n_2, ..., n_s)$, número total de lotes s, instância de treinamento $\mathbf{x}$ do tipo $< (x_l, y_l) >$ com rótulo $y_l \in Y$, número de classificadores k, parâmetro λ para controlar a diversidade;

1 $k \leftarrow \sum_{i=1}^{s} n_i$
2 **for** $j \leftarrow 1$ **to** k **do**
3 Calcula o peso da instância atual utilizando a distribuição de *Poisson* e o parâmetro λ.
4 Inicializa os classificadores base $h(\mathbf{x})$
5 Treina os modelos baseado no valor de λ.
6 Treina o meta-classificador (apenas para o MFASE).

O algoritmo geral lê o fluxo de exemplos formado por instância $\mathbf{x}$ e constrói o conjunto composto por diferentes classificadores base. Assim como nos métodos originais, cada classificador base é composto por um classificador e um detector de mudanças. A estratégia pode ser utilizada com diferentes modelos. Além disso, o tamanho do conjunto k depende do número de classificadores base com os quais é projetado, como representado na linha 1.

A linha 3 abstrai o uso de uma distribuição de *Poisson* para criar diversidade baseado no parâmetro λ, enviando cópias de cada instância de treinamento $\mathbf{x}$ para atualizar os k classificadores base, seguindo a abordagem de *Bagging* para MOACE$_{BA}$ e MFASE, ou a abordagem de *Boosting* para MOACE$_{BO}$. A linha 5 representa o treinamento dos conjuntos usando diferentes pesos baseados em λ. Note que, no caso do MOACE$_{BO}$, o valor de λ também depende dos resultados de classificação dos modelos anteriores.

No caso do MFASE (linha 6), uma meta-instância[1] é gerada combinando as saídas dos k classificadores de primeiro nível e o rótulo de classe verdadeiro da instância $\mathbf{x}$. O

[1] Meta-instância é uma representação em que as saídas preditas pelos classificadores bases são utilizadas como novas características (entradas) para treinar um meta-classificador, mantendo-se o rótulo original como o alvo de predição.

meta-classificador é treinado com essas meta-instâncias, ou seja, as saídas previstas dos classificadores de primeiro nível são consideradas como novas características, e os rótulos originais são mantidos como os rótulos no novo conjunto de dados com base nas saídas desses classificadores (AGGARWAL, 2014). A meta-instância é construída com o mesmo tamanho e ordem das previsões que o conjunto usou durante o procedimento de votação (Algoritmo 2).

Além disso, o método também propõe modificar a forma como os conjuntos calculam o voto final, permitindo que apenas um subconjunto dos classificadores base seja selecionado com base em seu desempenho no conjunto. Quando essa opção é utilizada, ordenamos os membros do conjunto pelo seu desempenho atual e votamos com um percentual p dos classificadores $h(\mathbf{x})$, os que obtiveram os melhores resultados. O Algoritmo 2 mostra um pseudo-código abstrato correspondente à implementação realizada no *framework* MOA.

Algoritmo 2: Procedimento geral para votar na instância $\mathbf{x}$ pelos comitês modificados.

Input: comitê H, número de classificadores k, instância de treinamento $\mathbf{x}$ do tipo $< (x_l, y_l) >$ com rótulo $y_l \in Y$, porcentagem de classificadores que participam da votação p;

1 **if** $p < 100$ **then**
2 **Ordena** H **pela** acurácia **em** ordem descendente.
3 $np \leftarrow k \times p/100$
4 **for** $j \leftarrow 1$ **to** np **do**
5 Combina o classificador $h(\mathbf{j})$ para votar na instância $\mathbf{x}$.
6 **return** Maior votação combinada para $\mathbf{x}$.

No Algoritmo 2, primeiro é verificado se o percentual de classificadores para votar p, é menor que 100% do tamanho do conjunto de classificadores (linha 1). Nesse caso, os membros do conjunto são ordenados em ordem decrescente pela performance de precisão do classificador base (linha 2). Assim, a primeira posição corresponde ao classificador base com maior valor de precisão e menor taxa de erro.

Em seguida, o número de classificadores a votar np é calculado (linha 4). Após isso, os votos são combinados usando os primeiros np classificadores (linhas 5–7). Observe que a linha 6 abstrai o fato de que a votação real difere nos métodos modificados MFASE, MOACE$_{BA}$ e MOACE$_{BO}$, assim como nos métodos originais.

No MOACE$_{BA}$, o número de votos para cada rótulo é contado entre os membros do conjunto e o rótulo com mais votos é o rótulo final previsto do conjunto (AGGARWAL, 2014). Nesse caso, cada membro do conjunto é tratado como igualmente preciso e, portanto, a votação não diferencia entre eles.

No MOACE$_{BO}$ e no MFASE, um peso é atribuído a cada classificador com o objetivo de que classificadores mais precisos recebam pesos mais altos, de modo que a previsão final possa ser influenciada pelos classificadores mais precisos. Os pesos são inferidos a partir da performance dos classificadores base ou do classificador combinado.

6.2.2 Análise da Aplicação da Metodologia Proposta

Esta seção apresenta informações relevantes sobre os experimentos realizados em cenários supervisionados, com o objetivo de encontrar parâmetros de inclusão de diversidade que aumentem o desempenho dos comitês de classificadores. A configuração geral dos experimentos é apresentada no Capítulo 4; contudo, destacaremos brevemente alguns pontos a seguir.

Como os métodos comparados possuem parâmetros comuns, eles foram configurados de forma semelhante para uma avaliação justa dos resultados. Assim, dois classificadores base foram escolhidos, HT e NB, entre outros classificadores possíveis, pois são algoritmos rápidos, amplamente utilizados em ambientes com fluxos de dados (FRÍAS-BLANCO et al., 2016; BARROS et al., 2017; HIDALGO; MACIEL; BARROS, 2019) e suas implementações estão disponíveis no *framework* MOA. O número máximo de especialistas foi definido como 10.

Os classificadores em conjunto foram configurados por padrão com HDDM$_A$ como detector de mudanças. Com base em sua configuração padrão no *framework* MOA, os níveis significativos de HDDM$_A$ foram configurados com $\alpha_D = 0.001$ para mudanças e $\alpha_W = 0.005$ para avisos.

Para realizar uma avaliação adequada e comparações entre os métodos, conjuntos de dados foram construídos com mudanças de conceito controladas e conhecidas. Os conjuntos de dados sintéticos foram construídos com três tamanhos diferentes: 10K, 50K e 100K instâncias, e mudanças de conceito abruptas e graduais foram introduzidas em intervalos regulares. Para as mudanças de conceito graduais, simulamos uma janela de mudança onde a probabilidade de uma dada instância pertencer ao conceito atual ou ao novo conceito segue uma função sigmoide, como implementado no *framework* MOA.

Nos conjuntos de dados sintéticos (36) e do mundo real (8), cada algoritmo foi testado e treinado usando HT e NB. Em resumo, 44 experimentos foram realizados para avaliar o desempenho de cada método e suas variações, utilizando a métrica de acurácia.

6.2.2.1 Resultados Experimentais e Análise

Esta seção compara os algoritmos FASE, OACE$_{BA}$ e OACE$_{BO}$ com suas variações implementadas com o método proposto, em conjuntos de dados artificiais e do mundo real.

Para ilustrar o comportamento da metodologia, apresentamos os resultados dos testes usando o OACE$_{BA}$ em conjuntos de dados artificiais com mudanças graduais de conceito, incluindo o aumento da diversidade do comitê usando diferentes tipos de classificadores e a possível seleção de um subconjunto dos membros para votar, aqueles que apresentam melhor desempenho no comitê, mostrando o impacto das diferentes configurações utilizadas. Selecionamos este método porque ele usa votação por maioria simples, que é menos complexa do que a votação por maioria ponderada. A última introduz peso como outro componente a ser observado no processo.

Os gráficos na Figura 9 mostram a relação entre a porcentagem de classificadores no comitê que votam, o tipo de classificador base e a acurácia obtida por OACE$_{BA}$ e sua variação chamada MOACE$_{BA}$ nas bases de dados artificiais configuradas com mudanças graduais de conceito. Note que OACE$_{BA}$ é um caso particular de MOACE$_{BA}$ onde $p = 100\%$. Usando 10 classificadores, combinamos dois tipos de classificadores base (HT e NB) e cinco porcentagens (25%, 50%, 75%, 95% e 100%). Especificamente, os classificadores combinados nos conjuntos testados foram: 7 HT + 3 NB, 3 HT + 7 NB, 5 HT + 5 NB, e os métodos originais, 10 HT e 10 NB, respectivamente. Por fim, vale ressaltar que os resultados com mudanças abruptas foram semelhantes.

A análise das mudanças abruptas e graduais nas bases de dados criadas pelos geradores *Mixed* e *SEA*, quando são utilizados comitês configurados com todos ou a maioria dos classificadores NB, indica que os melhores resultados são alcançados utilizando 50% ou 75% dos classificadores na votação. Em *Agrawal*, (p75% e 7 HT) e Sine (p50% e p75%), as escolhas com o maior número de classificadores HT apresentam melhores resultados. Em *Waveform*, observa-se que, a partir de um limite X de classificadores envolvidos, há uma relação inversa de desempenho.

A partir da observação anterior, similar nos outros métodos estudados, para cada um, a configuração mais estável em todos os cenários foi selecionada para ser comparada com os métodos originais. Assim, a Tabela 8 apresenta as taxas de acurácia dos algoritmos originais e da variação com a melhor configuração de desempenho em conjuntos de dados

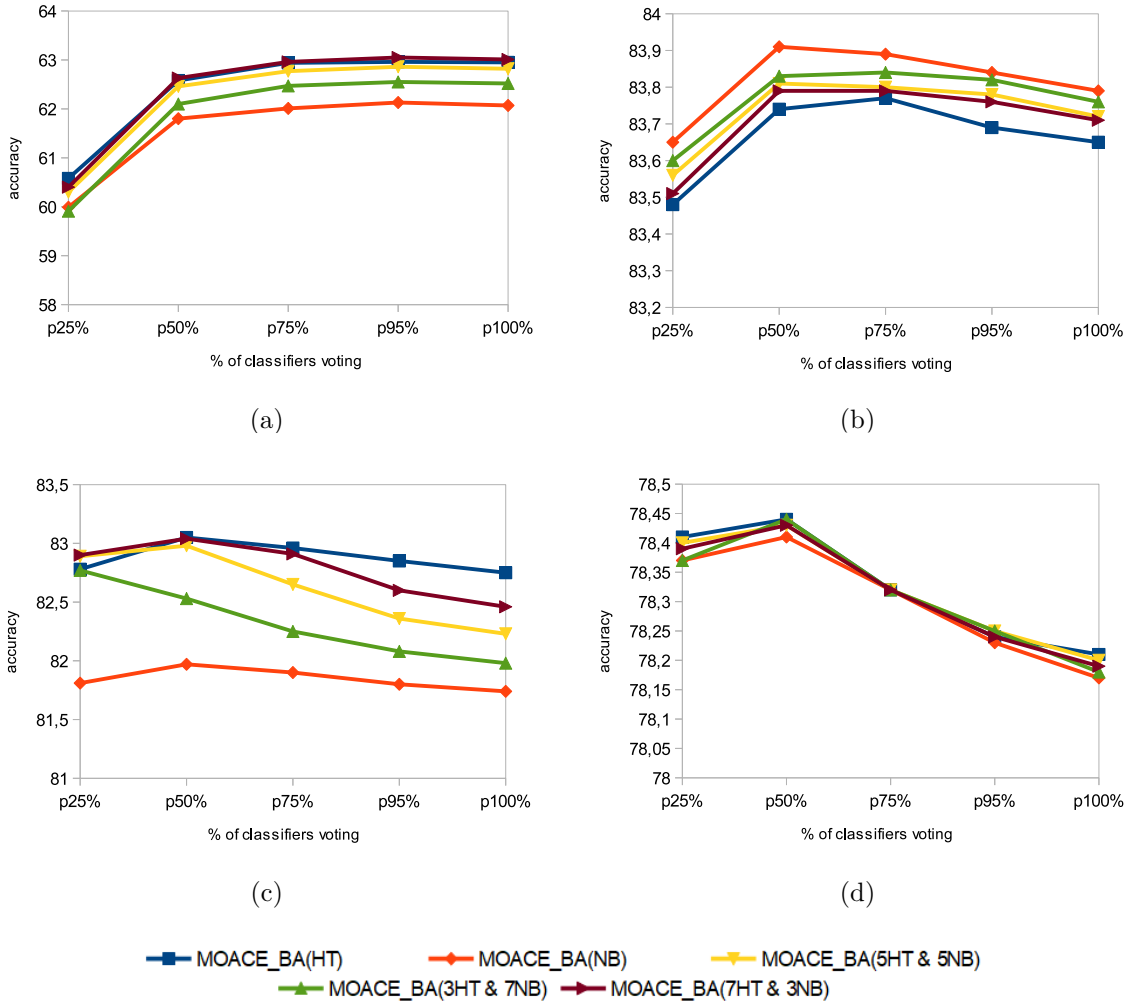


Figura 9 – Acurácias dos métodos OACE$_{BA}$ considerando a relação entre p e o tipo de classificador base: (a) Agrawal_Grad_Mudanças, (b) Mixed_Grad_Mudanças, (c) Sine_Grad_Mudanças e (d) Wave_Grad_Mudanças.

artificiais e reais, utilizando tanto NB quanto HT como classificadores base.

Os primeiros valores que aparecem em cada tabela referem-se às métricas de acurácia nos conjuntos de dados artificiais. As últimas linhas nas tabelas referem-se aos resultados com os dados reais. Entre parênteses está o classificador utilizado (NB, HT ou ambos). Os resultados que indicam melhorias em relação aos métodos originais estão em negrito. Valores mais altos de acurácia indicam melhor desempenho.

Em relação ao FASE e ao MFASE, os melhores resultados nos conjuntos de dados com mudanças abruptas foram alcançados quando o FASE é utilizado com HT: ele vence ou empata em 10 dos 18 conjuntos de dados gerados. Nos conjuntos de dados com mudanças graduais, o MFASE apresenta desempenhos semelhantes: cada um vence em 8 conjuntos de dados, embora o MFASE vença em 5 dos 6 conjuntos de dados com 50K instâncias.

Tabela 8 – Médias de acurácia em (%), com intervalos de confiança de 95%, em cenários de mudanças de conceito abruptas e graduais, utilizando bases de dados artificiais e reais com NB e HT.

Config	BASE	FASE (NB)	FASE (HT)	MFASE p50% (7HT-3NB)	OACE$_{BA}$ (NB)	OACE$_{BA}$ (HT)	MOACE$_{BA}$p95% (HT)	OACE$_{BO}$ (NB)	OACE$_{BO}$ (HT)	MOACE$_{BO}$ p95% (7HT-3NB)
	Agrawal	63.97±0.26	**64.88±0.35**	64.86±0.37	63.43±0.25	64.82±0.35	**64.89±0.35**	62.22±0.33	**62.55**±0.55	62.22±0.55
	Mixed	**89.95±0.18**	89.88±0.19	89.88±0.20	**90.22±0.21**	90.01±0.19	90.04±0.19	90.18±0.21	89.83±0.19	**90.29±0.22**
Synthetic	RandomRBF	31.58±0.37	**32.50±0.40**	32.30±0.39	30.61±0.35	31.81±0.39	**31.92±0.43**	18.21±0.40	17.95±0.34	**23.53±1.01**
Abr-10k	SEA	84.33±0.31	83.99±0.24	**84.41±0.26**	**84.72±0.27**	83.96±0.29	83.98 ±0.28	84.96±0.26	84.02±0.28	**85.09±0.26**
	Sine	86.42±0.20	88.50±0.20	**88.52±0.20**	86.51±0.22	88.38±0.18	**88.39±0.18**	88.42±0.15	**89.71±0.19**	89.22±0.23
	Waveform	79.00±0.45	78.94±0.45	**79.29±0.42**	78.85±0.45	78.88±0.45	**78.92±0.45**	79.80±0.42	**80.64±0.38**	79.68±0.72
	Agrawal	66.01±0.11	**72.87±0.26**	72.25 ±0.20	65.40 ±0.15	**70.94±0.21**	70.92±0.22	67.10±0.16	**69.34±0.35**	66.04±0.63
	Mixed	91.57±0.10	**92.56±0.09**	92.43±0.09	91.61 ±0.11	92.28±0.09	**92.34±0.09**	91.93 ±0.11	**93.21±0.15**	92.71±0.17
Synthetic	RandomRBF	32.26±0.23	**33.40±0.23**	**33.40±0.22**	30.92 ±0.31	32.61 ±0.25	**32.74±0.25**	17.73±0.16	17.67 ±0.15	**22.80±0.92**
Abr-50k	SEA	**87.38±0.17**	86.61±0.08	87.32±0.13	**87.40±0.16**	86.58±0.10	86.53±0.10	87.04±0.15	85.94±0.16	**87.12±0.12**
	Sine	87.26 ±0.11	**91.98±0.10**	91.82±0.12	87.29±0.11	91.91±0.12	**91.96±0.13**	89.67±0.11	**94.41±0.16**	93.35±0.16
	Waveform	80.42±0.15	**81.59±0.17**	81.56 ±0.16	80.17 ±0.14	81.50±0.17	**81.51±0.18**	81.13 ±0.15	**82.31±0.11**	80.89±0.63
	Agrawal	65.83±0.07	**71.94±0.66**	69.92±0.52	65.26 ±0.12	68.79 ±0.36	**68.81±0.38**	**67.82±0.08**	67.38 ±0.48	66.04±0.30
	Mixed	91.82 ±0.05	**93.93±0.05**	93.73±0.05	91.84±0.05	93.71 ±0.05	**93.75±0.05**	92.22 ±0.07	**95.67±0.09**	94.95±0.14
Synthetic	RandomRBF	27.98±0.19	29.12±0.12	**29.15±0.11**	25.03 ±0.25	28.75 ±0.12	**28.82±0.12**	17.68±0.09	17.65 ±0.08	**20.66±0.53**
Abr-100k	SEA	88.00±0.11	87.68±0.05	**88.07±0.07**	**87.99±0.10**	87.60 ±0.06	87.58 ±0.06	**87.83±0.08**	86.95 ±0.10	87.71±0.07
	Sine	85.31 ±0.07	**93.45±0.14**	92.93±0.14	85.33 ±0.07	93.34±0.14	**93.38±0.14**	87.32 ±0.06	**96.25±0.12**	95.33±0.15
	Waveform	80.66±0.09	82.56 ±0.13	**82.58±0.13**	80.38±0.09	82.50±0.12	**82.53±0.13**	81.23 ±0.11	**82.89±0.08**	81.70±0.37
	Agrawal	62.26±0.24	62.80±0.30	**62.93±0.23**	62.07±0.21	62.95±0.29	**62.96±0.30**	**61.18±0.33**	60.34±0.57	60.51±0.40
	Mixed	**83.69±0.25**	83.42±0.28	83.41±0.27	**83.79±0.28**	83.65±0.27	83.69±0.26	83.53±0.22	83.19±0.25	**83.65±0.24**
Synthetic	RandomRBF	31.61±0.35	**32.59±0.39**	32.23 ±0.38	30.58±0.48	31.78±0.44	**31.83±0.42**	18.21±0.40	17.95±0.34	**23.75±1.12**
Grad-10k	SEA	**84.10±0.28**	83.67±0.26	84.09±0.28	**84.46±0.27**	83.77±0.30	83.78 ±0.29	84.67±0.28	83.79±0.30	**84.80±0.31**
	Sine	81.71±0.21	**82.93±0.18**	82.90±0.21	81.74±0.21	82.75±0.21	**82.85±0.22**	82.37±0.17	**82.83±0.25**	82.71±0.18
	Waveform	78.19±0.40	78.15±0.41	**78.54±0.43**	78.17±0.41	78.21±0.41	**78.24±0.41**	78.69±0.42	**80.11±0.38**	78.22±1.09
	Agrawal	64.21 ±0.10	**68.60±0.24**	68.25±0.17	63.62 ±0.14	**67.78±0.19**	67.70 ±0.20	65.27±0.18	**66.45±0.20**	64.13±0.59
	Mixed	85.49±0.11	85.71 ±0.09	**85.86±0.10**	85.13 ±0.13	**85.37±0.11**	85.35 ±0.12	85.15 ±0.13	85.26±0.13	**85.31±0.13**
Synthetic	RandomRBF	32.19 ±0.21	33.32 ±0.24	**33.38±0.23**	31.15 ±0.29	32.63 ±0.28	**32.75±0.28**	17.73 ±0.16	17.67 ±0.15	**23.20±1.02**
Grad-50k	SEA	86.47 ±0.20	86.06 ±0.13	**86.54±0.15**	**86.55±0.18**	86.09 ±0.13	86.04 ±0.12	86.31±0.13	85.44 ±0.13	**86.52±0.12**
	Sine	82.90±0.09	85.54±0.11	**85.58±0.12**	82.63 ±0.09	85.06 ±0.11	**85.07±0.11**	83.61±0.14	**86.21±0.17**	85.60±0.19
	Waveform	79.46 ±0.17	81.33 ±0.17	**81.46±0.23**	79.21 ±0.16	81.39 ±0.13	**81.41±0.14**	79.89±0.18	**82.15±0.12**	80.35±0.65
	Agrawal	64.21 ±0.08	**67.40±0.49**	66.12±0.24	63.63±0.10	**65.84±0.30**	65.81 ±0.32	**66.07±0.08**	64.25±0.19	64.19±0.26
	Mixed	85.84 ±0.05	**87.30±0.05**	87.28 ±0.05	85.10±0.08	**86.69±0.06**	86.66 ±0.06	85.77±0.07	**88.01±0.10**	87.89±0.10
Synthetic	RandomRBF	27.83 ±0.17	**29.06±0.12**	29.02 ±0.13	25.09 ±0.21	28.77 ±0.15	**28.84±0.15**	17.68±0.09	17.65±0.08	**20.39±0.47**
Grad-100k	SEA	87.07 ±0.09	87.16 ±0.08	**87.36±0.07**	87.07±0.08	**87.11±0.07**	87.09 ±0.07	**87.03±0.12**	86.48 ±0.07	87.01±0.09
	Sine	80.68±0.06	**86.99±0.14**	86.50 ±0.13	80.19±0.06	86.08 ±0.12	**86.09±0.13**	81.54±0.05	**88.68±0.17**	88.07±0.19
	Waveform	79.61±0.16	**82.54±0.14**	82.50 ±0.13	79.29 ±0.16	82.52 ±0.11	**82.54±0.11**	79.84±0.12	**82.69±0.10**	81.68±0.35
	Electricity	84.99	86.23	**86.31**	84.67	86.08	**86.09**	89.99	**90.36**	89.60
	Kdd99Joined	89.15	98.1	**98.15**	89.02	98.11	**98.13**	93.52	**99.19**	99.04
	Sick	90.46	**93.53**	93.51	92.72	**93.83**	**93.83**	94.71	93.99	**94.73**
Real	Spam_data	91.97	92.74	**93.09**	91.85	92.68	**92.73**	97.15	96.84	**97.21**
	Usenet1	**73.74**	71.87	72.13	70.07	71.01	**71.2**	73.97	75.75	**76.53**
	Weather	72.85	72.01	**73.03**	**74.12**	73.71	73.54	75.26	74.69	**75.96**
	WineRed	50.6	54.02	**55.28**	55.05	55.58	**55.76**	51.77	**54.32**	41.20
	WineWhite	46.73	46.35	**48.46**	**52.62**	51.35	51.21	20.71	16.92	**23.18**

De forma semelhante, nos conjuntos de dados do mundo real, este método melhorou o original em 5 dos 8 conjuntos testados.

O método proposto MOACE$_{BA}$ apresenta melhores resultados do que o original em conjuntos de dados artificiais com mudanças abruptas (13 dos 18) e graduais (10 dos 18), assim como em conjuntos de dados do mundo real (6 dos 8). OACE$_{BO}$ com HT vence em 10 dos 18 conjuntos de dados artificiais com mudanças abruptas. Além disso, superou o MOACE$_{BO}$ nos experimentos com NB em 8 dos 18 conjuntos com mudanças graduais. No entanto, o MOACE$_{BO}$ tem resultados muito próximos desses, e o método proposto melhorou o original em 5 dos 8 conjuntos de dados reais. De forma geral, OACE$_{BO}$ com HT superou os outros métodos em conjuntos de dados artificiais, enquanto o MOACE$_{BO}$ obteve o melhor resultado em dados do mundo real.

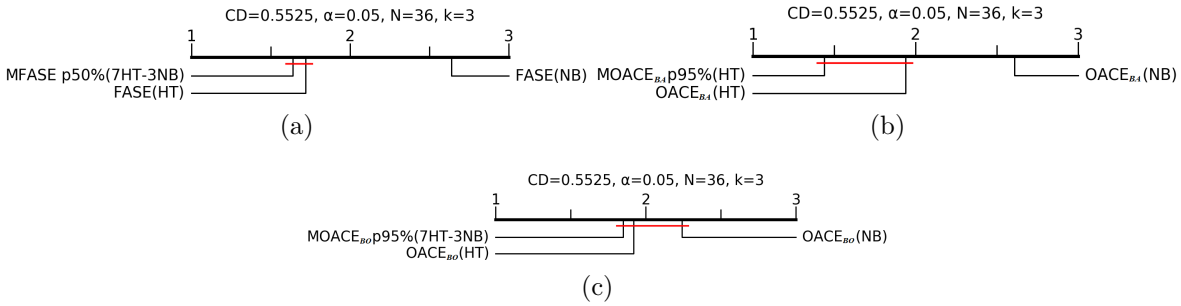


Figura 10 – Comparação das acurácias dos métodos utilizando os testes de *Nemenyi* com um nível de significância de 5% em conjuntos de dados artificiais: (a) Ranks_Método$_{Base}$_FASE, (b) Ranks_Método$_{Base}$_OACE$_{BA}$ e (c) Ranks_Método$_{Base}$_OACE$_{BO}$.

6.2.2.2 Análise Estatística

Para realizar uma análise estatística dos métodos derivados (MFASE, MOACE$_{BA}$ e MOACE$_{BO}$) em relação aos métodos de controle (os originais), foram aplicados o teste *Friedman*(FRIEDMAN, 1937; DEMSAR, 2006) e o teste *post-hoc Nemenyi*(NEMENYI, 1963). Uma explicação detalhada sobre o uso destes testes, bem como a apresentação gráfica do resultado por meio de diagramas simples e da CD, pode ser encontrada no Capítulo 4, na Seção 4.3.2.

A representação gráfica dos resultados, exibida na Figura 10, mostra as melhores medidas localizadas nas primeiras posições. Em particular, as Figuras 10(a), 10(b) e 10(c) mostram uma comparação dos métodos FASE, OACE$_{BA}$ e OACE$_{BO}$ utilizando o teste de *Nemenyi* em conjuntos de dados artificiais.

Em relação aos métodos baseados em FASE, o melhor posicionado foi o MFASE, que é significativamente melhor que o FASE com o classificador NB, considerando os conjuntos de dados artificiais. Nos dados reais, as diferenças observadas não foram estatisticamente significativas. O mesmo ocorre em relação aos métodos baseados em OACE$_{BA}$, onde o MOACE$_{BA}$ foi o melhor posicionado: apresentou desempenho significativamente melhor do que o OACE$_{BA}$ com o classificador NB em dados artificiais. Por outro lado, em relação aos métodos baseados em OACE$_{BO}$, o MOACE$_{BO}$ é o melhor colocado em conjuntos de dados artificiais e reais, mas não apresenta diferenças significativas.

Diante de tudo o que foi apresentado até o momento no estudo da diversidade e do comportamento dos classificadores base, ao analisar a influência no desempenho dos comitês em relação à taxa de acurácia, de maneira geral, pode-se afirmar que os algoritmos modificados MFASE, MOACE$_{BA}$ e MOACE$_{BO}$ apresentaram um desempenho ligeira-

mente melhor do que os métodos originais em muitas situações, embora não tenham sido significativamente superiores em todos os cenários testados. Em particular, o método proposto teve um desempenho superior quando aplicado a comitês que utilizam votação por maioria simples para realizar a classificação. Outra observação relevante a ser feita é que, de forma geral, os métodos que utilizam 7HT_3NB apresentaram os melhores resultados.

O desempenho robusto obtido pela integração do NB e do HT pode ser atribuído a três fatores principais: 1) ambos os classificadores tendem a apresentar bons resultados, contribuindo de forma equilibrada para o esquema de votação desde o início do aprendizado; 2) a variante do HT utilizada incorpora o NB em suas folhas, aproveitando a estrutura probabilística do NB para estimar classes com base nas distribuições condicionais dos atributos, tornando-o mais preciso em situações nas quais os atributos interagem de maneira complexa; e 3) a combinação de NB e HT permite explorar as forças complementares de ambos os classificadores, contribuindo para a adaptabilidade do comitê resultante.

6.3 IMPACTO DE DETECTOR DE MUDANÇAS DE CONCEITO E DIVERSIDADE NOS COMITÊS EM CENÁRIOS SEMI-SUPERVISIONADOS

Nesta seção, pretende-se demonstrar o impacto que os detectores de mudanças de conceito e a inclusão de diversidade têm na performance dos comitês de classificadores em cenários semi-supervisionados. Para isso, foi selecionado o método apresentado por (PINAGÉ; SANTOS; GAMA, 2020) que é conhecido como DSDD. Como descrito na Seção 2.5.9, o DSDD é um comitê que implementa uma versão modificada de *Online Bagging* e utiliza *self-training*. O DSDD foi escolhido por ter apresentado o pior desempenho geral na comparação realizada no Capítulo 5. Essa escolha serve como ponto de referência para demonstrar como as alterações, com a substituição do detector e a inclusão de diversidade, podem aprimorar os resultados.

A partir do comitê base DSDD (para mais detalhes veja 2.5.9) que é composto por dez classificadores do tipo NB, os quais têm associado um detector DDM, e que será nomeado de DSDD_NB_DDM pelo resto da pesquisa, foram geradas oito novas versões: duas surgem da mudança do detector utilizado pelo HDDM_A ou RDDM resultando nas versões DSDD_NB_HDDM_A e DSDD_NB_RDDM. Outras três versões surgem da alteração do classificador base NB para HT, e são nomeadas de DSDD_HT_DDM, DSDD_HT_HDDM_A e DSDD_HT_RDDM. As três restantes incorporam diversidade

no comitê utilizando sete classificadores HT e três NB, com um detector de mudanças de conceito diferente para cada versão gerada, sendo nomeadas DSDD_7HT_3NB_DDM, DSDD_7HT_3NB_HDDM$_A$ e DSDD_7HT_3NB_RDDM. Observe que os nomes dos métodos foram reduzidos nas tabelas e figuras para melhorar a formatação, sem perder o sentido lógico apresentado acima. Outros pontos a serem mencionados são: 1) a escolha da combinação de 7HT_3NB para aplicar diversidade e 2) a não restrição da quantidade de classificadores que participam na votação. A primeira escolha foi realizada com base no fato de que, de forma geral, essa combinação apresentou os melhores resultados quando aplicada aos comitês de classificadores. A segunda escolha é fundamentada na observação de que as combinações demonstraram um desempenho superior quando aplicadas a comitês que utilizam votação por maioria simples para realizar a classificação. As duas justificativas anteriores foram baseadas nos resultados apresentados na Seção 6.2 e no artigo (PÉREZ; MARIÑO; BARROS, 2021).

Os resultados dos experimentos usando bases com mudanças de conceito abruptas e graduais são apresentados nas Tabelas 9 e 10, respectivamente. Nessas tabelas, foram criados três blocos separando os algoritmos por tipo de classificador base usado. Além disso, o método original é sinalizado com * nas tabelas e figuras.

Analisando os resultados, pode-se observar que praticamente a totalidade dos métodos apresenta resultados melhores do que a versão original DSDD_NB_DDM, com exceção do método DSDD_HT_RDDM, que teve resultados inferiores na maioria dos cenários, especialmente nas bases com mudanças abruptas e com 15% dos rótulos. Os resultados de todos os métodos que incorporam diversidade foram o destaque positivo, sobressaindo-se de forma geral em todos os cenários, especialmente a versão DSDD_7HT_3NB_HDDM$_A$ nas bases com 15% de rótulos, que apresentou as melhores acurácias de forma geral.

Afirmamos que os resultados de acurácia reportados são influenciados pelo uso de detectores de mudanças de conceito e que a integração dos métodos RDDM e HDDM$_A$ aos comitês de classificadores impacta positivamente seu desempenho, ao permitir uma adaptação mais eficiente tanto a mudanças abruptas quanto graduais, em comparação com métodos de adaptação implícita. No entanto, RDDM e HDDM$_A$ dependem do acompanhamento sequencial do número de erros cometidos pelo classificador base, o que se torna inviável em cenários semi-supervisionados com um grande número de instâncias não rotuladas. Essa limitação quanto à necessidade de mais instâncias rotuladas é abordada por meio da incorporação de uma nova abordagem de *self-training* (Capítulo 7).

Tabela 9 – Médias de acurácias em (%) usando o comitê DSDD, com 95% de intervalo de confiança nas bases de dados artificiais com mudanças abruptas.

			DSDD_HT			DSDD_7HT_3NB			DSDD_NB		
%RÓT.	TAM.	BASE	DDM	HDDM$_A$	RDDM	DDM	HDDM$_A$	RDDM	DDM*	HDDM$_A$	RDDM
15%	20k	Agrawal(F1-F5)	51.03±1.10	53.88±0.91	53.20±0.96	54.34±1.05	**55.36±0.77**	55.03±0.80	53.55±1.09	52.28±1.25	54.72±1.03
		Agrawal(F6-F10)	66.57±2.61	67.89±1.45	**68.89±1.25**	67.29±1.97	67.28±1.64	68.81±1.04	63.27±2.19	60.15±2.81	64.60±1.84
		LED	35.39±3.56	37.61±4.24	36.26±3.36	43.34±1.67	**47.17±1.93**	45.41±1.51	41.30±0.74	45.78±1.52	42.76±0.88
		Mixed	57.00±0.74	56.45±0.80	54.92±0.84	57.30±0.77	56.53±0.82	55.04±0.76	**58.56±0.47**	57.92±0.41	57.86±0.86
		Sine	60.20±0.86	59.28±0.65	58.10±1.09	60.26±0.85	59.33±0.60	58.28±1.05	**60.49±0.83**	59.34±0.80	60.20±0.94
		Waveform	58.37±2.46	**63.90±1.47**	61.14±1.36	61.27±1.39	63.36±1.26	61.17±1.38	61.09±1.70	60.47±1.23	60.36±1.41
		RandomRBF	26.52±1.38	26.48±1.37	25.63±1.26	27.22±1.30	26.91±1.20	26.34±1.07	27.17±1.21	**27.40±1.12**	26.60±1.10
	50k	Agrawal(F1-F5)	54.24±1.05	55.32±1.14	54.41±0.61	55.21±1.01	54.35±1.47	**55.99±0.55**	53.21±1.11	54.18±1.03	54.51±1.06
		Agrawal(F6-F10)	70.37±0.80	69.95±1.25	**71.09±0.95**	70.12±1.88	70.02±1.28	71.05±1.06	65.75±2.08	63.00±2.43	66.55±1.57
		LED	36.81±5.22	36.21±4.87	32.11±4.60	48.68±1.98	**51.80±1.97**	48.99±1.68	45.96±2.00	51.17±2.05	50.33±1.71
		Mixed	**63.50±0.91**	61.70±1.13	58.48±1.14	63.08±0.96	61.34±1.34	57.48±1.11	58.13±0.49	57.66±0.43	58.06±0.59
		Sine	66.91±0.97	66.82±0.97	63.88±1.25	**67.48±0.94**	66.12±1.12	64.19±1.20	61.91±1.84	59.54±0.86	61.14±1.29
		Waveform	62.43±1.57	64.76±1.37	59.24±1.82	64.05±1.21	**65.49±1.34**	61.16±0.98	58.52±1.39	61.07±1.87	60.43±1.23
		RandomRBF	25.00±1.36	25.59±1.35	23.73±1.12	25.27±1.27	25.70±1.28	24.73±0.89	25.70±1.19	**26.21±1.28**	24.76±0.95
	100k	Agrawal(F1-F5)	55.55±0.66	**57.83±0.42**	55.91±0.43	56.84±0.88	57.42±0.52	57.05±0.38	52.19±1.33	54.32±1.17	54.21±0.95
		Agrawal(F6-F10)	72.78±1.20	73.71±1.04	72.96±0.83	72.64±0.86	73.08±0.97	**73.73±0.80**	66.96±1.48	66.30±1.75	68.06±0.88
		LED	27.41±4.39	38.99±4.41	35.85±4.23	52.53±1.66	55.05±1.37	54.14±1.33	53.37±1.50	**56.57±1.25**	54.15±1.44
		Mixed	71.13±1.08	70.65±1.10	61.85±0.97	70.96±1.10	**71.50±1.01**	62.14±1.09	57.73±0.62	57.50±0.80	58.90±1.01
		Sine	75.33±0.64	**77.12±0.74**	72.45±1.23	75.82±0.83	76.57±0.58	71.01±1.23	62.44±1.85	59.17±1.22	63.29±1.82
		Waveform	61.34±1.41	63.30±0.94	59.67±1.45	**63.71±1.12**	63.61±0.97	60.65±0.79	61.62±1.57	58.72±2.08	63.48±1.54
		RandomRBF	23.12±0.74	23.51±1.04	22.28±0.65	24.80±0.73	26.25±0.66	24.82±0.74	26.28±1.09	**26.32±1.16**	25.79±0.76
		Rank 15%	5.60	4.33	6.48	3.57	**3.02**	4.76	5.88	5.81	5.55
30%	20k	Agrawal(F1-F5	56.22±1.05	57.84±0.78	57.51±0.84	58.73±0.58	**59.04±0.63**	58.84±0.51	55.93±0.89	56.20±0.95	56.19±0.96
		Agrawal(F6-F10)	72.58±1.19	74.22±1.27	75.19±1.03	75.20±1.06	74.94±0.99	**75.97±1.03**	67.28±2.07	68.98±1.22	69.20±1.39
		LED	51.16±5.63	59.66±4.93	54.99±5.52	60.42±0.70	64.28±0.94	62.02±0.70	60.38±0.75	**64.35±0.82**	61.99±0.68
		Mixed	73.83±1.13	68.25±0.72	70.35±0.93	74.23±0.96	68.83±0.80	70.86±0.71	**74.64±0.75**	70.86±0.81	72.88±0.70
		Sine	77.80±0.83	74.81±0.84	75.86±0.92	**78.56±0.65**	75.63±0.72	76.64±0.64	76.84±0.58	75.47±0.56	77.40±0.71
		Waveform	74.73±0.77	74.51±1.00	74.74±1.06	75.18±0.49	74.41±0.89	74.98±0.57	74.81±1.00	**75.38±0.57**	75.12±0.73
		RandomRBF	27.07±1.05	26.29±1.01	26.82±0.97	27.20±1.06	27.11±0.94	**27.47±0.91**	26.93±1.18	26.61±1.04	27.29±1.03
	50k	Agrawal(F1-F5)	59.78±0.69	59.97±0.53	59.64±0.47	60.00±0.44	**60.81±0.32**	60.56±0.33	54.79±1.20	56.07±1.18	55.67±1.24
		Agrawal(F6-F10)	81.07±0.87	81.21±0.44	81.11±0.53	80.31±1.15	81.56±0.46	**81.64±0.48**	71.38±1.69	73.00±1.16	72.98±1.79
		LED	60.08±5.49	65.84±4.60	54.40±6.87	68.54±0.47	69.80±0.51	69.36±0.34	68.92±0.34	**69.91±0.50**	69.52±0.26
		Mixed	85.29±0.58	81.02±0.59	81.76±0.58	**85.40±0.79**	81.63±0.63	82.45±0.74	80.10±0.69	77.36±0.53	77.80±0.65
		Sine	85.61±0.33	84.11±0.49	84.32±0.44	**85.71±0.28**	84.33±0.38	84.42±0.42	81.24±0.40	80.76±0.44	81.23±0.37
		Waveform	76.57±0.38	76.61±0.41	76.55±0.42	76.45±0.43	76.80±0.34	76.62±0.40	76.63±0.56	**76.93±0.41**	76.78±0.46
		RandomRBF	26.05±1.08	25.42±0.93	24.49±0.95	26.50±1.03	**27.36±0.76**	26.82±0.87	25.61±1.10	26.44±0.82	25.96±0.94
	100k	Agrawal(F1-F5)	60.40±0.66	**61.65±0.77**	60.22±0.29	61.33±0.31	61.16±0.40	60.88±0.23	55.23±1.10	56.15±0.87	56.47±0.82
		Agrawal(F6-F10)	83.19±0.28	83.04±0.25	82.46±0.27	82.78±0.83	**83.28±0.33**	83.00±0.43	73.50±1.38	74.28±1.51	76.21±1.71
		LED	64.66±4.90	65.11±4.58	57.84±6.41	70.49±0.27	71.31±0.22	70.77±0.27	70.81±0.23	**71.67±0.20**	71.08±0.23
		Mixed	88.63±0.38	86.05±0.43	85.78±0.47	**88.95±0.34**	86.26±0.30	86.15±0.39	79.79±0.88	76.42±0.67	77.52±0.81
		Sine	**87.65±0.19**	86.97±0.40	86.56±0.46	87.47±0.26	86.97±0.28	86.66±0.28	82.00±0.30	80.78±0.62	81.63±0.37
		Waveform	77.00±0.20	76.82±0.23	76.32±0.31	76.82±0.24	76.93±0.14	76.14±0.40	77.23±0.23	**77.34±0.18**	77.02±0.29
		RandomRBF	25.09±1.04	25.34±0.96	23.58±0.77	**26.13±0.93**	25.84±0.70	26.12±0.71	25.40±1.07	26.06±0.81	26.03±0.99
		Rank 30%	4.86	5.90	6.52	**3.36**	3.55	3.55	6.29	5.50	5.48
		Rank	5.23	5.12	6.50	3.46	**3.29**	4.15	6.08	5.65	5.51

Portanto, baseando-se nesses resultados, pode-se afirmar que a inclusão de diversidade nos métodos contribui significativamente para a melhoria da performance em cenários dinâmicos e semi-supervisionados. Para finalizar, é válido observar que foi identificado que a escolha do detector pode fazer a diferença.

6.3.1 Avaliação estatística

Para a avaliação estatística, assim como na Seção 6.2.2.2, foi utilizado o teste F_F (DEMSAR, 2006) para comparação e o pós-teste *Nemenyi* (DEMSAR, 2006) para definir explicitamente as diferenças estatísticas encontradas.

A análise estatística permite observar que, em termos absolutos, os métodos que introduzem diversidade são os três melhores colocados. Também é válido apontar que, nos

Tabela 10 – Médias de acurácias em (%) usando o comitê DSDD, com 95% de intervalo de confiança nas bases de dados artificiais com mudanças graduais.

			DSDD_HT			DSDD_7HT_3NB			DSDD_NB		
%RÓT.	TAM.	BASE	DDM	HDDM$_A$	RDDM	DDM	HDDM$_A$	RDDM	DDM*	HDDM$_A$	RDDM
15%	20k	Agrawal(F1-F5)	51.82±1.14	54.33±0.94	52.63±1.01	54.26±1.00	54.96±0.87	54.79±0.80	52.95±1.21	52.20±1.23	**55.09±0.86**
		Agrawal(F6-F10)	66.60±2.63	67.59±1.25	68.89±1.21	66.10±2.34	67.92±0.85	**69.01±1.02**	63.22±2.05	59.03±2.95	64.43±1.91
		LED	32.98±3.61	38.15±3.81	37.99±3.73	44.07±1.74	**45.09±1.87**	42.50±1.47	41.19±0.80	44.67±1.63	42.57±0.81
		Mixed	56.81±0.70	56.42±0.81	54.97±0.88	57.03±0.68	56.46±0.80	54.94±0.74	**58.08±0.38**	57.65±0.40	57.57±0.79
		Sine	59.07±0.79	57.92±0.45	57.75±1.09	59.29±0.80	58.20±0.54	58.66±1.06	**60.08±1.02**	57.93±0.63	60.06±0.97
		Waveform	58.49±2.52	**63.78±1.26**	59.77±1.41	61.89±1.34	62.41±1.10	59.87±0.93	60.33±1.43	60.68±1.18	60.99±1.24
		RandomRBF	26.47±1.38	26.34±1.43	25.62±1.27	27.07±1.24	26.90±1.19	26.25±1.10	27.16±1.20	**27.52±1.09**	26.54±1.09
	50k	Agrawal(F1-F5)	54.10±0.87	55.80±0.60	54.67±0.95	55.33±0.99	54.69±1.41	**56.20±0.68**	54.05±1.13	54.23±1.00	55.57±0.64
		Agrawal(F6-F10)	70.33±0.98	70.08±1.07	71.24±0.76	69.72±1.79	70.04±0.71	**71.66±0.98**	65.79±1.95	63.36±2.22	66.54±1.61
		LED	32.00±4.81	36.54±5.15	32.97±4.24	49.47±2.02	**51.04±1.50**	50.14±1.89	46.24±1.56	50.26±1.80	48.21±1.91
		Mixed	62.55±1.04	61.55±1.11	57.69±0.92	**62.69±0.99**	61.87±1.18	57.74±0.82	58.85±0.88	57.82±0.55	57.78±0.82
		Sine	66.81±1.13	66.28±1.18	63.58±1.09	**67.51±1.14**	65.50±1.04	63.32±0.85	60.76±1.22	60.45±1.01	61.22±1.49
		Waveform	61.27±1.98	63.61±1.16	61.51±1.38	**63.87±1.07**	63.25±1.25	62.53±1.46	59.95±1.73	63.26±1.81	60.40±1.35
		RandomRBF	25.62±1.05	25.53±1.13	23.54±0.82	26.08±0.97	**26.53±1.03**	24.81±0.84	26.03±0.97	26.38±0.96	25.65±0.76
	100k	Agrawal(F1-F5)	55.89±0.69	**57.56±0.62**	56.12±0.51	56.76±0.95	57.53±0.43	57.30±0.46	52.71±1.23	54.19±1.12	54.40±0.96
		Agrawal(F6-F10)	72.24±0.89	73.49±1.13	72.46±0.95	73.06±1.11	**74.72±1.25**	73.09±0.73	65.25±2.38	65.31±1.82	68.63±1.07
		LED	25.38±3.81	34.91±4.46	35.83±4.42	53.92±1.49	54.94±1.38	53.22±1.28	52.86±1.33	**56.15±1.45**	54.08±1.26
		Mixed	71.44±1.07	70.44±1.18	62.90±1.04	**71.56±1.14**	70.60±1.31	62.46±0.92	57.55±0.66	57.34±0.95	59.66±1.06
		Sine	75.74±0.77	**76.79±0.70**	72.76±1.18	75.18±0.71	76.08±0.73	71.05±1.00	60.99±1.48	61.62±1.83	64.39±1.75
		Waveform	61.74±1.15	63.70±1.04	60.18±0.92	63.09±1.00	**64.41±0.99**	61.34±0.90	60.38±1.60	61.24±2.36	61.78±1.39
		RandomRBF	23.48±0.92	23.14±0.89	22.47±0.74	24.96±0.74	25.81±0.85	25.10±0.71	25.85±1.08	**26.51±1.13**	25.74±0.78
		Rank 15%	5.81	4.43	6.71	3.48	**2.90**	4.95	6.19	5.43	5.10
30%	20k	Agrawal(F1-F5)	56.22±1.15	58.16±0.70	57.35±0.98	58.62±0.66	**58.64±0.41**	58.33±0.63	54.82±1.16	54.59±1.25	56.49±0.65
		Agrawal(F6-F10)	71.90±1.30	72.05±1.26	72.80±1.20	74.24±1.31	74.80±1.12	**75.24±1.14**	67.17±1.98	68.70±1.06	68.69±1.16
		LED	53.56±5.16	57.74±4.76	52.15±5.26	60.57±0.70	**63.02±0.72**	61.05±0.73	59.65±0.82	62.46±0.94	61.63±0.56
		Mixed	70.70±0.99	67.62±1.17	68.05±0.88	71.93±0.99	68.37±0.83	69.94±0.87	**73.07±0.80**	69.78±0.76	71.20±0.70
		Sine	77.17±0.78	73.63±0.74	73.46±0.99	**77.49±0.65**	74.43±0.85	74.95±0.70	76.28±0.51	73.20±1.17	75.14±0.51
		Waveform	74.53±0.97	74.83±0.56	**75.30±0.51**	74.42±1.06	75.24±0.40	74.76±0.86	74.34±1.38	74.41±0.78	74.84±0.93
		RandomRBF	27.06±1.09	26.28±0.98	26.67±0.95	27.21±1.03	27.05±0.96	**27.40±0.93**	26.77±1.22	26.37±1.00	27.03±1.09
	50k	Agrawal(F1-F5)	59.61±0.42	59.88±0.67	59.62±0.47	60.15±0.32	60.09±0.28	**60.24±0.32**	55.26±1.14	55.68±0.91	56.22±1.03
		Agrawal(F6-F10)	81.05±0.80	80.81±0.45	80.72±0.59	79.69±1.08	81.40±0.64	**81.60±0.47**	70.07±2.15	73.17±1.34	72.14±1.69
		LED	58.62±5.98	66.39±3.51	54.91±7.47	67.87±0.56	69.00±0.50	68.80±0.39	68.54±0.29	**69.88±0.36**	69.20±0.27
		Mixed	**84.81±0.52**	80.31±0.70	80.62±0.56	84.07±0.76	80.69±0.67	81.70±0.70	78.70±0.80	76.67±0.65	77.28±0.68
		Sine	**85.26±0.29**	83.50±0.43	84.05±0.47	85.15±0.32	83.78±0.31	84.46±0.38	80.99±0.41	80.07±0.35	80.57±0.32
		Waveform	76.49±0.39	76.75±0.46	76.92±0.32	76.56±0.32	76.80±0.21	77.03±0.20	76.47±0.46	76.84±0.24	**77.06±0.35**
		RandomRBF	25.95±0.95	25.27±1.16	24.64±0.95	26.35±0.97	26.59±0.87	**26.65±0.97**	26.14±1.07	25.56±0.97	25.94±0.97
	100k	Agrawal(F1-F5)	61.16±0.85	61.11±0.43	60.30±0.37	61.19±0.35	**61.21±0.29**	60.80±0.23	54.02±1.47	56.37±0.63	56.42±0.64
		Agrawal(F6-F10)	82.91±0.24	82.81±0.39	82.68±0.31	82.81±0.82	82.94±0.38	**83.14±0.37**	72.96±1.67	73.30±1.50	75.25±1.74
		LED	61.64±5.46	67.32±3.10	60.30±5.05	70.28±0.37	71.06±0.24	70.49±0.33	70.67±0.20	**71.30±0.23**	71.22±0.21
		Mixed	88.09±0.34	85.68±0.39	85.66±0.50	**88.18±0.42**	86.27±0.38	86.35±0.46	79.08±0.73	76.51±0.71	77.58±0.78
		Sine	**87.50±0.27**	86.69±0.29	86.61±0.38	87.26±0.16	86.65±0.33	86.23±0.37	81.74±0.31	80.70±0.39	81.61±0.28
		Waveform	76.89±0.25	77.04±0.18	76.30±0.36	76.82±0.22	76.88±0.20	76.50±0.28	**77.15±0.26**	77.13±0.23	76.69±0.36
		RandomRBF	24.88±1.15	24.95±0.78	23.60±0.79	26.22±0.70	26.30±0.64	**26.47±0.76**	25.48±1.04	26.31±0.92	26.18±0.86
		Rank 30%	4.62	5.69	6.29	3.60	3.29	**3.24**	6.43	6.33	5.52
		Rank	5,21	5,06	6,50	3,54	**3,10**	4,10	6,31	5,88	5,31

algoritmos que usam NB como classificador, a simples alteração do detector já aprimorou os resultados do método original, sendo DSDD_NB_RDDM a melhor versão para ambos os tipos de mudanças. Já nos algoritmos que usam HT como classificador base, assim como nos que usam diversidade, o algoritmo que usa HDDM$_A$ obtém os melhores resultados em seus respectivos blocos.

Para proporcionar uma compreensão mais clara dos resultados, os *ranks* foram calculados e os diagramas de CD dos testes de *Nemenyi* foram gerados. Para as mudanças abruptas, podemos observar na Figura 11(c) que os métodos melhor ranqueados no cálculo que agrupa os resultados de 15% e 30% são métodos que incluem diversidade, com o DSDD_7HT_3NB_HDDM$_A$ como melhor posicionado, e não apresentando diferença significativa para os demais métodos que incluem diversidade, i.e. DSDD_HT_HDDM$_A$ e DSDD_HT_DDM. Na análise do cenário com presença de mudanças graduais, os métodos

melhores colocados continuam sendo os que inserem diversidade, mantendo-se como melhor posicionado o DSDD_7HT_3NB_HDDM$_A$, mas neste cenário o mesmo só não apresenta diferença significativa com DSDD_7HT_3NB_DDM e DSDD_7HT_3NB_RDDM como pode ser observado na Figura 12(c).

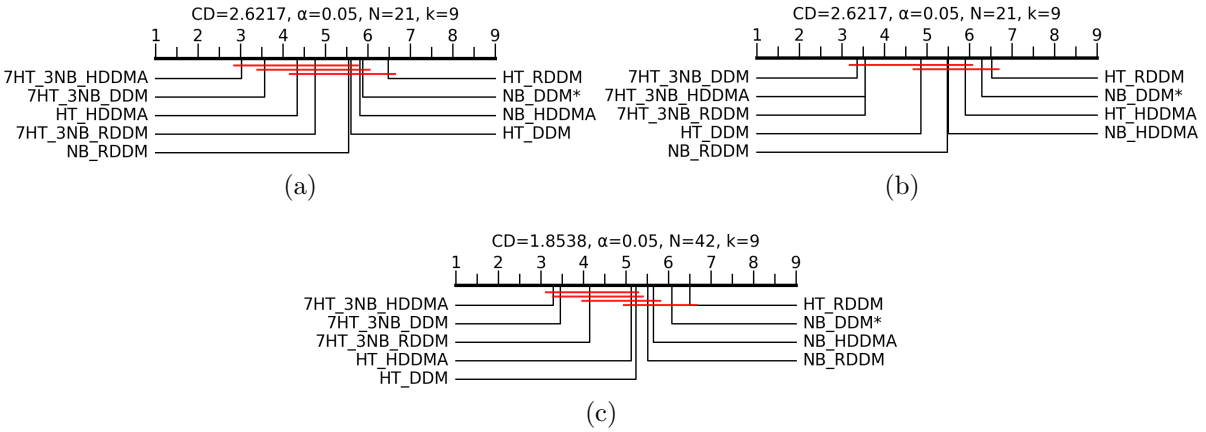


Figura 11 – Comparação estatística das acurácias de todos os métodos em todos os cenários do número de rótulos: (a) 15%, (b) 30% e (c) todos, através do Teste F_F e do Pós-Teste *Nemenyi*, com 95% de confiança, usando o comitê DSDD e bases de dados artificiais com mudanças abruptas.

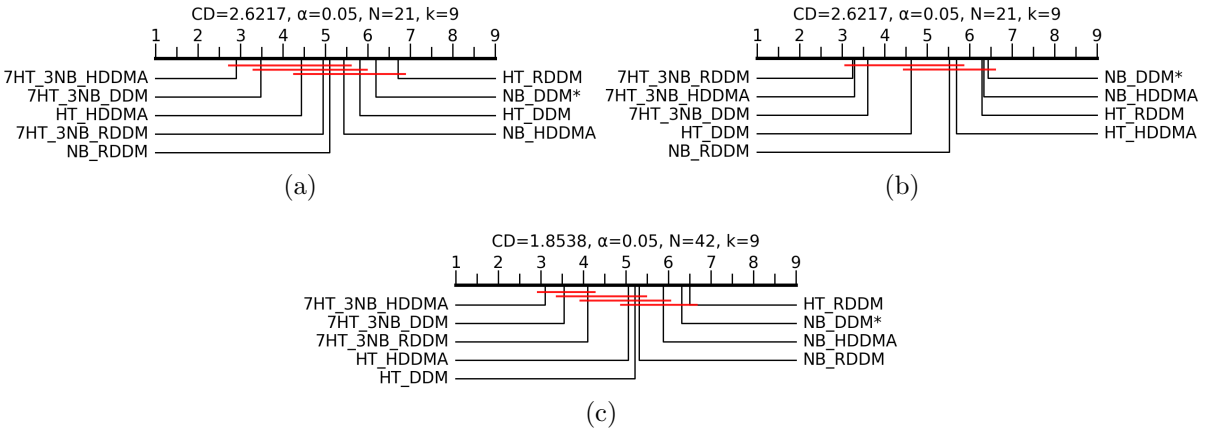


Figura 12 – Comparação estatística das acurácias de todos os métodos em todos os cenários do número de rótulos: (a) 15%, (b) 30% e (c) todos, através do Teste F_F e do Pós-Teste *Nemenyi*, com 95% de confiança, usando o comitê DSDD e bases de dados artificiais com mudanças graduais.

Com base no exposto, podemos afirmar que o uso da diversidade nos comitês de classificadores contribui significativamente para a precisão dos resultados. Além disso, foi constatado que a escolha do detector pode influenciar os resultados. Esta Seção 6.3.1 reafirma que, com o aumento do número de rótulos, os algoritmos tendem a ter um desempenho melhor, conforme mencionado no Capítulo 5. Os testes com bases reais para os métodos acima apresentados na Seção 6.3 foram desconsiderados, pois os testes nas bases artificiais forneceram as evidências necessárias para demonstrar o impacto dos detectores e a diversidade nos comitês de classificadores em cenários de aprendizado semi-supervisionado.

6.4 CONSIDERAÇÕES FINAIS

No decorrer do capítulo, foram efetuadas comparações empíricas e estatísticas de comitês de classificadores que incorporam mecanismo de detecção de mudanças de conceito, tendo a acurácia como métrica de avaliação. Depois de uma minuciosa análise, demonstrou-se que os comitês construídos para cenários supervisionados podem ser usados em ambientes semi-supervisionados. Além disso, foi demonstrado que certas alterações nos comitês podem melhorar o desempenho dos mesmos, resultando em melhorias perceptíveis na acurácia. Especificamente:

- incremento da diversidade de classificadores com a combinação de vários tipos de classificadores;
- mudança nos esquemas de votação e
- incorporação de detectores de mudanças de conceito.

Tudo o que foi apresentado até o momento valida a segunda hipótese levantada nesta tese, a qual defende que promover diversidade nos comitês de classificadores com a combinação de vários tipos de métodos resulta em maior robustez e capacidade de generalização, contribuindo para um desempenho superior em cenários com escassez de rótulos. O capítulo deixa empiricamente demonstrado que a combinação de 7HT + 3NB (com votação maioritária) com a incorporação do detector de mudanças de conceito HDDM$_A$ é uma opção robusta para a criação de comitês de classificadores para ser utilizada em ambientes supervisionados e semi-supervisionados.

Os resultados apresentados na Seção 6.2, referentes à abordagem supervisionada, foram previamente publicados em (PÉREZ; MARIÑO; BARROS, 2021). Por sua vez, os resultados dos experimentos conduzidos em cenários semi-supervisionados, descritos na Seção 6.3, foram reportados em (PÉREZ; BARROS; SANTOS, 2025).

7 ABORDAGEM DE SELF-TRAINING

O presente capítulo busca responder as hipóteses 3 e 4 formuladas nesta tese. A primeira delas defende que o uso de *self-training* para pseudo-rotular os dados não rotulados aumenta a quantidade de informações disponíveis para o treinamento, mitigando os efeitos da escassez de rótulos e melhorando a acurácia dos métodos de classificação. A segunda hipótese, por sua vez, defende que o desempenho dos métodos de classificação supervisionados em cenários semi-supervisionados pode ser significativamente melhorado, em termos de acurácia, pela integração das três abordagens: detecção de mudanças de conceito, introdução de diversidade nos comitês de classificadores e *self-training*.

Outro ponto abordado neste capítulo leva em consideração a análise apresentada nos Capítulos 5 e 6, onde se constatou que detectores de mudanças de conceito com maior acesso aos dados (neste caso, se a predição do classificador foi correta ou incorreta) tendem a detectar mudanças mais rapidamente. Isso pode resultar na reconstrução do modelo, contribuindo para a melhoria do desempenho do método (comitê ou classificador).

Diante do exposto, este capítulo apresenta uma abordagem de *self-training* que pode ser aplicada na etapa de treinamento de comitês ou classificadores, visando fornecer mais dados aos detectores de mudanças de conceito. A nova abordagem introduz ideias específicas com o potencial de melhorar o desempenho e a adaptabilidade dos comitês e classificadores em cenários dinâmicos e semi-supervisionados.

Para a realização dos testes, a nova abordagem foi aplicada ao comitê FASE e ao classificador HT, resultando em novos métodos que serão apresentados nas próximas seções. É importante ressaltar que o FASE foi utilizado com o detector (HDDM$_A$), conforme sugerido pelos autores na publicação (FRÍAS-BLANCO et al., 2016). No caso do HT, o RDDM foi o detector escolhido, uma vez que essa combinação demonstrou o melhor desempenho nos testes em cenários semi-supervisionados (ver Capítulo 5).

7.1 PROPOSTA DA ABORDAGEM *SELF-TRAINING*

O conceito de *Self-training* – uma técnica específica associada ao aprendizado semi-supervisionado – tem sido discutido e utilizado na comunidade científica há várias décadas, ganhando mais atenção principalmente a partir da década de 1990 (YAROWSKY, 1995;

BLUM; MITCHELL, 1998; MITCHELL, 1999); nessa época, métodos de aprendizado semi-supervisionado começaram a se tornar mais populares, especialmente com o aumento da disponibilidade de dados não rotulados e o reconhecimento de seu potencial para melhorar o desempenho dos modelos de aprendizado de máquina.

O artigo *Semi-Supervised Learning Literature Survey* apresentado por (ZHU, 2005), trouxe uma visão abrangente das técnicas semi-supervisionadas, incluindo o *Self-training*. É possível destacar também publicações e conferências recentes, que vêm discutindo e refinando técnicas de *self-training* e outras abordagens semi-supervisionadas (MONTEIRO; SOARES; BARROS, 2021; PÉREZ; BARROS; SANTOS, 2023; SHI; LIU, 2023; AMINI et al., 2024; ODONNAT; FEOFANOV; REDKO, 2024). Além disso, é importante observar que o *self-training* continua sendo uma área ativa de pesquisa, com novas variações e melhorias sendo propostas regularmente.

Nesta seção, uma nova abordagem é proposta, com o objetivo de contribuir na evolução das técnicas utilizadas no *self-training*. Essa nova abordagem é baseada em uma dupla verificação na etapa de treinamento, que decide se um exemplo não rotulado deve ser rotulado e incorporado. O processo utiliza o classificador KNN e um método de classificação definido pelo usuário (parametrizável), que são treinados (nessa ordem) com todos os exemplos rotulados. Esse fluxo exato pode ser visto nas linhas 3 e 4 do pseudo-código 3, que descreve o procedimento geral de treinamento.

Para exemplos de entrada não rotulados, o algoritmo utiliza uma verificação em duas etapas (linha 6) para estimar a qualidade da previsão feita pelo classificador e decidir se essa previsão deve ser considerada correta e utilizada para treinar o classificador. Na primeira etapa, a função *verificaClasse* verifica se a classe prevista para o exemplo corresponde à classe de pelo menos 50% dos vizinhos mais próximos identificados pelo algoritmo KNN. É importante notar que o KNN é usado exclusivamente para obter os vizinhos mais próximos. Embora um limite de 50% seja utilizado durante a fase de verificação para realizar a votação majoritária, esse valor não é tratado como absoluto na abordagem. Esse número é sustentado por validações empíricas realizadas em diversos conjuntos de dados artificiais, onde outros limites, como 75% e 100%, também foram considerados.

Outro ponto a ser destacado é que a escolha do valor de k pode ser crucial para o desempenho do modelo: um valor muito pequeno pode tornar o modelo sensível ao ruído, enquanto um valor muito grande pode suavizar excessivamente as previsões. Nesta abordagem, iniciamos com cinco vizinhos, conforme sugerido em outros estudos (PINAGÉ;

SANTOS; GAMA, 2020), até atingir dez vizinhos, o valor padrão utilizado pelo KNN.

Para mais detalhes sobre a implementação desta primeira etapa, o leitor deve consultar a primeira função do Algoritmo 4. Uma vez satisfeita a primeira etapa, a segunda etapa – confirmação – utiliza a função *verificaComQuartil* para finalizar a validação do potencial de treinamento. Se essa condição também for atendida, a classe prevista pelo classificador é considerada correta e atribuída como o rótulo do exemplo. Consequentemente, o método de classificação é treinado com este exemplo (ver linha 7).

Algoritmo 3: Procedimento geral integrado ao fluxo de treinamento.

```
1 A cada exemplo de entrada x_i, instância predita Pred, Knn_k: contêm as k instâncias mais
    próximas de Pred com 5 < k < 10;
2 if classeExiste(x_i) then
3 |   TreinaKnn(x_i)
4 |   TreinaClassificador(x_i)
5 else
6 |   if verificaClasse(Pred, Knn_k) & verificaComQuartil(Pred, Knn_k) then
7 |   |   TreinaClassificador(x_i)
```

A função *verificaComQuartil* é responsável pela implementação da etapa de confirmação da abordagem proposta e é apresentada no Algoritmo 4 a partir da linha 9. Inicialmente, com relação à distância inversa utilizada $(1/d)$, é importante destacar que ao invertê-la e utilizá-la como referência, pontos mais próximos exercem uma influência maior do que aqueles mais distantes (SHEPARD, 1968). Com base nessa possibilidade de influência, a função na linha 12 calcula a distância inversa para cada vizinho retornado pelo KNN. Os valores resultantes são ordenados (linha 13) para obter o limite do terceiro quartil, que é comparado (linha 14) com a média das distâncias inversas (Equação 7.1). Se a média for maior ou igual ao limite do terceiro quartil, a função confirma que o algoritmo deve treinar o método de classificação.

$$\text{Média das inversas} = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{d_i} \tag{7.1}$$

O procedimento proposto tem como objetivo aumentar a disponibilidade de dados de entrada rotulados para o detector de mudanças de conceito, permitindo avaliar a qualidade das previsões realizadas pelo classificador e, assim, considerar que, para um subconjunto de exemplos, os rótulos atribuídos pelo classificador são corretos. Essa estratégia tem potencial para melhorar os resultados da maioria dos detectores de mudanças de conceito existentes, já que estes dependem de testes estatísticos e/ou do número de previsões

Algoritmo 4: Funções usadas na decisão de treinamento

Input: $Pred$: instância predita; Knn_k: contêm as k instâncias mais próximas de $Pred$ com $5 < k < 10$;

```
1  Function verificaClasse(Pred, Knn_k):
2      numeroIguais ← 0
3      for l ← 1 to k do
4          if Pred == Knn_l then
5              numeroIguais++
6      if numeroIguais/k > k/2 then
7          return verdadeiro
8      return falso
9  Function verificaComQuartil(Pred, Knn_k):
10     distanciasInversas[k] ← 0.0
11     for l ← 1 to k do
12         distanciasInversas[l] ← 1/calculaDistancia(Knn_l, x_i)
13     ordena(distanciasInversas)
14     if media(distanciasInversas) >= distanciasInversas[k * 3/4] then
15         return verdadeiro
16     return falso
```

corretas e incorretas para identificar tais mudanças. Uma avaliação experimental desta abordagem utilizando um classificador tradicional (HT) e um método de comitê (FASE), é apresentada nos experimentos da Seção 7.2.

7.2 RESULTADOS DA APLICAÇÃO DA ABORDAGEM

Nesta seção, a abordagem proposta será aplicada a dois algoritmos de classificação (HT e FASE) que incorporam um mecanismo de detecção de mudanças de conceito. A configuração dos experimentos segue as práticas já estabelecidas ao longo da tese. Mais informações podem ser encontradas no Capítulo 4.

Na experimentação apresentada nesta seção, foi escolhida a combinação do classificador HT com o detector de mudanças de conceito RDDM devido ao seu desempenho superior entre os métodos desenvolvidos para ambientes supervisionados e testados em ambientes semi-supervisionados, conforme resultados detalhados no Capítulo 5. Ao longo desta seção, em figuras e tabelas, a combinação HT + RDDM será referida apenas como HT, e sendo usado o nome HT_S para a versão que incorpora abordagem de *self-training*.

No caso do FASE, a experimentação incorporou quatro versões diferentes, todas utilizando o HDDM$_A$ como detector de mudanças, conforme sugerido pelos autores (FRÍAS-BLANCO et al., 2016). O FASE foi selecionado como comitê base por ser um dos métodos de ponta, amplamente conhecido e utilizado em pesquisas sobre fluxos de dados (BARROS; SANTOS, 2019; MAHDI et al., 2024). Além da versão original, a abordagem proposta de *self-training* também foi testada em uma versão que incorporou o mecanismo de diversidade, combinando sete classificadores HT e três classificadores NB, denominada FASE$_D$. As outras duas versões foram denominadas FASE$_S$ e FASE$_{DS}$. A primeira é o resultado da aplicação da abordagem de *self-training* no FASE, enquanto a segunda surge da aplicação combinada da estratégia de diversidade utilizada para criar o FASE$_D$ com a ideia de *self-training* aplicada na geração da versão FASE$_S$.

Além dos métodos mencionados anteriormente, utilizados para avaliar o *self-training* proposto, os experimentos também incluem BDF e DSDD que são métodos de comitê semi-supervisionados que servem como referência, uma vez que implementam seus próprios mecanismos de *self-training*. Escolhemos a variação BDF$_{RDDM}$ – que apresentou o melhor desempenho entre os métodos semi-supervisionados em experimentos anteriores (Capítulo 5) (PÉREZ; BARROS; SANTOS, 2023) – e a versão DSDD_7HT_3NB_HDDM$_A$, que obteve o melhor desempenho nos resultados dos experimentos apresentados no Capítulo 6. É válido sinalizar que todos os comitês usados foram configurados com os valores padrão de seus parâmetros.

As Tabelas 11 e 12 apresentam os resultados dos experimentos utilizando os conjuntos de dados configurados com mudanças de conceito abruptas e graduais, respectivamente. Nos cenários com mudanças abruptas, o método FASE$_{DS}$ foi identificado como o de melhor desempenho, apresentando resultados de alta precisão nos conjuntos de dados *Agrawal* (F6-F10), especialmente com 15% dos rótulos. No conjunto de dados criado pelo gerador *RandomRBF*, o método FASE$_{DS}$ alcançou resultados significativos com 30% de dados rotulados. Por outro lado, para o gerador *Mixed*, resultados notáveis foram obtidos para ambas as porcentagens de dados rotulados, mantendo desempenho estável nos demais conjuntos de dados.

O segundo método mais bem posicionado foi o HT$_S$, que apresentou seus melhores desempenhos nos conjuntos de dados *Agrawal* (F1-F5) e *LED* para ambas as porcentagens de dados rotulados, além de obter bons resultados no conjunto *Agrawal* (F6-F10), particularmente com 30% de dados rotulados. Vale destacar que o HT$_S$ é um método

simples, consistindo em um único classificador. Seus bons resultados, quando comparados às versões do comitê FASE sem *self-training*, evidenciam que a abordagem proposta de *self-training* é eficaz.

O próximo método mais bem posicionado foi o FASE$_S$, que ficou muito próximo ao HT$_S$ e apresentou bons resultados na maioria dos conjuntos de dados gerados, especialmente nos conjuntos Sine. Observa-se que o FASE$_S$ não foi posicionado em segundo lugar devido aos seus resultados inferiores nos conjuntos com 15% de dados rotulados, em comparação aos resultados alcançados pelo HT$_S$. Também é importante mencionar o desempenho inferior do comitê DSDD em comparação com quase todos os outros métodos na maioria dos conjuntos de dados.

Nos cenários envolvendo mudanças graduais de conceito, os três primeiros métodos nos resultados experimentais são os mesmos dos posicionados nos conjuntos de dados com mudanças abruptas, como mostrado na Tabela 12. O método mais bem posicionado, FASE$_{DS}$, alcançou seus melhores resultados nos conjuntos de dados *RandomRBF* com 30% de dados rotulados e nos conjuntos *Waveform* com 15% dos rótulos. Além disso, o HT$_S$ manteve a segunda posição, beneficiando-se de seu desempenho nos conjuntos de dados *LED*, *Agrawal* (F1-F5) e *Agrawal* (F6-F10). Esses resultados do HT$_S$ confirmam as descobertas da seção 6.3, onde foi observado que métodos utilizando RDDM são eficazes em cenários com mudanças graduais, embora seu uso dependa de uma análise prévia dos conjuntos de dados. Concluindo os três primeiros, o FASE$_S$ apresentou desempenho estável na maioria dos conjuntos de dados com ambas as porcentagens de rótulos, sendo especialmente eficaz nos conjuntos *Sine*.

O análise dos resultados dos experimentos em conjuntos de dados do mundo real, apresentados na Tabela 13 confirmam, o FASE$_{DS}$ como o método mais bem posicionado para os experimentos com 15% e 30% de dados rotulados e, consequentemente, no ranking geral. O desempenho do FASE$_{DS}$ foi particularmente bom nos conjuntos de dados *Covertype*, *Spam_data* e *Weather*. O segundo método mais bem posicionado foi o FASE$_S$, que apresentou resultados consistentes em todos os conjuntos de dados do mundo real testados para ambas as porcentagens de rótulos. Por fim, completando os três primeiros, o FASE$_D$ mostrou desempenho consistente. No entanto, ao analisar os experimentos com 15% de dados rotulados, esse método perde a terceira posição para o BDF, que se destaca por seu desempenho robusto nos conjuntos de dados *Airlines* e *Connect4* para ambas as porcentagens de rótulos.

Tabela 11 – Médias de acurácia em (%), com intervalos de confiança de 95%, em base de dados artificiais com mudanças de conceito abruptas.

%RÓT.	TAM.	COMITÊ CLASSIFICADOR BASE BASE \ DETECTOR	— HT RDDM	— HT_S RDDM	BDF HT RDDM	DSDD 7HT_3NB $HDDM_A$	FASE HT $HDDM_A$	$FASE_D$ 7HT_3NB $HDDM_A$	$FASE_S$ HT $HDDM_A$	$FASE_{DS}$ 7HT_3NB $HDDM_A$
15%	20k	Agrawal(F1-F5)	61.07±0.39	**63.14±0.31**	62.32±0.39	55.36±0.77	60.15±0.33	60.51±0.28	62.15±0.26	62.63±0.23
		Agrawal(F6-F10)	77.59±1.93	82.46±0.49	68.50±1.28	67.28±1.64	79.75±0.43	80.17±0.40	82.31±0.25	**82.52±0.25**
		LED	64.95±0.33	**67.50±0.24**	61.71±0.41	47.17±1.93	63.32±0.33	63.21±0.32	65.41±0.26	65.62±0.27
		Mixed	87.88±0.23	87.89±0.43	87.69±0.26	56.53±0.82	89.01±0.23	87.54±0.19	87.55±0.20	**89.03±0.21**
		Sine	85.48±0.32	86.40±0.31	84.81±0.40	59.33±0.60	84.54±0.27	85.03±0.20	**86.53±0.19**	86.39±0.18
		Waveform	78.66±0.29	**79.72±0.22**	77.72±0.42	63.36±1.26	78.37±0.29	78.36±0.29	79.03±0.24	79.29±0.25
		RandomRBF	31.07±0.46	30.67±0.37	**33.92±0.52**	26.91±1.20	30.46±0.34	31.02±0.34	31.64±0.33	32.71±0.30
	50k	Agrawal(F1-F5)	63.67±0.37	**65.37±0.33**	64.07±0.34	54.35±1.47	63.21±0.29	63.46±0.33	64.30±0.25	64.51±0.24
		Agrawal(F6-F10)	82.77±0.97	84.55±0.39	71.51±0.58	70.02±1.28	83.24±0.33	83.55±0.31	84.41±0.16	**84.65±0.12**
		LED	69.81±0.16	**71.19±0.18**	67.98±0.36	51.80±1.97	68.85±0.19	68.82±0.21	69.93±0.19	70.09±0.17
		Mixed	**90.69±0.20**	90.06±0.31	89.76±0.21	61.34±1.34	90.44±0.17	90.45±0.16	90.17±0.21	90.24±0.23
		Sine	87.92±0.17	88.17±0.14	86.50±0.29	66.12±1.12	86.58±0.18	88.35±0.12	**88.52±0.12**	88.36±0.11
		Waveform	79.58±0.22	**80.24±0.15**	79.09±0.25	65.49±1.34	79.51±0.22	79.53±0.21	80.01±0.19	80.19±0.16
		RandomRBF	31.38±0.38	31.48±0.30	**35.88±0.50**	25.70±1.28	31.20±0.25	32.02±0.30	31.51±0.27	32.98±0.25
	100k	Agrawal(F1-F5)	66.89±0.40	**68.72±0.44**	66.36±0.31	57.42±0.52	64.96±0.13	66.62±0.33	67.67±0.42	67.90±0.34
		Agrawal(F6-F10)	84.51±0.37	**85.78±0.21**	73.09±0.30	73.08±0.97	84.56±0.23	84.74±0.24	85.37±0.18	85.59±0.10
		LED	71.57±0.14	**72.62±0.11**	70.59±0.20	55.05±1.37	71.10±0.13	70.99±0.11	71.56±0.11	71.69±0.11
		Mixed	90.96±0.11	90.14±0.17	90.76±0.17	71.50±1.01	90.99±0.08	90.99±0.08	90.90±0.11	**91.05±0.10**
		Sine	89.04±0.16	88.88±0.17	88.16±0.21	76.57±0.58	86.84±0.17	89.51±0.11	**89.77±0.09**	89.40±0.09
		Waveform	79.71±0.12	80.25±0.12	79.52±0.16	63.61±0.97	80.13±0.12	80.34±0.11	80.45±0.16	**80.60±0.12**
		RandomRBF	31.86±0.32	31.83±0.28	**36.91±0.39**	26.25±0.66	31.63±0.20	32.51±0.22	31.82±0.22	33.39±0.21
		Rank 15%	4.41	2.86	5.52	8.00	5.74	4.41	3.10	**1.95**
30%	20k	Agrawal(F1-F5)	62.81±0.38	**64.04±0.25**	63.94±0.29	59.04±0.63	62.44±0.27	62.49±0.28	63.07±0.24	63.57±0.26
		Agrawal(F6-F10)	80.64±1.04	83.05±0.41	69.56±0.98	74.94±0.99	81.74±0.39	82.05±0.31	82.94±0.21	**83.32±0.19**
		LED	68.47±0.23	**69.55±0.22**	61.85±0.49	64.28±0.94	67.01±0.20	66.96±0.22	68.23±0.21	68.35±0.19
		Mixed	89.65±0.22	89.84±0.25	87.80±0.27	68.83±0.80	89.42±0.16	89.44±0.15	89.90±-0.19	**89.95±0.18**
		Sine	87.08±0.19	87.51±0.18	85.22±0.32	75.63±0.72	86.05±0.30	87.33±0.20	**87.85±0.20**	87.69±0.20
		Waveform	79.28±0.29	**79.77±0.26**	78.83±0.32	74.41±0.89	79.01±0.27	79.03±0.27	79.26±0.22	79.55±0.21
		RandomRBF	31.37±0.40	31.14±0.41	31.23±0.51	27.11±0.94	30.90±0.37	31.64±0.39	31.91±0.33	**32.55±0.32**
	50k	Agrawal(F1-F5)	66.87±0.40	**67.93±0.40**	66.28±0.27	60.81±0.32	64.85±0.18	66.49±0.34	67.39±0.38	67.27±0.34
		Agrawal(F6-F10)	84.24±0.30	**85.28±0.14**	72.61±0.37	81.56±0.46	84.32±0.21	84.49±0.20	85.11±0.14	85.14±0.15
		LED	71.46±0.15	**72.20±0.16**	63.26±0.30	69.80±0.51	70.93±0.14	70.92±0.14	71.30±0.15	71.36±0.13
		Mixed	90.87±0.12	90.63±0.14	88.55±0.21	81.63±0.63	90.84±0.11	90.87±0.12	91.10±0.11	**91.21±0.11**
		Sine	88.88±0.16	88.93±0.15	86.06±0.16	84.33±0.38	86.89±0.17	89.45±0.12	**89.81±0.11**	89.53±0.10
		Waveform	79.76±0.20	80.06±0.15	79.49±0.28	76.80±0.34	80.01±0.21	80.27±0.19	80.33±0.16	**80.43±0.15**
		RandomRBF	31.88±0.27	31.92±0.17	31.45±0.29	27.36±0.76	31.72±0.22	32.49±0.20	32.08±0.16	**33.08±0.20**
	100k	Agrawal(F1-F5)	70.26±0.27	**71.08±0.29**	68.30±0.27	61.16±0.40	65.60±0.15	70.06±0.35	70.52±0.25	70.54±0.29
		Agrawal(F6-F10)	85.39±0.38	**86.22±0.11**	73.41±0.39	83.28±0.33	85.36±0.13	85.88±0.09	86.20±0.10	86.18±0.08
		LED	72.66±0.13	**73.06±0.12**	63.48±0.25	71.31±0.22	72.42±0.12	72.39±0.13	72.63±0.12	72.68±0.12
		Mixed	91.08±0.09	91.08±0.09	88.84±0.14	86.26±0.30	91.41±0.08	91.60±0.08	**91.79±0.10**	91.78±0.09
		Sine	90.25±0.15	89.59±0.20	86.76±0.14	86.97±0.28	87.12±0.11	90.53±0.08	**90.93±0.10**	90.42±0.07
		Waveform	79.88±0.12	80.20±0.12	79.83±0.15	76.93±0.14	80.51±0.11	**81.00±0.12**	80.74±0.12	80.83±0.10
		RandomRBF	32.35±0.21	32.27±0.28	31.69±0.24	25.84±0.70	32.03±0.18	32.78±0.20	32.12±0.16	**33.33±0.17**
		Rank 30%	4.33	2.88	6.91	7.67	5.67	3.93	2.67	**1.95**
		Rank	4.38	2.87	6.22	7.83	5.70	4.17	2.88	**1.95**

Um ponto a ser considerado é que os resultados de acurácia reportados são influenciados pelo uso de detectores de mudanças de conceito e que a integração do RDDM e $HDDM_A$ com comitês de classificadores impacta positivamente seu desempenho, permitindo uma adaptação mais eficiente a mudanças abruptas e graduais em comparação com métodos de adaptação implícita. No entanto, RDDM e $HDDM_A$ dependem do acompanhamento sequencial do número de erros cometidos pelo método de classificação base, o que se torna inviável em cenários semi-supervisionados com um grande número de instâncias não rotuladas. Essa necessidade por um maior número de instâncias rotuladas é resolvida pela incorporação da nova abordagem de *self-training*.

Os resultados demonstram que essa proposta é eficaz, o que é mais evidente ao comparar o FASE com o HT. O maior acesso a rótulos permite que o método de classificação

Tabela 12 – Médias de acurácia em (%), com intervalos de confiança de 95%, em base de dados artificiais com mudanças de conceito graduais.

%RÓT.	TAM.	COMITÊ CLASSIFICADOR BASE BASE \ DETECTOR	— HT RDDM	— HT_S RDDM	BDF HT RDDM	DSDD 7HT_3NB $HDDM_A$	FASE HT $HDDM_A$	$FASE_D$ 7HT_3NB $HDDM_A$	$FASE_S$ HT $HDDM_A$	$FASE_{DS}$ 7HT_3NB $HDDM_A$
15%	20k	Agrawal(F1-F5)	60.62±0.37	**62.33±0.35**	61.92±0.31	54.96±0.87	59.68±0.38	59.93±0.22	61.47±0.26	61.73±0.23
		Agrawal(F6-F10)	76.15±2.01	**81.34±0.39**	67.47±1.04	67.92±0.85	78.33±0.62	79.35±0.43	80.71±0.21	81.16±0.25
		LED	64.13±0.38	**66.18±0.23**	60.88±0.38	45.09±1.87	62.71±0.39	62.62±0.39	64.66±0.30	64.65±0.27
		Mixed	**86.23±0.16**	85.28±0.28	85.65±0.30	56.46±0.80	85.62±0.17	85.64±0.16	86.02±0.18	85.92±0.18
		Sine	**83.99±0.25**	83.91±0.30	83.31±0.33	58.20±0.54	82.97±0.19	83.20±0.22	83.45±0.19	83.39±0.20
		Waveform	78.15±0.33	**79.02±0.28**	77.22±0.38	62.41±1.10	77.91±0.33	77.87±0.34	78.29±0.23	78.62±0.28
		RandomRBF	31.14±0.49	30.90±0.43	**33.87±0.69**	26.90±1.19	30.47±0.37	30.83±0.40	31.61±0.41	32.46±0.30
	50k	Agrawal(F1-F5)	63.66±0.40	**64.80±0.35**	64.04±0.32	54.69±1.41	63.01±0.27	62.96±0.28	63.99±0.32	64.43±0.23
		Agrawal(F6-F10)	81.62±1.37	**84.08±0.45**	70.98±0.66	70.04±0.71	82.57±0.32	82.73±0.39	83.62±0.16	84.02±0.13
		LED	69.66±0.19	**70.69±0.18**	67.36±0.36	51.04±1.50	68.59±0.17	68.57±0.19	69.51±0.19	69.69±0.17
		Mixed	**89.46±0.14**	88.54±0.25	88.77±0.21	61.87±1.18	89.24±0.13	89.21±0.14	89.04±0.15	89.02±0.16
		Sine	87.24±0.16	87.28±0.17	86.05±0.20	65.50±1.04	85.85±0.19	87.36±0.17	**87.46±-0.12**	87.25±0.11
		Waveform	79.38±0.22	79.79±0.16	78.93±0.27	63.25±1.25	79.37±0.20	79.38±0.20	79.68±0.19	**79.89±0.19**
		RandomRBF	31.61±0.36	31.60±0.30	**36.09±0.44**	26.53±1.03	31.30±0.29	31.96±0.33	31.66±0.28	33.05±0.29
	100k	Agrawal(F1-F5)	66.80±0.42	**68.46±0.35**	66.15±0.35	57.53±0.43	64.69±0.19	66.65±0.40	67.15±0.42	67.36±0.37
		Agrawal(F6-F10)	84.52±0.25	**85.52±0.10**	73.02±0.58	74.72±1.25	84.39±0.18	84.46±0.22	85.05±0.14	85.26±0.09
		LED	71.63±0.14	**72.42±0.13**	70.37±0.16	54.94±1.38	71.04±0.12	71.00±0.11	71.53±0.12	71.66±0.13
		Mixed	**90.69±0.07**	89.56±0.11	90.44±0.15	70.60±1.31	90.56±0.06	90.57±0.06	90.43±0.07	90.41±0.08
		Sine	88.93±0.11	88.71±0.11	87.71±0.21	76.08±0.73	86.41±0.14	89.06±0.11	**89.15±0.10**	88.79±0.10
		Waveform	79.83±0.14	80.08±0.10	79.49±0.15	64.41±0.99	79.97±0.18	80.29±0.12	80.35±0.13	**80.51±0.11**
		RandomRBF	31.80±0.31	31.94±0.21	**36.73±0.34**	25.81±0.85	31.65±0.18	32.46±0.20	31.86±0.21	33.35±0.18
		Rank 15%	3.88	2.95	5.24	7.90	5.71	4.50	3.14	**2.67**
30%	20k	Agrawal(F1-F5)	62.27±0.41	63.39±0.23	**63.41±0.30**	58.64±0.41	61.82±0.21	61.78±0.21	62.34±0.27	62.95±0.23
		Agrawal(F6-F10)	79.12±1.40	**81.90±0.48**	68.69±1.01	74.80±1.12	80.23±0.50	80.72±0.40	81.28±0.26	81.67±0.23
		LED	67.85±0.22	**68.49±0.22**	60.92±0.41	63.02±0.72	66.35±0.24	66.33±0.26	67.41±0.23	67.55±0.22
		Mixed	**86.94±0.20**	86.63±0.29	85.85±0.29	68.37±0.83	86.84±0.18	86.85±0.18	86.60±0.18	86.61±0.17
		Sine	85.00±0.20	84.79±0.21	83.45±0.29	74.43±0.85	83.98±0.23	84.92±0.18	**85.07±0.17**	84.95±0.17
		Waveform	78.60±0.26	**79.05±0.26**	78.10±0.34	75.24±0.40	78.67±0.28	78.61±0.28	78.77±0.25	78.88±0.23
		RandomRBF	31.31±0.39	31.48±0.43	31.28±0.49	27.05±0.96	30.88±0.33	31.62±0.34	31.92±0.41	**32.47±0.33**
	50k	Agrawal(F1-F5)	66.39±0.35	**67.25±0.32**	65.82±0.23	60.09±0.28	64.50±0.15	66.14±0.26	66.94±0.35	66.76±0.32
		Agrawal(F6-F10)	83.98±0.31	**84.93±0.14**	72.52±0.39	81.40±0.64	83.93±0.25	84.03±0.16	84.52±0.12	84.70±0.11
		LED	71.27±0.15	**71.76±0.16**	63.06±0.32	69.00±0.50	70.68±0.15	70.62±0.16	70.97±0.18	71.07±0.17
		Mixed	89.83±0.12	89.22±0.12	87.74±0.17	80.69±0.67	89.86±0.09	89.85±0.09	89.88±0.09	**89.90±0.10**
		Sine	88.39±0.14	88.18±0.11	85.57±0.22	83.78±0.31	86.24±0.15	88.61±0.12	**88.74±0.13**	88.47±0.13
		Waveform	79.55±0.19	79.74±0.14	79.37±0.24	76.80±0.21	79.73±0.21	80.10±0.17	80.09±0.19	**80.24±0.14**
		RandomRBF	31.95±0.32	31.87±0.24	31.42±0.26	26.59±0.87	31.63±0.22	32.42±0.20	32.05±0.26	**33.02±0.22**
	100k	Agrawal(F1-F5)	70.12±0.32	**70.72±0.27**	68.19±0.27	61.21±0.29	65.47±0.15	69.94±0.29	70.30±0.27	70.35±0.22
		Agrawal(F6-F10)	85.21±0.40	**86.15±0.08**	73.31±0.49	82.94±0.38	85.16±0.14	85.66±0.11	85.90±0.12	85.98±0.07
		LED	72.60±0.13	72.92±0.11	63.23±0.22	71.06±0.24	72.29±0.13	**73.22±0.11**	72.44±0.13	72.52±0.13
		Mixed	90.67±0.07	90.58±0.08	88.56±0.15	86.27±0.38	90.97±0.07	91.05±0.06	**91.12±0.07**	91.10±0.07
		Sine	90.10±0.12	89.60±0.17	86.50±0.16	86.65±0.33	86.87±0.13	90.14±0.09	**90.42±0.10**	89.92±0.09
		Waveform	79.90±0.12	79.99±0.12	79.85±0.13	76.88±0.20	80.45±0.10	**82.26±0.10**	80.66±0.11	80.80±0.12
		RandomRBF	32.12±0.20	32.20±0.20	31.73±0.27	26.30±0.64	31.87±0.20	32.7±40.16	32.07±0.17	**33.23±0.18**
		Rank 30%	4.14	3.05	6.90	7.67	5.38	3.52	2.95	**2.38**
		Rank	4.01	3.00	6.07	7.79	5.55	4.01	3.05	**2.52**

Tabela 13 – Médias de acurácia em (%), com intervalos de confiança de 95%, em base de dados reais.

%RÓT.	COMITÊ CLASSIFICADOR BASE BASE \ DETECTOR	— HT RDDM	— HT_S RDDM	BDF HT RDDM	DSDD 7HT_3NB $HDDM_A$	FASE HT $HDDM_A$	$FASE_D$ 7HT_3NB $HDDM_A$	$FASE_S$ HT $HDDM_A$	$FASE_{DS}$ 7HT_3NB $HDDM_A$
15%	Airlines	64.77	65.64	**66.35**	60.60	65.91	65.87	65.27	65.55
	Connect4	70.77	71.29	**73.55**	63.10	69.37	69.80	71.76	71.69
	Convertype	78.42	75.42	78.01	67.31	79.50	79.77	85.84	**85.93**
	Spam_data	89.57	89.06	89.99	74.87	89.71	89.93	91.14	**91.28**
	Weather	68.88	70.02	68.94	65.34	69.19	69.85	70.99	**71.32**
	Rank 15%	6.00	5.00	3.40	8.00	4.60	4.00	2.80	**2.20**
30%	Airlines	64.82	65.51	**68.24**	64.52	66.49	66.49	65.89	66.37
	Connect4	72.46	73.00	**73.58**	69.14	71.38	71.74	72.71	72.68
	Convertype	80.24	80.01	68.07	77.38	81.55	81.70	86.46	**86.53**
	Spam_data	90.47	90.27	88.93	89.82	90.61	90.96	91.70	**91.75**
	Weather	69.99	70.86	70.35	66.51	69.19	70.85	71.24	**71.47**
	Rank 30%	5.60	4.60	4.60	7.60	4.90	3.70	2.80	**2.20**
	Rank	5.80	4.80	4.00	7.80	4.75	3.85	2.80	**2.20**

simples (HT) supere o comitê de classificadores (FASE).

Uma avaliação estatística utilizando o teste *pós-hoc* de *Nemenyi* (DEMSAR, 2006) também foi realizada para determinar se havia diferenças significativas entre os métodos comparados. Os resultados estão apresentados graficamente nas Figuras 13 e 14 para cenários com mudanças abruptas e graduais, respectivamente. A análise revela que, em todos os cenários testados, os métodos que incorporam a abordagem proposta de *self-training* estão consistentemente posicionados acima dos outros, com o método FASE_{DS} liderando em todos os casos.

As Figuras 13(c) e 14(c) apresentam os resultados gerais, agregando os valores de 15% e 30%. Na Figura 13(c), o FASE_{DS} não apresenta uma diferença estatisticamente significativa em relação ao HT_S e FASE_S. Vale destacar que o HT_S é um método simples, utilizando apenas um classificador e um detector de mudança. Apesar de sua simplicidade, o método HT_S geralmente garantiu a segunda posição nos rankings.

Nos testes utilizando conjuntos de dados com mudanças graduais, como ilustrado na Figura 14(c), o FASE_{DS} permanece como o método de melhor desempenho, e a ordem dos rankings se mantém bastante semelhante. No entanto, o FASE_{DS} não apresenta uma diferença significativa em relação aos métodos HT_S, FASE_S, HT e FASE_D.

Por fim, a Tabela 13 e a Figura 15 apresentam os resultados dos experimentos conduzidos com conjuntos de dados do mundo real. Observa-se que, tanto nos cenários com 15% quanto com 30% de dados rotulados, o método de melhor desempenho foi o FASE_{DS}. É importante destacar que os métodos que aplicaram a nova abordagem de *self-training* novamente se classificaram acima de suas versões base. No entanto, de forma geral, não foi observada uma diferença significativa entre os métodos, exceto para os métodos HT e $\text{DSDD\_7HT\_3NB\_HDDM}_A$, que foram os piores classificados.

A ideia de combinar as abordagens melhorou o desempenho dos modelos de classificação em termos de acurácia, como pode-se perceber nos resultados acima apresentados. Nós consideramos que a sinergia dessas técnicas é dada por: os detectores de mudanças de conceito são mecanismos que sinalizam quando o desempenho do classificador está se degradando, e substituem o modelo de aprendizado; ao combinar com *self-training*, os detectores podem evitar a propagação de erro causada pela rotulagem errada de exemplos. Por sua vez, o *self-training* traz o aumento no acesso à informação de entrada aos detectores. Já os comitês de classificadores reduzem o risco de decisões erradas por causa de classificadores fracos ou desatualizados, explorando a diversidade para aumentar a

robustez da decisão.

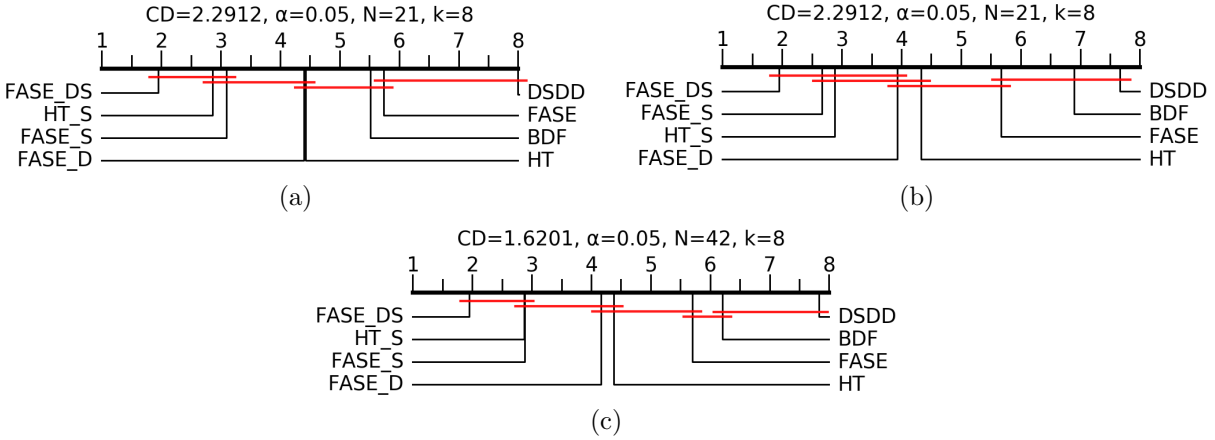


Figura 13 – Comparação estatística das acurácias em todos os cenários de rotulação: (a) 15%, (b) 30% e (c) geral, utilizando o teste F_F e o pós-teste de *Nemenyi*, com intervalos de confiança de 95%, em bases de dados artificiais com mudanças abruptas de conceito.

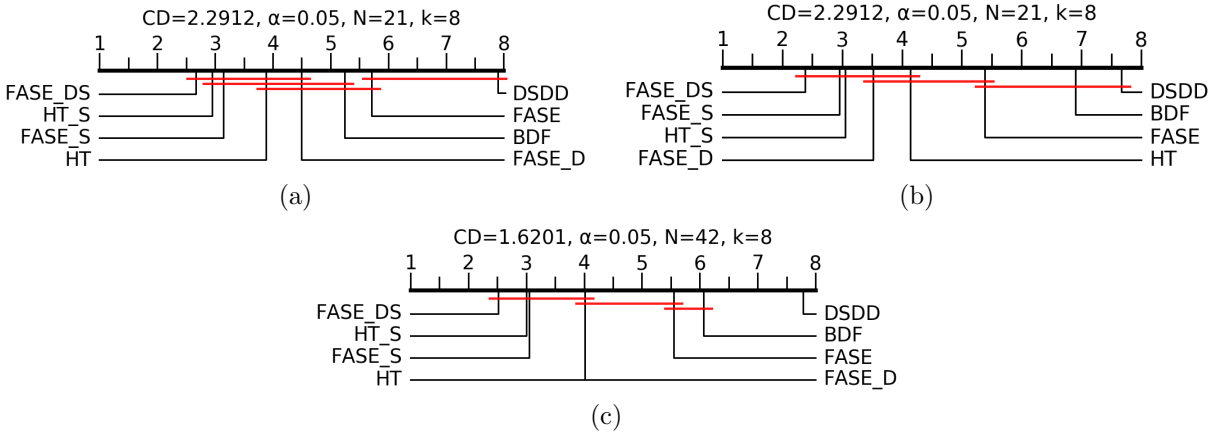


Figura 14 – Comparação estatística das acurácias em todos os cenários de rotulação: (a) 15%, (b) 30% e (c) geral, utilizando o teste F_F e o pós-teste de *Nemenyi*, com intervalos de confiança de 95%, em bases de dados artificiais com mudanças gradual de conceito.

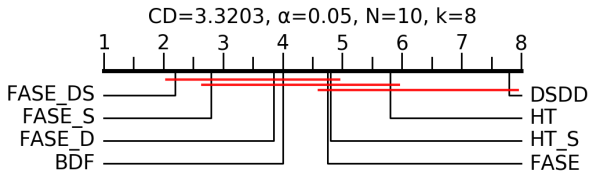


Figura 15 – Comparação estatística das acurácias em todos os cenários de rotulação, utilizando o teste F_F e o pós-teste de *Nemenyi*, com intervalos de confiança de 95%, em bases de dados do mundo real.

7.3 CONSIDERAÇÕES FINAIS

Neste capítulo, foi investigada a aplicação de uma estratégia de *self-training* para aumentar a quantidade de rótulos disponíveis aos métodos de classificação. Essa abordagem

demonstrou beneficiar diretamente o desempenho dos detectores de mudanças de conceito, o que, por sua vez, contribuiu para melhorar a acurácia dos modelos. Para validar as hipóteses 3 e 4 propostas nesta tese, foram realizadas análises empíricas e estatísticas que compararam algoritmos de classificação incorporando a nova estratégia de *self-training*, mecanismos de detecção de mudanças de conceito e diversidade (no caso dos comitês). A acurácia foi utilizada como a principal métrica de avaliação.

Os resultados obtidos evidenciaram que a integração das três abordagens – detecção de mudanças de conceito, introdução de diversidade nos comitês de classificadores e *self-training* – constitui uma alternativa viável e promissora. Essa combinação entre aprendizado supervisionado e técnicas semi-supervisionadas demonstrou ser eficaz em melhorar o desempenho dos modelos, oferecendo soluções robustas para lidar com fluxos de dados sujeitos a mudanças na distribuição. No entanto, é importante ressaltar que a adoção dessa estratégia implica custos adicionais de tempo e recursos computacionais, os quais devem ser considerados em sua implementação prática. Outro ponto a mencionar é que as informações sobre os testes e a análise dos resultados deste capítulo foram publicadas em (PÉREZ; BARROS; SANTOS, 2025).

8 CONCLUSÕES

Esta tese teve como objetivo geral explorar a viabilidade do uso de detectores de mudanças de conceito, de técnicas para promover a diversidade em comitês de classificadores, e de uma abordagem de *self-training*, bem como de sua combinação, para aumentar a eficácia e a adaptabilidade de métodos de classificação supervisionada em ambientes semi-supervisionados. Alinhado como este objetivo geral, ao longo da tese foram realizados experimentos para validar as quatro hipotes levantadas: 1) A utilização de métodos de detecção de mudanças de conceito permite a adaptação dos métodos de classificação às alterações na distribuição dos dados, melhorando o desempenho em cenários de aprendizado semi-supervisionado; 2) Promover diversidade nos comitês de classificadores com a combinação de vários tipos de métodos resulta em maior robustez e capacidade de generalização, contribuindo para um desempenho superior em cenários de escassez de rótulos e presença de mudanças de conceitos; 3) O uso de *self-training* para pseudo-rotular dados não rotulados aumenta a quantidade de informações disponíveis para o treinamento, mitigando os efeitos da escassez de rótulos e melhorando a acurácia dos métodos de classificação; e 4) A acurácia dos métodos de classificação supervisionados em cenários semi-supervisionados pode ser significativamente melhorada pela integração das três abordagens experimentadas. i.e. detecção de mudanças de conceito, uso de diversidade nos comitês de classificadores, e *self-training*.

Na tese, tornou-se evidente a necessidade de ferramentas que facilitassem a experimentação e implementação da ideia central. Para viabilizar essa ideia na prática e avaliar corretamente os resultados, foi necessário adicionar novas funcionalidades ao *Massive Online Analysis* (MOA). Os recursos desenvolvidos permitiram ocultar os rótulos de um subconjunto das instâncias das bases de dados a serem testadas, tanto para métodos classificadores quanto para detectores. Esses recursos permitem controlar a ocultação por meio de parâmetros e possibilitam a geração de bases de dados de cenários semi-supervisionados no formato *arff*. A ideia é facilitar a execução em outros ambientes, ampliando a experimentação e a comparação com métodos desenvolvidos fora da plataforma MOA.

Os experimentos realizados foram divididos em três grupos com o objetivo de validar as quatro hipóteses levantadas. O primeiro dos grupos é focado em validar a primeira hipótese, para a qual realizam-se os testes com dois classificadores – *Hoeffding Tree* (HT) e

Naïve Bayes (NB) – combinados com quatro métodos detectores de mudanças de conceito supervisionados e dois métodos semi-supervisionados. Foi utilizado um número razoavelmente grande de bases de dados artificiais e reais, configuradas com 15%, 30% das instâncias com rótulos. Nos resultados, os desempenhos dos métodos supervisionados RDDM, HDDM_A e FHDDM foram melhores nas bases de dados com mudanças abruptas e graduais em quase todas as configurações testadas. Quanto aos métodos semi-supervisionados, os melhores resultados foram obtidos pelo BDF_{RDDM}. Além disso, testes com 25% e 100% dos rótulos foram realizados com o objetivo de servir como suporte adicional, e seus resultados confirmam o comportamento dos métodos. Os resultados das análises e testes estatísticos realizados neste grupo permitem afirmar que a utilização de detectores supervisionados em ambientes semi-supervisionados é válida e pode apresentar excelentes resultados, sinalizando uma possível mudança de paradigma para pesquisas futuras na área.

O segundo grupo dos experimentos se concentra em validar a segunda das hipóteses, fornecendo parâmetros para a inclusão de diversidade nos comitês de classificadores, além de demonstrar como o uso desses parâmetros e dos detectores de mudanças de conceito pode melhorar o desempenho dos modelos de aprendizado. Os testes da diversidade introduzida pela variação do número e do tipo de classificadores presentes nos comitês demonstram que, quando HT e NB são combinados, a melhor configuração ocorre com a mistura de 7 HT + 3 NB com o uso de esquema de votação majoritária simples. Esses parâmetros foram validados para cenários supervisionados e semi-supervisionados, onde é demonstrado que promover diversidade resulta em maior robustez e capacidade de generalização, contribuindo para um desempenho superior neste cenário de escassez de rótulos.

O terceiro e último grupo de experimentos foi realizado para responder se as hipóteses terceira e quarta seriam aceitas ou refutadas. Os testes mostram o bom desempenho da nova proposta de *self-training*, introduzida com o objetivo de mitigar os efeitos da escassez de rótulos e melhorar a acurácia dos métodos de classificação. Também neste grupo foi validada a combinação das estratégias de diversidade com a técnica de *self-training*, resultando no método FASE_{DS} que apresenta notável performance nos cenários semi-supervisionados.

Finalmente, se pode afirmar que os resultados obtidos nas experimentações nesta tese permitiram validar as quatro hipóteses levantadas, as quais foram criadas em concordância com o objetivo geral do estudo. Hipóteses formuladas com base na premissa de que

a criação, utilização e integração de estratégias como detecção de mudanças de conceito, introdução de diversidade em comitês de classificadores e *self-training* poderiam melhorar o desempenho dos modelos de aprendizado em ambientes semi-supervisionados. A análise empírica e estatística realizada demonstrou que essas abordagens não apenas atenderam às expectativas, mas também reforçaram a relevância do objetivo geral da tese, que buscava explorar métodos eficazes e robustos para lidar com cenários semi-supervisionados dinâmicos.

8.1 CONTRIBUIÇÕES

A seguir, são listadas as principais contribuições desta tese:

- Nova proposta de *self-training* para aumentar os dados de treinamento.
- Validação que a combinação da diversidade nos comitês aliada ao *self-training* resulta em métodos de desempenho superior em ambientes semi-supervisionados.
- Validação do uso de detectores de mudanças de conceito em cenários de aprendizado com dados semi-supervisionados.
- Estabelece parâmetros para introduzir diversidade nos comitês de classificadores em cenários supervisionados e semi-supervisionados.
- Atualização e adaptação do *framework Massive Online Analysis* (MOA). As novas funcionalidades implementadas permitem a simulação do aprendizado em cenários semi-supervisionados na presença de mudanças de conceito, gerando a ferramenta *MOAManagerSS*.

8.2 TRABALHOS FUTUROS

Abaixo estão listadas algumas possíveis direções futuras para esta pesquisa:

- Realizar experimentos adicionais em cenários de aprendizado semi-supervisionado, utilizando outros métodos, como os classificadores KNN (EVELYN; JOSEPH, 1951) e SVM (CORTES; VAPNIK, 1995), além dos comitês BOLE (BARROS; SANTOS; GONÇALVES JR., 2016), OACE (VERDECIA-CABRERA; FRÍAS-BLANCO; CARVALHO, 2018)

e os métodos propostos por Santos e Barros (2020), OABM1 e *Online AdaBoost-based M2* (OABM2). Assim como, utilizando os métodos *Dynamic Classifier Selection with Local Accuracy* (DCL-LA) e *Dynamic Classifier Selection based on Multiple Classifier Behavior* (DS-MCB), ambos propostos em (PINAGÉ; SANTOS; GAMA, 2020)

- Aplicar os comitês de detectores *Drift Detection Ensemble* (DDE) (MACIEL; SANTOS; BARROS, 2015) e *Statistical Tests Ensemble Drift Detector* (STED) (PÉREZ; BARROS; SANTOS, 2020) nos métodos de classificação utilizados e analisar o desempenho em ambientes semi-supervisionados.

- Utilizar outras estratégias, como *Q-Statistic* ou medidas de discordância (ABUASSBA et al., 2017), para garantir e quantificar a diversidade dos comitês.

- O método de *self-training* proposto, assim como sua combinação com estratégias de diversidade quando aplicado a comitês, pode ser adaptado para outros comitês de classificadores, como *Leveraging Bagging* (LevBag) (BIFET et al., 2010) e aqueles mencionados no primeiro item desta lista. Além disso, a experimentação pode incluir diferentes combinações de esquemas de votação para comitês, a fim de avaliar o comportamento da abordagem proposta.

- Incluir testes de performance focados na análise da complexidade computacional, bem como a aplicação de técnicas de otimização de algoritmos que visem reduzir o custo computacional sem comprometer a acurácia.

- Utilizar testes para avaliar o comportamento do método *self-training* proposto com dados ruidosos, corrompidos ou *Out-of-Distribution* (OOD). Alterações nos dados como adicionar pertubações ou remover subconjuntos de dados permitem medir e avaliar a degradação do desempenho do modelo, assim como, a capacidade do modelo rejeitar entradas desconhecidas.

REFERÊNCIAS

ABUASSBA, A. O.; ZHANG, D.; LUO, X.; SHAHERYAR, A.; ALI, H. Improving classification performance through an advanced ensemble based heterogeneous extreme learning machines. *Computational intelligence and neuroscience*, Wiley Online Library, v. 2017, n. 1, p. 3405463, 2017.

AGGARWAL, C. C. *Data classification: algorithms and applications*. [S.l.]: CRC Press, 2014.

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.

AHMADI, Z.; BEIGY, H. Semi-supervised ensemble learning of data streams in the presence of concept drift. *Hybrid Artificial Intelligent Systems*, Springer, p. 526–537, 2012.

AMINI, M.; FEOFANOV, V.; PAULETTO, L.; HADJADJ, L.; DEVIJVER, E.; MAXIMOV, Y. *Self-Training: A Survey*. 2024.

ANGLUIN, D.; LAIRD, P. Learning from noisy examples. *Machine learning*, Springer, v. 2, p. 343–370, 1988.

ANKERST, M.; BREUNIG, M. M.; KRIEGEL, H.-P.; SANDER, J. Optics: ordering points to identify the clustering structure. *SIGMOD Rec.*, Association for Computing Machinery, v. 28, n. 2, p. 49–60, 1999.

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

ATALLAH, D. M.; BADAWY, M.; EL-SAYED, A. Intelligent feature selection with modified k-nearest neighbor for kidney transplantation prediction. *SN Applied Sciences*, Springer, v. 1, p. 1–17, 2019.

ATTAR, V.; CHAUDHARY, P.; RAHAGUDE, S.; CHAUDHARI, G.; SINHA, P. An instance-window based classification algorithm for handling gradual concept drifts. In: . [S.l.]: Springer Berlin Heidelberg, 2012. p. 156–172.

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.

BANDYOPADHYAY, S.; MAULIK, U. An evolutionary technique based on k-means algorithm for optimal clustering in rn. *Information Sciences*, v. 146, n. 1, p. 221–237, 2002.

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

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

BARROS, R. S. M.; SANTOS, S. G. T. C. A Large-scale Comparison of Concept Drift Detectors. *Information Sciences*, Elsevier, v. 451-452, n. C, p. 348–370, 2018.

BARROS, R. S. M.; SANTOS, S. G. T. C. An overview and comprehensive comparison of ensembles for concept drift. *Information Fusion*, Elsevier, v. 52, n. C, p. 213–244, 2019.

BARROS, R. S. M.; SANTOS, S. G. T. C.; GONÇALVES JR., P. M. A boosting-like online learning ensemble. In: *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN)*. Vancouver, Canada: [s.n.], 2016. p. 1871–1878.

BASU, S. *Semi-supervised clustering: probabilistic models, algorithms and experiments*. [S.l.]: The University of Texas at Austin, 2005.

BASU, S.; BANERJEE, A.; MOONEY, R. Semi-supervised clustering by seeding. In: CITESEER. *In Proceedings of 19th International Conference on Machine Learning (ICML-2002*. [S.l.], 2002.

BENNETT, K. P.; DEMIRIZ, A. Semi-supervised support vector machines. In: *Advances in Neural Information processing systems*. [S.l.: s.n.], 1999. p. 368–374.

BENNETT, K. P.; DEMIRIZ, A.; MACLIN, R. Exploiting unlabeled data in ensemble methods. In: ACM. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2002. p. 289–296.

BENTLEY, J. L. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, ACM New York, NY, USA, v. 18, n. 9, p. 509–517, 1975.

BIEMANN, C. Chinese whispers-an efficient graph clustering algorithm and its application to natural language processing problems. In: *Proceedings of TextGraphs: the first workshop on graph based methods for natural language processing*. [S.l.: s.n.], 2006. p. 73–80.

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.; 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.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *Journal of machine Learning research*, v. 3, n. Jan, p. 993–1022, 2003.

BLUM, A.; CHAWLA, S. Learning from labeled and unlabeled data using graph mincuts. *In Proceeding of Eighteenth International Conference on Machine Learning*, p. 19–26, 2001.

BLUM, A.; MITCHELL, T. Combining labeled and unlabeled data with co-training. In: ACM. *Proceedings of the eleventh annual conference on Computational learning theory*. [S.l.], 1998. p. 92–100.

BOQIANG, R.; CHUANWEN, J. A review on the economic dispatch and risk management considering wind power in the power market. *Renewable and Sustainable Energy Reviews*, v. 13, n. 8, p. 2169–2174, 2009.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: ACM. *Proceedings of the fifth annual workshop on Computational learning theory*. [S.l.], 1992. p. 144–152.

BREIMAN, L. *Bias, variance, and arcing classifiers*. [S.l.], 1996.

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).

BRUCE, R. F. A bayesian approach to semi-supervised learning. In: *NLPRS*. [S.l.: s.n.], 2001. p. 57–64.

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

CAMPELLO, R. J. G. B.; MOULAVI, D.; SANDER, J. Density-based clustering based on hierarchical density estimates. In: *Advances in Knowledge Discovery and Data Mining*. [S.l.]: Springer Berlin Heidelberg, 2013. p. 160–172.

COLLINS, M.; SINGER, Y. Unsupervised models for named entity classification. In: *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. [S.l.: s.n.], 1999.

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.

COVER, T.; HART, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, v. 13, n. 1, p. 21–27, 1967.

CRUZ, D. P. F. *Agrupamento e classificação de dados utilizando um algoritmo inspirado no comportamento de abelhas*. 2015. M.Sc. Dissertation, Universidade Presbiteriana Mackenzie.

CUNNINGHAM, P.; DELANY, S. J. K-nearest neighbour classifiers-a tutorial. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 54, n. 6, p. 1–25, 2021.

DAWID, A. P. Present position and potential developments: Some personal views: Statistical theory: The prequential approach. *Journal of the Royal Statistical Society. Series A (General)*, JSTOR, p. 278–292, 1984.

DAWID, A. P.; VOVK, V. G. Prequential probability: Principles and properties. *Bernoulli*, Bernoulli Society for Mathematical Statistics and Probability, 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.

DEMIRIZ, A.; BENNETT, K. P. Optimization approaches to semi-supervised learning. In: *Complementarity: Applications, Algorithms and Extensions*. [S.l.]: Springer, 2001. p. 121–141.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, JSTOR, p. 1–38, 1977.

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

DITZLER, G.; POLIKAR, R. Semi-supervised learning in nonstationary environments. In: IEEE. *Neural Networks (IJCNN), The 2011 International Joint Conference on*. [S.l.], 2011. p. 2741–2748.

DONGEN, S. V. Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications*, v. 30, n. 1, p. 121–141, 2008.

DU, L.; SONG, Q.; JIA, X. Detecting concept drift: An information entropy based method using an adaptive sliding window. *Intelligent Data Analysis*, 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.

DUARTE, J. M.; BERTON, L. A review of semi-supervised learning for text classification. *Artificial intelligence review*, Springer, v. 56, n. 9, p. 9401–9469, 2023.

DUNN, J. C. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. Taylor & Francis, 1973.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *kdd*. [S.l.: s.n.], 1996. v. 96, n. 34, p. 226–231.

EVELYN, F.; JOSEPH, L. H. *Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties*. [S.l.], 1951.

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.

FERRER-TROYANO, F. J.; AGUILAR-RUIZ, J. S.; RIQUELME-SANTOS, J. C. 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.

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

FREUND, Y. Boosting a weak learning algorithm by majority. *Inform. and Computation*, v. 121, n. 2, p. 256–285, 1995.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, v. 55, n. 1, p. 119–139, 1997. ISSN 0022-0000.

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

FRÍAS-BLANCO, I.; CAMPO-ÁVILA, J. del; RAMOS-JIMÉNEZ, G.; MORALES-BUENO, R.; ORTIZ-DÍAZ, A.; CABALLERO-MOTA, Y. Online and non-parametric drift detection methods based on hoeffding's bounds. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 27, n. 3, p. 810–823, 2015.

FRÍAS-BLANCO, I.; CAMPO-ÁVILA, J.; RAMOS-JIMÉNEZ, G.; CARVALHO, A. C. P. L. F.; ORTIZ-DÍAZ, A.; MORALES-BUENO, R. Online adaptive decision trees based on concentration inequalities. *Knowledge-Based Systems*, v. 104, p. 179 – 194, 2016.

FRÍAS-BLANCO, I.; VERDECIA-CABRERA, A.; ORTIZ-DÍAZ, A.; CARVALHO, A. C. P. L. F. Fast adaptive stacking of ensembles. In: *Proceedings of the 31st ACM Symposium on Applied Computing (SAC)*. Pisa, Italy: [s.n.], 2016. p. 929–934.

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.; 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.

GARRIDO-LABRADOR, J. L.; SERRANO-MAMOLAR, A.; MAUDES-RAEDO, J.; RODRÍGUEZ, J. J.; GARCÍA-OSORIO, C. Ensemble methods and semi-supervised learning for information fusion: A review and future research directions. *Information Fusion*, Elsevier, v. 107, p. 102310, 2024.

GEMAQUE, R. N.; COSTA, A. F. J.; GIUSTI, R.; SANTOS, E. M. An overview of unsupervised drift detection methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley Online Library, v. 10, n. 6, p. e1381, 2020.

GOLDMAN, S.; ZHOU, Y. Enhancing supervised learning with unlabeled data. In: CITESEER. *ICML*. [S.l.], 2000. p. 327–334.

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

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.

GONÇALVES JR., 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.

GRABNER, H.; LEISTNER, C.; BISCHOF, H. Semi-supervised on-line boosting for robust tracking. *Computer Vision–ECCV 2008*, Springer, p. 234–247, 2008.

GUPTA, S.; GUPTA, A. Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, Elsevier, v. 161, p. 466–474, 2019.

GUTIÉRREZ, V. A. L. *Classificação semi-supervisionada baseada em desacordo por similaridade*. Tese (Doutorado) — Universidade de São Paulo, Brazil, 2010.

HADY, M. F. A.; SCHWENKER, F. Co-training by committee: a new semi-supervised learning framework. In: IEEE. *2008 IEEE International Conference on Data Mining Workshops*. [S.l.], 2008. p. 563–572.

HAN, M.; LI, A.; GAO, Z.; MU, D.; LIU, S. Hybrid sampling and dynamic weighting-based classification method for multi-class imbalanced data stream. *Applied Sciences*, v. 13, n. 10, 2023. ISSN 2076-3417.

HAQUE, A.; KHAN, L.; BARON, M. Sand: Semi-supervised adaptive novel class detection and classification over data stream. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2016. v. 30, n. 1.

HAQUE, A.; KHAN, L.; BARON, M.; THURAISINGHAM, B.; AGGARWAL, C. Efficient handling of concept drift and concept evolution over stream data. In: IEEE. *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. [S.l.], 2016. p. 481–492.

HICKEY, R. J. Noise modelling and evaluating learning from examples. *Artificial Intelligence*, Elsevier, v. 82, n. 1-2, p. 157–179, 1996.

HIDALGO, J. I. G.; MACIEL, B. I. F.; BARROS, R. S. M. Experimenting with prequential variations for data stream learning evaluation. *Computational Intelligence*, Wiley, v. 35, p. 670–692, 2019.

HIDALGO, J. I. G.; SANTOS, S. G. T. C.; BARROS, R. S. M. Dynamically adjusting diversity in ensembles for the classification of data streams with concept drift. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, ACM New York, NY, v. 16, n. 2, p. 1–20, 2021.

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.

HUANG, Z. A fast clustering algorithm to cluster very large categorical data sets in data mining. *Dmkd*, v. 3, n. 8, p. 34–39, 1997.

HUANG, Z. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, Springer, v. 2, n. 3, p. 283–304, 1998.

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. *Discovery Science*. [S.l.], 2013. (LNCS, v. 8140), p. 79–93.

IVANOV, Y.; BLUMBERG, B.; PENTLAND, A. Expectation maximization for weakly labeled data. In: *ICML*. [S.l.: s.n.], 2001. p. 218–225.

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.

KARABOGA, D.; OZTURK, C. A novel clustering approach: Artificial bee colony (abc) algorithm. *Applied Soft Computing*, v. 11, n. 1, p. 652–657, 2011.

KATAKIS, I.; TSOUMAKAS, G.; VLAHAVAS, I. Tracking recurring contexts using ensemble classifiers: an application to email filtering. *nowledge and Information Systems*, Springer-Verlag, v. 22, p. 371–391, 2010. ISSN 0219-1377.

KAUFMAN, L.; ROUSSEEUW, P. J. *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley-Interscience, 1990.

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.

KLINKENBERG, R. Using labeled and unlabeled data to learn drifting concepts. In: *Workshop notes of the IJCAI-01 Workshop on Learning from Temporal and Spatial Data*. [S.l.: s.n.], 2001. p. 16–24.

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.

LAAN, M. Van-der; POLLARD, K.; BRYAN, J. A new partitioning around medoids algorithm. *Journal of Statistical Computation and Simulation*, Taylor & Francis, v. 73, n. 8, p. 575–584, 2003.

LE, T.; NGUYEN, K.; NGUYEN, V.; NGUYEN, V.; PHUNG, D. Scalable semi-supervised learning with graph-based kernel machine. 2016.

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

LI, M.; ZHOU, Z.-H. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, IEEE, v. 37, n. 6, p. 1088–1098, 2007.

LI, P.; WU, X.; HU, X. Mining recurring concept drifts with limited labeled streaming data. In: *Proceedings of 2nd Asian Conference on Machine Learning*. [S.l.: s.n.], 2010. p. 241–252.

LITTLESTONE, N.; WARMUTH, M. K. The weighted majority algorithm. *Information and Computation*, v. 108, n. 2, p. 212–261, 1994. ISSN 0890-5401.

LU, T. T. *Fundamental limitations of semi-supervised learning*. Dissertação (Mestrado) — University of Waterloo, 2009.

LUONG, A. V.; NGUYEN, T. T.; LIEW, A. W.; WANG, S. Heterogeneous ensemble selection for evolving data streams. *Pattern Recognition*, Elsevier, v. 112, p. 107743, 2021.

MACÁRIO, V. *Um novo algoritmo de agrupamento semi-supervisionado baseado no Fuzzy C-Means*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2009.

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.

MACIEL, B. I. F.; SANTOS, S. G. T. C.; BARROS, R. S. M. MOAManager: a tool to support data stream experiments. *Software: Practice and Experience*, Wiley, v. 50, n. 4, p. 325–334, 2020.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. [S.l.], 1967. v. 1, p. 281–297.

MAHDI, O. A.; ALI, N.; PARDEDE, E.; ALAZAB, A.; AL-QURAISHI, T.; DAS, B. Roadmap of concept drift adaptation in data stream mining, years later. *IEEE Access*, v. 12, p. 21129–21146, 2024.

MAHDI, O. A.; PARDEDE, E.; ALI, N.; CAO, J. Fast reaction to sudden concept drift in the absence of class labels. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 10, n. 2, p. 606, 2020.

MARIÑO, L. M. P. *Variants of the Fast Adaptive Stacking of Ensembles algorithm*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2019.

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

MATSUBARA, E. T. *O algoritmo de aprendizado semi-supervisionado co-training e sua aplicação na rotulação de documentos*. Tese (Doutorado) — Universidade de São Paulo, Brazil, 2004.

MAULIK, U.; BANDYOPADHYAY, S. Genetic algorithm-based clustering technique. *Pattern Recognition*, v. 33, n. 9, p. 1455–1465, 2000. ISSN 0031-3203.

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, 2011.

MITCHELL, T. M. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, v. 45, n. 37, p. 870–877, 1997.

MITCHELL, T. M. The role of unlabeled data in supervised learning. Citeseer, 1999.

MONTEIRO, P.; SOARES, E.; BARROS, R. S. M. Co-op training: a semi-supervised learning method for data streams. In: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC)*. Melbourne: [s.n.], 2021. p. 933–938.

MVULA, P. K.; BRANCO, P.; JOURDAN, G.-V.; VIKTOR, H. L. A survey on the applications of semi-supervised learning to cyber-security. *ACM Computing Surveys*, ACM New York, NY, v. 56, n. 10, p. 1–41, 2024.

NEGRI, R. G.; SANT'ANNA, S. J. S.; DUTRA, L. V. Aplicação de modelos de aprendizado semissupervisionado na classificação de imagens de sensoriamento remoto. *Revista de Informática Teórica e Aplicada*, v. 20, n. 2, p. 32–55, 2013.

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

NGUYEN, M. H. L.; GOMES, H. M.; BIFET, A. Semi-supervised learning over streaming data using moa. In: IEEE. *2019 IEEE International Conference on Big Data (Big Data)*. [S.l.], 2019. p. 553–562.

NIGAM, K.; MCCALLUM, A. K.; THRUN, S.; MITCHELL, T. Text classification from labeled and unlabeled documents using em. *Machine learning*, Springer, v. 39, n. 2, p. 103–134, 2000.

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.

ODONNAT, A.; FEOFANOV, V.; REDKO, I. Leveraging ensemble diversity for robust self-training in the presence of sample selection bias. In: PMLR. *International Conference on Artificial Intelligence and Statistics*. [S.l.], 2024. p. 595–603.

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

ONAN, A.; KORUKOĞLU, S.; BULUT, H. A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Systems with Applications*, Elsevier, v. 62, p. 1–16, 2016.

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

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

OZA, N. C.; RUSSELL, S. Online Bagging and Boosting. In: *Artif.˜Intellig.˜and Stat.* [S.l.]: Morgan Kauf, 2001. p. 105–112.

PARK, H.-S.; JUN, C.-H. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, v. 36, n. 2, Part 2, p. 3336–3341, 2009.

PÉREZ, J. L. M. *Comitê de métodos estatísticos para detecção de mudanças de conceito*. 2018. M.Sc. Dissertation, Centro de Informática, Universidade Federal de Pernambuco, Brazil.

PÉREZ, J. L. M.; BARROS, R. S. M.; SANTOS, S. G. T. C. Statistical tests ensemble drift detector. In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. [S.l.: s.n.], 2020. p. 1021–1028.

PÉREZ, J. L. M.; BARROS, R. S. M.; SANTOS, S. G. T. C. Experimenting with supervised drift detectors in semi-supervised learning. In: *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. [S.l.: s.n.], 2023. p. 730–735.

PÉREZ, J. L. M.; BARROS, R. S. M.; SANTOS, S. G. T. C. Enhancing semi-supervised learning with concept drift detection and self-training: A study on classifier diversity and performance. *IEEE Access*, v. 13, p. 24681–24697, 2025.

PÉREZ, J. L. M.; MARIÑO, L. M. P.; BARROS, R. S. M. Improving diversity in concept drift ensembles. In: *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. [S.l.: s.n.], 2021. p. 1–8.

PERVEZ, M. S.; FARID, D. M. Feature selection and intrusion classification in nsl-kdd cup 99 dataset employing svms. In: IEEE. *Software, Knowledge, Information Management and Applications (SKIMA), 2014 8th International Conference on*. [S.l.], 2014. p. 1–6.

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.

PINAGÉ, F.; SANTOS, E. M.; GAMA, J. A drift detection method based on dynamic classifier selection. *Data Mining and Knowledge Discovery*, Springer, v. 34, n. 1, p. 50–74, 2020.

QUINLAN, R. *Thyroid Disease*. 1986. UCI Machine Learning Repository.

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.

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.

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.

SANCHES, M. K. *Aprendizado de máquina semi-supervisionado: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados*. Tese (Doutorado) — Universidade de São Paulo, Brazil, 2003.

SANTOS, S. G. T. C.; BARROS, R. S. M. Online adaboost-based methods for multiclass problems. *Artificial Intelligence Review*, Springer, v. 53, p. 1293–1322, 2020.

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.

SANTOS, S. G. T. C.; JR., P. M. G.; SILVA, G.; BARROS, R. S. M. Speeding up recovery from concept drifts. In: *Machine Learning and Knowl. Discovery in Databases*. [S.l.]: Springer, 2014, (LNCS, v. 8726). p. 179–194.

SHARMA, D.; JONES, M. Efficiently learning the graph for semi-supervised learning. In: PMLR. *Uncertainty in Artificial Intelligence*. [S.l.], 2023. p. 1900–1910.

SHEPARD, D. A two-dimensional interpolation function for irregularly-spaced data. In: *Proceedings of the 1968 23rd ACM national conference*. [S.l.: s.n.], 1968. p. 517–524.

SHI, L.; LIU, W. Adversarial self-training improves robustness and generalization for gradual domain adaptation. In: *Advances in Neural Information Processing Systems*. [S.l.]: Curran Associates, Inc., 2023. v. 36, p. 37321–37333.

SHI, X.; YUE, C.; QUAN, M.; LI, Y.; SAM, H. N. A semi-supervised ensemble clustering algorithm for discovering relationships between different diseases by extracting cell-to-cell biological communications. *Journal of Cancer Research and Clinical Oncology*, Springer, v. 150, n. 1, p. 3, 2024.

SILVA, W. L. *Métodos de agrupamentos com restrições e com busca em vizinhança variável com aplicações em séries temporais de imagens NDVI*. Tese (Doutorado) — Universidade Estadual de Campinas, Brazil, 2017.

SOARES, V. H. A.; CAMPELLO, R. J. G. B.; NOURASHRAFEDDIN, S.; MILIOS, E.; NALDI, M. C. Combining semantic and term frequency similarities for text clustering. *Knowledge and Information Systems*, Springer, v. 61, p. 1485–1516, 2019.

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.

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

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.

TAN, C. H.; LEE, V.; SALEHI, M. Online semi-supervised concept drift detection with density estimation. 2019.

TANHA, J. *Ensemble approaches to semi-supervised learning*. Tese (Doutorado) — Universiteit van Amsterdam, Nederland, 2013.

TRAT, M.; BENDER, J.; OVTCHAROVA, J. *Sensitivity-Based Optimization of Unsupervised Drift Detection for Categorical Data Streams*. [S.l.]: Karlsruher Institut für Technologie (KIT), 2022.

VERDECIA-CABRERA, A.; FRÍAS-BLANCO, I.; CARVALHO, A. C. P. L. F. An online adaptive classifier ensemble for mining non-stationary data streams. *Intelligent Data Analysis*, IOS Press, v. 22, n. 4, p. 787–806, 2018.

WAGSTAFF, K.; CARDIE, C.; ROGERS, S.; SCHRÖDL, S. Constrained k-means clustering with background knowledge. In: *ICML*. [S.l.: s.n.], 2001. v. 1, p. 577–584.

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.

WANG, S.; WU, L.; JIAO, L.; LIU, H. Improve the performance of co-training by committee with refinement of class probability estimations. *Neurocomputing*, v. 136, p. 30–40, 2014. ISSN 0925-2312.

WITTEN, I. H.; FRANK, E. Data mining: practical machine learning tools and techniques with java implementations. *Acm Sigmod Record*, ACM New York, NY, USA, v. 31, n. 1, p. 76–77, 2002.

WOLPERT, D. H. Stacked generalization. *Neural networks*, Elsevier, v. 5, n. 2, p. 241–259, 1992.

WOOLAM, C.; MASUD, M. M.; KHAN, L. Lacking labels in the stream: Classifying evolving stream data with few labels. In: SPRINGER. *ISMIS*. [S.l.], 2009. p. 552–562.

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.

YAROWSKY, D. Unsupervised word sense disambiguation rivaling supervised methods. In: *33rd annual meeting of the association for computational linguistics*. [S.l.: s.n.], 1995. p. 189–196.

ZAHN, C. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, C-20, n. 1, p. 68–86, 1971.

ZHANG, P.; ZHU, X.; GUO, L. Mining data streams with labeled and unlabeled training examples. In: IEEE. *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*. [S.l.], 2009. p. 627–636.

ZHOU, Y.; GOLDMAN, S. Democratic co-learning. In: IEEE. *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*. [S.l.], 2004. p. 594–602.

ZHOU, Z.; LI, M. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, IEEE, v. 17, n. 11, p. 1529–1541, 2005.

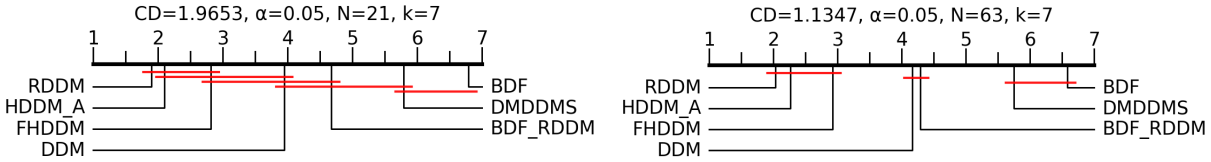
ZHU, X. *Semi-Supervised Learning Literature Survey*. [S.l.], 2005.

ZHU, X.; WU, X. Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, Springer, v. 22, p. 177–210, 2004.

ŽLIOBAITĖ, I. Learning under concept drift: an overview. 2010.

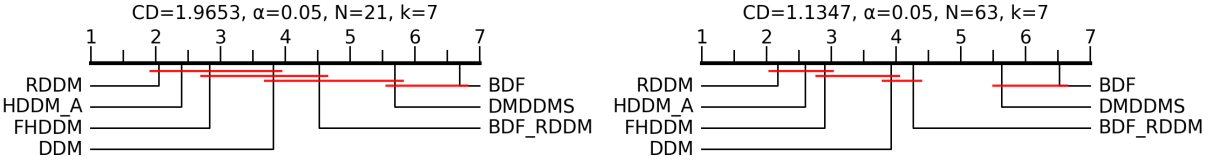
APÊNDICE A – COMPARAÇÃO COM 25% DOS RÓTULOS

Neste apêndice, apresentamos os resultados com 25% de presença dos rótulos. Essa informação foi usada na apresentação da qualificação. No entanto, seguindo a sugestão dos revisores do artigo publicado (PÉREZ; BARROS; SANTOS, 2023), substituímos o uso de 25% por 15% com o objetivo de observar se uma menor presença dos rótulos influencia significativamente os resultados, permitindo uma análise mais robusta e detalhada do impacto dessa variação. No anexo, para cada tabela de comparação das bases de dados artificiais em cenários de mudanças abruptas ou graduais com o uso dos classificadores HT e NB, foi adicionada uma figura contendo a comparação de CD das bases com 25% dos rótulos. Os resultados do método DSDD não foram incluídos nas tabelas deste apêndice, pois, com base nos experimentos realizados com 15% e 30% dos dados rotulados, o método apresentou o pior desempenho nos cenários semi-supervisionados.



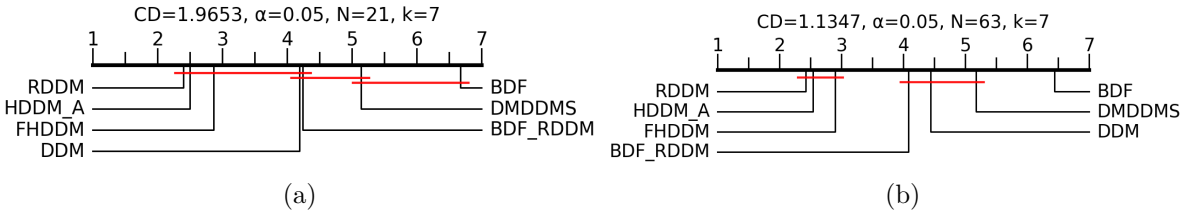
(a) (b)

Figura 16 – Comparação estatística das acurácias de todos os métodos em todos os cenários do número de rótulos: (a) 25% e (c) todos, através do Teste F_F e do Pós-Teste *Nemenyi*, com 95% de confiança, usando o classificador HT e bases de dados artificiais com mudanças abruptas.



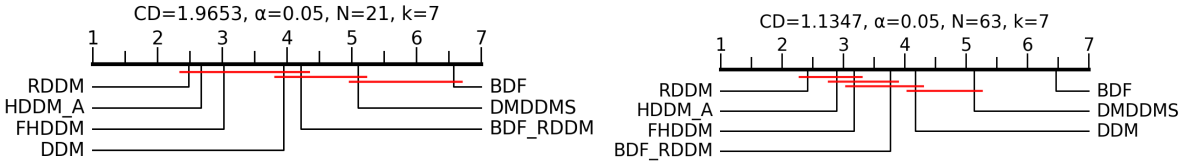
(a) (b)

Figura 17 – Comparação estatística das acurácias de todos os métodos em todos os cenários do número de rótulos: (a) 25% e (d) todos, através do Teste F_F e do Pós-Teste *Nemenyi*, com 95% de confiança, usando o classificador HT e bases de dados artificiais com mudanças graduais.



(a) (b)

Figura 18 – Comparação estatística das acurácias de todos os métodos em todos os cenários do número de rótulos: (a) 25% e (d) todos, através do Teste F_F e do Pós-Teste *Nemenyi*, com 95% de confiança, usando o classificador NB e bases de dados artificiais com mudanças abruptas.



(a) (b)

Figura 19 – Comparação estatística das acurácias de todos os métodos em todos os cenários do número de rótulos: (a) 25% e (d) todos, através do Teste F_F e do Pós-Teste *Nemenyi*, com 95% de confiança, usando o classificador NB e bases de dados artificiais com mudanças graduais.

Tabela 14 – Médias de acurácias em (%) usando o classificador HT, com 95% de intervalo de confiança nas bases de dados artificiais com mudanças abruptas.

%RÓT.	TAM.	BASE	DDM	*Hoeffding-based Drift Detection Method* (HDDM)$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$
25%	20K	Agrawal(F1-F5)	60.64±0.55	61.80±0.48	62.70±0.37	62.55±0.47	59.63±0.17	58.66±0.29	**63.32±0.30**
		Agrawal(F6-F10)	72.51±2.04	79.77±0.83	**81.34±0.58**	79.44±1.52	77.39±0.72	62.42±0.41	68.51±1.21
		LED	67.62±0.44	67.51±0.44	67.39±0.43	**68.06±0.22**	55.84±0.24	49.19±0.47	60.59±0.67
		Mixed	89.32±0.22	**89.76±0.24**	89.45±0.22	89.67±0.22	59.71±1.00	64.66±0.45	87.48±0.28
		Sine	86.17±0.52	**87.22±0.34**	87.13±0.27	87.20±0.29	67.77±2.16	63.00±0.43	85.00±0.23
		Waveform	78.21±0.39	78.41±0.38	78.61±0.27	**78.98±0.27**	76.81±0.27	76.62±0.47	78.35±0.44
		RandomRBF	**31.60±0.52**	31.58±0.46	30.93±0.50	31.56±0.44	30.15±0.80	30.17±1.56	31.13±0.46
	50K	Agrawal(F1-F5)	64.30±0.86	65.96±0.45	65.78±0.47	**66.05±0.38**	61.46±0.28	60.77±0.54	65.90±0.30
		Agrawal(F6-F10)	80.24±1.50	83.80±0.35	**84.34±0.34**	83.13±0.96	79.37±0.38	64.27±0.29	71.61±0.73
		LED	70.66±0.17	70.67±0.19	70.58±0.24	**70.93±0.17**	56.51±0.49	51.94±0.98	61.49±0.42
		Mixed	89.34±1.23	**90.78±0.14**	90.75±0.13	90.70±0.13	66.53±1.25	65.29±0.37	88.29±0.23
		Sine	88.19±0.24	**89.09±0.12**	88.98±0.12	88.54±0.16	81.37±0.55	63.59±0.59	85.65±0.17
		Waveform	79.02±0.22	79.52±0.28	79.56±0.25	**79.76±0.19**	77.55±0.14	77.24±0.24	79.33±0.26
		RandomRBF	31.84±0.39	31.94±0.34	31.06±0.26	**31.95±0.30**	30.40±0.58	29.97±1.33	30.88±0.29
	100K	Agrawal(F1-F5)	67.00±1.35	69.32±0.31	69.07±0.46	**69.59±0.40**	62.05±0.15	59.70±0.14	67.63±0.28
		Agrawal(F6-F10)	83.69±0.40	85.46±0.26	**85.88±0.22**	85.31±0.35	81.26±0.18	66.00±0.43	73.28±0.49
		LED	72.13±0.23	72.31±0.15	71.99±0.25	**72.59±0.13**	63.03±0.60	52.23±1.15	62.09±0.37
		Mixed	88.00±1.25	90.78±0.09	90.96±0.08	**90.97±0.08**	79.42±0.36	67.06±0.93	88.47±0.19
		Sine	89.95±0.13	90.38±0.12	**90.41±0.13**	90.08±0.15	83.26±0.41	63.99±0.44	86.40±0.16
		Waveform	79.13±0.21	79.84±0.17	79.90±0.17	**79.91±0.13**	78.93±0.11	78.34±0.16	79.78±0.14
		RandomRBF	31.77±0.38	**32.28±0.20**	31.06±0.20	32.22±0.19	31.71±0.40	31.71±1.47	31.09±0.20

Tabela 15 – Médias de acurácias em (%) usando o classificador HT, com 95% de intervalo de confiança nas bases de dados artificiais com mudanças graduais.

%RÓT.	TAM.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$
25%	20K	Agrawal(F1-F5)	60.30±0.47	61.25±0.32	61.83±0.27	61.69±0.44	59.24±0.28	58.74±0.26	**62.94±0.33**
		Agrawal(F6-F10)	69.94±1.85	77.70±0.76	**79.11±0.75**	77.35±1.88	77.78±0.68	62.23±0.40	67.19±1.30
		LED	66.93±0.39	66.54±0.18	65.95±0.45	**67.05±0.22**	55.78±0.25	49.44±0.52	59.90±0.63
		Mixed	86.72±0.20	**86.96±0.19**	86.93±0.20	86.69±0.19	59.58±0.93	64.63±0.37	85.11±0.36
		Sine	84.77±0.22	**84.91±0.23**	84.75±0.21	84.81±0.24	67.18±1.86	63.16±0.38	82.97±0.23
		Waveform	77.69±0.42	77.79±0.45	78.05±0.35	**78.61±0.27**	76.74±0.26	76.68±0.38	78.04±0.41
		RandomRBF	**31.49±0.43**	31.38±0.42	30.71±0.44	31.39±0.41	29.77±0.72	30.14±1.55	31.10±0.52
	50K	Agrawal(F1-F5)	64.40±0.81	65.37±0.43	65.20±0.41	**65.49±0.42**	61.06±0.26	60.16±0.42	65.38±0.30
		Agrawal(F6-F10)	79.91±1.36	83.34±0.31	**84.03±0.19**	82.80±0.99	79.36±0.43	64.46±0.28	71.36±0.73
		LED	70.55±0.17	70.40±0.20	70.05±0.21	**70.79±0.19**	56.42±0.29	52.01±0.96	61.31±0.47
		Mixed	89.72±0.14	**89.90±0.12**	89.89±0.13	89.84±0.12	67.95±0.73	65.22±0.42	87.57±0.26
		Sine	88.11±0.11	**88.36±0.12**	**88.36±0.13**	88.18±0.12	80.97±0.62	63.25±0.38	85.14±0.18
		Waveform	78.87±0.19	79.12±0.23	79.27±0.22	**79.65±0.19**	77.50±0.12	77.25±0.25	79.15±0.25
		RandomRBF	31.47±0.39	**31.85±0.33**	30.89±0.30	31.74±0.27	30.12±0.62	30.12±1.16	30.83±0.30
	100K	Agrawal(F1-F5)	66.57±1.29	69.54±0.40	68.76±0.45	**69.89±0.38**	62.10±0.16	59.66±0.14	67.72±0.25
		Agrawal(F6-F10)	83.71±0.42	85.14±0.30	**85.53±0.24**	85.41±0.26	81.06±0.17	65.96±0.44	73.50±0.38
		LED	72.23±0.13	72.14±0.13	71.69±0.20	**72.53±0.13**	63.43±0.31	52.17±1.29	62.20±0.30
		Mixed	89.47±0.55	90.37±0.08	**90.47±0.07**	90.44±0.08	79.61±0.43	65.82±0.64	88.22±0.21
		Sine	89.75±0.11	89.99±0.09	**90.01±0.01**	89.85±0.13	82.50±0.89	63.93±0.36	86.08±0.14
		Waveform	79.17±0.18	79.71±0.16	79.60±0.20	**79.88±0.14**	78.89±0.10	78.31±0.16	79.70±0.14
		RandomRBF	31.56±0.39	**32.17±0.27**	31.10±0.19	32.08±0.28	31.68±0.48	31.74±1.40	31.15±0.20

Tabela 16 – Médias de acurácias em (%) usando o classificador NB, com 95% de intervalo de confiança nas bases de dados artificiais com mudanças abruptas.

%RÓT.	TAM.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$
25%	20K	Agrawal(F1-F5)	59.86±0.60	60.96±0.42	61.76±0.37	**61.94±0.38**	59.63±0.17	57.07±0.36	60.94±0.39
		Agrawal(F6-F10)	70.02±0.99	79.61±0.44	**80.26±0.66**	76.58±1.40	77.39±0.72	61.50±0.36	65.15±1.04
		LED	67.76±0.31	67.57±0.44	67.44±0.43	**68.09±0.21**	55.84±0.24	49.21±0.40	60.63±0.67
		Mixed	89.40±0.22	**89.89±0.21**	89.58±0.19	89.78±0.20	59.71±1.00	63.87±0.16	87.54±0.31
		Sine	84.35±1.24	86.42±0.31	86.35±0.27	**86.45±0.28**	67.77±2.16	62.23±0.18	84.65±0.28
		Waveform	78.28±0.36	78.42±0.38	78.62±0.27	**79.02±0.28**	76.81±0.27	76.65±0.47	78.41±0.45
		RandomRBF	30.37±0.58	29.97±0.55	29.59±0.52	29.90±0.49	30.15±0.80	30.28±1.55	**31.47±0.44**
	50K	Agrawal(F1-F5)	62.34±0.60	64.26±0.30	64.09±0.28	**64.38±0.20**	61.46±0.28	58.58±0.25	63.67±0.24
		Agrawal(F6-F10)	76.40±0.94	82.67±0.52	**83.89±0.39**	82.23±0.94	79.37±0.38	62.77±0.24	69.97±0.92
		LED	70.66±0.17	70.67±0.19	70.60±0.24	**70.93±0.17**	56.51±0.49	50.95±0.24	61.62±0.41
		Mixed	89.74±0.80	**90.84±0.11**	90.80±0.11	90.75±0.12	66.53±1.25	64.30±0.35	88.07±0.25
		Sine	83.43±1.62	86.67±0.18	**86.72±0.17**	86.63±0.20	81.37±0.55	62.34±0.15	85.13±0.17
		Waveform	79.04±0.21	79.55±0.27	79.60±0.24	**79.75±0.19**	77.55±0.14	77.21±0.25	79.32±0.26
		RandomRBF	30.36±0.37	29.91±0.42	29.05±0.29	29.98±0.33	30.40±0.58	29.78±1.36	**31.18±0.28**
	100K	Agrawal(F1-F5)	63.29±0.66	**65.32±0.19**	64.75±0.19	65.26±0.15	62.05±0.15	59.01±0.16	64.61±0.17
		Agrawal(F6-F10)	81.23±1.04	84.88±0.42	**85.07±0.24**	84.96±0.32	81.26±0.18	63.35±0.22	72.63±0.47
		LED	72.20±0.19	72.33±0.15	72.01±0.25	**72.59±0.13**	63.03±0.60	51.62±0.25	62.15±0.36
		Mixed	90.72±0.62	91.44±0.10	**91.52±0.08**	91.43±0.09	79.42±0.36	64.39±0.28	88.21±0.15
		Sine	83.99±1.37	87.14±0.14	**87.26±0.13**	87.12±0.16	83.26±0.41	62.34±0.12	85.21±0.14
		Waveform	79.24±0.26	80.09±0.18	**80.14±0.15**	80.09±0.17	78.93±0.11	77.34±0.13	79.72±0.15
		RandomRBF	30.55±0.49	30.52±0.39	29.27±0.37	30.40±0.37	**31.71±0.40**	30.00±1.62	31.42±0.18

Tabela 17 – Médias de acurácias em (%) usando o classificador NB, com 95% de intervalo de confiança nas bases de dados artificiais com mudanças graduais.

%RÓT.	TAM.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$
25%	20K	Agrawal(F1-F5)	59.50±0.55	60.38±0.42	60.80±0.29	**61.25±0.32**	59.24±0.28	57.17±0.34	60.63±0.36
		Agrawal(F6-F10)	68.24±0.96	74.85±1.57	76.20±1.06	73.52±1.68	**77.78±0.68**	61.38±0.39	64.16±1.02
		LED	66.94±0.39	66.45±0.31	66.01±0.45	**67.06±0.22**	55.78±0.25	49.56±0.41	60.03±0.60
		Mixed	86.89±0.20	**87.09±0.19**	87.07±0.18	86.84±0.19	59.58±0.93	64.10±0.17	85.29±0.32
		Sine	83.10±1.34	**84.25±0.19**	84.16±0.17	84.19±0.21	67.18±1.86	62.41±0.22	82.58±0.24
		Waveform	**77.80±0.42**	**77.80±0.45**	78.06±0.35	78.60±0.27	76.74±0.26	76.72±0.38	78.06±0.41
		RandomRBF	30.11±0.53	29.87±0.51	29.12±0.44	29.72±0.42	29.77±0.72	30.24±1.54	**31.43±0.50**
	50K	Agrawal(F1-F5)	62.05±0.55	**63.96±0.21**	63.63±0.25	63.93±0.17	61.06±0.26	58.60±0.23	63.36±0.23
		Agrawal(F6-F10)	71.95±2.20	80.35±1.26	**82.11±1.00**	79.32±1.59	79.36±0.43	62.64±0.26	69.05±0.91
		LED	70.56±0.17	70.42±0.21	70.07±0.21	**70.80±0.19**	56.42±0.29	51.02±0.24	61.37±0.43
		Mixed	89.80±0.12	**89.95±0.12**	89.94±0.12	89.89±0.12	67.95±0.73	64.31±0.34	87.42±0.24
		Sine	84.49±1.10	86.27±0.17	86.33±0.16	**86.35±0.18**	80.97±0.62	62.36±0.17	84.56±0.22
		Waveform	78.87±0.20	79.17±0.23	79.29±0.21	**79.63±0.20**	77.50±0.12	77.23±0.26	79.16±0.26
		RandomRBF	30.46±0.43	30.10±0.35	29.12±0.36	29.88±0.30	30.12±0.62	29.98±1.28	**31.24±0.27**
	100K	Agrawal(F1-F5)	63.10±0.67	65.13±0.14	64.64±0.14	**65.22±0.15**	62.10±0.16	59.05±0.17	64.60±0.17
		Agrawal(F6-F10)	79.74±1.29	83.03±0.75	**84.32±0.35**	83.53±0.72	81.06±0.17	63.37±0.17	71.81±0.55
		LED	72.23±0.13	72.09±0.14	71.71±0.20	**72.53±0.13**	63.43±0.31	51.69±0.27	62.18±0.28
		Mixed	90.88±0.24	91.03±0.07	**91.06±0.07**	91.02±0.07	79.61±0.43	64.43±0.26	87.95±0.19
		Sine	86.06±0.49	86.89±0.13	**86.95±0.12**	86.93±0.13	82.50±0.89	62.39±0.13	84.95±0.14
		Waveform	79.34±0.25	79.92±0.17	79.83±0.18	**80.09±0.13**	78.89±0.10	77.34±0.14	79.72±0.14
		RandomRBF	30.46±0.36	30.46±0.30	29.27±0.31	30.37±0.28	**31.68±0.48**	30.07±1.62	31.47±0.19

Tabela 18 – Acurácias (%) usando o classificador HT nas bases de dados reais.

%RÓT.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$
25%	Airlines	63.67	64.73	65.31	64.88	65.13	67.51	**68.39**
	Connect4	71.20	71.73	71.94	71.91	69.81	69.25	**73.15**
	Covertype	78.15	79.57	**79.89**	79.80	73.72	58.97	67.05
	Spam_data	**90.83**	90.69	90.30	90.14	89.85	83.83	88.49
	Weather	69.78	69.46	69.82	69.53	69.09	69.87	**70.19**

Tabela 19 – Acurácias (%) usando o classificador NB nas bases de dados reais.

%RÓT.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$
25%	Airlines	65.42	66.04	65.83	66.30	65.13	**67.53**	66.86
	Connect4	70.95	71.50	71.68	71.75	69.81	68.60	**72.42**
	Covertype	75.94	79.59	**79.63**	79.46	73.72	59.11	67.35
	Spam_data	87.53	89.62	88.95	89.09	**89.85**	82.53	86.38
	Weather	68.18	68.93	70.01	69.41	69.09	69.37	**70.15**

APÊNDICE B – COMPARAÇÃO COM 100% DOS RÓTULOS

No apêndice, são apresentados os resultados obtidos com 100% dos rótulos disponíveis durante o treinamento. Esses resultados servem como referência na tese, pois representam o desempenho dos classificadores em um cenário plenamente supervisionado. A utilização dessa configuração de referência é fundamental, pois permite uma comparação direta do desempenho dos métodos projetados para aprendizado supervisionado em contextos semi-supervisionados.

Tabela 20 – Médias de acurácias em (%), com intervalo de confiança de 95% e classificador base HT em conjuntos de dados artificiais com mudanças abruptas de conceito.

%RÓT.	TAM.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$	DSDD
100%	20K	Agrawal(F1-F5)	64.93±1.28	68.12±0.48	67.26±0.59	68.19±0.45	62.81±0.21	60.62±0.17	66.92±0.30	**78.78±0.26**
		Agrawal(F6-F10)	81.19±1.60	84.44±0.26	84.68±0.17	83.11±1.31	81.17±0.24	66.53±0.35	72.43±0.81	**88.42±0.33**
		LED	71.31±0.25	71.52±0.18	71.26±0.32	71.73±0.17	59.71±0.88	61.27±0.32	71.13±0.19	**76.91±0.20**
		Mixed	88.96±0.54	90.29±0.15	90.58±0.14	90.66±0.14	80.87±1.73	75.66±0.37	90.58±0.16	**92.79±0.12**
		Sine	89.31±0.14	89.89±0.13	89.87±0.12	89.46±0.14	83.69±0.52	69.43±0.53	89.09±0.13	**93.42±0.15**
		Waveform	78.89±0.22	79.41±0.25	79.52±0.26	79.64±0.26	78.61±0.23	78.00±0.22	79.51±0.29	**83.08±0.28**
		RandomRBF	31.82±0.43	32.40±0.34	31.39±0.32	32.30±0.27	31.34±0.61	37.97±0.46	37.46±0.37	**43.18±0.44**
	50K	Agrawal(F1-F5)	68.03±1.98	72.57±0.33	71.50±0.28	72.43±0.31	63.44±0.15	59.27±0.16	69.69±0.20	**82.14±0.20**
		Agrawal(F6-F10)	83.60±1.14	85.76±0.32	86.54±0.13	86.09±0.19	82.23±0.13	65.09±0.52	74.33±0.27	**90.49±0.20**
		LED	71.93±0.48	72.81±0.16	72.10±0.29	72.88±0.15	64.70±0.15	64.71±0.30	72.56±0.13	**78.30±0.15**
		Mixed	91.28±0.37	92.11±0.07	92.13±0.08	91.60±0.14	85.50±0.45	68.40±0.20	91.39±0.12	**94.67±0.11**
		Sine	91.06±0.15	91.52±0.14	91.52±0.13	91.19±0.14	85.00±0.17	70.93±0.59	90.73±0.11	**94.89±0.11**
		Waveform	79.28±0.20	79.58±0.16	79.81±0.14	79.94±0.16	79.61±0.13	78.81±0.15	79.88±0.16	**85.05±0.46**
		RandomRBF	32.54±0.37	32.57±0.30	31.35±0.25	32.40±0.28	31.90±0.31	39.36±0.34	38.48±0.30	**42.14±0.33**
	100K	Agrawal(F1-F5)	71.01±2.08	74.74±0.34	72.91±0.25	74.85±0.28	63.59±0.10	59.79±0.24	71.33±0.20	**83.54±0.24**
		Agrawal(F6-F10)	84.81±0.83	87.44±0.14	87.69±0.05	87.17±0.19	82.77±0.05	65.01±0.52	74.98±0.27	**91.65±0.14**
		LED	72.65±0.30	73.37±0.11	72.51±0.23	73.39±0.12	66.24±0.22	66.07±0.28	73.18±0.11	**78.86±0.11**
		Mixed	92.79±0.12	93.12±0.07	93.15±0.06	92.46±0.21	88.14±0.37	79.45±0.37	92.17±0.19	**95.46±0.08**
		Sine	92.31±0.09	92.57±0.11	92.59±0.10	92.41±0.11	85.41±0.13	76.63±0.43	91.89±0.10	**95.95±0.09**
		Waveform	79.35±0.24	79.52±0.17	79.85±0.13	79.97±0.14	79.86±0.10	79.34±0.16	79.92±0.14	**86.24±0.41**
		RandomRBF	33.64±0.26	32.87±0.27	31.30±0.13	32.80±0.20	32.06±0.23	40.32±0.31	39.22±0.21	**42.10±0.30**

Tabela 21 – Médias de acurácias em (%), com intervalo de confiança de 95% e classificador base HT em conjuntos de dados artificiais com mudanças gradual de conceito.

%RÓT.	TAM.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$	DSDD
100%	20K	Agrawal(F1-F5)	64.10±1.17	66.30±0.43	65.73±0.50	66.84±0.40	62.31±0.22	60.55±0.17	65.81±0.25	**78.27±0.32**
		Agrawal(F6-F10)	79.00±1.91	82.98±0.28	82.98±0.17	82.64±0.96	80.16±0.19	66.20±O.38	72.20±0.59	**87.29±0.22**
		LED	70.54±0.19	70.42±0.19	69.79±0.34	70.66±0.19	59.38±0.81	61.15±0.29	69.92±0.19	**76.18±0.19**
		Mixed	87.29±0.19	87.23±0.18	87.52±0.17	87.53±0.18	81.44±1.76	74.46±0.52	87.93±0.18	**90.50±0.14**
		Sine	86.68±0.14	86.79±0.13	87.03±0.13	86.83±0.13	80.90±0.64	68.72±0.29	86.82±0.12	**91.93±0.13**
		Waveform	78.52±0.25	78.78±0.28	78.84±0.26	79.10±0.28	78.53±0.22	77.94±0.23	79.01±0.31	**83.29±0.30**
		RandomRBF	31.80±0.54	32.32±0.37	31.26±0.35	32.19±0.35	31.01±0.62	37.93±0.52	37.54±0.44	**43.43±0.39**
	50K	Agrawal(F1-F5)	68.46±1.74	71.39±0.26	70.35±0.31	71.43±0.31	63.21±0.12	59.27±0.16	69.02±0.24	**81.98±0.24**
		Agrawal(F6-F10)	83.17±1.16	85.19±0.32	85.87±0.13	85.69±0.22	82.04±0.09	65.04±0.51	74.07±0.29	**90.17±0.18**
		LED	72.33±0.23	72.47±0.14	71.66±0.26	72.62±0.15	64.76±0.18	64.82±0.29	72.23±0.14	**78.16±0.15**
		Mixed	90.84±0.10	90.78±0.09	90.93±0.09	90.74±0.09	85.23±0.45	68.28±0.18	90.61±0.11	**94.26±0.10**
		Sine	90.27±0.10	90.33±0.11	90.42±0.10	90.34±0.09	83.83±0.20	70.75±0.58	90.02±0.09	**94.54±0.09**
		Waveform	79.18±0.23	79.44±0.18	79.49±0.16	79.71±0.14	79.59±0.13	78.81±0.15	79.67±0.15	**84.86±0.39**
		RandomRBF	32.53±0.34	32.58±0.29	31.34±0.23	32.38±0.28	31.72±0.39	39.22±0.42	38.35±0.34	**42.16±0.24**
	100K	Agrawal(F1-F5)	71.72±1.76	74.25±0.29	72.61±0.31	74.43±0.33	63.56±0.09	59.79±0.24	70.95±0.22	**83.52±0.22**
		Agrawal(F6-F10)	84.47±0.82	87.14±0.15	87.31±0.05	86.97±0.18	82.58±0.05	65.01±0.52	74.85±0.26	**91.52±0.15**
		LED	72.54±0.34	73.21±0.12	72.30±0.23	73.30±0.12	66.24±0.22	66.00±O.27	73.06±0.11	**78.81±0.11**
		Mixed	92.42±0.08	92.43±0.08	92.52±0.07	92.37±0.08	88.07±0.35	74.72±0.78	92.00±0.13	**95.38±0.07**
		Sine	92.00±0.09	91.98±0.09	92.04±0.08	91.99±0.09	84.74±0.15	75.77±0.36	91.56±0.09	**95.71±0.08**
		Waveform	79.37±0.25	79.47±0.12	79.67±0.14	79.81±0.13	79.85±0.10	79.33±0.16	79.78±0.13	**86.19±0.36**
		RandomRBF	33.67±0.24	32.92±0.26	31.30±0.13	32.84±0.19	32.01±0.24	40.30±0.30	39.21±0.22	**42.03±0.26**

Tabela 22 – Médias de acurácias em (%), com intervalos de confiança de 95% e classificador base HT em conjuntos de dados do mundo real.

%RÓT.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$	DSDD
100%	Airlines	65.30	65.00	65.37	66.01	65.70	68.12	**68.15**	80.47
	Connect4	74.12	75.04	75.28	75.43	72.74	74.35	**78.21**	80.43
	Covertype	**87.33**	87.22	85.05	86.39	77.92	66.30	74.06	90.62
	Spam_data	92.11	91.98	**92.60**	92.27	91.14	86.29	90.99	95.18
	Weather	**73.71**	72.31	72.26	72.56	70.41	71.71	73.42	82.32

Tabela 23 – Médias de acurácias em (%), com intervalo de confiança de 95% e classificador base NB em conjuntos de dados artificiais com mudanças abruptas de conceito.

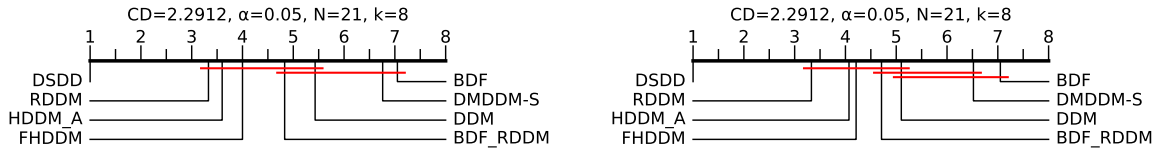
%RÓT.	TAM.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$	DSDD
100%	20K	Agrawal(F1-F5)	63.08±0.59	64.82±0.17	64.41±0.18	64.89±0.15	62.81±0.21	58.89±0.15	64.39±0.16	**75.94±0.20**
		Agrawal(F6-F10)	79.00±0.87	83.00±0.50	84.05±0.23	83.18±0.56	81.17±0.24	63.11±0.20	71.58±0.67	**87.48±0.42**
		LED	71.32±0.25	71.52±0.18	71.28±0.33	71.74±0.16	59.71±0.88	58.64±0.17	71.15±0.19	**76.91±0.21**
		Mixed	90.26±0.67	91.10±0.12	91.11±0.13	91.03±0.15	80.87±1.73	64.76±0.08	90.89±0.15	**92.13±0.12**
		Sine	83.67±1.77	87.08±0.18	87.17±0.19	87.02±0.19	83.69±0.52	63.64±0.08	86.86±0.18	**88.58±0.19**
		Waveform	78.98±0.29	79.60±0.26	79.69±0.29	79.78±0.29	78.61±0.23	77.26±0.24	79.53±0.31	**82.18±0.31**
		RandomRBF	30.79±0.47	30.69±0.41	29.76±0.39	30.50±0.41	31.34±0.61	38.02±0.46	37.24±0.37	**45.31±0.28**
	50K	Agrawal(F1-F5)	63.64±0.63	65.67±0.16	65.03±0.14	65.73±0.11	63.44±0.15	59.45±0.10	65.43±0.11	**74.65±0.17**
		Agrawal(F6-F10)	82.40±1.16	85.34±0.22	85.63±0.15	85.38±0.20	82.23±0.13	63.69±0.10	73.54±0.26	**88.56±0.23**
		LED	71.66±0.71	72.81±0.16	72.12±0.30	72.89±0.15	64.70±0.15	59.42±0.11	72.57±0.13	**78.29±0.15**
		Mixed	90.85±0.96	91.63±0.11	91.68±0.10	91.57±0.10	85.50±0.45	65.10±0.05	91.48±0.10	**92.89±0.09**
		Sine	84.21±1.32	87.26±0.10	87.39±0.12	87.22±0.13	85.00±0.17	63.69±0.06	87.07±0.14	**88.60±0.15**
		Waveform	79.60±0.18	80.13±0.15	80.18±0.14	80.16±0.14	79.61±0.13	77.47±0.14	79.97±0.16	**81.89±0.13**
		RandomRBF	31.06±0.50	30.91±0.40	29.58±0.36	30.73±0.36	31.90±0.31	39.01±0.42	38.18±0.29	**45.16±0.26**
	100K	Agrawal(F1-F5)	64.17±0.68	66.06±0.08	65.27±0.11	66.08±0.08	63.59±0.10	59.73±0.07	65.84±0.07	**73.93±0.19**
		Agrawal(F6-F10)	84.29±0.62	86.14±0.09	86.14±0.07	86.13±0.04	82.77±0.05	63.95±0.07	74.27±0.07	**88.62±0.07**
		LED	72.54±0.40	73.37±0.11	72.53±0.24	73.39±0.12	66.24±0.22	59.87±0.08	73.19±0.11	**78.86±0.11**
		Mixed	90.70±1.17	91.81±0.07	91.88±0.06	91.78±0.06	88.14±0.37	65.26±0.03	91.73±0.07	**92.93±0.07**
		Sine	83.77±1.40	87.27±0.10	87.42±0.09	87.31±0.10	85.41±0.13	63.73±0.05	87.14±0.12	**88.45±0.12**
		Waveform	79.67±0.22	80.27±0.11	80.31±0.11	80.25±0.11	79.86±0.10	77.57±0.10	80.12±0.12	**81.54±0.12**
		RandomRBF	31.38±0.42	31.13±0.34	29.53±0.26	31.16±0.28	32.06±0.23	39.82±0.32	39.00±0.20	**44.93±0.27**

Tabela 24 – Médias de acurácias em (%), com intervalo de confiança de 95% e classificador base NB em conjuntos de dados artificiais com mudanças gardual de conceito.

%RÓT.	TAM.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$	DSDD
100%	20K	Agrawal(F1-F5)	62.62±0.51	63.92±0.14	63.40±0.15	63.98±0.13	62.31±0.22	58.90±0.15	63.53±0.13	**75.67±0.28**
		Agrawal(F6-F10)	75.89±1.02	80.47±0.51	81.07±1.01	80.79±0.91	80.16±0.19	63.03±0.20	69.82±0.90	**85.96±0.63**
		LED	70.54±0.19	70.43±0.18	69.81±0.35	70.66±0.18	59.38±0.81	58.67±0.17	69.93±0.19	**76.19±0.19**
		Mixed	87.85±0.17	87.80±0.18	88.19±0.18	88.01±0.18	81.44±1.76	64.95±0.07	88.34±0.18	**90.26±0.11**
		Sine	84.64±0.20	84.97±0.15	85.07±0.17	84.98±0.15	80.90±0.64	63.71±0.09	85.15±0.14	**87.54±0.15**
		Waveform	78.46±0.29	78.90±0.30	78.78±0.25	79.22±0.31	78.53±0.22	77.23±0.24	79.05±0.31	**81.73±0.32**
		RandomRBF	30.89±0.51	30.68±0.44	29.56±0.42	30.53±0.47	31.01±0.62	37.97±0.51	37.26±0.45	**45.30±0.34**
	50K	Agrawal(F1-F5)	63.92±0.57	65.43±0.11	64.68±0.15	65.38±0.11	63.21±0.12	59.50±0.10	65.08±0.11	**74.66±0.19**
		Agrawal(F6-F10)	82.41±0.96	84.57±0.31	84.81±0.12	84.77±0.22	82.04±0.09	63.67±0.10	73.20±0.26	**88.29±0.26**
		LED	72.30±0.26	72.48±0.15	71.68±0.27	72.63±0.15	64.76±0.18	59.42±0.10	72.24±0.14	**78.17±0.14**
		Mixed	90.42±0.11	90.45±0.11	90.52±0.11	90.50±0.09	85.23±0.45	65.13±0.05	90.63±0.10	**92.29±0.09**
		Sine	86.31±0.26	86.76±0.10	86.76±0.11	86.78±0.11	83.83±0.20	63.70±0.06	86.80±0.10	**88.27±0.14**
		Waveform	79.51±0.21	79.89±0.18	79.75±0.19	79.95±0.13	79.59±0.13	77.47±0.14	79.75±0.16	**81.81±0.13**
		RandomRBF	30.94±0.49	30.92±0.37	29.61±0.36	30.81±0.31	31.72±0.39	38.76±0.50	38.10±0.34	**45.10±0.32**
	100K	Agrawal(F1-F5)	64.06±0.63	65.93±0.09	65.07±0.10	65.92±0.08	63.56±0.09	59.73±0.07	65.67±0.07	**73.41±0.18**
		Agrawal(F6-F10)	83.83±0.81	85.70±0.13	85.70±0.07	85.79±0.06	82.58±0.05	63.95±0.07	74.07±0.07	**88.59±0.07**
		LED	72.34±0.51	73.22±0.12	72.31±0.24	73.30±0.12	66.24±0.22	59.86±0.08	73.06±0.11	**78.81±0.11**
		Mixed	91.22±0.07	91.25±0.07	91.30±0.07	91.29±0.06	88.07±0.35	65.25±0.03	91.32±0.07	**92.69±0.05**
		Sine	86.58±0.29	87.14±0.09	87.17±0.09	87.16±0.09	84.74±0.15	63.72±0.05	87.09±0.09	**88.31±0.10**
		Waveform	79.52±0.18	80.20±0.11	80.18±0.12	80.13±0.11	79.85±0.10	77.57±0.10	80.01±0.12	**81.54±0.12**
		RandomRBF	31.41±0.40	31.16±0.33	29.52±0.26	31.23±0.30	32.01±0.24	39.77±0.34	38.95±0.22	**44.77±0.31**

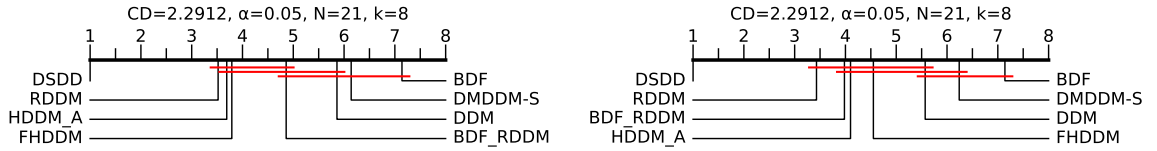
Tabela 25 – Médias de acurácias em (%), com intervalos de confiança de 95% e classificador base NB em conjuntos de dados do mundo real.

%RÓT.	BASE	DDM	HDDM$_A$	FHDDM	RDDM	DMDDM-S	BDF	BDF$_{RDDM}$	DSDD
100%	Airlines	65.35	67.23	65.82	67.50	65.70	**69.19**	68.61	75.84
	Connect4	74.47	74.95	75.23	75.23	72.74	69.19	**77.87**	80.80
	Covertype	**88.00**	87.39	85.07	86.84	77.92	66.38	74.00	90.87
	Spam_data	89.34	90.96	**91.51**	91.39	91.14	87.04	88.87	93.43
	Weather	70.99	72.44	72.74	72.72	70.41	69.45	**73.10**	81.42



(a) (b)

Figura 20 – Comparação estatística das acurácias de todos os métodos em todos os cenários do número de rótulos: (a) 100% mudanças abruptas, (b) 100% mudanças graduais, através do Teste F_F e do Pós-Teste *Nemenyi*, com 95% de confiança, usando o classificador HT e bases de dados artificiais.



(a) (b)

Figura 21 – Comparação estatística das acurácias de todos os métodos em todos os cenários do número de rótulos: (a) 100% mudanças abruptas, (b) 100% mudanças graduais, através do Teste F_F e do Pós-Teste *Nemenyi*, com 95% de confiança, usando o classificador NB e bases de dados artificiais.