

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO – BCC

FERNANDA MARIA DE SOUZA

**ADAPTAÇÃO DA BIBLIOTECA AXGBOOST PARA REGRESSÃO EM FLUXOS DE
DADOS COM DETECÇÃO E RESPOSTA À MUDANÇA DE CONCEITOS
REALIZADA POR MÚLTIPLOS DETECTORES**

JOINVILLE

2022

FERNANDA MARIA DE SOUZA

**ADAPTAÇÃO DA BIBLIOTECA AXGBOOST PARA REGRESSÃO EM FLUXOS DE
DADOS COM DETECÇÃO E RESPOSTA À MUDANÇA DE CONCEITOS
REALIZADA POR MÚLTIPLOS DETECTORES**

Trabalho de conclusão de curso submetido à Universidade do Estado de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Fabiano Baldo

JOINVILLE

2022

Para gerar a ficha catalográfica de teses e
dissertações acessar o link:
<https://www.udesc.br/bu/manuais/ficha>

,
Adaptação da Biblioteca AXGBoost para Regressão em
Fluxos de Dados com Detecção e Resposta à Mudança de
Conceitos Realizada por Múltiplos Detectores / Fernanda
Maria de Souza. - Joinville, 2022.
54 p. : il. ; 30 cm.

Orientador: Fabiano Baldo.

.
Dissertação - Universidade do Estado de Santa
Catarina, Centro de Ciências Tecnológicas, Bacharelado
em Ciência Da Computação, Joinville, 2022.

1. Regressão. 2. Mudança de conceito. 3. Fluxos
de dados. 4. AXGB. I. Baldo, Fabiano. II. , . III.
Universidade do Estado de Santa Catarina, Centro
de Ciências Tecnológicas, Bacharelado em Ciência Da
Computação. IV. Título.

FERNANDA MARIA DE SOUZA

**ADAPTAÇÃO DA BIBLIOTECA AXGBOOST PARA REGRESSÃO EM FLUXOS DE
DADOS COM DETECÇÃO E RESPOSTA À MUDANÇA DE CONCEITOS
REALIZADA POR MÚLTIPLOS DETECTORES**

Trabalho de conclusão de curso submetido à Universidade do Estado de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Fabiano Baldo

BANCA EXAMINADORA:

Fabiano Baldo - Doutor
UDESC

Membros:

Rafael Stubs Parpinelli - Doutor
UDESC

Fernando José Braz - Doutor
IFC

Joinville, 03 de fevereiro de 2022

RESUMO

Com cada vez mais dados sendo gerados de maneira exponencial, a ciência de dados transformou-se em uma área essencial para compreensão e identificação de padrões de informação. Esses dados, muitas vezes podem vir de forma contínua e sem limitações, sendo conhecidos como fluxo de dados. Atualmente, para fluxo de dados, embora alguns sistemas de aprendizado de máquina tenham sido propostos, a regressão atraiu menos atenção como área de estudo. Conciliado a isso, existindo hoje a técnica de AXGB focada na classificação de fluxos de dados, neste trabalho o objetivo será portar o AXGB para regressão de fluxos de dados. Além disso, no contexto da classificação e regressão em fluxo de dados, o modelo preditivo precisa se adaptar às mudanças de conceito. A mesma ocorre quando as distribuições de dados mudam ao longo do tempo de maneira imprevisível, consequentemente tornando o resultados dos algoritmos de aprendizado de máquina inacurados. Para identificação rápida dessas mudanças, será utilizado uma abordagem para detecção ativa por meio da união de três detectores: KSWIN, ADWIN e DDM. A partir da detecção, o mecanismo para adaptação utilizado será a reinicialização do tamanho da janela deslizante do algoritmo. Portanto, tendo em vista que em fluxo de dados contínuos as informações chegam a todo momento e estão suscetíveis a passar por mudança de conceito, da mesma forma que o processo de regressão precisa se adaptar para um modelo condizente com novas amostras e tendo em vista que a técnica AXGB para regressão ainda não foi elaborada, este trabalho apresenta a seguinte pergunta de pesquisa: Como a aplicação de diferentes técnicas de detecção de mudança de conceito pode diminuir o tempo de adaptação dos modelos às mudanças em regressão de fluxos de dados utilizando o algoritmo AXGB?

Palavras-chave: Regressão, Mudança de conceito, Fluxos de dados, AXGB.

ABSTRACT

With more and more data being generated exponentially, data science has become an essential area for understanding and identifying information patterns. This data can often come in a continuous and unconstrained form, and is known as data streaming. Currently, for data streams, although some machine learning systems have been proposed, regression has attracted less attention as an area of study. Conciliated with that, existing today the AXGB technique focused on classification of data streams, in this paper the aim will be to port AXGB to regression for data streams. Moreover, in the context of data stream classification and regression, the predictive model needs to adapt to concept drift. The same occurs when data distributions change over time in unpredictable ways, consequently rendering the results of machine learning algorithms inaccurate. To quickly identify these changes, an approach will be used for active detection by joining three detectors: KSWIN, ADWIN, and DDM. From the detection, the mechanism for adaptation used will be to reset the size of the sliding window of the algorithm. Therefore, considering that in continuous data flow the information arrives all the time and is susceptible to undergo concept drift, in the same way that the regression process needs to adapt to a model that is consistent with new samples, and considering that the AXGB technique for regression has not yet been elaborated, this work presents the following research question: How can the application of different types of concept drift detectors decrease the time for detecting the incidence of concept drift in data flow, verifying when the model needs to be retrained, applying regression in continuous data flow carrying the AXGB technique?

Keywords: Regression, Concept Drift, Data Streams, AXGB.

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxo do Aprendizado Supervisionado	16
Figura 2 – Fluxo do Aprendizado Não-Supervisionado	16
Figura 3 – Fluxo do XGBoost.	18
Figura 4 – Funcionamento do XGB - parte 1.	19
Figura 5 – Funcionamento do XGB - parte 2.	19
Figura 6 – Funcionamento do XGB - parte 3.	19
Figura 7 – Tipos de mudança de conceito	21
Figura 8 – Diagrama geral da detecção de mudança de conceito	23
Figura 9 – Detecção de mudança de conceito pelo método KSWIN.	24
Figura 10 – Fluxo de dados do projeto	37
Figura 11 – Comparativo entre os algoritmos utilizando o gerador <i>AGRAWAL</i> com mu- dança de conceito na posição 10.000	42
Figura 12 – Segmento do processamento do fluxo onde ocorreu a mudança de conceito utilizando o gerador <i>AGRAWAL</i>	42
Figura 13 – Comparativo dos classificadores utilizando o <i>dataset Hyperplane-fast</i>	44
Figura 14 – Comparativo dos classificadores utilizando o <i>dataset Airlines</i>	45
Figura 15 – Comparativo dos classificadores utilizando o <i>dataset SEA</i>	46

LISTA DE TABELAS

Tabela 1 – Resumo dos trabalhos relacionados	31
Tabela 2 – Bases de dados	39
Tabela 3 – Tempo de execução de 100.000 dados (em segundos)	41
Tabela 4 – Aprofundamento das acurácias em cada instante do fluxo de dados do gerador AGRAWAL para 100000 instâncias	43
Tabela 5 – Tempo de execução de 100.000 dados (em segundos)	43
Tabela 6 – Tempo de execução de 100.000 dados (em segundos)	45
Tabela 7 – Tempo de execução de 100.000 dados (em segundos)	46
Tabela 8 – Cronograma das próximas etapas	49

LISTA DE ABREVIATURAS E SIGLAS

AXGB	Adaptive eXtreme Gradient Boosting
XGB	eXtreme Gradient Boosting
ADWIN	Adaptive Windowing
KSWIN	Kolmogorov-Smirnov Windowing
DDM	Drift Detection Method
SEA	Streaming Ensemble Algorithm
HYP	Hyperplane Generator
IOT	Internet Of Things
MSE	Mean Squared Error
AM	Aprendizado de Máquina
PHT	Page Hincley
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
KNN	K-Nearest Neighbors Algorithm
PWPAE	Performance Weighted Probability Averaging Ensemble
ARF	Adaptive Random Fores
SRP	Streaming Random Patches

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVO GERAL	13
1.2	OBJETIVOS ESPECÍFICOS	13
1.3	METODOLOGIA	13
1.4	ESTRUTURA DO TRABALHO	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	APRENDIZADO DE MÁQUINA	15
2.1.1	<i>Feedback</i>	15
2.1.2	Objetivo	16
2.1.3	Conjuntos de Classificadores	17
2.1.4	XGBoost	18
2.2	REGRESSÃO DE FLUXO DE DADOS	19
2.2.1	Mudança de conceito	20
2.2.2	Métodos de detecção ativa de mudança de conceito	22
2.2.2.1	<i>ADWIN</i>	22
2.2.2.2	<i>KSWIN</i>	24
2.2.2.3	<i>DDM</i>	25
2.2.2.4	<i>Page Hincley</i>	25
2.2.3	Métodos de Validação e Medidas de Avaliação	26
2.3	TRABALHOS RELACIONADOS	28
2.3.1	Considerações sobre os Trabalhos Relacionados	30
2.4	CONSIDERAÇÕES SOBRE O CAPÍTULO	32
3	DESENVOLVIMENTO	34
3.1	PROPOSTA	34
3.2	PARÂMETROS	35
3.3	ALGORITMO	35
3.4	IMPLEMENTAÇÃO	38
4	ANÁLISE DOS RESULTADOS PARCIAIS	39
4.1	METODOLOGIA DE AVALIAÇÃO	39
4.1.1	Parâmetros	40
4.2	RESULTADOS	40
4.2.1	Base AGRAWAL – Mudança Abrupta	40
4.2.1.1	<i>Análise do Tempo de Execução</i>	41
4.2.1.2	<i>Análise da Acurácia e Estatística Kappa na Presença de Mudança de Conceito Abrupta</i>	41

4.2.2	Base HYP – Mudança Incremental	43
4.2.2.1	<i>Análise do Tempo de Execução</i>	43
4.2.2.2	<i>Análise da Acurácia e Estatística Kappa na Presença de Mudança de Conceito Incremental</i>	44
4.2.3	Base Airlines – Sem Mudança	44
4.2.3.1	<i>Análise do Tempo de Execução</i>	44
4.2.3.2	<i>Análise da Acurácia e Estatística Kappa Sem a Presença de Mudança de Conceito</i>	45
4.2.4	Base SEA – Mudança Gradual	46
4.2.4.1	<i>Análise do Tempo de Execução</i>	46
4.2.4.2	<i>Análise da Acurácia e Estatística Kappa na Presença de Mudança de Conceito Gradual</i>	46
5	CONSIDERAÇÕES PARCIAIS	48
5.1	CRONOGRAMA DAS PRÓXIMAS ETAPAS	49
	REFERÊNCIAS	50

1 INTRODUÇÃO

Atualmente, com a evolução das tecnologias de armazenamento em nuvem, a melhoria da infraestrutura de comunicações e da popularização dos dispositivos móveis, dados são produzidos continuamente em escala crescente. A partir da observação dessa abundância de dados é que surgiu o conceito de *Big Data*, cuja ciência de dados tenta extrair potenciais benefícios de sua análise para a sociedade (LARSON; CHANG, 2016). O termo *Big Data* visa descrever o crescimento exponencial de dados com base em cinco princípios: i) o volume de dados, ii) a variabilidade dos tipos de dados, iii) a velocidade que os dados são gerados, capturados e processados, iv) a sua veracidade, de forma a obter dados verídicos, de acordo com a realidade e v) o valor, os custos envolvidos nessa operação e o valor agregado de todo esse trabalho (LANEY, 2001).

Considerando os potenciais benefícios da extração de padrões e informações relevantes a partir de grandes volumes de dados, dentro da ciência de dados se consolidou o conceito de mineração de dados. Nele, a partir de algoritmos de aprendizado de máquina, é possível explorar um conjunto de dados com o objetivo de estabelecer relações difíceis de visualizar manualmente.

No contexto do aprendizado de máquina, a grande maioria das pesquisas se concentra em construir modelos estáticos, que são treinados uma vez sobre um conjunto de dados de treinamento e depois são aplicados a análise de novos dados (GAMAGE; PREMARATNE, 2017). Contudo, com a crescente geração de dados por sensores, dispositivos IoT (Internet of Things), tráfego de rede de computadores, conversas telefônicas e transações financeiras, a demanda por processamento de fluxos de dados não estacionários, que mudam continuamente com o tempo, vêm crescendo consideravelmente. Os fluxos de dados podem ser entendidos como sequências de dados não estacionários gerados de forma constante e por tempo indeterminado (YU; LU; ZHANG, 2021). Essas características fazem com que os algoritmos de mineração não possam armazenar os dados e que eles precisem lidar com as mudanças de conceitos inerentes à característica não estacionária dos fluxos (KRAWCZYK et al., 2017a).

Com isso, a mineração de fluxos de dados lida com desafios sobre as seguintes dimensões: i) rapidez: os dados são processados em tempo limitado; ii) eficiência de memória: os dados processados anteriormente precisam ser descartados; iii) não iterativo: os dados são processados uma única vez; e iv) adaptativo: o modelo de aprendizagem deve ser ajustado às mudanças de conceito (LU et al., 2018).

No contexto de aprendizado de máquina existem duas abordagens principais utilizadas pelos algoritmos para aprender sobre os dados processados. A primeira é o aprendizado não supervisionado, onde não há um conjunto anotado de treinamento e, portanto, a própria estrutura dos dados é utilizada para realizar o aprendizado. A segunda é o aprendizado supervisionado, cujo objetivo é projetar um modelo usando exemplos anotados de treinamento para prever corretamente os rótulos para novos dados não rotulados (MARSLAND, 2011). Para ambas as abordagens, quando os rótulos a serem previstos são discretos, o problema é resolvido como

uma tarefa de classificação. Entretanto, quando os rótulos a serem previstos são contínuos, então, ela é resolvida como uma tarefa de regressão.

Assim como na classificação, no contexto da regressão em fluxo de dados, o modelo preditivo precisa se adaptar às mudanças de conceito. Portanto, para dados que mudam dinamicamente, os métodos tradicionais de aprendizagem supervisionada são inadequados e podem causar perda no desempenho do modelo. Para tentar manter a melhor precisão possível, o modelo treinado precisa se adaptar usando novos dados de entrada. Portanto, os modelos preditivos precisam ser treinados de forma incremental, seja por atualização contínua ou por retreinamento usando lotes recentes de dados (GAMA et al., 2014b).

Embora existam algumas propostas de abordagens para regressão de fluxos de dados, quando comparadas às iniciativas de classificação de fluxos de dados, ela atrai menos atenção da comunidade científica, pois é considerada um problema mais complexo e, portanto, requer mais esforço para solução (YU; LU; ZHANG, 2021).

No aprendizado de máquina, a mudança de conceito significa que as propriedades estatísticas da variável de destino, que o modelo está tentando prever, mudam ao longo do tempo de maneira imprevista (LIAO; WANG, 2018). Portanto, isso resulta em problemas nas predições, pois o modelo se torna menos preciso com o tempo. As formas de detecção de mudança de conceito para aprendizado de máquina adaptativo podem ser classificadas em duas categorias principais, passivas e ativas. Nas formas passivas, os modelos de aprendizagem são atualizados sem a detecção prévia da mudança de conceito (DITZLER et al., 2015), portanto, tendem a serem mais lentas na detecção da mudança. Nas ativas, as abordagens são focadas em detectar primeiro o desvio/mudança de conceito e, posteriormente, acelerar o processo de atualização do modelo para que a alteração seja adsorvida pelo modelo o quanto antes, tornando o processo de detecção mais rápido do que na passiva (ELWELL; POLIKAR, 2011).

Para lidar com a mudança de conceito em dados não estacionários, as abordagens passivas são capazes de se adaptar com atualizações contínuas, mas exigem maior esforços computacionais por serem mais lentas (MEHMOOD et al., 2021). Entretanto, as ativas tentam a serem mais rápidas, porém, introduzem ao processo de aprendizado mais uma tarefa a ser executada, que é a execução do algoritmo de detecção. Neste trabalho, será explorada a utilização de detecção ativa de mudança de conceito a fim de explorar seus potenciais benefícios e limitações em relação a detecção passiva na regressão de fluxos de dados.

Dentre as abordagens para a classificação e regressão de fluxos de dados, os conjuntos de classificadores se destacam com os mais robustos e precisos. Analisando especificamente as abordagens de conjuntos de classificadores, os algoritmos de *boosting* representam uma abordagem promissora, pois são capazes de produzir um modelo geral preciso a partir da combinação de vários modelos moderadamente imprecisos (SCHAPIRE, 2003). A ideia básica desses algoritmos é aplicar iterativamente classificadores e combinar suas soluções para obter um resultado com melhor predição (MAYR et al., 2014). Dois de seus algoritmos mais famosos são o *AdaBoost* e o *Gradient Boosting*, que trabalham adicionando sequencialmente modelos que

corrigem os erros residuais de predição do modelo anterior, geralmente constituídos de árvores de decisão.

Porém, a construção de modelos baseados na abordagem *boosting* pode ser lenta, especialmente para grandes conjuntos. Portanto, para mitigar esse problema foi proposto o algoritmo *eXtreme Gradient Boosting* (XGBoost), o qual cria árvores individuais usando vários processadores e os dados são organizados para minimizar o tempo de pesquisa, o que diminui o tempo de treinamento e melhora o desempenho (RAMRAJ et al., 2016). O algoritmo XGBoost é eficaz para uma ampla gama de problemas de modelagem preditiva de classificação e regressão (CHEN et al., 2015), porém, sua aplicação em fluxo de dados ainda foi pouco explorada. Os principais trabalhos são dedicados a classificação de fluxos de dados, tais como, o *Adaptative Extreme Gradient Boosting* (AXGB). O AXGB usa uma estratégia incremental para criar o conjunto, ao invés de usar os mesmos dados para selecionar cada função, ele usa sub amostras de dados obtidos de janelas não sobrepostas (MONTIEL et al., 2020).

Portanto, sabendo que em fluxo de dados as informações chegam de forma contínua e estão suscetíveis a apresentarem mudança de conceito, e tendo em vista que a técnica AXGB demonstrou bons resultados na classificação mas ainda não foi aplicada na regressão de fluxos, este trabalho apresenta a seguinte pergunta de pesquisa: Como a aplicação de diferentes técnicas de detecção de mudança de conceito pode diminuir o tempo de adaptação dos modelos às mudanças em regressão de fluxos de dados utilizando o algoritmo AXGB?

1.1 OBJETIVO GERAL

Este trabalho tem como objetivo portar o algoritmo AXGB para regressão de fluxo de dados, e introduzir nele técnicas de detecção ativa de mudança de conceito para acelerar o processo de adaptação dos modelos.

1.2 OBJETIVOS ESPECÍFICOS

Os seguintes objetivos específicos foram definidos baseados no objetivo geral:

- Adaptar o algoritmo AXGB para regressão;
- Adaptar o algoritmo AXGB para suportar a execução de técnicas ativas de detecção de mudança de conceito.
- Implementar as adaptações projetadas na biblioteca AXGB.

1.3 METODOLOGIA

O procedimento para realizar este trabalho se inicia com a revisão e estudo dos conceitos abordados pela temática da pesquisa, sendo eles: algoritmos de regressão, aprendizado

supervisionado, fluxo de dados, mudança de conceito, algoritmos de *boosting*, detectores ativos de mudança de conceito e métricas de regressão para validação. Além disso, os trabalhos relacionados encontrados na literatura também serão revisados.

Após o levantamento bibliográfico, é feito o projeto do que será necessário alterar para portar a biblioteca AXGB para regressão em fluxo de dados. O algoritmo implementado deve ser capaz de suportar fluxo de dados e lidar com a mudança de conceito. Além disso, os testes para verificação de mudança de conceito serão implementados por meio de testes de detecção ativa. Após detalhamento, o algoritmo será implementado utilizando a linguagem de programação Python.

Após terminada a implementação, o algoritmo será testado e validado em bases de dados reais e sintéticas, utilizando métricas como MSE e R2. Os resultados da utilização da detecção ativa serão comparados aos sem a utilização da detecção ativa para verificar se houve alguma vantagem na sua utilização.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está dividido da seguinte maneira. O Capítulo 2 traz uma revisão dos principais conceitos envolvidos no entendimento do trabalho, bem como uma revisão dos trabalhos relacionados. O Capítulo 3 apresenta a proposta de algoritmo baseado na união de detectores ativos de mudança de conceito visando aumento de acurácia na regressão de fluxos de dados. O Capítulo 4 apresenta uma análise dos resultados parciais. Por fim, o Capítulo 5 apresenta as considerações parciais e os próximos passos.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos necessários para o entendimento deste trabalho. Serão abordados os conceitos de Aprendizado de Máquina e suas técnicas, os desafios existentes envolvendo fluxos de dados, bem como uma análise dos trabalhos relacionados.

2.1 APRENDIZADO DE MÁQUINA

O Aprendizado de Máquina (AM) é um campo da ciência da computação, derivado da Inteligência Artificial (IA), que introduz nos sistemas formas para automaticamente aprenderem conceitos e conhecimentos sem serem explicitamente programados para isso (DARGAN et al., 2019).

Ainda, segundo Provost e Kohavi (1998), o AM explora o estudo e construção de algoritmos que podem aprender e fazer previsões sobre dados. Com isso, esses algoritmos superam instruções de programas estritamente estáticas, realizando previsões ou decisões por meio da construção de um modelo baseado em conjuntos de dados, que possuem características (atributos) comuns.

Em um modelo de aprendizado de máquina, parte dos dados são usados para treinar o modelo, enquanto outra parte para testar o modelo. Neste cenário, *overfitting* (super-especialização) e *underfitting* (sub-especialização) são algumas das maiores causas de má generalização dos algoritmos de aprendizado de máquina. Koehrsen (2018) define esses dois conceitos: "o *overfitting* se caracteriza pelo momento onde o modelo se adaptou muito bem aos dados com os quais está sendo treinado, porém para novos dados de teste que chegam isso não acontece da mesma forma. Inversamente, a sub-especialização *underfitting* é definida quando o modelo não se adapta bem sequer aos dados com os quais foi treinado".

Ademais, algoritmos de aprendizado de máquina podem ser divididos por duas características principais, de acordo com a categorização feita por Verbraeken et al. (2020),:

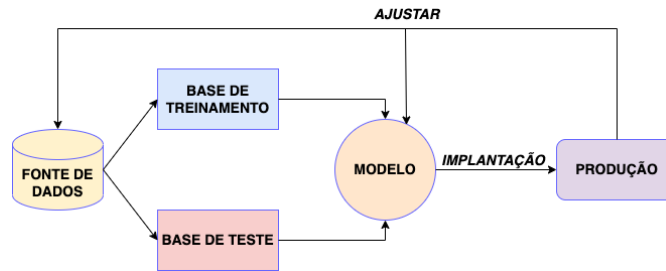
- *Feedback* - o tipo de experiência que é dada ao algoritmo durante o aprendizado.
- Objetivo - o resultado final desejado do algoritmo.

2.1.1 *Feedback*

Nesse tipo de abordagem, no treinamento do algoritmo ele requer um *feedback* para que possa melhorar gradativamente a qualidade do modelo. Existem diferentes tipos de *feedback*:

- **Aprendizado Supervisionado:** nesse método, o algoritmo é treinado usando exemplos rotulados, como uma entrada onde a saída desejada é conhecida. Portanto, o objetivo principal desse aprendizado é fazer com que a função que mapeia os dados de entrada possa ser aplicada a novos dados de entrada visando prever a saída. A Figura 1 apresenta o fluxograma do processo de aprendizado supervisionado.

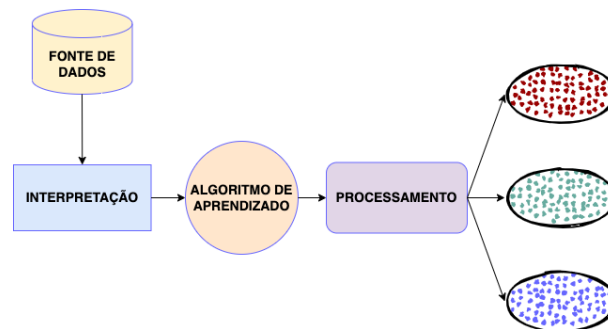
Figura 1 – Fluxo do Aprendizado Supervisionado



Fonte: O autor.

- **Aprendizado Não-Supervisionado:** as instâncias do conjunto de entrada não possuem classificação, ou seja, usa dados de treinamento que consistem em objetos de entrada sem valores de saída. Um ponto importante a ser destacado é que esse tipo de técnica de aprendizado não gera modelos. Como ilustrado na Figura 2, quando novos dados são introduzidos, ele usa todos as instâncias do conjunto de entrada para reconhecer a classe dos dados.

Figura 2 – Fluxo do Aprendizado Não-Supervisionado



Fonte: O autor.

- **Aprendizado Semi-Supervisionado:** é uma combinação dos dois métodos anteriores, usando dados rotulados e não rotulados para treinamento e construção de modelos.

2.1.2 Objetivo

De acordo com Kwon et al. (2019), os algoritmos de AM podem ser usados para uma ampla variedade de propósitos, porém normalmente são utilizados para dois objetivos principais: a classificação e a regressão.

- **Classificação:** a classificação de dados é o processo de formação do modelo ou classificador para prever os rótulos categóricos, que são os rótulos responsáveis por tornar as classes de dados distintas (HAN; PEI; KAMBER, 2011). A classificação é um processo de duas etapas. Durante a primeira etapa, o modelo é criado aplicando-se o algoritmo de classificação no conjunto de dados de treinamento. Durante a segunda etapa o modelo extraído deve ser validado contra um conjunto de dados teste predefinido, visando avaliar o desempenho e precisão do modelo treinado (NIKAM, 2015).

- **Regressão:** seu objetivo é estimar valores numéricos ao invés de classes. Modelos de regressão fazem uso de atributos ou recursos de dados de entrada e seus valores de saída numéricos contínuos correspondentes, visando aprender relações e associações específicas entre as entradas e suas saídas correspondentes (GOLLAPUDI, 2016).

Portanto, enquanto a classificação é a tarefa de prever um rótulo de classe discreta, a regressão é a tarefa de prever uma quantidade contínua. (MAULUD; ABDULAZEEZ, 2020).

2.1.3 Conjuntos de Classificadores

O aprendizado por conjuntos de classificadores ganhou uma popularidade significativa na comunidade de mineração de dados ao longo dos últimos anos (KRAWCZYK; PFAHRINGER; WOŹNIAK, 2018). De acordo com Krawczyk et al. (2017a), um *ensemble*, também chamado de conjunto de classificadores, é a união de componentes individuais cujas previsões são combinadas para prever novas instâncias de entrada. Eles têm se mostrado uma maneira eficiente de melhorar a acurácia preditiva, bem como decompor um problema de aprendizado complexo e difícil em subproblemas mais fáceis (DONG et al., 2020).

Para Ho (2000), os principais objetivos no projeto de um conjunto de classificadores são:

- A otimização da cobertura deve focar na geração de um conjunto de classificadores mutuamente complementares, que podem ser combinados para alcançar a acurácia ideal usando uma função de combinação de decisão fixa.
- A otimização de decisões deve concentrar-se em projetar e treinar uma função de combinação de decisão apropriada, responsável pela decisão final do conjunto, que deve explorar os pontos fortes dos classificadores.

Um conjunto ideal inclui classificadores individuais mutuamente complementares que são caracterizados por alta diversidade e acurácia (KROGH; VEDELSBY et al., 1995). Ainda, segundo Zenobi e Cunningham (2001), é geralmente aceito que não apenas a precisão, mas também a diversidade dos classificadores, são ingredientes chave para aumentar a precisão do conjunto. Com o objetivo de combinar os resultados desses classificadores surgiram técnicas de destaque, tais como:

- **Bagging:** Breiman (1996) introduziu o conceito de *Bootstrapped Aggregating* para construir conjuntos de classificadores. Galar et al. (2011) definiu que o método de *Bagging* consiste em treinar diferentes classificadores com réplicas *bootstrap* do conjunto de dados de treinamento original. Assim, quando uma instância desconhecida é apresentada a cada classificador, a maioria define qual a classe mais prevista por eles.
- **Boosting:** O método de Boosting foi proposto por Schapire (1990), que provou que um aprendiz fraco (que é ligeiramente melhor do que uma adivinhação aleatória) pode ser

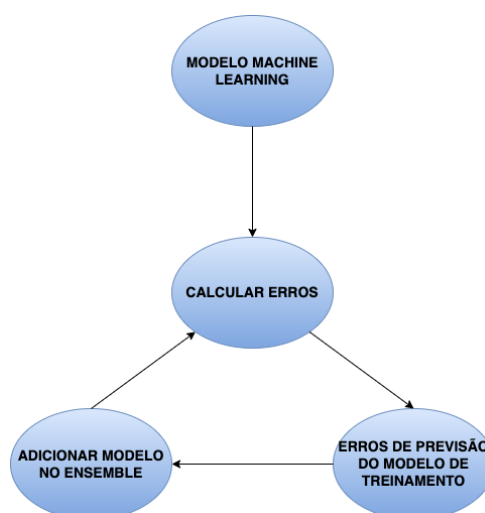
transformado em um aprendiz forte. Para isso, novos classificadores são influenciados pela saída de classificadores anteriores, de forma que a mudança nos dados de treinamento para os novos modelos é baseada nas instâncias mal classificadas anteriormente. Com isso, o *Boosting* tenta melhorar o desempenho de cada novo modelo de forma iterativa com base em modelos anteriores.

- ***Gradient Boosting***: Este método, assim como o algoritmo de *Boosting*, treina novos modelos baseados nos anteriores com o objetivo de aumentar a acurácia do classificador. A principal diferença é como os algoritmos identificam as deficiências dos aprendizes fracos. Para o *Gradient Boosting*, existe uma função de perda a ser otimizada, que indica quão bons são os coeficientes do modelo em ajustar os dados subjacentes.

2.1.4 XGBoost

O XGBoost é um algoritmo de aprendizado de máquina escalável desenvolvido sobre a abordagem *Boosting* proposto por Chen e Guestrin (2016). Ele é o resultado de uma otimização do *Gradient Boosting* que explora os benefícios da paralelização na criação dos modelos do conjunto e possui uma biblioteca de código aberto projetado para ser altamente eficiente e flexível.

Figura 3 – Fluxo do XGBoost.

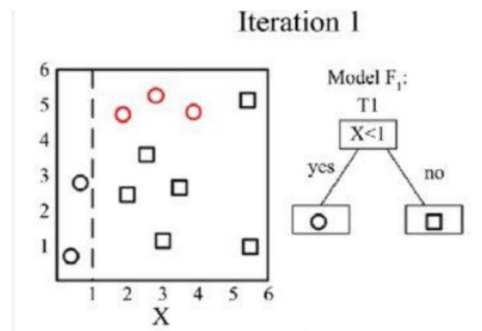


Fonte: O autor.

Assim com no *Gradient Boosting*, no XGBoost a cada iteração o resíduo do preditor anterior é usado para corrigir o próximo preditor de maneira eficiente pela otimização da função de perda, conforme Figura 3. Dessa forma, um peso maior é dado às instâncias que o XGBoost tem maior dificuldade de prever visando estabelecer a função objetiva com melhor desempenho para o modelo. A flexibilidade do *XGBoost* se dá pela utilização de hiper-parâmetros passíveis de aperfeiçoamento, onde é possível ajustar o *XGBoost* para utilização em diferentes problemas.

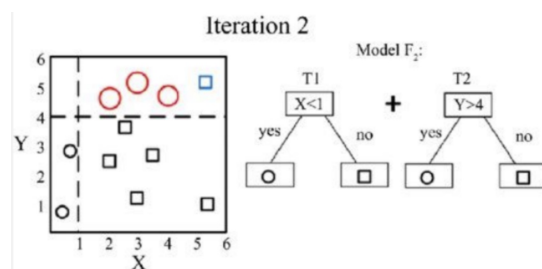
As Figuras 4, 5 e 6 demonstram o funcionamento com o processo iterativo de aprendizado da mesma forma que o XGB utiliza. O objetivo é combinar vários aprendizes fracos para produzir

Figura 4 – Funcionamento do XGB - parte 1.



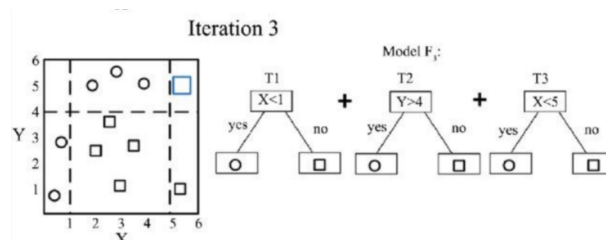
Fonte: Fafalios, Charonyktakis e Tsamardinos (2020).

Figura 5 – Funcionamento do XGB - parte 2.



Fonte: Fafalios, Charonyktakis e Tsamardinos (2020).

Figura 6 – Funcionamento do XGB - parte 3.



Fonte: Fafalios, Charonyktakis e Tsamardinos (2020).

um aprendiz poderoso. Os aprendizes são construídos sequencialmente e os resultados finais são somados tentando reduzir o viés. A partir da primeira iteração o modelo é testado e os resíduos considerados para aprendizados futuros, de modo a construir novos modelos de forma paralelizada com árvores de decisão.

2.2 REGRESSÃO DE FLUXO DE DADOS

A aplicação do aprendizado de máquina em fluxos de dados tem crescido em importância nos últimos anos devido à grande quantidade de dados em tempo real gerados por redes, telefones celulares e a grande variedade de sensores atualmente disponíveis (DONG et al., 2020). Fluxo de dados pode ser definido como uma sequência de exemplos, contextualmente chamados de eventos, que ocorrem no tempo em uma taxa que não permite o seu armazenamento permanente em memória (REIS, 2017).

O processo de mineração de fluxo de dados difere do processo de mineração de dados tradicional devido ao ambiente e às características dos dados serem diferentes (HAN; PEI; KAMBER, 2011). Bifet (2009) descreve as principais restrições para os algoritmos que processam fluxos de dados:

- Não é possível armazenar todos os dados do fluxo: somente uma pequena parte pode ser processada e armazenada;
- A velocidade de chegada dos dados do fluxo exige que cada elemento seja processado em tempo real para em seguida ser descartado;
- A distribuição dos dados pode mudar com o tempo, logo, os dados que foram processados anteriormente e fazem parte do modelo podem se tornar irrelevantes e até mesmo prejudicar os conceitos atuais.

Junto com a classificação, a tarefa de regressão é um dos tópicos mais comuns na área de aprendizado de máquina. No entanto, a regressão baseada em fluxos de dados é ainda pouco explorada no meio acadêmico, embora sua pesquisa seja bastante significativa para a era do aprendizado de máquina Liao e Wang (2018). Muitos métodos que utilizam conjuntos de classificadores podem ser encontrados na literatura para resolver tarefas de classificação em fluxos de dados, entretanto, para a tarefa de regressão são poucos. Dentre eles, os que merecem destaque são: **AMRules** (DUARTE; GAMA; BIFET, 2016), **eFIMT-DD** (IKONOMOVSKA; GAMA; DŽEROSKI, 2015), **Análise de regressão linear multidimensional de fluxos de dados de séries temporais** (CHEN et al., 2002), e **Rival Learner** (LIAO; WANG, 2018).

Outra característica importante do fluxo de dados é a mudança de conceito. Os modelos de aprendizagem devem prever mudanças à medida que os dados evoluem, por isso é necessário adaptar o modelo aos dados mais recentes.

2.2.1 Mudança de conceito

Um dos principais pressupostos dos tradicionais métodos de mineração de dados é que cada conjunto de dados é gerado a partir de uma única função estática, ou seja, a função usada para gerar os dados de treinamento é a mesma usada para testes (HOENS; POLIKAR; CHAWLA, 2012). No modelo de fluxo de dados isso não necessariamente é verdade, visto que a função pode variar durante o tempo de geração do fluxo. Essa (potencial) variação é conhecida como mudança de conceito.

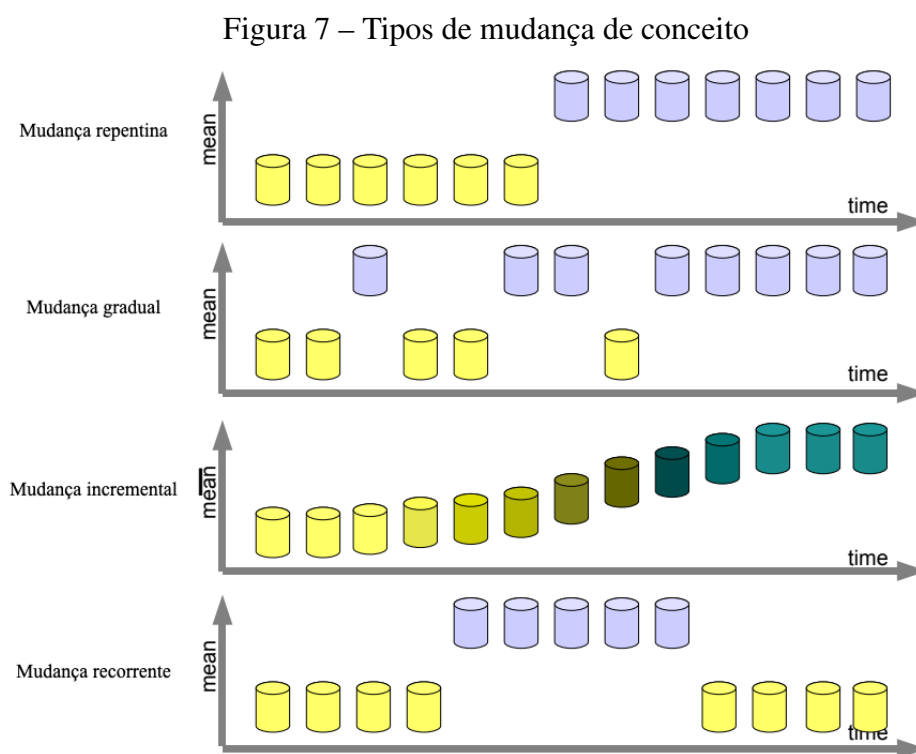
Logo, a mudança de conceito ocorre quando as distribuições de dados mudam ao longo do tempo de maneira imprevisível, tornando os resultados consequentemente imprecisos (PINAGÉ et al., 2017). Widmer (1994) define a mudança de conceito da seguinte forma: “Em muitos domínios do mundo real, o contexto em que alguns conceitos de interesse dependem pode mudar, resultando em mudanças mais ou menos abruptas e radicais na definição do conceito-alvo. A mudança no conceito alvo é conhecido como mudança de conceito”.

Um exemplo da ocorrência de mudança de conceito em situações reais é referente aos padrões de preferências de compras de clientes que muitas vezes podem alterar com o tempo, dependendo do dia da semana, avaliação de alternativas, inflação e outros fatores (NUNES, 2017). Outro domínio real são os ambientes de *IoT* (*Internet of Things*) não estacionários, onde a análise de fluxo de dados muitas vezes enfrenta desafios de mudança de conceito em que as distribuições de dados mudam ao longo do tempo.

Segundo Lu et al. (2018), a mudança de conceito pode ser caracterizada em quatro tipos:

- **Repentina:** onde um determinado conceito é instantaneamente alterado, sendo substituído por outro;
- **Gradual:** ocorrem alternâncias entre os conceitos, com um novo conceito substituindo gradualmente o antigo ao longo do tempo;
- **Incremental:** as mudanças ocorrem de forma mais lenta, com o conceito sendo inserido de forma incremental;
- **Recorrente:** é aquela quando um conceito que existia anteriormente e foi extinto reaparece depois de algum tempo.

A Figura 7 resume os tipos de mudança existentes.



Fonte: Adaptado de Žliobaitė (2010).

Desse modo, visando lidar com os diversos tipos de mudanças de conceito em fluxo de dados, de acordo com Bose et al. (2013), as soluções de mudança de conceito devem se

concentrar em duas direções: como detectar desvios (mudanças); e como adaptar o modelo preditivo aos desvios.

Segundo Krawczyk et al. (2017a), a detecção de mudanças de conceito pode ser feita de duas maneiras:

- **Ativa:** As abordagens ativas são projetadas para detectar mudanças de conceito usando diferentes tipos de detectores. Se existir desvio de conceito, o modelo é atualizado (MAHDI et al., 2020).
- **Passiva:** As abordagens passivas atualizam continuamente o modelo sempre que novos dados se tornam disponíveis, independentemente se a mudança está ocorrendo ou não (MAHDI et al., 2020).

Neste trabalho, serão explorados diferentes métodos de detecção ativa de mudança de conceito, cuja abordagem e funcionamento são descritos na subseção 2.2.2.

2.2.2 Métodos de detecção ativa de mudança de conceito

A detecção refere-se às técnicas e mecanismos que caracterizam e quantificam a mudança de conceito por meio da identificação de pontos de mudança ou intervalos de mudança de tempo (BASSEVILLE; NIKIFOROV et al., 1993). A Figura 8 aborda uma estrutura geral de como funcionam os *frameworks* de detecção de mudança de conceito.

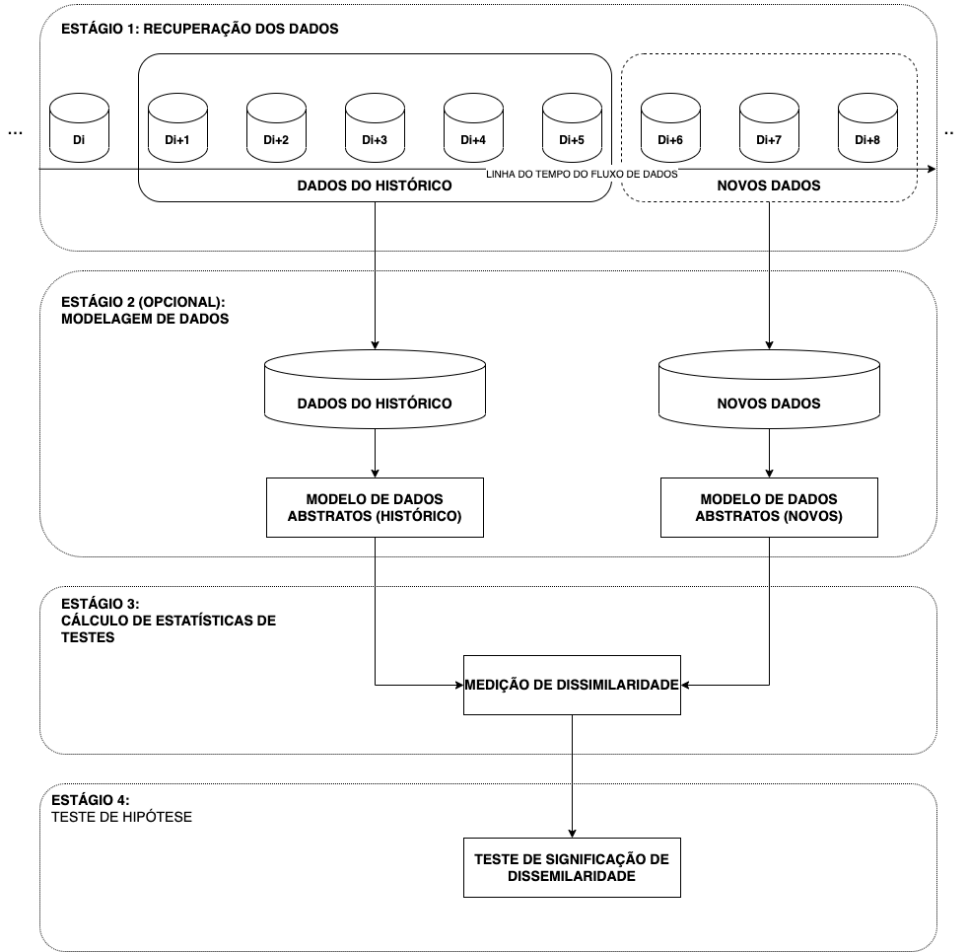
No Estágio 1 o objetivo é recuperar blocos de dados do fluxo de dados, enquanto no Estágio 2 é abstraído dos dados recuperados os principais recursos que mais impactam um sistema se eles sofrem com mudança de conceito. Adiante, no Estágio 3 é calculado a medida de dissimilaridade, ou estimativa de distância: é quantificado a gravidade da mudança de conceito e formando estatísticas de teste para o teste de hipótese. A Fase 4 (Teste de Hipótese) usa um teste de hipótese específico para avaliar a significância estatística da mudança observada no Estágio 3, ou o valor-p.

Sem o Estágio 4, as estatísticas de teste adquiridas no Estágio 3 não fazem sentido para detecção de mudança de conceito, porque eles não podem determinar o intervalo de confiança para a mudança de conceito, ou seja, quão provável é que a mudança é causada por uma mudança de conceito e não por um ruído ou uma amostra aleatória (LU et al., 2018).

2.2.2.1 ADWIN

O detector Adaptive Sliding Window (ADWIN) monitora uma sequência de entradas de valor real usando janelas deslizantes (BARDDAL, 2019). Ele usa duas janelas: (i) uma janela de referência e (ii) uma janela de teste. A janela de referência ajuda a calcular a taxa de erro e outros parâmetros quando não há alteração e o valor de saída é tomado como referência. A janela de teste representa o novo lote de dados e o algoritmo monitora sua taxa de erro na detecção da mudança de conceito (MANICKASWAMY; BHUVANESWARI, 2020). O ADWIN

Figura 8 – Diagrama geral da detecção de mudança de conceito



Fonte: Adaptado de Lu et al. (2018).

identifica a ocorrência de mudança de conceito por uma grande diferença entre as médias das duas sub-janelas. Uma vez que um ponto de mudança é detectado, todas as amostras de dados antigas antes desse ponto de tempo de mudança são descartados (YANG; MANIAS; SHAMI, 2021).

Além disso, como parâmetros o ADWIN não exige que os usuários definam antecipadamente o tamanho das janelas comparadas, apenas é configurado o valor de confiança $\delta \in (0, 1)$ para ajustar o valor de sensibilidade da detecção de mudança de conceito (GRULICH et al., 2018).

Sejam n_0 e n_1 os tamanhos de W_0 e W_1 e n o tamanho de W (janela deslizando), então $n = n_0 + n_1$. Ainda, μ_{w_0} e μ_{w_1} sejam os valores médios em W_0 e W_1 , e μ_{w_0} e μ_{w_1} seus valores esperados. Para obter uma performance rigorosa, é definido no algoritmo:

$$m = \frac{1}{\frac{1}{n_0} + \frac{1}{n_1}}; \quad (1)$$

$$\delta' = \frac{\delta}{n}; \quad (2)$$

$$\epsilon_{cut} = \sqrt{\frac{1}{2m} \cdot \ln \frac{4}{\delta'}} \quad (3)$$

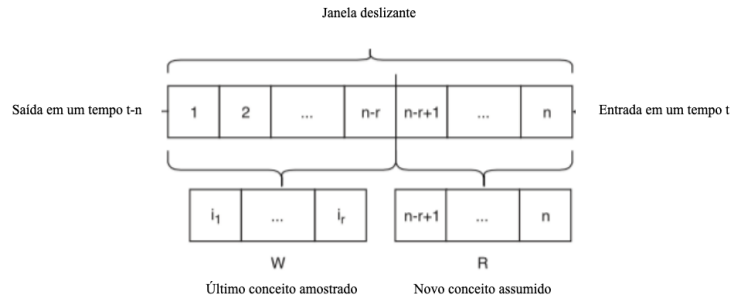
O teste estatístico para diferentes distribuições em W_0 e W_1 simplesmente verifica se a média observada em ambas as sub janelas difere por mais do que o limite de corte ϵ_{cut} (BIFET; GAVALDA, 2007).

O método de detecção ativa ADWIN pode detectar efetivamente desvios graduais, pois a janela deslizante pode ser estendida para uma janela de grande porte para identificar mudanças de longo prazo (YANG; MANIAS; SHAMI, 2021).

2.2.2.2 KSWIN

KSWIN é um método recente para detecção de mudança de conceito que é baseado no teste estatístico de Kolmogorov-Smirnov (KS) (LOPES; REID; HOBSON, 2007). O teste de KS é um teste não paramétrico que não precisa de nenhuma suposição da distribuição de dados subjacente. Semelhante ao ADWIN, o KSWIN também implementa as duas funções: adicionar os novos dados unidimensionais na janela deslizante local; e a segunda função é a que informa se o desvio foi detectado ou não. A primeira janela tem um tamanho fixo. Assim, antes de adicionar qualquer novo dado nessa janela, os dados antigos são removidos e os novos são adicionado na fila da janela deslizante (TOGBE et al., 2021). A Figura 9 exhibe esse processo.

Figura 9 – Detecção de mudança de conceito pelo método KSWIN.



Fonte: Adaptado de Togbe et al. (2021).

Para o KSWIN, a janela deslizante local L é dividida em duas partes. A primeira chamada de R contém o parâmetro predefinido r com os dados mais recentes que chegaram:

$$R = \{x_i \in L\}_{i=n-r+1}^n \quad (4)$$

A segunda janela W seleciona uniformemente dados dos dados antigos em L . A distribuição uniforme usada é:

$$U(x|1, n-r) = \frac{1}{n-r} \quad (5)$$

Portanto,

$$W = \{x_i \in L | i < n - r + 1, p(x) = U(x_i | 1, n - r)\} \quad (6)$$

A mudança de conceito é detectada pelo KSWIN quando a $dist(R, W) > \sqrt{-\frac{\ln(\alpha)}{r}}$, onde α é a probabilidade para a estatística do teste KS.

2.2.2.3 DDM

O DDM (*Drift Detection Method*) monitora a taxa de erro, assumindo que ela diminuirá enquanto o número de instâncias aumentar e a distribuição de dados for estacionária no processo de aprendizagem incremental (YAN, 2020). Ele usa o conceito de distribuição binomial e quando a distribuição de dados é estável, o erro do modelo diminui com o tempo e vice-versa (ABBASZADEH; AMIRI; KHANTEYMOORI, 2015). O método possui dois modos: o nível de aviso, no qual os dados de entrada são armazenados na memória temporária; e o nível de mudança, em que o modelo do algoritmo funciona reaprendendo a partir dos dados na memória temporária (KRAWCZYK et al., 2017b).

O DDM foi o primeiro algoritmo a definir um nível de alerta e o nível de mudança para detecção de mudança de conceito. Quando uma nova instância de dados fica disponível para avaliação, o DDM detecta se a taxa de erro geral dentro da janela de tempo aumentou significativamente. Se o nível de confiança da taxa de erro observada atingir o nível de aviso, o DDM começa a construir um novo aprendiz ao usar o antigo aprendiz para previsões. Se a mudança atingiu o nível de mudança de conceito, o antigo aprendiz será substituído pelo novo aprendiz para outras tarefas de previsão (LU et al., 2018).

O DDM funciona melhor com dados com mudança repentina em que gradualmente conceitos podem passar sem acionar o nível de mudança (GAMA et al., 2014a).

2.2.2.4 Page Hincley

O detector *Page-Hinckley* (PHT) é uma abordagem acumulativa sem memória para monitorar o desempenho dos aprendizes ao longo do tempo. O PHT é uma adaptação da detecção de mudança abrupta na média de um sinal gaussiano, e funciona acumulando a diferença entre os valores observados e sua média até o momento atual (BARDDAL, 2019). Sempre que a média extrapola um limite dado pelo usuário, um desvio é sinalizado.

O teste de PHT considera uma variável acumulativa m_T , definida como a diferença acumulada entre os valores observados e suas médias até o momento atual:

$$m_{t+1} = \sum_1^t (x_t - x_t + \alpha) \quad (7)$$

onde, $x = 1/t \sum_{l=1}^t x_l$ e α corresponde a magnitude das mudanças que são permitidas.

O menor valor dessa variável é também computado pela seguinte fórmula:

$$M_T = \min(m_t, t = 1 \dots T) \quad (8)$$

Então, o teste monitora a diferença entre M_T e m_T :

$$PHT = m_T - M_T \quad (9)$$

Quando a diferença é maior que o valor limite definido por δ , é disparado um alarme de mudança de conceito na distribuição.

2.2.3 Métodos de Validação e Medidas de Avaliação

Diversas técnicas foram criadas e adaptadas ao longo dos anos para medir a eficiência de algoritmos de classificação e regressão. Por isso, um conceito amplamente aceito é escolher a estratégia de avaliação do desempenho baseado no objetivo da aplicação.

Os modelos de previsão atribuem rótulos a uma determinada instância que quando comparado com o rótulo real pode ser:

- **True Positives (TP)**: Refere-se à quantidade de instâncias verdadeiras previstas corretamente;
- **False Negatives (FN)**: Refere-se à quantidade de instâncias falsas previstas incorretamente;
- **True Negatives (TN)**: Refere-se à quantidade de instâncias falsas previstas corretamente;
- **False Positives (FP)**: Refere-se à quantidade de instâncias verdadeiras previstas incorretamente.

Com base nisso, é possível utilizar as seguintes relações para se analisar um classificador:

- **True Positive Rate (TPR)**: é a quantidade de classificações verdadeiras feitas corretamente dividido pelo mesmo valor somado a quantidade de classificações falsas incorretas.

$$TPR = \frac{TP}{TP + FN} \quad (10)$$

- **False Positive Rate (FPR)**: é a quantidade de classificações verdadeiras previstas incorretamente sobre o mesmo valor somado com a quantidade de verdadeiros negativos.

$$FPR = \frac{FP}{FP + TN} \quad (11)$$

- **Matthews Correlation Coefficient (MCC):** é a única taxa de classificação binária que consegue produzir uma pontuação alta se o preditor foi capaz de obter bons resultados nas quatro categorias da matriz de confusão (CHICCO; JURMAN, 2020).

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \quad (12)$$

Krawczyk et al. (2017a) destaca os meios mais populares de se avaliar a capacidade preditiva de um algoritmo de classificação:

- **Acurácia:** é a proporção das previsões corretas feitas pelo algoritmo;

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \quad (13)$$

- **Sensibilidade:** diz respeito a precisão de uma determinada classe;

$$Sensibilidade = \frac{TP}{TP + FN} \quad (14)$$

- **Média G:** similar a sensibilidade mas voltado a fluxos de dados desbalanceados, que consiste em uma ou mais classes predominantes no decorrer do fluxo;
- **Estatística Kappa:** considera a acurácia e a probabilidade de um classificador aleatório fazer uma predição correta;
- **Estatística Kappa Generalizadas:** similar a Estatística Kappa, mas voltada a fluxos de dados desbalanceados.

Para a regressão, a predição resulta em um valor numérico contínuo, envolvendo assim um grau de incerteza quanto a exatidão do valor predito. Com isso, ao invés de focar no valor exato, o objetivo é minimizar a distância entre o valor predito (y') e o valor real (y). Algumas métricas relacionadas com a regressão são:

- **Erro Médio Absoluto ou Mean Absolute Error (MAE):** refere-se à magnitude da diferença entre a previsão de uma observação e o valor real dessa observação:

$$\frac{1}{n} \sum_{i=1}^n |y - y'| \quad (15)$$

- **Erro Médio Quadrado ou Mean Squared Error (MSE):** é comumente usado para verificar a acurácia de modelos e dá um maior peso aos maiores erros.

$$\frac{1}{n} \sum_{i=1}^n |y - y'|^2 \quad (16)$$

- R^2 : R-quadrado é uma medida estatística de quão próximos os dados estão da linha de regressão ajustada. Também é conhecido como coeficiente de determinação, ou coeficiente de determinação múltipla para regressão múltipla. R-quadrado está sempre entre 0 e 100 %: 0% indica que o modelo não explica nenhuma variabilidade dos dados de resposta em torno de sua média. 100% indica que o modelo explica toda a variabilidade dos dados de resposta em torno de sua média.
- **Erro Médio Absoluto Percentual ou *Mean Absolute Percentage Error* (MAPE)**: é uma medida de quão preciso é um algoritmo de regressão. Ele mede essa precisão como uma porcentagem e é a média de todos os erros absolutos percentuais, fornecendo uma indicação do tamanho médio do erro.

2.3 TRABALHOS RELACIONADOS

Os trabalhos relacionados foram selecionados com base em temáticas envolvendo o *AXGBoost*, regressão de fluxo de dados e também em estratégias para lidar com a mudança de conceito. Esta seção apresenta o resumo da revisão da literatura que foi realizada.

A metodologia utilizada para encontrar artigos pertinentes ao tema se baseou na pesquisa nas principais bibliotecas digitais, como ACM, IEEE e *Google Scholar*. As palavras-chave utilizadas foram "*regression data stream*", "*concept drift*" e "*drift detectors*".

Chen e Guestrin (2016) propôs o algoritmo XGBoost, baseado no aprendizado de máquina escalável para *Boosting* de árvore. Segundo os autores, o XGBoost executa até dez vezes mais rápido que as soluções populares existentes em uma única máquina e escala para bilhões de exemplos em distribuição ou configurações de memória limitada. As principais contribuições do artigo foram: desenvolvimento de uma árvore de ponta a ponta altamente escalável com sistema de *Boosting*; introdução de um novo algoritmo com reconhecimento de esparsidade para aprendizado de árvores paralelas; e por fim, uma estrutura de bloco com reconhecimento de *cache* eficaz para aprendizado em árvore fora do núcleo.

Montiel et al. (2020) propuseram o algoritmo chamado AXGB (*Adaptive eXtreme Gradient Boosting*), como uma adaptação do XGBoost, para a classificação de fluxo de dados com mudança de conceito. A ideia central do AXGB é a criação/atualização incremental do conjunto de classificadores, ou seja, aprendizes fracos são treinados em mini-lotes de dados e então adicionados ao conjunto. O tamanho máximo do conjunto é fixo, mas quando ele é atingido os classificadores mais antigos são descartados para dar lugar aos novos modelos para garantir o bom desempenho do conjunto. A atualização do conjunto pode ser feita de duas formas: *Push*: com modelos novos sendo adicionados na mesma medida que os antigos são removidos; e *Replacement*, os modelos mais antigos são substituídos pelos novos modelos. Segundo os autores, a partir dos testes realizados, o método de *Replacement* apresentou o melhor desempenho em termos de acurácia, tempo de treinamento e uso de memória.

Bonassa (2021) propôs uma adaptação ao AXGB (MONTIEL et al., 2020) com intuito de diminuir o seu tempo de treinamento, eliminando da proposta original o conjunto de classificadores. Para resolver esse problema, foi utilizada uma característica presente no XGBoost que permite salvar o modelo que está sendo treinado. Dessa forma, sempre que a janela deslizante de entrada de dados está cheia, o modelo treinado salvo é carregado e atualizado com os dados do novo lote de treinamento, e salvo posteriormente. Utilizando essa estratégia, o modelo se torna incremental sem a necessidade de utilizar um conjunto de modelos e também evitando a super-especialização. Além disso, Bonassa (2021) também adaptou o AXGB para o aprendizado semi-supervisionado. Essa adaptação foi feita pela implementação de um *wrapper* para pseudo-rotulagem dos dados não classificados utilizando o algoritmo KNN. Em relação aos testes, a acurácia permaneceu a mesma em comparação ao AXGB original proposto por Montiel et al. (2020), porém com tempo de execução até 10 vezes menor que o original. Isso ocorreu devido à retirada do conjunto de classificadores utilizados na proposta original do AXGB.

Liao e Wang (2018) elaborou um trabalho envolvendo a regressão em fluxo de dados com adaptação a mudança de conceito por meio da criação de um novo algoritmo, denominado *Rival Learner*. O algoritmo utiliza um conjunto de classificadores e possui dois submodelos, um baseado em amostras históricas, chamado de modelo global, e o outro baseado em amostras de janelas deslizantes atuais, chamado de modelo local. Esses dois modelos competem entre si para ver quem tem o melhor desempenho e o vencedor assume como modelo global. Nos testes realizados pelos autores, quando ocorre um desvio de conceito, o algoritmo proposto mantém o modelo efetivamente atualizado sem necessidade da detecção ativa, mesmo com a necessidade de inicializar o submodelo com frequência, o desempenho do modelo também se mantém eficaz.

Yang, Manias e Shami (2021) descreveu um *framework* de conjuntos de classificadores para adaptação à mudança de conceito em fluxo de dados por meio da união de detectores ativos chamado *Performance Weighted Probability Averaging Ensemble* (PWPAE). O *framework* proposto é baseado no aprendizado de um conjunto de modelos que usa as combinações de dois métodos de detecção, ADWIN e DDM, e dois métodos de adaptação à deriva, *Adaptive Random Forest* (ARF) e *Streaming Random Patches* (SRP), para construir aprendizes básicos. Os aprendizes são ponderados de acordo com seu desempenho em tempo real e integrados para construir um modelo robusto. Os resultados apresentados pelos autores mostram que o algoritmo alcançou altos índices de acurácia comparado com os trabalhos de última geração.

Boulegane et al. (2020) introduziu uma nova abordagem para previsão de séries temporais em fluxo de dados usando o método de seleção dinâmica do conjunto. A seleção de conjunto visa escolher um subconjunto de um conjunto de modelos considerados os melhores preditores para uma instância. No trabalho foi formulado uma tarefa de Regressão *Multi target* onde um meta-aprendiz é responsável por simultaneamente prever o erro de cada modelo base no conjunto, dada uma instância de teste. Ao contrário dos métodos existentes, que não lidam com as dependências entre os modelos base, o método explora as informações de dependência em uma fase inicial ao prever os erros.

Gomes et al. (2018) propuseram uma adaptação ao *Adaptive Random Forest* (ARF) para a tarefa de regressão em fluxo de dados, chamado ARF-Reg . O algoritmo funciona de modo a possuir uma votação responsável por realizar a média das previsões individuais para obter a previsão final. A diversidade também é uma característica importante induzida na floresta, treinando as árvores em diferentes sub-conjuntos de dados e limitando as decisões de divisão a um subconjunto de recursos selecionado aleatoriamente dos recursos de entrada originais. Por sua vez, o aprendiz base é uma árvore de regressão, com o algoritmo de FIMT-DD. Finalmente, a dinâmica de atualização no ARF-Reg depende tanto interna quanto externamente dos detectores de mudança de conceito para cada árvore e pelo crescimento de árvores no *background* quando um aviso é detectado. Como resultado, o ARF-Reg obteve taxas de erro consideravelmente pequenas, principalmente em cenários no mundo real, mostrando assim a sua eficácia.

2.3.1 Considerações sobre os Trabalhos Relacionados

A Tabela 1 apresenta uma comparação entre os trabalhos relacionados revisados na seção 2.3. Para melhor compreensão sobre os critérios utilizados na comparação, a seguir é apresentada uma breve descrição dos critérios utilizados:

1. **Tipo de aprendizagem:** se o algoritmo visa realizar classificação ou regressão;
2. **Fluxo:** se o algoritmo suporta classificação ou regressão de fluxo de dados;
3. **Mudança de conceito:** se o algoritmo trata o problema de mudança de conceito em fluxos;
4. **XGBoost:** se o algoritmo utiliza o XGBoost como base para a classificação ou regressão;

Tabela 1 – Resumo dos trabalhos relacionados

Trabalho	Tipo de aprendizagem	Fluxo de dados	Mudança de conceito	XGBoost
Chen e Guestrin (2016)	Classificação	Não	Não	Sim
Montiel et al. (2020)	Classificação	Sim	Substituição do modelo + ADWIN	Sim
Bonassa (2021)	Classificação	Sim	Substituição do modelo + ADWIN	Sim
Liao e Wang (2018)	Regressão	Sim	Adaptação passiva a mudança de conceito	Não
Yang, Maniás e Shami (2021)	Classificação	Sim	Substituição do modelo + união de detectores ativos de mudança de conceito	Não
Boulegane et al. (2020)	Regressão	Sim	Não	Não
Gomes et al. (2018)	Regressão	Sim	Não	Não

A Tabela 1 apresenta um resumo dos trabalhos relacionados. É possível notar uma escassez de propostas que envolvam a regressão de fluxos de dados utilizando a biblioteca XGBoost

Chen et al. (2016) propôs o XGBoost para classificação e regressão de conjuntos de dados estacionários. Com a adaptação do XGBoost para fluxo de dados feita no trabalho de Montiel et al. (2020), ele pode também ser utilizado em conjuntos de dados não estacionários, pois ele apresenta a possibilidade de lidar com a mudança de conceito, por meio da utilização do detector ativo ADWIN.

Bonassa (2021), por sua vez, propôs uma adaptação ao trabalho de Montiel et al. (2020) para o aprendizado semi-supervisionado, com a classificação rápida realizada por um único classificador, ao invés de utilizar um conjunto como proposto por Montiel et al. (2020). A proposta de Bonassa (2021) melhorou consideravelmente a velocidade de treinamento do modelo sem diminuir a acurácia do algoritmo, também evitando a super-especialização do modelo.

Liao e Wang (2018) é um dos trabalhos mais recentes envolvendo a regressão em fluxo de dados. O algoritmo desenvolvido utiliza uma abordagem incremental para adaptação a mudança de conceito de forma passiva.

Na temática de detecção ativa da mudança de conceito, o trabalho de Yang, Manias e Shami (2021) utiliza diferentes detectores, e tira proveito de suas particularidades e complementariedades, de forma que a depender do tipo de mudança de conceito, alguns detectores específicos podem ser mais propícios para detectar a mudança e atualizar o modelo.

Os trabalhos de Gomes et al. (2018) e Boulegane et al. (2020) são os principais trabalhos sobre regressão em fluxos de dados, porém os algoritmos utilizados por cada um deles não propôs uma alternativa eficaz para adaptação e detecção de mudança de conceito. Por essa razão, ainda é uma área com poucos trabalhos e mais pesquisas sobre essa temática precisam ser realizadas.

2.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Nesse capítulo foram revisados os principais conceitos necessários para o entendimento deste trabalho, sendo eles: aprendizado de máquina, classificação e regressão de dados, classificação e regressão de fluxo de dados, mudança de conceito, detectores ativos para mudança de conceito, XGBoost e métricas de avaliação.

O aprendizado de máquina é uma área que visa aprender automaticamente conceitos e reconhecer padrões sem necessariamente ser programado especificamente para uma dessas tarefas. Dentro dessa área, esses algoritmos possuem duas características principais, que são o *feedback* e objetivo do algoritmo. Os diferentes tipos de *feedback* são o aprendizado supervisionado, aprendizado não-supervisionado e o semi-supervisionado.

O objetivo dos algoritmos pode ser dividido em Classificação e Regressão. Enquanto a primeira é definida para previsão de rótulos categóricos, a segunda foca na previsão de valores numéricos. Por sua vez, dentro da classificação, existe a possibilidade da utilização

de conjuntos de classificadores, conhecidos como *ensembles*. Um conjunto de classificadores inclui classificadores individuais mutuamente complementares, e essa abordagem vem ganhando destaque nos últimos anos dentro da área de aprendizado de máquina.

Com o surgimento de fluxos de dados, o XGBoost foi adaptado para suportá-los, porém originalmente proposta para utilizar um conjunto de classificadores, essa abordagem foi substituída por uma outra cujo algoritmo treina um único classificador. Contudo, o tema da mudança de conceito ainda não foi abordado de forma extensa na literatura sobre XGBoost.

Por ser uma característica de dados não estacionários, a detecção da mudança de conceito também foi estudada nos fluxos de dados. A mudança de conceito pode acontecer de forma repentina, gradual, incremental ou recorrente. Suas formas de detecção se dividem em ativas e passivas. Como o trabalho em tela tem por objetivo explorar a detecção ativa, os detectores ADWIN, KSWIN, DDM e PHT foram abordados e estudados.

Com as fundamentações abordadas foi possível ampliar o conhecimento a respeito das técnicas que serão utilizadas na contribuição deste trabalho, bem como a motivação para sua utilização.

3 DESENVOLVIMENTO

Este capítulo apresenta a proposta de adaptação do algoritmo AXGB para realizar a tarefa de regressão de fluxos de dados. É importante destacar que o AXGB foi originalmente proposta para realizar apenas a classificação de fluxo de dados. Além disso, também é proposta uma adaptação visando prover ao algoritmo uma abordagem ativa de detecção de mudança de conceito. Para isso, a união da aplicação de diferentes detectores ativos de mudança de conceito será explorada.

A Seção 3.1 esboça as adaptações projetadas sobre o algoritmo AXGB original, tanto para realização de regressão como para detecção ativa de mudança de conceito. A Seção 3.2 exhibe os parâmetros necessários na execução do algoritmo. A Seção 3.3 detalha o algoritmo do AXGB e as adaptações nele realizadas. Por fim, a Seção 3.4 explica o processo de implementação.

3.1 PROPOSTA

Como observado na revisão de trabalhos relacionado apresentada na Seção 2.3, a tarefa de regressão, apesar de ser um dos tópicos mais comuns na área de aprendizado de máquina, não é um tema amplamente abordado e estudado na análise de fluxos de dados. Portanto, este trabalho visa adaptar o algoritmo AXGB para suportar regressão de fluxo de dados com detecção ativa de mudança de conceito.

A proposta original do AXGB utilizava um conjunto de classificadores, entretanto isso prejudicava consideravelmente o desempenho do modelo. Por isso, será utilizado como base do trabalho o algoritmo proposto por Bonassa (2021), uma adaptação ao AXGB que obteve menor tempo de treinamento mantendo a mesma acurácia obtida pelo AXGB original. Na proposta de Bonassa (2021), apenas um modelo XGBoost é treinado e atualizado de forma incremental. Entretanto, após esse classificador ultrapassar um limite de dados treinados, um outro classificador é treinado para substituí-lo, evitando assim a super-especialização (*overfitting*) do primeiro.

Embora a estratégia incremental de atualização do modelo AXGB proposta por Bonassa (2021) consiga lidar com a detecção passiva de mudança de conceito, pois o modelo é atualizado com base em dados mais recentes, o ajuste às mudanças abruptas de conceito não são adequadamente suportadas devido sua demora na percepção da mudança. Por isso, este trabalho propõe a inclusão de técnicas ativas de detecção de mudança de conceito ao algoritmo AXGB modificado por Bonassa (2021).

Portanto, este trabalho se propõe a explorar as seguintes técnicas ativas de detecção de mudança de conceito: ADWIN, KSWIN e DDM. Esses detectores têm o objetivo de identificar as mudanças de conceito e acionar o mecanismo de atualização do modelo. Esse mecanismo consiste em diminuir o tamanho da janela de dados vinda do fluxo utilizada para atualizar o modelo, fazendo com que a atualização do modelo seja antecipada, resultando em uma percepção da mudança de conceito mais rápida.

3.2 PARÂMETROS

O XGBoost conta com diversos parâmetros que podem ser alterados para melhorar seu desempenho com base nas características dos dados e do cenário de mineração. O AXGB de (MONTIEL et al., 2020) utiliza alguns deles e inclui alguns próprios, sendo o último parâmetro adicionado pelo trabalho atual desenvolvido com a união de detectores:

- **Tamanho do ensemble ($n_estimators$):** número máximo de modelos dentro do conjunto.
- **Taxa de aprendizado (η):** Um modelo XGBoost envolve a criação de árvores sequencialmente. As novas árvores são criadas para reduzir o erro das anteriores. Quanto mais próximo de 0 menor o aprendizado, e inversamente quanto mais próximo de 1 maior ele é.
- **Profundidade máxima (max_depth):** a profundidade máxima que uma árvore pode atingir. Quanto maior esse valor mais o modelo cresce em complexidade e consumo de memória.
- **Tamanho máximo da janela (max_window_size):** Tamanho máximo do *buffer* que armazena os dados do fluxo. O algoritmo só atualiza os modelos quando o tamanho máximo é atingido.
- **Mudança de conceito ($detect_drift$):** Esse booleano determina se a mudança de conceito com o *array* de detectores vai ser aplicada durante a aprendizagem.

3.3 ALGORITMO

Um pseudocódigo para o algoritmo proposto pode ser visualizado no Algoritmo 1. As modificações implementadas no algoritmo AXGB de Bonassa (2021), destacadas em amarelo, nesta etapa do projeto iniciaram pela inclusão de uma etapa de detecção ativa de mudança de conceito realizada pelo conjunto de detectores composto por: ADWIN, KSWIN e DDM. O modelo de regressão deverá ser portado na continuação deste trabalho.

Como entrada do algoritmo, estão presentes o conjunto x e y pertencente ao fluxo de dados: com x sendo os dados de entrada enquanto y é o valor que irá ser predito. No algoritmo, quando a janela deslizante de dados vindos do fluxo está cheia, o algoritmo verifica se existe algum modelo de regressão XGBoost já treinado (Linha 4). Caso não exista, um regressor XGBoost é treinado (Linha 17). Caso já exista, outras condições são testadas com a proposta de verificar o tempo de vida do regressor (Linhas 5 a 11) e, na sequência o regressor é carregado (Linha 12) e atualizado (Linha 13).

A adaptação para portar o XGBoost para regressão é realizada de maneira a mudar a função objetivo do algoritmo nos parâmetros configuráveis de *boosting*.

Algoritmo 1: Adaptação ao AXGB com a união de detectores ativos

Entrada: $(x, y) \in$ Fluxo de dados

Data: w = quantidade máxima de exemplos no *buffer*, M = Modelo de classificadores XGBoost, W = janela, SW = janela deslizante dos dados rotulados, *life_time* = Tempo de vida de cada regressor, *training_time* = tempo de treinamento de cada regressor, D = array de detectores

```

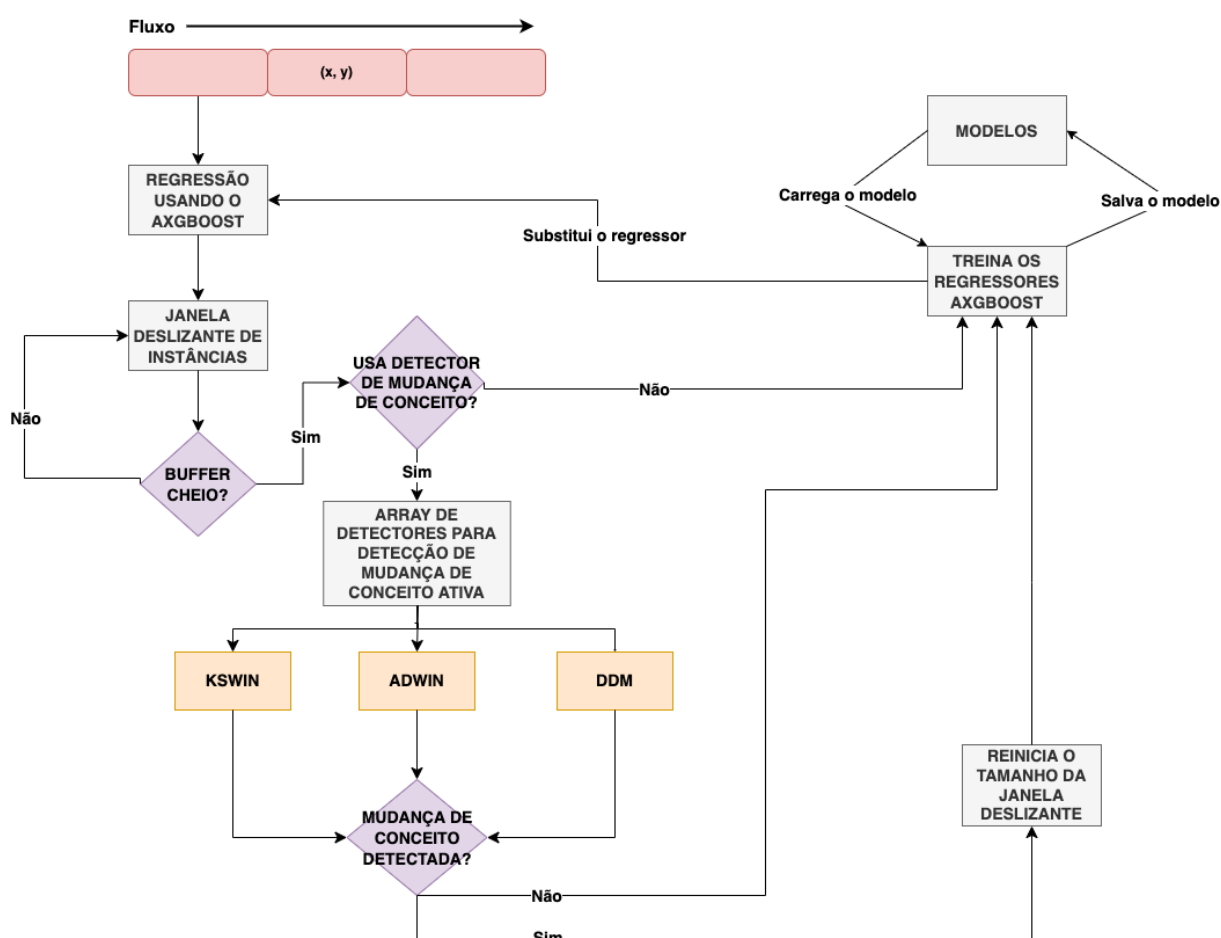
1 Realiza a predição de  $(x, y)$  usando o modelo XGBoost treinado
2 Adicione  $(x, y)$  na janela ( $W$ )
3  $cont = 0$ 
4 se  $M \neq Null$  então
5     se training_time == life_time -  $cont$  então
6         Verifica se esta dentro do tempo de treinamento do próximo
            regressor.
7          $nextM =$  Treina o próximo regressor XGBoost
8     se  $cont \geq life\_time$  então
9         Reinicia o contador e começa a utilizar o novo regressor.
10         $cont = 0$ 
11         $M = nextM$ 
12        se reset_ativo == 'S' então
13             $tamanho\_janela = 0$ 
14         $M =$  Carrega o modelo anterior;
15         $M' =$  Treina regressor XGBoost com  $W'$  usando  $M$  e adiciona a  $M$ ;
16         $M = M'$ 
17        Salva o novo modelo  $M$  treinado;
18 senão
19      $M =$  Treina o regressor XGBoost com  $W'$ ;
20     Salva o novo modelo  $M$  treinado;
21     se detect_drift == True então
22         Calcula o erro absoluto
23          $error = abs(self.predict(X) - y)$ 
24         para cada detector_ativo de  $D$  faça
25             se detectar_mudanca(detector_ativo, error) então
26                 Se algum detector ativo de mudança de conceito detectar alguma
                    mudança com base no erro absoluto, o tamanho da janela é zerado.
27                  $tamanho\_janela = 0$ 
28                 Na primeira detecção de mudança de conceito por algum detector o
                    laço é quebrado, passando para próximas instâncias
29                 break
30      $W = W - W'$ ;
31  $cont = cont + 1$  conta o número de vezes que a janela reiniciou
32 retorna  $M$ ;
  
```

Com o modelo carregado, um *array* de detectores ativos é iterado visando detectar uma mudança de conceito. Cada detector deve ser complementar um ao outro, de forma a agregar em acurácia ou R^2 , não aumentando consideravelmente o tempo de execução ou peso em memória do modelo (Linhas 20 a 23). Para esta implementação, foram escolhidos os detectores ADWIN, KSWIN e DDM. Para o TCC-2 serão estudadas mais a fundo as particularidades de cada um e novos tipos de detecção visando aumentar o desempenho do modelo.

A cada vez que uma mudança de conceito é detectada por algum detector, o tamanho da janela deslizante é zerado.

De maneira semelhante, a partir da Figura 10 é exposto o fluxo de dados do projeto. O principal destaque do diagrama se dá com a implementação do *array* de detectores, onde a partir de uma variável *booleana* é verificado se a estratégia deve ser utilizada. Caso identificada a variável com valor *True* definida pelo usuário, a partir de qualquer mudança de conceito detectada por qualquer detector, é disparado o mecanismo para reinicialização da janela deslizante. Inversamente, caso não for detectado nenhuma mudança de conceito, o fluxo continua para treinar os regressores AXGB.

Figura 10 – Fluxo de dados do projeto



Fonte: O autor.

Logo, as propostas mencionadas neste capítulo e principalmente, o método da união de

detectores, têm como objetivo se adaptar mais rapidamente a qualquer mudança de conceito detectada, consequentemente visando aumentar a acurácia para classificação e o R^2 para regressão, sem prejudicar consideravelmente o tempo de execução completa da predição e o treinamento dos dados presentes no fluxo.

É importante ressaltar a grande quantidade de parâmetros disponíveis pelo XGBoost. É imprescindível alinhar todos os parâmetros da melhor forma possível para assim obter o resultado desejado.

3.4 IMPLEMENTAÇÃO

As adaptações propostas ao AXGB de Bonassa (2021) foram implementadas a partir do código original obtido do próprio autor. O AXGB foi desenvolvido em linguagem de programação *Python*. O *Python* é uma linguagem de programação de alto nível e orientada a objetos, de tipagem dinâmica e forte. Sua utilização é amplamente difundida nas áreas de *Big Data* e Aprendizado de Máquina, possuindo diversas bibliotecas de código aberto disponibilizadas por sua comunidade, bem como o próprio AXGBoost (KADIYALA; KUMAR, 2018).

Uma das bibliotecas *Python* utilizadas é a *scikit-multiflow*. Ela contém vários pacotes com algoritmos e ferramentas de aprendizado de máquina para fluxo de dados, dentre elas, é possível destacar:

- **Mecanismos para geração de fluxos de dados com diferentes aspectos** como: *SEA Generator*, *Agrawal Stream Generator*, *Regression Generator*, *Random Tree Generator*.
- **Monitor com gráficos e métricas para comparação de diferentes algoritmos.**
- **Diferentes algoritmos de conjunto de classificadores** como: *Adaptive Random Forest Regressor* e *Streaming Random Patches Ensemble Classifier*.
- **Detectores de mudança de conceito ativa** como: ADWIN, DDM, KSWIN e Page-Hinkley.

4 ANÁLISE DOS RESULTADOS PARCIAIS

Este capítulo apresenta os resultados dos testes parciais feitos com o algoritmo ainda em desenvolvimento. Esta etapa é fundamental para qualquer trabalho de pesquisa pois é responsável por validar a proposta bem como seus pontos fortes e fracos. Com isso, a Seção 4.1 esboça a metodologia utilizada para a avaliação e a Seção 4.2 detalha os resultados obtidos.

4.1 METODOLOGIA DE AVALIAÇÃO

Para realizar a avaliação do algoritmo proposto, ele foi comparado com o AXGB adaptado de Bonassa (2021) e o AXGB original de Montiel et al. (2020). Esses três algoritmos são descritos abaixo:

- AXGB com conjunto de classificadores *ensemble*: algoritmo proposto por Montiel et al. (2020) utilizando a estratégia *replace*, que foi considerada a melhor pelos próprios autores;
- AXGB com classificadores alternados: proposto por Bonassa (2021), uma adaptação do algoritmo original AXGB que utiliza um único classificador atuando de maneira alternada;
- AXGB com classificadores alternados e união de detectores ativos de mudança de conceito: algoritmo proposto neste trabalho que utiliza a abordagem de classificadores alternados em conjunto com a detecção ativa de mudanças de conceito.

Os aspectos avaliados incluem o tempo de execução, a acurácia e a capacidade de se adaptar a diferentes tipos de mudança de conceito. O método de avaliação selecionado é o *Prequential evaluation*, onde os dados são analisados de forma sequencial quando chegam ao modelo.

As bases de dados utilizadas para os testes foram retiradas da biblioteca *scikit-multiflow* disponibilizadas por Montiel et al. (2018), sendo uma base real chamada *Airlines*, e o restante bases sintéticas por meio dos métodos de geração *Hyperplane Generator*, *AGRAWAL Generator* e *SEA Generator*. A Tabela 2 apresenta as bases geradas ou selecionadas para a avaliação.

Tabela 2 – Bases de dados

Nome	Instâncias	Classes	Tipo da Base	Mudança
AGRAWAL	100000	2	Sintética	Abrupto
HYP	100000	2	Sintética	Incremental
AIRLINES	100000	2	Real	Não especificado
SEA	100000	2	Sintética	Gradual

Fonte: O autor.

4.1.1 Parâmetros

Antes de iniciar os testes é necessário definir os valores para os parâmetros que serão utilizados pelos algoritmos a fim de extrair o melhor desempenho deles. Os valores dos parâmetros foram definidos a partir dos trabalhos de Montiel et al. (2020) e Bonassa (2021). Posteriormente, eles foram calibrados por meio de testes empíricos.

O número de modelos do conjunto aumenta a acurácia do classificador, mas também aumento seu tempo de treinamento. Por isso, é necessário encontrar um equilíbrio entre a quantidade de modelos e o tempo necessário para treiná-los. Com base no trabalho de Bonassa (2021) e em testes empíricos realizados, constatou-se que o número adequado de modelos seria 30.

Outro fator importante é a taxa de aprendizado do modelo, que varia entre 0 e 1: valores próximos a 1 tendem a ocasionar uma perda de acurácia do modelo com o passar do treinamento, pois o algoritmo faz correções muito grandes que podem prejudicar o aprendizado, porém diminui o tempo de processamento. Em contraste, valores próximos a 0 tendem a aumentar a acurácia, mas o tempo de processamento aumenta consideravelmente. Com os testes feitos também por Bonassa (2021), o valor escolhido para o parâmetro taxa de aprendizado foi 0,3.

Para o tamanho da janela, ou seja, o *buffer* que armazena os dados a serem utilizados no treinamento, com base nos testes empíricos realizados o número selecionado foi 1000 instâncias.

A profundidade máxima de uma árvore (*max_depth*) foi definida como 6 devido a uma pequena melhora na acurácia em comparação com os demais valores próximos.

Portanto, os valores selecionados para os parâmetros utilizados pelos algoritmos foram:

- **Classificadores:** 30
- **Taxa de aprendizado:** 0,3
- **Tamanho da janela:** 1000
- **Profundidade máxima de uma árvore:** 6

4.2 RESULTADOS

Esta seção apresenta os resultados de tempo de processamento, acurácia e Kappa alcançados pelo algoritmo adaptado proposto neste trabalho, em relação aos dois outros algoritmos utilizados na comparação. As próximas subseções apresentam os resultados para cada uma das bases de dados utilizadas nos testes.

4.2.1 Base AGRAWAL – Mudança Abrupta

O primeiro teste realizado foi sobre uma base sintética gerada pelo gerador AGRAWAL, com simulação de uma mudança de conceito abrupta na posição 10.000 do fluxo. As próximas

subseções apresentam a análise do tempo de execução, e das métricas de acurácia e Kappa, respectivamente.

4.2.1.1 Análise do Tempo de Execução

Os algoritmos foram executados quatro vezes sobre a mesma base para extrair o tempo médio de execução entre elas. A Tabela 3 compara o tempo de treinamento e de predição entre os 3 algoritmos, onde:

- AXGB e: AXGB com conjunto de classificadores proposto por Montiel et al. (2020);
- AXGB c: AXGB com classificadores alternados proposto por (BONASSA, 2021);
- AXGB d: AXGB com classificadores alternados e união de detectores de mudança proposto neste trabalho.

Tabela 3 – Tempo de execução de 100.000 dados (em segundos)

Classificador	Predição	Treinamento	Total
<i>AXGB e</i>	148,89	1,20	150,09
<i>AXGB c</i>	15,01	1,48	16,49
<i>AXGB d</i>	16,10	31,66	47,76

Fonte: O autor.

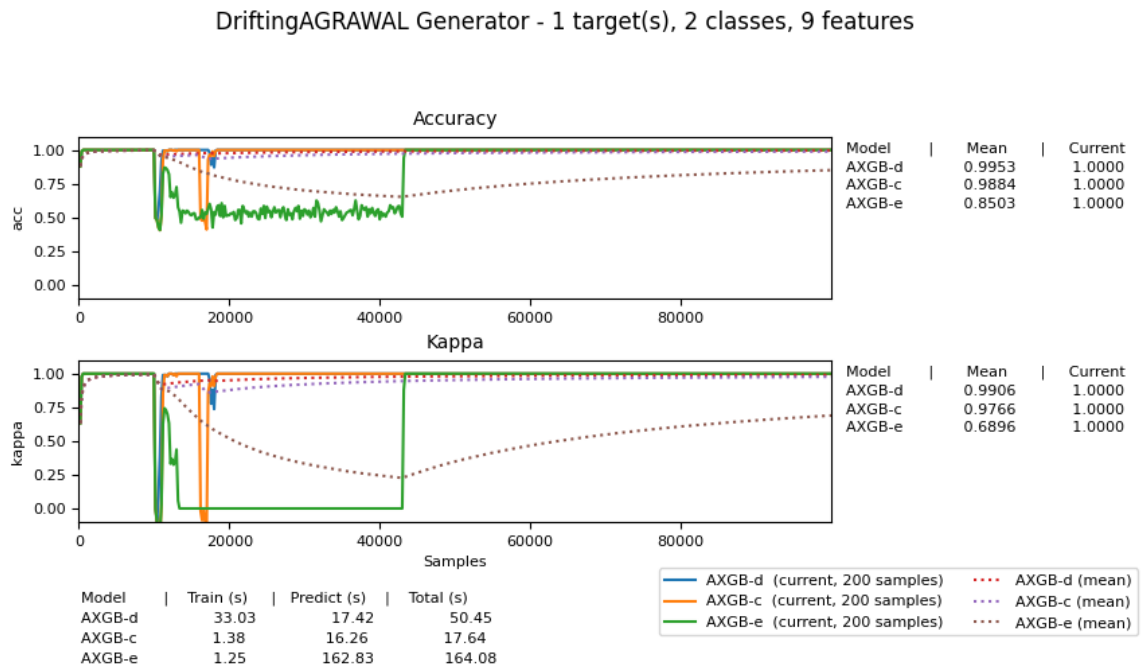
Como pode ser observado na Tabela 3, o *AXGB d* apresentou um ganho significativo no tempo total em comparação com o *AXGB e*, entretanto, ele não foi melhor que o *AXGB c*. Isso aconteceu principalmente porque o *AXGB d* teve um tempo de treinamento consideravelmente superior a ambos. Essa questão pode ser explicada porque é na etapa de treinamento que os detectores de mudança de conceito são executados, como o *AXGB e* e o *AXGB c* não realizam essa tarefa, seus tempos de treinamento são menores. Em relação ao tempo de predição, o *AXGB d* tem tempo semelhante ao *AXGB c*, justamente porque eles compartilham a mesma estratégia de atualização dos modelos, que é realizada por meio da alternância entre modelos.

4.2.1.2 Análise da Acurácia e Estatística Kappa na Presença de Mudança de Conceito Abrupta

Para esse teste foi realizada a geração de 100.000 instâncias utilizando o gerador *AGRAWAL*. Esse gerador pode ser configurado por meio de parâmetros específicos para geração de um fluxo de dados. Para esse teste foi configurada a geração de uma mudança de conceito abrupta na posição 10.000 do fluxo. A Figura 11 mostra os valores da acurácia e da estatística Kappa para os três algoritmos avaliados.

O algoritmo *AXGB e* foi o que apresentou a maior dificuldade para recuperar sua acurácia em relação aos outros. Ele precisou ser treinado com mais de 30.000 instâncias de treinamento para recuperar sua acurácia antes da mudança de conceito. Já o algoritmo *AXGB c* reagiu a

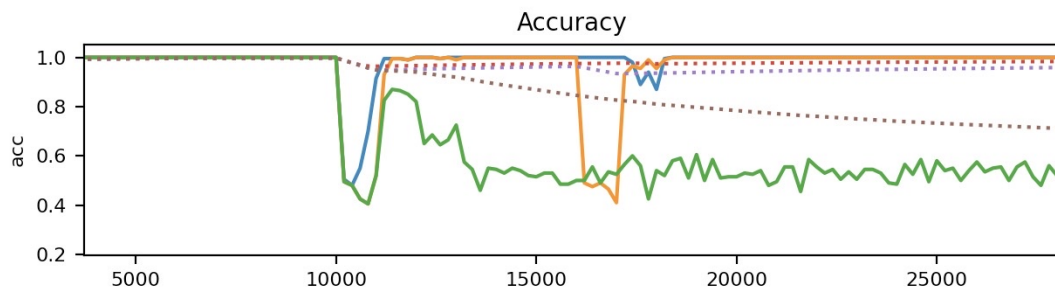
Figura 11 – Comparativo entre os algoritmos utilizando o gerador *AGRAWAL* com mudança de conceito na posição 10.000



Fonte: O autor.

mudança de conceito praticamente junto com o algoritmo *AXGB d*, entretanto, ele apresentou por volta da instância 16.000 mais uma queda acentuada da acurácia, o que não aconteceu com o algoritmo *AXGB d*. A Figura 12 apresenta uma aplicação da Figura 11 focada na área onde aconteceu a mudança de conceito, onde a percepção sobre a capacidade de adaptação dos algoritmos pode ser melhor observada. Por fim, é importante destacar que mesmo apresentado acurácia e estatística Kappa média parecidas, quando se observa a acurácia ou Kappa durante todo o tempo de processamento do fluxo, é possível observar como a detecção ativa consegue acelerar o processo de adaptação dos modelos.

Figura 12 – Segmento do processamento do fluxo onde ocorreu a mudança de conceito utilizando o gerador *AGRAWAL*



Fonte: O autor.

A Tabela 4 detalha as acurácias dos algoritmos no processamento de instâncias após a ocorrência da mudança de conceito na posição 10.000. Como é possível observar, a partir do instante 10799, o *AXGB d* recupera a sua acurácia devido ao mecanismo que diminui o

tamanho da janela deslizante de treinamento dos dados de entrada. No instante 16199, devido à mudança de conceito ainda não ter sido completamente absorvida, o *AXGB c* teve uma nova queda acentuada de acurácia, fato que não aconteceu com o *AXGB d*.

Ainda, vale destacar que essa queda acentuada de acurácia do *AXGB c* no instante 18999, mesmo após ele já ter se adaptado à mudança é uma característica que está relacionada a memória do classificador. Como ele demorou mais tempo para se atualizar, ainda existem partes do modelo que não foram adequadamente atualizadas e quando são acessadas causam erro na predição.

Tabela 4 – Aprofundamento das acurácias em cada instante do fluxo de dados do gerador AGRAWAL para 100000 instâncias

Instante	Acurácia média			Acurácia atual		
	<i>AXGB d</i>	<i>AXGB c</i>	<i>AXGB e</i>	<i>AXGB d</i>	<i>AXGB c</i>	<i>AXGB e</i>
10199	0.9877	0.9875	0.9877	0.5050	0.4950	0.4950
10799	0.9649	0.9569	0.9569	0.7000	0.4050	0.4050
11399	0.9659	0.9494	0.9454	0.9950	0.9950	0.8700
13599	0.9704	0.9572	0.9025	1.0000	1.0000	0.4600
16199	0.9751	0.9577	0.8414	1.0000	0.4900	0.5000
18999	0.9754	0.9390	0.7974	1.0000	1.0000	0.6050

Fonte: O autor.

4.2.2 Base HYP – Mudança Incremental

4.2.2.1 Análise do Tempo de Execução

A segunda base de dados utilizada nos experimentos foi gerada pelo gerador *Hyperplane-fast* (HYP), no qual o Hiperplano com D dimensões muda de posição e orientação. Outra característica dessa base de dados é que ela possui uma mudança de conceito incremental.

A Tabela 5 mostra os tempos de execução da base HYP nos três algoritmos analisados. Nesse teste, os tempos apresentam comportamento similar aos observados nos experimentos da Seção 4.2.1.1. O principal ponto observado é a discrepância no tempo de predição do algoritmo *AXGB e* e o maior tempo de treinamento do algoritmo *AXGB d*.

Tabela 5 – Tempo de execução de 100.000 dados (em segundos)

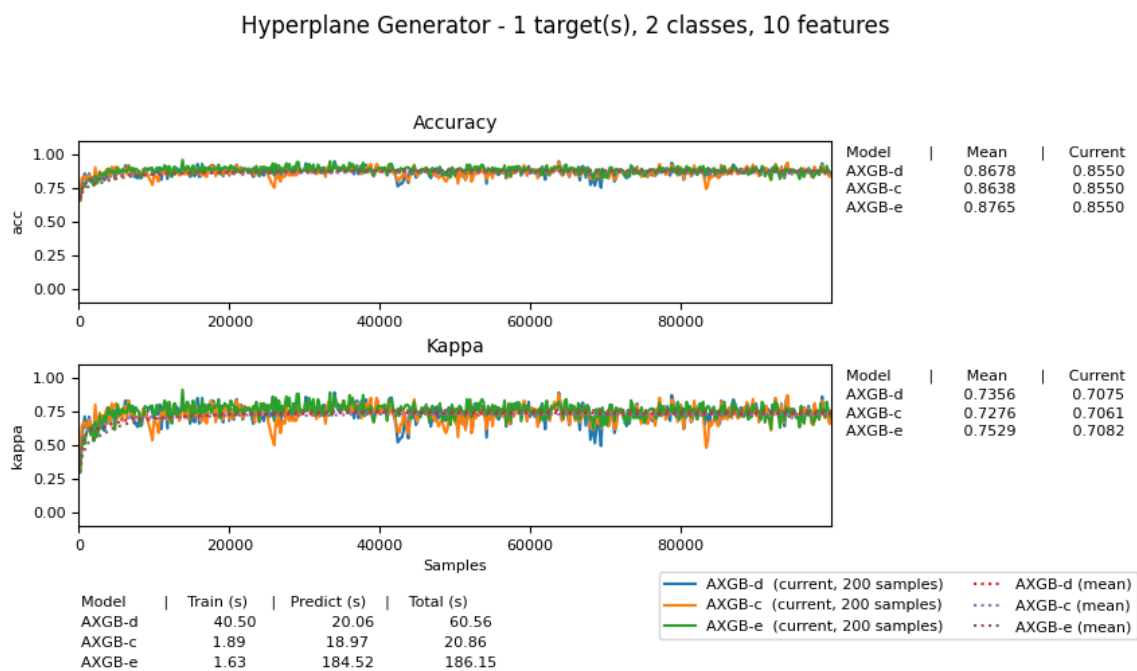
Classificador	Predição	Treinamento	Total
<i>AXGB e</i>	143,89	1,31	145,20
<i>AXGB c</i>	14,38	1,81	16,19
<i>AXGB d</i>	15,38	32,84	48,22

Fonte: O autor.

4.2.2.2 Análise da Acurácia e Estatística Kappa na Presença de Mudança de Conceito Incremental

Os resultados dos três algoritmos comparados podem ser observados na Figura 13. Neste fluxo é possível notar que os modelos possuíram acurácia semelhante. Com relação ao algoritmo AXGB (d), o mesmo ficou com acurácia média de 86,78%, sendo maior comparado ao AXGB (c) com acurácia de 86,38%. Porém, o modelo de *ensembles* obteve uma maior acurácia de 87,65%.

Figura 13 – Comparativo dos classificadores utilizando o *dataset Hyperplane-fast*



Fonte: O autor.

Alguns pontos para se destacar no gráfico de acurácia são as etapas de deterioração de cada classificador. O AXGB (d) apresentou dois pontos consideráveis de queda, enquanto o AXGB (c) resultou em três pontos de queda. Como conclusão, destaca-se a neutralidade do AXGB (e) com *ensemble*, que apresentou melhor acurácia com a mudança de conceito para mudanças de conceito incrementais.

Com isso, o AXGB (d) com união de detectores ainda não é uma estratégia tão eficaz para *drifts* incrementais pois o detector se torna sensível a qualquer mudança com uma força muito maior do que se deve agir, perdendo para a estratégia tradicional com *ensembles*.

4.2.3 Base Airlines – Sem Mudança

4.2.3.1 Análise do Tempo de Execução

O terceiro *dataset* usado nos testes foi o *Airlines*. O mesmo utiliza dados do mundo real contendo informações de partidas programadas de voos comerciais dentro dos EUA.

A partir da Tabela 6 os tempos de execução para cada classificador foram considerados, resultando em um tempo médio após 10 execuções.

Tabela 6 – Tempo de execução de 100.000 dados (em segundos)

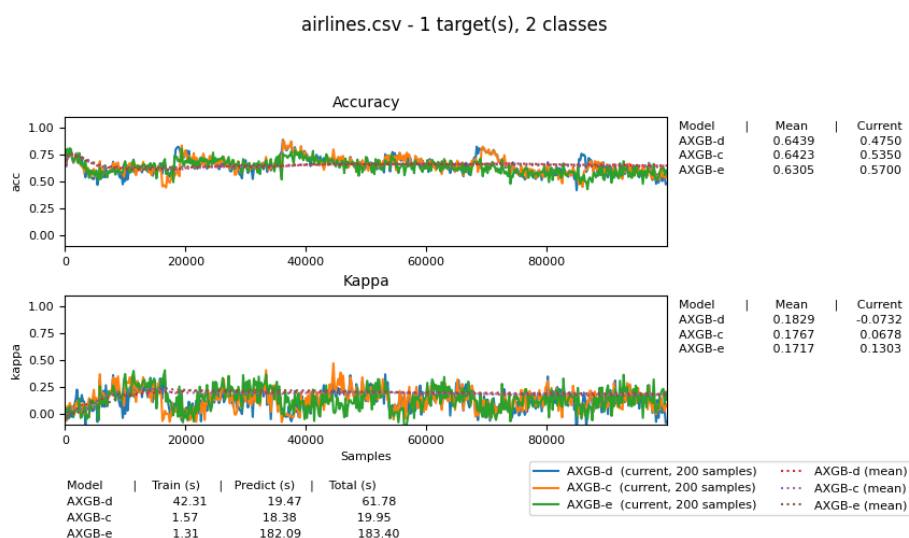
Classificador	Predição	Treinamento	Total
AXGB e	157,75	1,28	159,03
AXGB c	16,83	3,78	20,62
AXGB d	17,71	66,10	83,11

Fonte: O autor.

Com base nos tempos observados, a maior diferença se deve ao tempo de treinamento do AXGB (d), pois a condição para detecção ou não de mudança de conceito pelos detectores iniciados no algoritmo exige um maior tempo do classificador.

4.2.3.2 Análise da Acurácia e Estatística Kappa Sem a Presença de Mudança de Conceito

Os resultados observados podem ser constatados na Figura 14. Os dados representados na Figura 14 demonstram um comportamento estatisticamente semelhante de acurácia e estatística Kappa das versões que apresentam ensemble e somente classificação alternada. A acurácia do AXGB (d) ficou com acurácia de 64,39%, enquanto a do AXGB (c) com 64,23% e por fim, a acurácia do AXGB (e) resultou em 63,05%.

Figura 14 – Comparativo dos classificadores utilizando o *dataset* Airlines

Fonte: O autor.

Com relação ao AXGB (d) com a união de detectores de mudança de conceito, o algoritmo se manteve constante comparado aos outros modelos. Considerando a ansiedade do classificador para se adaptar a novas mudanças de conceito, percebe-se no gráfico da acurácia que qualquer desvio de conceito, o AXGB (d) possui a reação mais rápida e consequente adaptação, zerando o tamanho da janela com as instâncias que possam vir a possuir mais *drifts*.

4.2.4 Base SEA – Mudança Gradual

4.2.4.1 Análise do Tempo de Execução

O quarto *dataset* usado nos testes é chamado SEA, o SEA é um gerador de dados sintéticos com três atributos numéricos em que apenas dois estão relacionados a classe de destino.

Tabela 7 – Tempo de execução de 100.000 dados (em segundos)

Classificador	Predição	Treinamento	Total
AXGB e	19,06	0,21	19,28
AXGB c	2,94	0,32	3,26
AXGB d	3,07	6,52	9,60

Fonte: O autor.

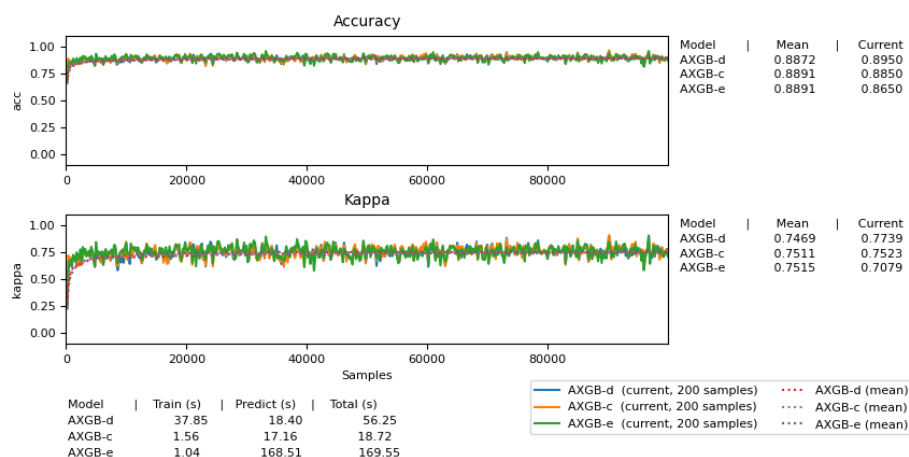
De maneira similar com o tempo de execução observado para a base de dados *Airlines*, o tempo de treinamento do classificador AXGB (d) se manteve acima dos demais pelos mesmos motivos enumerados anteriormente: a união dos detectores ativos de mudança de conceito. Porém, para o gerador de fluxo de dados *SEA*, a partir do tempo total resultante de cada classificador, não se teve uma diferença considerável.

4.2.4.2 Análise da Acurácia e Estatística Kappa na Presença de Mudança de Conceito Gradual

Para a geração do fluxo de dados, o percentual de ruído utilizado como parâmetro foi 0.1. Esse *dataset* compreende uma mudança de fluxo gradual e os resultados são apresentados na Figura 15.

Figura 15 – Comparativo dos classificadores utilizando o *dataset* SEA

SEA Generator - 1 target(s), 2 classes, 3 features



Fonte: O autor.

A partir dos resultados obtidos, todos os classificadores obtiveram resultados semelhantes, com a acurácia do AXGB (d) de 88,71%, em comparação com a precisão do AXGB (c) e do AXGB (e) de 88,91%.

Um ponto interessante a ser observado se deve a característica do classificador AXGB (d) possuir condições sensíveis para detecção de qualquer tipo de mudança de conceito. Conforme observado anteriormente, para uma mudança abrupta, a utilização do conjunto de detectores é extremamente favorável para identificação em menor tempo e recuperação da mudança de conceito. Porém, para mudanças de conceito que acontecem em ocorrências consideráveis de forma gradual, o mecanismo de reinicializar o tamanho da janela com um *drift* detectado pode comprometer a acurácia do modelo, pois desse modo, se perde o que se sabia antes devido a reinicialização do tamanho da janela para zero.

5 CONSIDERAÇÕES PARCIAIS

Com o grande crescimento da geração de fluxo de dados em tempo real, extrair informações e previsões úteis tornou-se uma parte importante em diversas áreas. Porém, por se tratarem de dados não estacionários, uma série de dificuldades surgiram durante o processo de classificação e regressão em fluxo de dados, principalmente com relação a rapidez do algoritmo, sua eficiência de memória e adaptação a diferentes tipos de mudança de conceito.

Logo, esse trabalho tem como objetivo propor a adaptação e avaliação do algoritmo de *boosting* AXGB da classificação para regressão em fluxos de dados, com suporte para se adaptar rapidamente a mudança de conceito por meio da união de detectores ativos.

O XGB, bem como sua adaptação para fluxo de dados chamada AXGB, com ambas estratégias utilizando ensembles e classificação alternada, são ferramentas poderosas com trabalhos difundidos apenas com a aprendizagem de máquina para classificação. Além disso, nos trabalhos anteriores com fluxos de dados, a mudança de conceito não foi um tópico devidamente explorado e com a devida importância destacada: os padrões que o modelo aprendeu podem não ser mais válidos, tornando as previsões do modelo inacuradas.

Devido a isso, pretendendo explorar o potencial do AXGB, este trabalho visa unir a eficiência do AXGB, os desafios em decorrência da utilização dos fluxos de dados, a falta de trabalhos para regressão e os malefícios da detecção tardia da mudança de conceito com o objetivo de propor um método eficaz para o melhor tratamento dessas questões.

Por ser um algoritmo altamente dependente dos parâmetros utilizados, diferentes comportamentos foram observados, porém, com base em trabalhos recentes foram escolhidos valores que obtiveram o melhor desempenho. Durante os testes feitos foi possível detectar a eficiência do algoritmo proposto para diferentes tipos de *drift*: abrupto, incremental e gradual. A união de detectores apresentou melhor resultado e acurácia para detecção de *drifts* abruptos devido a sua característica para detecção rápida e consequente adaptação com os três mecanismos utilizados: ADWIN, KSWIN e DDM. Consequentemente, por se mostrar sensível a qualquer mudança de conceito e reagir com a reinicialização do tamanho da janela deslizante, instâncias consideradas ultrapassadas não foram consideradas nessa nova janela, permitindo a melhor readaptação do modelo para novos conceitos.

Na próxima etapa será feita a avaliação e adaptação que permitirá a regressão em fluxo de dados utilizando a mesma estratégia para classificação. Para isso, outros trabalhos de diferentes algoritmos sendo portados da classificação para regressão serão estudados e considerados para o desenvolvimento deste. As principais dificuldades serão com relação a adaptação do conjunto de técnicas ativas de detecção de mudança de conceito, proposto nesta etapa do projeto, para regressão. Além disso, outros detectores ativos de mudança de conceito deverão ser explorados, visando tornar a utilização dos mesmos complementar, de forma a tornar cada mecanismo mais eficiente em determinado tipo de *drift*, e eficiente, com configuração de maneira a não prejudicar consideravelmente o tempo de execução e utilização de memória do modelo. Também será feito

a comparação com alguns algoritmos já existentes que possuem objetivos semelhantes.

Ademais, foi visto que a utilização do modelo proposto para os outros tipos de *drifts* não melhorou a performance, a manteve neutra. Para a segunda parte, pretende-se avaliar algum mecanismo ou tratamento para adaptação após o *drift*, capaz de aumentar a acurácia e possibilidades para esses outros tipos de mudança de conceito.

5.1 CRONOGRAMA DAS PRÓXIMAS ETAPAS

O cronograma a ser seguido para a próxima parte do trabalho se dá pela Tabela 8:

1. Implementação com a adaptação do algoritmo proposto para regressão em fluxo de dados;
2. Execução de uma série de testes com o algoritmo desenvolvido;
3. Análise dos resultados obtidos pelo algoritmo proposto e comparação com algoritmos clássicos da área de mineração de fluxo de dados;
4. Escrita de artigo científico;
5. Continuação da escrita do TCC.

Tabela 8 – Cronograma das próximas etapas

Etapas	2022					
	FEV	MAR	ABR	MAI	JUN	JUL
1	X	X	X	X	X	
2		X	X	X	X	
3		X	X	X	X	
4		X	X	X	X	
5			X	X	X	

Fonte: o Autor.

REFERÊNCIAS

- ABBASZADEH, Omid; AMIRI, Ali; KHANTEYMOORI, Ali Reza. An ensemble method for data stream classification in the presence of concept drift. **Frontiers of Information Technology & Electronic Engineering**, Springer, v. 16, n. 12, p. 1059–1068, 2015. Citado na página 25.
- BARDDAL, Jean Paul. Vertical and horizontal partitioning in data stream regression ensembles. In: IEEE. **2019 International Joint Conference on Neural Networks (IJCNN)**. Curitiba, BR, 2019. p. 1–8. Citado 2 vezes nas páginas 22 e 25.
- BASSEVILLE, Michele; NIKIFOROV, Igor V et al. **Detection of abrupt changes: theory and application**. France: prentice Hall Englewood Cliffs, 1993. v. 104. Citado na página 22.
- BIFET, Albert. Adaptive learning and mining for data streams and frequent patterns. **ACM SIGKDD Explorations Newsletter**, ACM New York, NY, USA, v. 11, n. 1, p. 55–56, 2009. Citado na página 20.
- BIFET, Albert; GAVALDA, Ricard. Learning from time-changing data with adaptive windowing. In: SIAM. **Proceedings of the 2007 SIAM international conference on data mining**. Catalunha, Espanha, 2007. p. 443–448. Citado na página 24.
- BONASSA, Gustavo. **Adaptação de classificador utilizando a biblioteca XGBoost para classificação rápida de fluxos de dados parcialmente classificados com mudança de conceito**. 2021. Citado 9 vezes nas páginas 29, 31, 32, 34, 35, 38, 39, 40 e 41.
- BOSE, RP Jagadeesh Chandra et al. Dealing with concept drifts in process mining. **IEEE transactions on neural networks and learning systems**, IEEE, v. 25, n. 1, p. 154–171, 2013. Citado na página 21.
- BOULEGANE, Dihia et al. Streaming time series forecasting using multi-target regression with dynamic ensemble selection. In: IEEE. **2020 IEEE International Conference on Big Data (Big Data)**. Atlanta, GA, USA, 2020. p. 2170–2179. Citado 3 vezes nas páginas 29, 31 e 32.
- BREIMAN, Leo. Bagging predictors. **Machine learning**, Springer, v. 24, n. 2, p. 123–140, 1996. Citado na página 17.
- CHEN, Tianqi; GUESTRIN, Carlos. Xgboost: A scalable tree boosting system. In: ACM. **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. Washington, USA, 2016. p. 785–794. Citado 3 vezes nas páginas 18, 28 e 31.
- CHEN, Tianqi et al. Xgboost: extreme gradient boosting. **R package version 0.4-2**, v. 1, n. 4, p. 1–4, 2015. Citado na página 13.
- CHEN, Wenbin et al. Radar emitter classification for large data set based on weighted-xgboost. **IET**, 2016. Citado na página 32.
- CHEN, Yixin et al. Multi-dimensional regression analysis of time-series data streams. In: ELSEVIER. **VLDB'02: Proceedings of the 28th International Conference on Very Large Databases**. Wright State University, USA, 2002. p. 323–334. Citado na página 20.
- CHICCO, Davide; JURMAN, Giuseppe. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. 2020. Citado na página 27.

DARGAN, Shaveta et al. A survey of deep learning and its applications: a new paradigm to machine learning. **Archives of Computational Methods in Engineering**, Springer, p. 1–22, 2019. Citado na página 15.

DITZLER, Gregory et al. Learning in nonstationary environments: A survey. **Comp. Intell. Mag.**, IEEE Press, v. 10, n. 4, p. 12–25, nov. 2015. ISSN 1556-603X. Disponível em: <<https://doi.org/10.1109/MCI.2015.2471196>>. Citado na página 12.

DONG, Xibin et al. A survey on ensemble learning. **Frontiers of Computer Science**, Springer, v. 14, n. 2, p. 241–258, 2020. Citado 2 vezes nas páginas 17 e 19.

DUARTE, João; GAMA, João; BIFET, Albert. Adaptive model rules from high-speed data streams. **ACM Trans. Knowl. Discov. Data**, Association for Computing Machinery, New York, NY, USA, v. 10, n. 3, jan 2016. ISSN 1556-4681. Disponível em: <<https://doi.org/10.1145/2829955>>. Citado na página 20.

ELWELL, Ryan; POLIKAR, Robi. Incremental learning of concept drift in nonstationary environments. **IEEE Transactions on Neural Networks**, IEEE, v. 22, n. 10, p. 1517–1531, 2011. Citado na página 12.

FAFALIOS, Stefanos; CHARONYKTAKIS, Pavlos; TSAMARDINOS, Ioannis. Gradient boosting trees. 2020. Citado na página 19.

GALAR, Mikel et al. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, IEEE, v. 42, n. 4, p. 463–484, 2011. Citado na página 17.

GAMA, João et al. A survey on concept drift adaptation. **ACM computing surveys (CSUR)**, ACM, v. 46, n. 4, p. 44, 2014. Citado na página 25.

GAMA, João et al. A survey on concept drift adaptation. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 46, n. 4, mar. 2014. ISSN 0360-0300. Disponível em: <<https://doi.org/10.1145/2523813>>. Citado na página 12.

GAMAGE, Sunanda; PREMARATNE, Upeka. Detecting and adapting to concept drift in continually evolving stochastic processes. In: **Proceedings of the International Conference on Big Data and Internet of Thing**. New York, NY, USA: Association for Computing Machinery, 2017. (BDIOT2017), p. 109–114. ISBN 9781450354301. Disponível em: <<https://doi.org/10.1145/3175684.3175723>>. Citado na página 11.

GOLLAPUDI, Sunila. **Practical machine learning**. EUA: Packt Publishing Ltd, 2016. Citado na página 17.

GOMES, Heitor Murilo et al. Adaptive random forests for data stream regression. In: IEEE. **ESANN**. Curitiba, Paraná, Brasil, 2018. Citado 3 vezes nas páginas 30, 31 e 32.

GRULICH, Philipp M et al. Scalable detection of concept drifts on data streams with parallel adaptive windowing. In: EDBT. **EDBT**. Berlim, Alemanha, 2018. p. 477–480. Citado na página 23.

HAN, Jiawei; PEI, Jian; KAMBER, Micheline. **Data mining: concepts and techniques**. Illinois, EUA: Elsevier, 2011. 460-500 p. Citado 2 vezes nas páginas 16 e 20.

HO, Tin Kam. Complexity of classification problems and comparative advantages of combined classifiers. In: SPRINGER. **International workshop on multiple classifier systems**. Murray Hill, USA, 2000. p. 97–106. Citado na página 17.

HOENS, T Ryan; POLIKAR, Robi; CHAWLA, Nitesh V. Learning from streaming data with concept drift and imbalance: an overview. **Progress in Artificial Intelligence**, Springer, v. 1, n. 1, p. 89–101, 2012. Citado na página 20.

IKONOMOVSKA, Elena; GAMA, João; DŽEROSKI, Sašo. Online tree-based ensembles and option trees for regression on evolving data streams. **Neurocomputing**, Elsevier, v. 150, p. 458–470, 2015. Citado na página 20.

KADIYALA, Akhil; KUMAR, Ashok. Applications of python to evaluate the performance of decision tree-based boosting algorithms. **Environmental Progress & Sustainable Energy**, Wiley Online Library, v. 37, n. 2, p. 618–623, 2018. Citado na página 38.

KOEHRSEN, Will. Overfitting vs. underfitting: A complete example. **Towards Data Science**, 2018. Citado na página 15.

KRAWCZYK, Bartosz et al. Ensemble learning for data stream analysis: A survey. **Information Fusion**, v. 37, p. 132 – 156, 2017. ISSN 1566-2535. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1566253516302329>>. Citado 4 vezes nas páginas 11, 17, 22 e 27.

KRAWCZYK, Bartosz et al. Ensemble learning for data stream analysis: A survey. **Information Fusion**, v. 37, p. 132–156, 2017. ISSN 1566-2535. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1566253516302329>>. Citado na página 25.

KRAWCZYK, Bartosz; PFAHRINGER, Bernhard; WOŹNIAK, Michał. Combining active learning with concept drift detection for data stream mining. In: IEEE. **2018 IEEE International Conference on Big Data (Big Data)**. Hamilton, New Zealand, 2018. p. 2239–2244. Citado na página 17.

KROGH, Anders; VEDELSBY, Jesper et al. Neural network ensembles, cross validation, and active learning. **Advances in neural information processing systems**, Morgan Kaufmann Publishers, v. 7, p. 231–238, 1995. Citado na página 17.

KWON, Donghwoon et al. A survey of deep learning-based network anomaly detection. **Cluster Computing**, Springer, v. 22, n. 1, p. 949–961, 2019. Citado na página 16.

LANEY, Douglas. **3D Data Management: Controlling Data Volume, Velocity, and Variety**. EUA, 2001. Disponível em: <<http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>>. Citado na página 11.

LARSON, Deanne; CHANG, Victor. A review and future direction of agile, business intelligence, analytics and data science. **International Journal of Information Management**, Elsevier, v. 36, n. 5, p. 700–710, 2016. Citado na página 11.

LIAO, Zhenwei; WANG, Yongheng. Rival learner algorithm with drift adaptation for online data stream regression. In: **Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence**. New York, NY, USA: Association

for Computing Machinery, 2018. (ACAI 2018). ISBN 9781450366250. Disponível em: <<https://doi.org/10.1145/3302425.3302475>>. Citado 5 vezes nas páginas 12, 20, 29, 31 e 32.

LOPES, Raul HC; REID, ID; HOBSON, Peter R. The two-dimensional kolmogorov-smirnov test. *Proceedings of Science*, 2007. Citado na página 24.

LU, Jie et al. Learning under concept drift: A review. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 31, n. 12, p. 2346–2363, 2018. Citado 4 vezes nas páginas 11, 22, 23 e 25.

Lu, J. et al. Learning under concept drift: A review. **IEEE Transactions on Knowledge and Data Engineering**, v. 31, n. 12, p. 2346–2363, 2018. Citado na página 21.

MAHDI, Osama A et al. 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. Citado na página 22.

MANICKASWAMY, Thangam; BHUVANESWARI, A. Concept drift in data stream classification using ensemble methods: Types, methods and challenges. **INFOCOMP Journal of Computer Science**, v. 19, n. 2, p. 163–174, 2020. Citado na página 22.

MARSLAND, Stephen. **Machine learning: an algorithmic perspective**. Berlin, Germany: Chapman and Hall/CRC, 2011. Citado na página 11.

MAULUD, Dastan; ABDULAZEEZ, Adnan M. A review on linear regression comprehensive in machine learning. **Journal of Applied Science and Technology Trends**, v. 1, n. 4, p. 140–147, 2020. Citado na página 17.

MAYR, Andreas et al. The evolution of boosting algorithms. **Methods of information in medicine**, Schattauer GmbH, v. 53, n. 06, p. 419–427, 2014. Citado na página 12.

MEHMOOD, Hassan et al. Concept drift adaptation techniques in distributed environment for real-world data streams. **Smart Cities**, Multidisciplinary Digital Publishing Institute, v. 4, n. 1, p. 349–371, 2021. Citado na página 12.

MONTIEL, Jacob et al. Adaptive xgboost for evolving data streams. In: IEEE. **2020 International Joint Conference on Neural Networks (IJCNN)**. Hamilton, New Zealand, 2020. p. 1–8. Citado 9 vezes nas páginas 13, 28, 29, 31, 32, 35, 39, 40 e 41.

MONTIEL, Jacob et al. Scikit-multiflow: A multi-output streaming framework. **Journal of Machine Learning Research**, v. 19, n. 72, p. 1–5, 2018. Disponível em: <<http://jmlr.org/papers/v19/18-251.html>>. Citado na página 39.

NIKAM, Sagar S. A comparative study of classification techniques in data mining algorithms. **Oriental journal of computer science & technology**, v. 8, n. 1, p. 13–19, 2015. Citado na página 16.

NUNES, André Luís. Um estudo investigativo de algoritmos de regressão para data streams. Universidade do Vale do Rio dos Sinos, 2017. Citado na página 21.

PINAGÉ, Felipe Azevedo et al. Handling concept drift based on data similarity and dynamic classifier selection. Universidade Federal do Amazonas, 2017. Citado na página 20.

- PROVOST, Foster; KOHAVI, R. Glossary of terms. **Journal of Machine Learning**, v. 30, n. 2-3, p. 271–274, 1998. Citado na página 15.
- RAMRAJ, S et al. Experimenting xgboost algorithm for prediction and classification of different datasets. **International Journal of Control Theory and Applications**, v. 9, p. 651–662, 2016. Citado na página 13.
- REIS, Denis Moreira dos. **Classificação de fluxos de dados com mudança de conceito e latência de verificação**. Tese (Doutorado) — Universidade de São Paulo, 2017. Citado na página 19.
- SCHAPIRE, Robert E. The strength of weak learnability. **Machine learning**, Springer, v. 5, n. 2, p. 197–227, 1990. Citado na página 17.
- SCHAPIRE, Robert E. The boosting approach to machine learning: An overview. **Nonlinear estimation and classification**, Springer, p. 149–171, 2003. Citado na página 12.
- TOGBE, Maurras Ulbricht et al. Anomalies detection using isolation in concept-drifting data streams. **Computers**, Multidisciplinary Digital Publishing Institute, v. 10, n. 1, p. 13, 2021. Citado na página 24.
- VERBRAEKEN, Joost et al. A survey on distributed machine learning. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 53, n. 2, p. 1–33, 2020. Citado na página 15.
- WIDMER, Gerhard. Combining robustness and flexibility in learning drifting concepts. In: CITESEER. **ECAI**. Vienna, Austria, 1994. p. 468–472. Citado na página 20.
- YAN, Myuu Myuu Wai. Accurate detecting concept drift in evolving data streams. **ICT Express**, Elsevier, v. 6, n. 4, p. 332–338, 2020. Citado na página 25.
- YANG, Li; MANIAS, Dimitrios Michael; SHAMI, Abdallah. Pwpae: An ensemble framework for concept drift adaptation in iot data streams. **arXiv preprint arXiv:2109.05013**, 2021. Citado 5 vezes nas páginas 23, 24, 29, 31 e 32.
- YU, Hang; LU, Jie; ZHANG, Guangquan. Morstreaming: A multioutput regression system for streaming data. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, p. 1–13, 2021. Citado 2 vezes nas páginas 11 e 12.
- ZENOBI, Gabriele; CUNNINGHAM, Padraig. Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error. In: SPRINGER. **European Conference on Machine Learning**. Dublin, Ireland, 2001. p. 576–587. Citado na página 17.
- ŽLIOBAITÉ, Indrė. Learning under concept drift: an overview. **arXiv preprint arXiv:1010.4784**, 2010. Citado na página 21.