Fatec Cruzeiro Prof. Waldomiro May
Revista Científica H-TEC Humanidades & Tecnologia

ANÁLISE DE DADOS PARA APRENDIZADO DE MÁQUINA: UM ESTUDO SOBRE ALGORITMOS DE DETECÇÃO DE MUDANÇA DE CONCEITO

Autores
Juan Canuto Hassam[1]
José Walmir Gonçalves Duque[2]
Anderson Anjos da Silva[3]
Renato Duarte Costa[4]

Resumo
Este trabalho tem por objetivo construir uma ferramenta geradora de dados (na forma de cenários) de Mudança de Conceito em ambiente controlado para a realização de análises de desempenho por meio de de algoritmos de Detecção de Mudança de Conceito. Uma das maiores dificuldades, quando se trata de Mudança de Conceito, é saber diferenciar ruído de mudança e qual o melhor algoritmo de Detecção de Mudança de Conceito para determinado cenário de Mudança e, para esse intento, que a ferramenta proposta se aplica. Foram criados quatro cenários de Mudança de Conceito: Gradual, Incremental, Abrupto e Recorrente. Em seguida, para cada um dos cenários, foram aplicados três algoritmos de Detecção de Mudança de Conceito: ADWIN, DDM e Page-Hinkley. Após aplicação, foi feita uma análise do desempenho de cada um dos algoritmos para cada cenário criado pelo gerador de cenários para definir o melhor algoritmo para cada cenário estudado. Ao final, foi percebido que a ferramenta geradora de cenários de Mudança de Conceito desenvolvida fornece ao usuário um ambiente controlado para execução e testes de novos algoritmos de Detecção de Mudança de Conceito. Nos cenários criados pela ferramenta, o algoritmo Page-Hinkley obteve melhores resultados nas Mudanças de Conceito Gradual e Incremental, o DMM se mostrou o melhor na Mudança de Conceito Abrupta e o ADWIN na Mudança de Conceito Recorrente.
Palavras-chave: Aprendizado de Máquina. Mudança de Conceito. ADWIN. Page-Hinkley. DDM

Abstract
This work aims to build a tool to generate data (in the form of scenarios) of Concept Drift in a controlled environment for performance analysis using Concept Drift Detection algorithms. One of the biggest difficulties, when it comes to Concept Dritf, is knowing how to differentiate noise from change and which is the best Concept Drift Detection algorithm for a given Change scenario and, for this purpose, which tool is applied. Four Concept Drift scenarios were created, namely: Gradual, Incremental, Abrupt and Recurrent. Then, for each of the scenarios, three Concept Drift Detection algorithms were applied: ADWIN, DDM and Page-Hinkley. After application, an analysis of the

[1] Tecnólogo em Análise e Desenvolvimento de Sistemas pela Faculdade de Tecnologia São José dos Campos – Prof. Jessen Vidal e Cientista de Dados pela COMPSIS - Computadores e Sistemas Ind. Com. Ltda. E-mail: juanhassam2014@gmail.com
[2] Mestre em Engenharia Eletrônica e Computação pelo Instituto Tecnológico de Aeronáutica – ITA, docente pela Faculdade de Tecnologia São José dos Campos – Prof. Jessen Vidal e pelo Centro Universitário Salesiano São Paulo – UNISAL – Lorena. E-mail: walmirduque@gmail.com
[3] Doutor em Engenharia Eletrônica e Computação pelo Instituto Tecnológico de Aeronáutica – ITA e atuando como Gerente de Projetos na COMPSIS - Computadores e Sistemas Ind. Com. Ltda. E-mail: ansoanjos@gmail.com
[4] Mestre em Engenharia Eletrônica e Computação pelo Instituto Tecnológico de Aeronáutica – ITA e atuando como Diretor de Novos Negócios na COMPSIS - Computadores e Sistemas Ind. Com. Ltda. E-mail: renato.costa@compsis.com.br

performance of each of the algorithms was made for each scenario created by the scenario generator to define the best algorithm for each scenario studied. In the end, it was noticed that the developed tool for generating scenarios of Concept Drift provides the user with a controlled environment for the execution and testing of new algorithms for Detection of Concept Drift. In the scenarios created by the tool, the Page-Hinkley algorithm obtained better results in the Gradual and Incremental Concept Drift , the DMM was the best in the Abrupt Concept Drift and ADWIN in the Recurrent Concept Drift.
***Keywords**: Machine Learning; Concept Drift; ADWIN; Page-Hinkley; DDM*

INTRODUÇÃO

Uma ferramenta geradora de cenários de Mudança de Conceito é capaz de reproduzir um ambiente para desenvolvimento e teste de algoritmos de detecção de Mudança de Conceito em ambientes que replicam condições de domínio de aplicações reais.

Em um ambiente controlado é possível proporcionar controle sobre as informações geradas para o desenvolvimento de metodologias para tratar alguns problemas, como por exemplo a criação de mecanismos para detecção e adaptação demodelos ao longo do tempo.

A ferramenta é destinada a usuários técnicos que desejam criar ambientes controlados para testar algoritmos de Detecção de Mudança de Conceito para problemas declassificação em ambientes dinâmicos**.**

O Aprendizado de Máquina (AM) é uma subárea da Inteligência Artificial (IA) destinada à pesquisa e aplicação de técnicas computacionais com o propósito de prover a um sistema computacional a habilidade de aprender e evoluir com exemplos sem a necessidade da programação de regras específicas obtidas do conhecimento de um ou mais especialistas. O AM consegue encontrar padrões em conjuntos de dados e extrair conhecimento a partir deles. Dentro do AM existem diversas técnicas de aprendizado, porém as três principais são: o Aprendizado Supervisionado, o Aprendizado Não-Supervisionado e o Aprendizado por Reforço.

Um dos desafios relacionados ao Aprendizado de Máquina é que os dados tendem a mudar bastante com o tempo (GAMA et al., 2014), sendo “dados” os fatos que geralmente são coletados e armazenados de alguma forma (AMARAL, 2016), em formato eletrônico, analógico, ou mesmo em formato não eletrônico.

Conforme (GAMA et al., 2014), as previsões dos modelos treinados com dados desatualizados podem se tornar obsoletas devido às mudanças inesperadas na distribuição dos dados. Esse problema é conhecido como Mudança de Conceito (KHAMASSI; SAYEDMOUCHAWEH, 2014). A fim de resolver esse problema, os modelos de

aprendizagem de máquina necessitam de mecanismos que propiciem a adaptação do modelo ao longo do tempo.

Para exemplificar como acontece uma Mudança de Conceito, supondo um cenário em que uma câmera é utilizada para identificar a face de uma pessoa e liberar ou não seu acesso a determinado lugar: a Mudança de Conceito nesse caso ocorre de maneira incremental, pois a pessoa aos poucos envelhece, mudando os padrões em sua face que foram utilizados para a criação do modelo de classificação.

Outro exemplo de Mudança de Conceito é o de modelos de previsão do tempo, em que os padrões podem mudar bruscamente.

Uma ferramenta geradora de cenários de Mudança de Conceito é capaz de reproduzir um ambiente para desenvolvimento e teste de algoritmos de detecção de Mudança de Conceito em ambientes que replicam condições de domínio de aplicações reais.

Em um ambiente controlado é possível proporcionar controle sobre as informações geradas para o desenvolvimento de metodologias para tratar alguns dos problemas citados na seção anterior, como por exemplo a criação de mecanismos para detecção e adaptação de modelos ao longo do tempo.

A ferramenta é destinada a usuários técnicos que desejam criar ambientes controlados para testar algoritmos de Detecção de Mudança de Conceito para problemas de classificação em ambientes dinâmicos.

Este trabalho tem por objetivo construir uma ferramenta geradora de dados (na forma de cenários) de Mudança de Conceito em ambiente controlado para a realização de análises de desempenho por meio de de algoritmos de Detecção de Mudança de Conceito.

1 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentados os principais temas que servem como base para o desenvolvimento do trabalho proposto: Inteligência Artificial (IA), Aprendizado de Máquina (AM), Aprendizado Supervisionado, Aprendizado Não-supervisionado, Aprendizado por Reforço, Mudança de Conceito, Tipos de Mudança de Conceito, Algoritmos de Detecção de Mudança e as ferramentas utilizadas neste trabalho.

1.1 Inteligência Artificial

Inteligência Artificial é uma área de estudo dentro da ciência da computação que pesquisa e desenvolve métodos, técnicas e aplicações para simular, estender e expandir a teoria da inteligência humana. IA é um ramo da ciência da computação que tenta entender a essência da inteligência e reproduzir, em uma máquina, reações similares ao comportamento de decisão humana. A pesquisa no campo da inteligência artificial inclui robótica, reconhecimento de fala, reconhecimento de imagem, processamento de linguagem natural e sistemas especialistas (NING; YAN, 2010).

1.2 Aprendizado de Máquina

Conforme Copeland (2016), o AM na forma mais básica é uma área de IA cujo objetivo é a prática de usar algoritmos para analisar dados, aprender com eles e depois fazer uma determinação ou previsão sobre algo no mundo. Assim, em vez de rotinas de software de codificação manual com um conjunto específico de instruções para realizar uma tarefa específica, a máquina é "treinada" usando grandes quantidades de dados e algoritmos que lhe dão a capacidade de aprender a executar a tarefa.

De acordo com Libralão et al., 2003), estudos do ramo do AM pesquisam o desenvolvimento de métodos capazes de extrair informação a partir de determinadas amostras de dados. Existem diversos algoritmos de AM cujo objetivo é, após um determinado treinamento com certo conjunto de dados cujas instâncias têm classificação conhecida, uma máquina seja capaz de interpretar novos dados e classificá-los de maneira correta para que seja feita uma generalização levando em consideração os dados que foram apresentados antes.

1.3 Mudança de Conceito

Nesta sub-seção são apresentados os principais temas que servem como base para o desenvolvimento do trabalho proposto: Inteligência Artificial (IA), Aprendizado de Máquina (AM), Aprendizado Supervisionado, Aprendizado Não-supervisionado, Aprendizado por Reforço, Mudança de Conceito, Tipos de Mudança de Conceito, Algoritmos de Detecção de Mudança e as ferramentas utilizadas neste trabalho.

1.3.1 Definição de Mudança de Conceito

O termo Mudança de Conceito pode ser definido informalmente como uma mudança na definição dos conceitos (dados) ao longo do tempo e, consequentemente, mudança na sua distribuição. Mudança de Conceito refere-se a um cenário de aprendizagem onde a relação entre os dados de entrada e a variável alvo muda ao longo do tempo (Gama et al., 2014). Um ambiente a partir do qual este tipo de dados é obtido é considerado um ambiente não estacionário.

Um exemplo prático de problema de classificação com Mudança de Conceito, mencionado em Kuncheva (2004), é a detecção e filtro de e-mails spam. As descrições das duas classes, "spam" e "não spam", podem variar ao longo do tempo: elas são específicas por usuário e as preferências dos usuários também variam ao longo do tempo. Além disso, as variáveis usadas no tempo t para classificar spam podem ser irrelevantes em t+k. Para dificultar ainda mais, a variabilidade do ambiente neste cenário é fortemente relacionada ao acaso.

Desta forma, o classificador deve lidar com os "spammers", que irão continuamente criar novas formas de tentar enganar o classificador para categorizar um spam como e-mail legítimo. Segundo Gama et al. (2014), existem duas maneiras de se lidar com Mudança de Conceito: uma é utilizando-se o resultado do modelo preditivo para se atualizar o modelo definindo uma métrica de performance como, por exemplo, acurácia, e a outra é analisando estatisticamente a distribuição dos dados de uma mesma classe sobre o tempo.

Este trabalho utiliza-se da segunda abordagem para detectar Mudança de Conceito nos cenários gerados pela ferramenta geradora de cenários de Mudança de Conceito. Ruídos também conhecidos como outlier e, segundo Hawkins(1980), são definidos por dados que se desviam muito dos outros dados de um conjunto (ou subconjunto de dados), despertando a suspeita de que não pertencem ao conjunto.

Vale a pena ressaltar que os ruídos não caracterizam Mudança de Conceito, sendo um dos desafios diferenciar ruídos de Mudança de Conceito. (GAMA et al., 2014)

1.3.2 Tipos de Mudança de Conceito

Trabalhos da literatura como Tsymbal, 2004; Gama et al., 2014; Zliobaite, 2009 geralmente classificam as Mudanças de Conceito que podem ocorrer no mundo real em quatro tipos, de acordo com os padrões de configuração das fontes de dados ao longo do tempo.

Supondo a existência de duas fontes de dados, os quatro tipos de Mudança de Conceito são descritos a seguir:

• **Abrupta**: ocorre a troca brusca de um conceito A por outro B. No tempo t a fonte S1 é repentinamente substituída por S2.

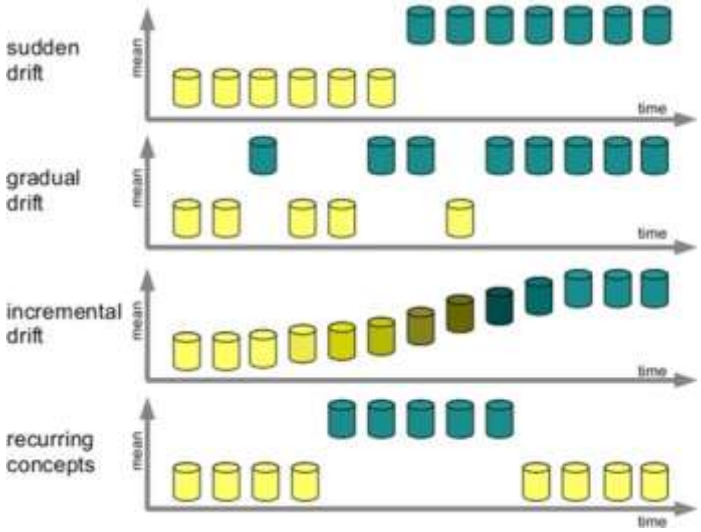
• **Gradual**: um conceito A vai sendo trocado pelo outro B aos poucos. Enquanto não ocorre a mudança definitiva do conceito A para o conceito B, observa-se cada vez mais ocorrências de B e menos ocorrências de A. Ambas as fontes S1 e S2 estão ativas, mas, à medida que o tempo passa, a probabilidade de amostragem da fonte S1 diminui enquanto a probabilidade de amostra da fonte S2 aumenta. No início desta mudança, antes que mais instâncias sejam observadas, uma instância da fonte S2 pode ser facilmente confundida com ruído aleatório.

• **Incremental**: presença de vários conceitos intermediários A1, A2, A3, ..., etc., entre dois pontos observados, indo gradativamente de um conceito A para outro B. Muitos autores consideram este tipo de mudança como um subtipo de Mudança Gradual. Este tipo de mudança inclui mais fontes intermediárias entre S1 e S2, mas como a diferença entre elas é muito pequena, a mudança é detectada apenas quando é observado por um longo período de tempo.

• **Recorrente**: Alguns trabalhos como o de Zliobate (2009) consideram um tipo particular de Mudança de Conceito, o chamado contexto recorrente, que ocorre quando um conceito ativo anteriormente reaparece depois de algum tempo. Este conceito difere-se da noção de sazonalidade comum porque não é periódico e nem há clareza de quando esta fonte pode reaparecer. Neste caso, o conceito A é trocado pelo B (a fonte S1 é substituída por S2) e, depois de algum tempo, volta para o conceito A (a fonte volta a ser S1).

É importante ressaltar que o ruído não é considerado um tipo de mudança, pois se refere a uma anomalia ou ocorrência isolada de um desvio aleatório. Neste caso, não há nenhuma necessidade de adaptação do modelo, que deve ser robusto ao ruído. A Figura 1 ilustra os tipos de Mudança de Conceito apresentados anteriormente.

Figura 1 - Representação de Mudança de Conceito Abrupta, Gradual, Incremental e Recorrente



Fonte: Adaptada de Zliobate (2009).

1.4 Detecção de Mudança

Muitos esforços foram realizados para minimizar as incertezas com relação à confiabilidade dos resultados apresentados pelos métodos utilizados em Data Mining e Machine Learning. Estas incertezas se referem ao fato de que os treinamentos são realizados sobre um volume de dados que pode ser menos relevante do que os dados reais.

Widmer e Kubat (1995) apresentam como exemplo a previsão do tempo, cujas regras podem variar de acordo com a estação. Já Tsymbal (2004) relata que "a maioria dos algoritmos tradicionais de Machine Learning está suscetível a Mudança de Conceito, pois a precisão da previsão reduz com o passar do tempo". Algoritmos com estas características são chamados de algoritmos em lote bons no aprendizado a partir de bases de dados armazenadas em lote, mas ineficientes quando estes dados crescem dinamicamente, ficando expostos à Mudança de Conceito.

Apresenta-se abaixo três algoritmos, selecionados do framework Scikit-Multiflow, que permitem a detecção da mudança e que foram experimentados neste trabalho avaliando-se as performances nos diferentes cenários de Mudança de Conceito simulados.

1.4.1 ADWIN

Para detectar uma Mudança de Conceito, Adaptive Windowing (ADWIN) (BIFET; GAVALDA, 2007) utilizam uma janela (W) que tem seu tamanho ajustado automaticamente.

Quanto mais tempo os dados permanecerem com a mesma distribuição de probabilidade, maior será o tamanho de W. Por outro lado, quando mudanças ocorrerem, W será reduzida. Com essa abordagem, o objetivo do ADWIN é manter em W somente instâncias com uma mesma distribuição.

1.4.2 DDM

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

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

1.4.3 Page-Hinkley

Page-Hinkley é um teste estatístico que monitora a diferença entre duas variáveis x' e y' onde x' são definidas com a diferença acumulada entre o valor observado e sua média até o momento (GAMA et al. 2004). A alteração da distribuição é informada quando a diferença das duas variáveis é superior a um dado limiar λ: $x' - y' > \lambda$. O valor do parâmetro λ determina a taxa de alarmes falsos e é informado pelo usuário. Um valor alto implica em menos alarmes falsos, porém algumas mudanças podem ser descartadas. Desta forma, em caso de mudanças na capacidade do indivíduo (considerando a média das capacidades anteriores e um limiar de aceitação), este ocasiona em um alarme informando que ocorreu Mudança de Conceito.

2 METODOLOGIA

Para atender os objetivos deste trabalho, a metodologia de pesquisa utilizada foi estruturada no seguinte formato:

A Seção 1 apresenta a fundamentação teórica do trabalho com a conceituação das áreas abordadas e as ferramentas utilizadas para o desenvolvimento do trabalho. Na Seção 2 é detalhado o desenvolvimento do trabalho proposto. Por fim, na Seção 3 são apresentados os

resultados alcançados e as discussões dos resultados com base nos objetivos propostos. As ferramentas utilizadas para o desenvolvimento foram:

- Python: é uma linguagem de programação de alto nível, de script, imperativa, interpretada, orientada a objeto, funcional, de tipagem dinâmica e forte, criada por Guido van Rossum, em 1991 (PYTHON, 2019). É uma linguagem que possui um modelo de desenvolvimento feito pela comunidade, tem seu código aberto e é gerenciada atualmente pela Python Software Foundation, uma instituição sem fins lucrativos.
- Scikit-Multiflow: é um framework de Aprendizado de Máquina para fluxo de dados de código aberto, inspirado em outros frameworks com funcionalidade parecidas (MONTIEL et al., 2018).

O objetivo desta seção é apresentar o desenvolvimento da ferramenta geradora de cenários de Mudança de Conceito passando pela geração dos dados que simularão o ambiente controlado e aplicação dos algoritmos de Detecção de Mudança de Conceito. Foi criado um algoritmo para criar os cenários de Mudança de Conceito Gradual, Incremental, Abrupta e Recorrente. A geração de cada um dos cenários também será explicada nesta seção. A Figura 2 apresenta a interface da ferramenta desenvolvida.

Figura 2 - Interface de Usuário

```
tipo_concept_drift = int(input("Escolha o tipo de concept drift que deseja simular:\n 1)Abrupto \n 2)Gradual \n"+
                               "3)Incremental \n 4)Recorrente \n"))
tipo_deteccao = int(input("Escolha o tipo de algoritmo de detecção: \n 1)Adwin \n 2)DDM \n 3)PageHinkley \n"))
```

```
Escolha o tipo de concept drift que deseja simular:
 1)Abrupto
 2)Gradual
3)Incremental
 4)Recorrente
1
Escolha o tipo de algoritmo de detecção:
 1)Adwin
 2)DDM
 3)PageHinkley
1
```

Fonte: Autores (2019).

O usuário da ferramenta escolhe o tipo de cenário desejado digitando 1 para a opção de cenário Abrupto, 2 para cenário Gradual, 3 para cenário Incremental e 4 para cenário Recorrente, após a escolha do cenário é inserida pelo usuário a escolha do tipo de algoritmo de Detecção de Mudança de Conceito sendo 1 para ADWIN, 2 para DDM e 3 Para Page-Hinkley.

Os dados utilizados para simular os ambientes de Mudança de Conceito são dados pseudoaleatórios do tipo inteiro gerados por algoritmos desenvolvidos na linguagem de programação Python, que criam um vetor com n instâncias, cuja quantidade de dados no vetor é definida pela variável len_data. Com a finalidade de se manipular de maneira clara como ocorre a Mudança de Conceito, é gerado apenas um atributo por instância, todos os atributos gerados são quantitativos como será explicado nas subseções a seguir.

2.1 Geração do cenário de Mudança Gradual

Nesta etapa foi criado um algoritmo que simula as condições de um ambiente real no qual ocorre Mudança Gradual: foi simulada uma entrada contendo 11.250 instâncias que representam o conjunto simulado. Foi criado um vetor com tamanho igual ao valor da variável len_data definido no algoritmo.

Após a criação do vetor de dados, foi feita a manipulação dos valores para simular a ocorrência da Mudança Gradual no cenário em questão. Na Figura 3 é apresentado o algoritmo desenvolvido na linguagem Python:

Figura 3 - Simulação do cenário de Mudança Gradual

```
len_data = 11250
data_stream = np.resize([1,0], len_data)
for i in range(1500, 1750):
    data_stream[i] = np.random.randint(2, high=4)
for i in range(1750, 3000):
    data_stream[i] = np.random.randint(0, high=2)
for i in range(3000, 3500):
    data_stream[i] = np.random.randint(2, high=4)
for i in range(3500, 4500):
    data_stream[i] = np.random.randint(0, high=2)
for i in range(4500, 5250):
    data_stream[i] = np.random.randint(2, high=4)
for i in range(5250, 6000):
    data_stream[i] = np.random.randint(0, high=2)
for i in range(6000, 7000):
    data_stream[i] = np.random.randint(2, high=4)
for i in range(7000, 7500):
    data_stream[i] = np.random.randint(0, high=2)
for i in range(7500, 9000):
    data_stream[i] = np.random.randint(2, high=4)
for i in range(9000, 9250):
    data_stream[i] = np.random.randint(0, high=2)
for i in range(9250, 11250):
    data_stream[i] = np.random.randint(2, high=4)
```

Fonte: Autoeres (2019).

Os valores são manipulados nas instâncias desejadas a fim de simular o ambiente desejado: são inseridos os dados nos índices desejados para simular o cenário criado.

2.2 Geração do cenário de Mudança Incremental

Nesta etapa foi criado um algoritmo que simula as condições de um ambiente real no qual ocorre Mudança Incremental; foi simulada uma entrada contendo 10.000 instâncias que representam o conjunto simulado. Foi criado um vetor com tamanho igual ao valor da variável len_data definido no algoritmo.

Após a criação do vetor de dados, foi feita a manipulação dos valores para simular a ocorrência do Mudança Incremental no cenário em questão. Na Figura 4, é apresentado o algoritmo desenvolvido na linguagem Python:

Figura 4 - Simulação do cenário de Mudança Incremental

```
len_data = 10000
data_stream = np.resize([1,0], len_data)
for i in range(1000, 2000):
    data_stream[i] = np.random.randint(0, high=3)
for i in range(2000, 3000):
    data_stream[i] = np.random.randint(0, high=4)
for i in range(3000, 4000):
    data_stream[i] = np.random.randint(0, high=5)
for i in range(4000, 5000):
    data_stream[i] = np.random.randint(1, high=6)
for i in range(5000, 6000):
    data_stream[i] = np.random.randint(1, high=7)
for i in range(6000, 7000):
    data_stream[i] = np.random.randint(1, high=8)
for i in range(7000, 8000):
    data_stream[i] = np.random.randint(2, high=9)
for i in range(8000, 9000):
    data_stream[i] = np.random.randint(2, high=10)
for i in range(9000, 10000):
    data_stream[i] = np.random.randint(3, high=11)
```

Fonte: Autores (2019).

Os valores são manipulados nas instâncias desejadas a fim de simular o ambiente desejado: são inseridos os dados nos índices desejados para simular o cenário criado.

2.3 Geração do cenário de Mudança Abrupta

Nesta etapa foi criado um algoritmo que simula as condições de um ambiente real no qual ocorre Mudança Abrupta, assim foi simulada uma entrada contendo 10.000 instâncias que representam o conjunto simulado.

Foi criado um vetor com tamanho igual ao valor da variável len_data definido no algoritmo. Após a criação do vetor de dados, foi feita a manipulação dos valores para simular a ocorrência do Mudança Abrupta no cenário em questão; na Figura 5 é apresentado o algoritmo desenvolvido na linguagem Python:

Figura 5 - Simulação do cenário de Mudança Abrupta

```
len_data = 10000
data_stream = np.resize([1,0], len_data)
for i in range(5000, 10000):
    data_stream[i] = np.random.randint(2, high=4)
```

Fonte: O autor (2019).

Os valores são manipulados nas instâncias desejadas a fim de simular o ambiente desejado: são inseridos os dados nos índices desejados para simular o cenário criado.

2.4 Geração do cenário de Mudança Recorrente

Nesta etapa foi criado um algoritmo que simula as condições de um ambiente real no qual ocorre Mudança Recorrente: foi simulada uma entrada contendo 10.000 instâncias que representam o conjunto simulado.

Foi criado um vetor com tamanho igual ao valor da variável len_data definido no algoritmo. Após a criação do vetor de dados, foi feita a manipulação dos valores para simular aocorrência do Mudança Recorrente no cenário em questão; na figura 6 é apresentado o algoritmo desenvolvido na linguagem Python:

Figura 6 - Simulação do cenário de Mudança Recorrente

```
len_data = 10000
data_stream = np.resize([1,0], len_data)
for i in range(1000, 1250):
    data_stream[i] = np.random.randint(2, high=3)
for i in range(2000, 2250):
    data_stream[i] = np.random.randint(2, high=3)
for i in range(3000, 3250):
    data_stream[i] = np.random.randint(2, high=3)
for i in range(4000, 4250):
    data_stream[i] = np.random.randint(2, high=3)
for i in range(5000, 5250):
    data_stream[i] = np.random.randint(2, high=3)
for i in range(6000, 6250):
    data_stream[i] = np.random.randint(2, high=3)
for i in range(7000, 7250):
    data_stream[i] = np.random.randint(2, high=3)
for i in range(8000, 8250):
    data_stream[i] = np.random.randint(2, high=3)
for i in range(9000, 9250):
    data_stream[i] = np.random.randint(2, high=3)
```

Fonte: Autores (2019).

Os valores são manipulados nas instâncias desejadas a fim de simular o ambiente desejado: são inseridos os dados nos índices desejados para simular o cenário criado.

2.5 Algoritmos de Detecção de Mudança de Conceito

Nesta etapa foram implementados os algoritmos de Detecção de Mudança de Conceito ADWIN, DDM e Page-Hinkley para analisar sua performance.

Nos diferentes cenários de Mudança de Conceito simulados neste trabalho, os três métodos de Detecção de Mudança de Conceito estão disponíveis na biblioteca Scikit-Multiflow e a escolha desses três algoritmos foi feita pelo autor por suas constantesutilizações nos trabalhos consultados para o desenvolvimento deste projeto.

2.5.1 Implementação do algoritmo de detecção ADWIN

Para a implementação da Detecção de mudança com ADWIN foi utilizado o framework Scikit-Multiflow. Após instanciar a classe do ADWIN, foi feita uma iteração sobre os dados e utilizado o método detected_change para detectar a Mudança de Conceito em cima do conjunto de dados, como é apresentado na Figura 7:

Figura 7 - Detecção de Mudança de Conceito com ADWIN

```
adwin = ADWIN()
    for i in range(10000):
        adwin.add_element(data_stream[i])
        if adwin.detected_change():
            print('Concept Drift detectado nos dados: ' + str(data_stream[i]) + ' - indíce: ' + str(i))
```

Fonte: Autores (2019).

2.5.2 Implementação do algoritmo de detecção DDM

Para a implementação da Detecção de Mudança com DDM, foi utilizado o framework Scikit-Multiflow. Após a instanciar a classe do DDM, foi feita uma iteração sobre os dados e a utilização dos métodos detected_warning_zone que indica que uma Mudança de Conceito está perto de ocorrer e detected_change para detectar o Mudança de Conceito em cima do conjunto de dados, como é apresentado na Figura 8:

Figura 8 - Detecção de Mudança de Conceito com DDM

```
ddm = DDM()
    for i in range(10000):
        ddm.add_element(data_stream[i])
        if ddm.detected_warning_zone():
            print('Zona de aviso : ' + str(data_stream[i]) + ' - no indíce: ' + str(i))
            if ddm.detected_change():
                print('Mudança detectada no dado: ' + str(data_stream[i]) + ' - no indíce: ' + str(i))
```

Fonte: O autor (2019).

2.5.3 Implementação do algoritmo de detecção Page-Hinkley

Para a implementação da Detecção de Mudança com Page-Hinkley, foi utilizado o framework Scikit-Multiflow. Após instanciar a classe do Page-Hinkley, foi feita uma iteração sobre os dados e a utilização dos métodos detected_warning_zone que indica que uma Mudança de Conceito está perto de ocorrer e detected_change para detectar o Mudançade Conceito em cima do conjunto de dados, como é apresentado na Figura 9:

Figura 9 - Detecção de Mudança de Conceito com Page-Hinkley

```
ph = PageHinkley()
# Adiciona os elementos e verifica se ocorreu drift
for i in range(10000):
    ph.add_element(data_stream[i])
    if ph.detected_change():
        print('Mudança detectada no dado: ' + str(data_stream[i]) + ' - no indíce: ' + str(i))
```

Fonte: O autor (2019)

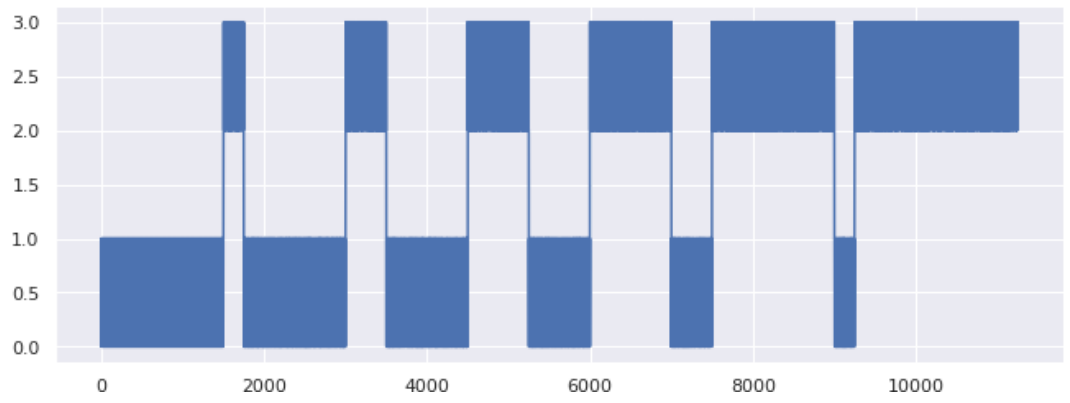
3 RESULTADOS E DISCUSSÃO

3.1 Análise dos cenários gerados de Mudança de Conceito

Foram simulados 4 cenários que representam os tipos de Mudança de Conceito descritos no escopo do trabalho: Mudança Gradual, Mudança Incremental, Mudança Abrupta e Mudança Recorrente. A Figura 10 mostra graficamente o cenário gerado pela ferramenta para a Mudança Gradual baseada na implementação proposta na seção 3:

Figura 10 - Gráfico da simulação de Mudança Gradual

```
sns.set(rc={'figure.figsize':(11, 4)})
plt.plot(data_stream)
plt.show()
```



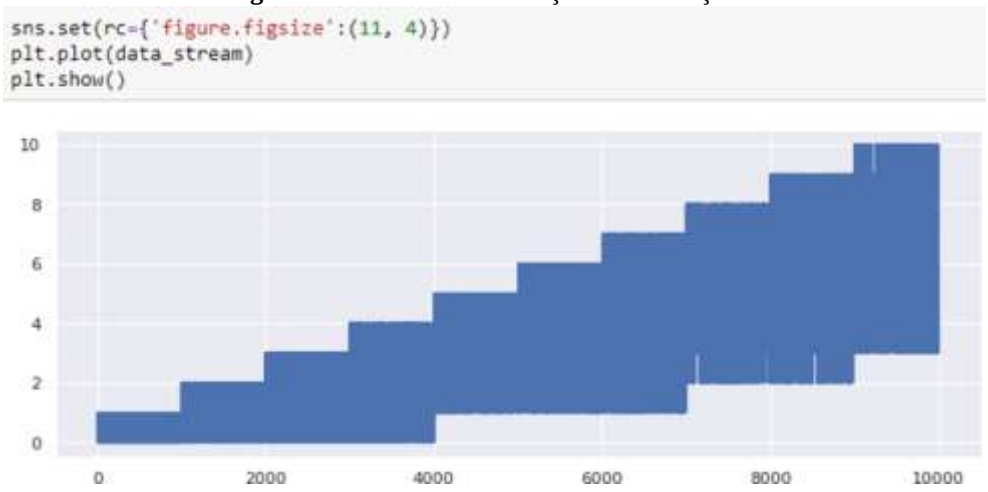
Fonte: Autores (2019).

O eixo X representa o número de instâncias e o eixo Y seus respectivos valores.

Observando a Figura 13, percebe-se que ocorre Mudança de Conceito a entre osíndices 1.500 e 1.750, 3.000 e 3.500, 4.500 e 5.250, 6.000 e 7.000, 7.500 e 9.000, 9.250 e 11.250 assumindo-se um novo padrão nos dados a partir do índice 6000.

A Figura 11 mostra graficamente o cenário gerado pela ferramenta para a Mudança Incremental baseado na implementação proposta:

Figura 11 - Gráfico da simulação de Mudança Incremental

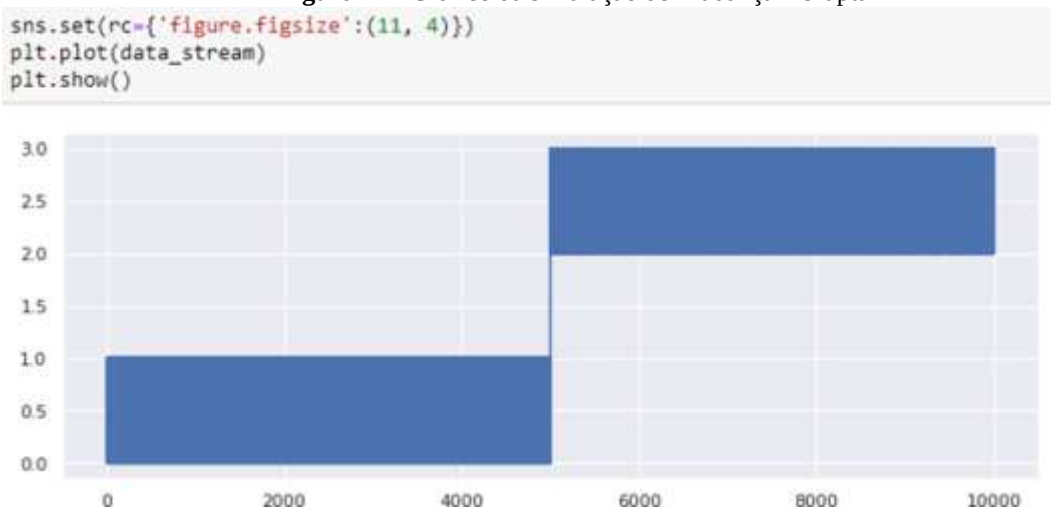


Fonte: Autores (2019).

O eixo X representa o número de instâncias e o eixo Y seus respectivos valores. Ocorre Mudança de Conceito a partir do índice 1.000 e a partir do índice 4.000. O padrão inicial que deu origem à simulação no índice 0 desaparece e novos padrões começam a existir e, com o passar do tempo, novos padrões surgem substituindo os padrões existentes anteriormente.

A Figura 12 mostra graficamente o cenário gerado pela ferramenta para a Mudança Abrupta baseada na implementação proposta:

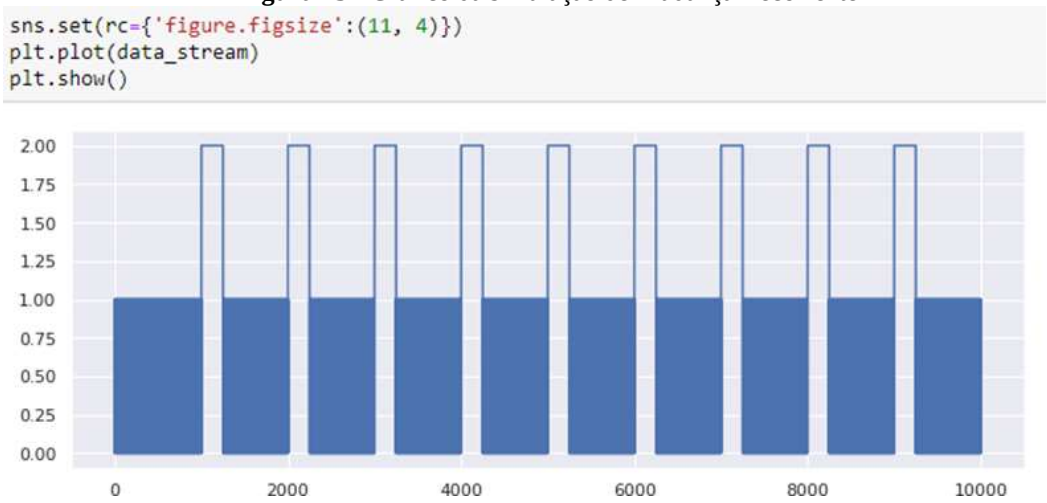
Figura 12 - Gráfico da simulação de Mudança Abrupta



Fonte: Autores (2019).

No caso acima ocorre Mudança de Conceito a partir do índice 5.000. A Figura 13 mostra graficamente o cenário gerado pela ferramenta para a Mudança Recorrente baseado na implementação proposta:

Figura 13 - Gráfico da simulação de Mudança Recorrente



Fonte: Autores (2019).

O eixo X representa o número de instâncias e o eixo Y seus respectivos valores. A cada 1.000 instâncias ocorre Mudança de Conceito nas próximas 250 instâncias dos dados, caracterizando a Mudança Recorrente.

3.2 Aplicação dos algoritmos de Detecção de Mudança

Para cada um dos cenários simulados foram aplicados os algoritmos ADWIN, DDM e Page-Hinkley a fim de comparar o desempenho na Detecção de Mudança de Conceito no ambiente controlado ao qual a ferramenta proporciona.

3.2.1 Detecção de Mudança com ADWIN

A Figura 14 apresenta o resultado da Detecção de Mudança Gradual com o ADWIN:

Figura 14 - Detecção de Mudança Gradual com ADWIN

```
Concept Drift detectado nos dados: 2 - indíce: 1055
Concept Drift detectado nos dados: 3 - indíce: 1087
Concept Drift detectado nos dados: 2 - indíce: 1151
Concept Drift detectado nos dados: 0 - indíce: 1567
Concept Drift detectado nos dados: 0 - indíce: 1599
Concept Drift detectado nos dados: 0 - indíce: 1663
Concept Drift detectado nos dados: 3 - indíce: 3103
Concept Drift detectado nos dados: 3 - indíce: 3135
Concept Drift detectado nos dados: 2 - indíce: 3167
Concept Drift detectado nos dados: 1 - indíce: 4095
Concept Drift detectado nos dados: 0 - indíce: 4159
Concept Drift detectado nos dados: 2 - indíce: 4575
Concept Drift detectado nos dados: 3 - indíce: 4607
Concept Drift detectado nos dados: 2 - indíce: 4639
Concept Drift detectado nos dados: 3 - indíce: 4671
Concept Drift detectado nos dados: 2 - indíce: 4735
```

Fonte: Autores (2019).

O algoritmo ADWIN foi capaz de identificar Mudança de Conceito a partir do índice 1055. A Figura 15 apresenta o resultado da Detecção de Mudança Incremental com o ADWIN:

Figura 15 - Detecção de Mudança Incremental com ADWIN

```
Concept Drift detectado nos dados: 1 - indíce: 1151
Concept Drift detectado nos dados: 0 - indíce: 1183
Concept Drift detectado nos dados: 0 - indíce: 1215
Concept Drift detectado nos dados: 0 - indíce: 1247
Concept Drift detectado nos dados: 1 - indíce: 1279
Concept Drift detectado nos dados: 1 - indíce: 1407
Concept Drift detectado nos dados: 0 - indíce: 1663
Concept Drift detectado nos dados: 1 - indíce: 2015
Concept Drift detectado nos dados: 1 - indíce: 2239
Concept Drift detectado nos dados: 0 - indíce: 2303
Concept Drift detectado nos dados: 3 - indíce: 2335
Concept Drift detectado nos dados: 3 - indíce: 2367
Concept Drift detectado nos dados: 1 - indíce: 2431
Concept Drift detectado nos dados: 2 - indíce: 3199
Concept Drift detectado nos dados: 4 - indíce: 3423
Concept Drift detectado nos dados: 2 - indíce: 3583
Concept Drift detectado nos dados: 2 - indíce: 4223
Concept Drift detectado nos dados: 2 - indíce: 4351
Concept Drift detectado nos dados: 5 - indíce: 4415
Concept Drift detectado nos dados: 3 - indíce: 4607
Concept Drift detectado nos dados: 1 - indíce: 4799
Concept Drift detectado nos dados: 7 - indíce: 6111
Concept Drift detectado nos dados: 2 - indíce: 6271
Concept Drift detectado nos dados: 2 - indíce: 6335
Concept Drift detectado nos dados: 7 - indíce: 6591
Concept Drift detectado nos dados: 3 - indíce: 7039
Concept Drift detectado nos dados: 6 - indíce: 7167
Concept Drift detectado nos dados: 7 - indíce: 7295
Concept Drift detectado nos dados: 8 - indíce: 7423
Concept Drift detectado nos dados: 5 - indíce: 7679
Concept Drift detectado nos dados: 4 - indíce: 8063
Concept Drift detectado nos dados: 8 - indíce: 8191
Concept Drift detectado nos dados: 5 - indíce: 8767
Concept Drift detectado nos dados: 3 - indíce: 9311
Concept Drift detectado nos dados: 3 - indíce: 9343
Concept Drift detectado nos dados: 6 - indíce: 9759
```

Fonte: Autores (2019).

A Figura 16 apresenta o resultado da Detecção de Mudança Abrupta com o ADWIN:

Figura 16 - Detecção de Mudança Abrupta com ADWIN

```
Concept Drift detectado nos dados: 2 - indíce: 5055
Concept Drift detectado nos dados: 3 - indíce: 5087
Concept Drift detectado nos dados: 3 - indíce: 5119
Concept Drift detectado nos dados: 2 - indíce: 5151
Concept Drift detectado nos dados: 2 - indíce: 5471
```

Fonte: Autores (2019).

Para o cenário de Mudança Abrupta, o algoritmo foi capaz de identificar Mudança deConceito a partir do índice 5.055.

A Figura 17 apresenta o resultado da Detecção de Mudança Recorrente com oADWIN*:*

Figura 17 - Detecção de Mudança Recorrente com ADWIN

```
Concept Drift detectado nos dados: 3 - indíce: 1567
Concept Drift detectado nos dados: 3 - indíce: 1599
Concept Drift detectado nos dados: 2 - indíce: 1663
Concept Drift detectado nos dados: 1 - indíce: 1791
Concept Drift detectado nos dados: 1 - indíce: 1823
Concept Drift detectado nos dados: 1 - indíce: 1887
Concept Drift detectado nos dados: 0 - indíce: 1919
Concept Drift detectado nos dados: 3 - indíce: 3039
Concept Drift detectado nos dados: 2 - indíce: 3071
Concept Drift detectado nos dados: 2 - indíce: 3103
Concept Drift detectado nos dados: 2 - indíce: 3135
Concept Drift detectado nos dados: 3 - indíce: 3167
Concept Drift detectado nos dados: 0 - indíce: 3551
Concept Drift detectado nos dados: 1 - indíce: 3583
Concept Drift detectado nos dados: 1 - indíce: 3615
Concept Drift detectado nos dados: 0 - indíce: 3647
Concept Drift detectado nos dados: 1 - indíce: 3743
Concept Drift detectado nos dados: 2 - indíce: 4575
Concept Drift detectado nos dados: 3 - indíce: 4607
Concept Drift detectado nos dados: 2 - indíce: 4639
Concept Drift detectado nos dados: 2 - indíce: 4831
Concept Drift detectado nos dados: 1 - indíce: 5343
Concept Drift detectado nos dados: 1 - indíce: 5375
Concept Drift detectado nos dados: 0 - indíce: 5407
Concept Drift detectado nos dados: 3 - indíce: 6111
Concept Drift detectado nos dados: 3 - indíce: 6143
Concept Drift detectado nos dados: 3 - indíce: 6175
Concept Drift detectado nos dados: 2 - indíce: 6335
Concept Drift detectado nos dados: 1 - indíce: 7071
Concept Drift detectado nos dados: 1 - indíce: 7103
Concept Drift detectado nos dados: 1 - indíce: 7135
Concept Drift detectado nos dados: 1 - indíce: 7167
Concept Drift detectado nos dados: 0 - indíce: 7231
Concept Drift detectado nos dados: 0 - indíce: 7295
Concept Drift detectado nos dados: 2 - indíce: 7583
Concept Drift detectado nos dados: 2 - indíce: 7615
Concept Drift detectado nos dados: 3 - indíce: 7647
Concept Drift detectado nos dados: 2 - indíce: 7679
Concept Drift detectado nos dados: 2 - indíce: 7807
Concept Drift detectado nos dados: 0 - indíce: 9119
Concept Drift detectado nos dados: 1 - indíce: 9151
Concept Drift detectado nos dados: 1 - indíce: 9247
Concept Drift detectado nos dados: 2 - indíce: 9503
Concept Drift detectado nos dados: 3 - indíce: 9567
```

Fonte: Autores (2019).

No cenário recorrente o algoritmo ADWIN detectou Mudança de Conceito a partir do índice 1.567. A Figura 18 apresenta o resultado da Detecção de Mudança Abrupta com o DDM:

Figura 18 - Detecção de Mudança Abrupta com DMM

```
Zona de aviso : 3 - no indíce: 5017
Zona de aviso : 2 - no indíce: 5018
Zona de aviso : 2 - no indíce: 5019
Zona de aviso : 2 - no indíce: 5020
Zona de aviso : 3 - no indíce: 5021
Zona de aviso : 3 - no indíce: 5022
Zona de aviso : 3 - no indíce: 5023
Zona de aviso : 3 - no indíce: 5024
Zona de aviso : 3 - no indíce: 5025
Zona de aviso : 3 - no indíce: 5026
Zona de aviso : 3 - no indíce: 5027
Zona de aviso : 3 - no indíce: 5028
Zona de aviso : 3 - no indíce: 5029
Zona de aviso : 2 - no indíce: 5030
Zona de aviso : 3 - no indíce: 5031
```

Fonte: Autores (2019).

Para o cenário de Mudança Abrupta, o algoritmo DMM detectou Mudança de Conceito a partir do índice 5.017.

A Figura 19 apresenta o resultado da Detecção de Mudança Recorrente com o DDM:

Figura 19 - Detecção de Mudança Recorrente com DMM

```
Zona de aviso : 3 - no indíce: 1511
Zona de aviso : 2 - no indíce: 1512
Zona de aviso : 2 - no indíce: 1513
Zona de aviso : 2 - no indíce: 1514
Zona de aviso : 2 - no indíce: 1515
Zona de aviso : 3 - no indíce: 1516
Zona de aviso : 3 - no indíce: 1517
Zona de aviso : 3 - no indíce: 1518
Zona de aviso : 3 - no indíce: 1519
Zona de aviso : 3 - no indíce: 3009
Zona de aviso : 2 - no indíce: 3010
Zona de aviso : 2 - no indíce: 3011
Zona de aviso : 2 - no indíce: 3012
Zona de aviso : 3 - no indíce: 3013
Zona de aviso : 2 - no indíce: 3014
Zona de aviso : 3 - no indíce: 3015
Zona de aviso : 2 - no indíce: 3016
Zona de aviso : 3 - no indíce: 3017
Zona de aviso : 2 - no indíce: 3018
Zona de aviso : 2 - no indíce: 3019
```

Fonte: Autores (2019).

O algoritmo DDM detectou Mudança Recorrente a partir do índice 1.511.

3.2.3 Detecção de Mudança com Page-Hinkley

A Figura 20 apresenta a Detecção de Mudança Gradual com o algoritmo Page-Hinkley:

Figura 20 - Detecção de Mudança Gradual com Page-Hinkley

```
Mudança detectada no dado: 3 - no indíce: 1027
Mudança detectada no dado: 3 - no indíce: 3032
Mudança detectada no dado: 3 - no indíce: 4572
```

Fonte: Autores (2019).

O algoritmo identificou Mudança Gradual a partir do índice 1.027.

A Figura 21 apresenta o resultado da Detecção de Mudança Incremental com o Page-Hinkley:

Figura 21 - Detecção de Mudança Incremental com Page-Hinkley

```
Mudança detectada no dado: 2 - no indíce: 1094
Mudança detectada no dado: 3 - no indíce: 2127
Mudança detectada no dado: 3 - no indíce: 3026
Mudança detectada no dado: 5 - no indíce: 4010
Mudança detectada no dado: 5 - no indíce: 4413
Mudança detectada no dado: 5 - no indíce: 5026
Mudança detectada no dado: 6 - no indíce: 6078
Mudança detectada no dado: 8 - no indíce: 7061
Mudança detectada no dado: 7 - no indíce: 7510
Mudança detectada no dado: 6 - no indíce: 7713
Mudança detectada no dado: 7 - no indíce: 8048
Mudança detectada no dado: 9 - no indíce: 8314
Mudança detectada no dado: 8 - no indíce: 8729
Mudança detectada no dado: 9 - no indíce: 8949
Mudança detectada no dado: 7 - no indíce: 9057
Mudança detectada no dado: 10 - no indíce: 9842
```

Fonte: Autores (2019).

No cenário incremental o algoritmo Page-Hinkley detectou Mudança de Conceito a partir do índice 1.094.

A Figura 22 apresenta o resultado da Detecção de Mudança Abrupta com o Page-Hinkley:

Figura 22 - Detecção de Mudança Abrupta com Page-Hinkley

```
Mudança detectada no dado: 2 - no indíce: 5025
```

Fonte: Autores (2019).

A Mudança de Conceito foi detectada pelo algoritmo a partir no índice 5.025.

A Figura 23 apresenta o resultado da Detecção de Mudança Recorrente com o algoritmo Page-Hinkley.

Figura 23 - Detecção de Mudança Recorrente com Page-Hinkley

```
Mudança detectada no dado: 3 - no indíce: 1524
Mudança detectada no dado: 3 - no indíce: 3028
Mudança detectada no dado: 3 - no indíce: 4542
Mudança detectada no dado: 2 - no indíce: 6056
Mudança detectada no dado: 3 - no indíce: 7580
Mudança detectada no dado: 3 - no indíce: 9455
```

Fonte: Autores (2019).

3.3 Análise de desempenho dos algoritmos de Detecção de Mudança

A seguir são apresentadas as tabelas com resultado dos algoritmos de Detecção de Mudança de Conceito para os cenários simulados: a primeira linha indica entre quais índices ocorreram Mudança de Conceito, para métrica de avaliação serão considerados como mais precisos os algoritmos que mais se aproximarem do índice inicial ao qual ocorreu Mudança de Conceito, a segunda linha indica os índices ao qual o ADWIN detectou Mudança de Conceito, a terceira linha representa a Detecção de Mudança do algoritmo DDM e a última linha, as detecções de Mudança de Conceito pelo algoritmo Page-Hinkley.

Caso o algoritmo não detecte Mudança de Conceito, será adicionado um X no lugar do valor de detecção.

Tabela 1 - Comparação de desempenho de Detecção de Mudança de Conceito para o cenário de Mudança Gradual

Índice de Mudançade Conceito:	1500 a 1750	3000 a 3500	4500 á 5250
ADWIN	1055	3103	4575
DDM	1008	3003	X
Page-Hinkley	1027	3032	4572

Fonte: O autor (2019)

A partir do índice 5250 o padrão inicial já não é o que se apresenta em maior volume e os algoritmos se adaptam à mudança e não detectam mais a Mudança de Conceito visto que essa não ocorre mais.

Para a Mudança de Conceito Gradual, o algoritmo Page-Hinkley obteve o resultado mais preciso na Detecção de Mudança de Conceito.

Tabela 2 - Comparação de desempenho de Detecção de Mudança de Conceito para o cenário de Mudança Incremental

Índice de Mudança de Conceito:	1000 a 2000	2000 a 3000	3000 a 4000	4000 a 5000	5000 a 6000	6000 a 7000	7000 a 8000	8000 a 9000	9000 a 10000
ADWIN	1151	2015	3199	4223	X	6111	7039	8063	9311
Page-Hinkley	1094	2127	3026	4010	5026	6078	7061	8048	9057

Fonte: O autor (2019)

Na Mudança Incremental, o DDM apresenta a melhor precisão para a primeira ocorrência de Mudança de Conceito, porém o algoritmo Page-Hinkley apresenta resultados mais confiáveis para o cenário gerado.

Tabela 3 - Comparação de desempenho de Detecção de Mudança de Conceito para o cenário de Mudança Abrupta

Índice de Mudança de Conceito:	5000:10000
ADWIN	5055
DDM	5017
Page-Hinkley	5025

Fonte: O autor (2019)

Para este cenário, o DDM provou ser o algoritmo com maior precisão para Detecção de Mudança de Conceito.

Tabela 4 - Comparação de desempenho de Detecção de Mudança para o cenário de Mudança Recorrente

Índice de Mudança de Conceito:	1000 a 1250	2000 a 2250	3000 a 3250	4000 a 4250	5000 a 5250	6000 a 6250	7000 a 7250	8000 a 8250	9000 a 9250
ADWIN	1567	3039	3039	4575	5343	6111	7071	9119	9119
DDM	1511	X	3009	X	X	X	X	X	X
Page-Hinkley	1524	3028	X	4542	X	6056	7580	X	9455

Fonte: Autores (2019)

Na Mudança Recorrente, o ADWIN apresenta os melhores resultados para o cenário gerado.

CONSIDERAÇÕES FINAIS

A proposta de desenvolvimento do projeto foi a criação de uma ferramenta geradora de cenário de Mudança de Conceito e a aplicação e comparação de algoritmos de Detecção de Mudança no ambiente simulado.

No contexto de inteligência, é importante entender-se o comportamento dos dados de acordo com o tempo e entender que os dados mudam: faz-se necessário que se exista mecanismos que mostrem quando essas mudanças ocorrem, seja para re-treinar um modelo de rede neural, para atualizar dados cadastrais de um cliente ou até mesmo para uma tomada de decisão de âmbito geral em uma empresa de grande porte.

A ferramenta geradora de cenário de Mudança de Conceito desenvolvida no presente trabalho fornece ao usuário um ambiente controlado para execução e testes de novos algoritmos de Detecção de Mudança de Conceito. Como forma de experimentação dos cenários, foi feita uma análise da eficácia de três métodos de Detecção de Mudanças existentes nos diferentes cenários de Mudança de Conceito simulados

Nos cenários criados pelo gerador, o algoritmo Page-Hinkley obteve melhores resultados nas Mudanças de Conceito Gradual e Incremental, o DMM se mostrou o melhor na Mudança de Conceito Abrupta e o ADWIN na Mudança de Conceito Recorrente.

Muito trabalho ainda pode ser feito em relação aos algoritmos de Detecção de Mudança de Conceito, seja testando com dados de um ambiente real ou a implementação de melhorias nos próprios algoritmos para detectar uma Mudança de Conceito mais precisa.

REFERÊNCIAS

AMARAL, F. Aprenda mineração de dados: teoria e prática. Rio de Janeiro: Alta Books, 2016.

BAENA-GARCÍA, Manuel;Campo-Ávila, José AU - Fidalgo-Merino, Raúl;Bifet, Albert ;Gavald, Ricard ;Morales-Bueno, Rafael , Early Drift Detection Method ,2006.

BIFET, A., Gavaldá, R.: Kalman filters and adaptive windows for learning in data streams. In: Todorovski, L., Lavrač, N., Jantke, K.P. (eds.) DS 2006. LNCS (LNAI). Springer, Heidelberg (2006).

DATAR, M., Gionis, A., Indyk, P., Motwani, R.: Maintaining stream statistics over sliding windows. SIAM Journal on Computing 14(1), 27–45 (2002).

DUDA, R.O., Hart, P.E. and Stork, D.G. (2001) Pattern Classification. John Wiley and Sons, New York.

HAWKINS, DOUGLAS M.; Identification of Outliers; Chapman and Hall. 1980.

Java Point. Aprendizado de Máquina Supervisionado. Disponível em: <https://www.javatpoint.com/supervised-machine-learning> Acesso em: 11/11/2019.

João Gama, Indrè ŽLiobaité, Albert Bifet, Mykola Pechenizkiy, Abdelhamid Bouchachia. A survey on concept drift adaptation. In: ACM COMUTIM SURVEYS. 2014, CSUR.

KHAMASSI, Imen; SAYED-MOUCHAWEH, Moamar. Drift detection and monitoring in non-stationary environments. In: IEEE CONFERENCE ON EVOLVING AND ADAPTIVE INTELLIGENT SYSTEMS (EAIS). 2014, IEEE.

LIBRALÃO, Giampaolo Luiz; OSHIRO, Rodrigo Mithuhiro; VALERIO NETTO, Antonio; CARVALHO, André Carlos Ponce de Leon Ferreira; OLIVEIRA, Maria Cristina Ferreira de. Técnicas de aprendizado de máquina para análise de imagens oftalmológicas. Anais. Fortaleza: [s. n.], 2003.

LUDMILA I. Kuncheva. Combining Pattern Classifiers: Methods and Algorithms, 2004.

MARTINS W, AFONSECA UR, NALINI LE, GOMES VM. Tutoriais inteligentes baseados em aprendizado por reforço: concepção, implementação e avaliação empírica. Anais do SBIE. 2007 Nov 1.

MIROSLAV Kubat ; Gerhard Widmer. Adapting to drift in continuous domains (extended abstract). In Proceedings of the 8th European Conference on Machine Learning, 1995.

MITCHELL, T. M. Machine Learning. McGraw-Hill. EUA. 1997.

MOHRI, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations of machine learning. Boston, MA: MIT Press.

SATHYA, R. and Abraham, A. (2013) Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. International Journal of Advanced Research in Artificial Intelligence, 2, 34-38.

Scikit-Multiflow: A Multi-outputStreaming Framework. Jacob Montiel, Jesse Read, Albert Bifet, Talel Abdessalem; Journal of Machine Learning Research, 19(72):1−5, 2018.

Tsymbal, Alexey. (2004). The Problem of Concept Drift: Definitions and Related Work.

Xaltius. Aprendizado por Reforço. Disponível em: <https://xaltius.tech/reinforcement- learning/> Acesso em: 11/11/2019.

ZLIOBATE, Indrè. Learning under Concept Drift: an Overview. 2009. - Vilnius University.