

Improving Fuzzy Rule-Based Classifier Systems Considering Data Streams and Concept Drift

Vanderleicio Carvalho Leite Junior¹, Matheus Giovanni Pires¹

¹Universidade Estadual de Feira de Santana (UEFS)
Avenida Transnordestina S/N, Novo Horizonte – 44036-900 – Feira de Santana, BA – Brasil

vanderleiciojr397@gmail.com, mgpires@uefs.br

Abstract. This study evaluates the performance of a Fuzzy Rule-Based Classification System in the context of data streams and concept drift. The Hoeffding Drift Detection Method (HDDM) is employed to identify concept changes within the data stream, which may occur either gradually or abruptly. Upon detection of concept drift, three distinct strategies for rule adaptation were investigated. In the first strategy, a new rule base is entirely generated. In the second, a utility value is computed for each rule, and those with utility below the average are discarded, and subsequently, new rules are induced using the current data. In the third strategy, new rules are first generated, followed by utility evaluation. Then, both existing and newly generated rules with utility below the average are removed. Experiments were conducted across all three strategies using both synthetic and real-world datasets, varying the window size between concepts from 100 to 1000 instances. The results indicate that the first and second strategies provided more effective system adaptation during concept transitions, for both synthetic and real data. Finally, the Friedman statistical test was applied, confirming the statistical significance of the results, with the exception of only two cases.

Resumo. Neste trabalho é avaliado o desempenho de um Sistema Classificador Baseado em Regras Fuzzy considerando fluxo de dados e concept drift. O método Hoeffding Drift Detection Method é aplicado para detectar as mudanças de conceito no fluxo de dados, as quais podem ser de dois tipos, gradual e abrupto. Após a detecção do concept drift, três cenários foram avaliados para o ajuste das regras do sistema. No primeiro, uma nova base de regras é gerada. No segundo, um valor de utilidade é calculado para as regras, e as que tiverem utilidade abaixo dessa média são descartadas. Após isso, novas regras são criadas utilizando os dados atuais. Por fim, no terceiro cenário, novas regras são geradas, e em seguida, a utilidade é calculada. As regras, novas e antigas, que tenham utilidade menor do que a média são descartadas. Experimentos foram executados nos três cenários considerando datasets artificiais e reais, variando o tamanho da janela de dados entre os conceitos em 100 e 1000 exemplos. Os resultados mostram que os cenários 1 e 2 ajustaram melhor o sistema durante a mudança dos conceitos, tanto para os dados artificiais quanto para os reais. Por fim, o teste estatístico de Friedman foi aplicado e constatou que os resultados são significativos, com exceção de apenas dois casos.

1. Introdução

Dados são o principal recurso no que diz respeito ao desenvolvimento de modelos de *Machine Learning* (ML), principalmente para modelos que se propõem a lidar e resolver problemas complexos, como por exemplo, modelos que analisam dados que variam ao longo do tempo. Neste caso, esses dados são considerados um fluxo de dados, que é caracterizado por ser um grande conjunto de dados organizados em *timestamps*, ou seja, com informações de data e hora, com velocidades variadas e que precisam ser analisados praticamente em tempo real [Agrahari and Singh 2022].

Considerando as características de um fluxo de dados, variações podem ocorrer nos padrões ao longo do tempo, alterando a distribuição estatística dos dados, caracterizando um *Concept Drift*, ou seja, atributos que antes caracterizavam um padrão de informação passam a caracterizar outro. Por exemplo, a mudança na escrita de textos de *spams* ao longo do tempo, representa um *Concept Drift* nos dados que esses e-mails representam [Alshalali and Josyula 2020]. Ao desenvolver modelos pensados para lidar com fluxo de dados, diferentes estratégias podem ser adotadas, desde treiná-los incrementalmente promovendo uma atualização contínua, ou até retreiná-los usando o conjunto de dados mais recente. A estratégia a ser usada dependerá do equilíbrio dos dados e da eficiência computacional desejada [Gama et al. 2014].

Diante da necessidade de adaptação dos modelos, devido a variação dos dados ao longo do tempo, é inerente a necessidade o desenvolvimento de algoritmos que consigam identificar um *concept drift*, para que medidas relativas ao aprimoramento do modelo sejam tomadas. [Frias-Blanco et al. 2014] propõe um detector de *concept drift* chamado *Hoeffding Drift Detection Method* (HDDM), que se baseia na métrica de avaliação do modelo que está sendo usada para determinar se os dados realmente sofreram *concept drift*. Mais especificamente, este detector irá alertar se houve um *concept drift* com base na variação do desempenho do modelo, ou seja, se esta variação foi estatisticamente significativa, ocorreu um *concept drift*. O HDDM possui duas versões, o HDDM_A e o HDDM_W. A primeira versão é adequada para a detecção de *concept drift* abruptos, isto é, que acontecem repentinamente em um determinado momento do tempo. A segunda versão foi projetada para *concept drift* graduais, ou seja, acontecem gradualmente durante um intervalo de tempo. O novo conceito vai surgindo enquanto o antigo desaparece. A principal diferença entre eles é que o HDDM_W dá um peso maior aos resultados mais recentes dentro de uma janela de resultados do modelo.

A tarefa de classificação de padrões é uma tarefa presente em diversos problemas do mundo real, como por exemplo, o diagnóstico de uma doença, a identificação de uma fraude em uma transação bancária e o reconhecimento de uma emergência em uma cidade, estado ou país (enchentes, tsunamis, terremotos, etc.). Na literatura é possível encontrar vários algoritmos de ML aplicados na resolução de problemas de classificação, dentre eles, podemos citar as Redes Neurais Artificiais [Emam et al. 2024], Árvores de Decisão [Tan et al. 2024], *Support Vector Machine* [Chagas et al. 2020] e os Sistemas Fuzzy [Charizanos et al. 2024].

Em um Sistema Fuzzy, a forma de representação do conhecimento é por meio de regras fuzzy, que permite a modelagem de conhecimento vago e incerto, muito comum em aplicações reais, e também possui a facilidade de expressar o comportamento do sistema em uma linguagem de fácil compreensão para os seres humanos. Desta forma, o conhe-

cimento não é uma caixa preta, facilitando a manutenção e alteração do conhecimento do sistema quando necessário. Sendo assim, os Sistemas Fuzzy, também conhecidos como Sistemas Baseados em Regras Fuzzy (SBRF), ou também, Sistemas de Classificação Baseados em Regras Fuzzy (SCBRF), quando aplicados à problemas de classificação, é uma abordagem muito utilizada na literatura [Lu et al. 2024].

Diante desse contexto, este trabalho se propõe avaliar o desempenho de um Sistema de Classificação Baseado em Regras Fuzzy considerando fluxo de dados. Para isso, utilizamos as duas versões do algoritmo HDDM, e sempre que um *concept drift* é detectado, a base de regras do sistema fuzzy é modificada, considerando três possíveis configurações. Na primeira, a base de regras é recriada usando somente os dados da janela atual. Na segunda, as regras com utilidade menor do que a utilidade média da base de regras são descartadas, e em seguida, a base é incrementada com novas regras. Por fim, na terceira configuração, a base é incrementada com novas regras e só depois a sua utilidade média é calculada e as regras com utilidade menor do que a média são descartadas.

Este artigo está organizado da seguinte forma: A seção 2 apresenta trabalhos que tratam sobre o aprimoramento de modelos de *machine learning* considerando fluxo de dados, principalmente modelos fuzzy. A seção 3 apresenta a metodologia desenvolvida neste trabalho. A seção 4 apresenta e discute os resultados dos experimentos, e por último, trazemos as conclusões acerca do trabalho.

2. Revisão da Literatura

O melhoramento de modelos de *Machine Learning* considerando fluxos de dados está estreitamente alinhado com a necessidade de detecção dos *concepts drifts*, visto que eles podem ser responsáveis pela queda das taxas de desempenho do modelo durante os testes em situações reais. Na literatura é possível encontrar variadas formas, tanto na detecção de *concepts drifts* quanto na forma de atualização do SCBRF, após a detecção do *concept drift*. Sendo assim, esta seção destaca estudos importantes sobre essas técnicas.

No artigo de [Agrahari and Singh 2022], os autores fazem uma revisão de literatura sobre *concept drift*, seus tipos e os métodos usados para detectá-lo. Com esse intuito, os autores revisitam o conceito de *concept drift* no contexto de fluxo de dados, que são de tamanho infinito e nos quais os dados são não estacionários, caracterizando o *concept drift* como a mudança na distribuição fundamental dos dados ao longo do tempo. Também são elucidados os tipos de *drifts*, sendo eles: o abrupto, quando o conceito muda repentinamente num determinado momento; o gradual, quando a mudança leva um determinado tempo para acontecer, durante o qual dois conceitos estão simultaneamente presentes; e o recorrente, no qual um conceito antigo pode reaparecer. Eles também discutem sobre os principais tipos de detectores de *drift*, seus pontos fortes, fracos e aplicações. Destacamos o método baseado em estatística, o HDDM, que independe de classificador, e permite capturar diferentes tipos de *concept drift*, embora possa cair em desempenho, se houver alta não linearidade nos limites de classificação.

O HDDM é proposto por [Frias-Blanco et al. 2014], frente a necessidade de um detector de *concept drift* que não fizesse suposições relacionadas às funções de densidade de probabilidade que geram os valores a serem medidos pelo detector. Dessa forma, ele se baseia na desigualdade de Hoeffding para determinar um limite, calculado a partir

das diferenças entre as médias de desempenho do classificador. O método apresenta duas variações, o HDDM_A, no qual as médias são analisadas diretamente, e por isso consegue detectar melhor as mudanças abruptas, e o HDDM_W, que utiliza um esquema de pesos para as médias, priorizando os resultados mais recentes, e sendo mais apropriado para a detecção de mudanças graduais. Para testar os métodos propostos, [Frias-Blanco et al. 2014] utilizam dados artificiais, gerados através do *Massive On-line Analysis* (MOA) pelos geradores LED, STAGGER e AGRAWAL, e cinco *datasets* reais: *Electricity Market* dataset (ELEC2), USENET1 e USENET2, *Spam Assassin collection* (SPAM), *Forest Cover Type* (COVERTYPE) e o *Nursery* dataset. Nos testes, os autores alinharam os métodos de detecção propostos com algum modelo de classificação (Naive-bayes e Perceptron), e perceberam que há melhora do desempenho dos modelos quando alinhados a uma forma de detecção de *concept drift*.

No que diz respeito ao melhoramento de classificadores baseados em regras fuzzy, [Shahparast et al. 2014] propõe um método diferente para atualizar os pesos das regras para lidar com o *concept drift* em fluxo de dados. A atualização do modelo é constante e com base em novos dados dentro de uma janela de tamanho fixo. Assim, a cada janela, os pesos de um pequeno subconjunto de regras são atualizados, com base no impacto que o peso de cada regra teve na classificação do novo dado. O conjunto de regras que será atualizado é definido com base no impacto que a alteração da regra teria na classificação daquele dado (se não há impacto a regra não é modificada), e no fato da classe do consequente da regra corresponder à classe do dado. O método foi testado com dados reais e sintéticos e mostrou que é capaz de melhorar o classificador fuzzy e, ao ser comparado com outros métodos (*Accuracy Weighted Ensemble* e *Hoeffding Tree* + o detector de *drift* DDM), obteve resultados comparáveis.

Já em [Varshney and Torra 2023], é feita uma revisão da literatura sobre as mais recentes aplicações de SBRF. Foram revisados trabalhos de 2010 até 2021, dos quais foram destacadas oito áreas: Sistemas Fuzzy Genéticos/Evolutivos (GFS/EFS), Sistemas Fuzzy Hierárquicos (HFS), Sistema de Inferência Fuzzy Neuro Adaptativo (ANFIS/NFS), Sistemas Fuzzy Evolutivos (eFS), SBRF para *Big Data*, SBRF para Conjuntos de Dados Desbalanceados, centroides de clusters como regras em SBRF e interpretabilidade/explicabilidade de SBRF. Especificamente sobre os eFS, são destacadas as estratégias de evolução do classificador com os novos dados que chegam, criando, atualizando ou deletando regras, caso os dados indiquem alguma necessidade de atualização. Além disso, alguns métodos levam em consideração o potencial da regra, representado por quanto o uso dela impacta na classificação dos dados. A revisão mostra que a área tem crescido, com uma média de 45 artigos por ano, com tendências recentes focadas em Neuro-eFS, atualizações estruturais no eFS, eFS tipo-2 e aplicações diversas.

Como os estudos revelam, há um grande interesse em trabalhar com o melhoramento de classificadores fuzzy considerando fluxos de dados, e principalmente, na ocorrência de *concept drift* que podem acontecer nesses casos, embora ainda não haja um consenso sobre os métodos usados tanto para a detecção de *concept drift* quanto na atualização do modelo. Frente a isso, este trabalho se propõe a analisar estratégias para o melhoramento de um SCBRF considerando fluxo de dados, e usando o detector de *concept drift* HDDM em suas duas versões, para determinar quando o modelo deve ser atualizado.

3. Metodologia

Essa seção descreve todos os passos usados neste trabalho na análise de desempenho de um SBRF aplicado na classificação de dados que variam no tempo, incluindo a determinação do algoritmo de detecção de *concept drift*, os *datasets* utilizados e a simulação do fluxo de dados, o processo de construção do SCBRF, os cenários propostos para comparação, a escolha da métrica usada para avaliar o modelo e os testes estatísticos usados para determinar se houve diferença significativa entre os resultados.

3.1. Algoritmo de detecção de *concept drift*

Como o objetivo do trabalho é analisar possíveis estratégias para o ajuste do SCBRF, o algoritmo de detecção de *concept drift* não foi o foco principal, embora seja muito importante, já que é o responsável por determinar quando o modelo será atualizado. Para essa tarefa foi escolhido o *Hoeffding Drift Detection Method* (HDDM) [Frias-Blanco et al. 2014] que, como já elucidado na Seção 2, é um detector de *drift* que se destaca com sua flexibilidade, por não depender do modelo que se está sendo avaliado e não fazer suposições estatísticas sobre a distribuição dos dados apresentados. Além disso, a família de detectores HDDM permite a identificação de dois tipos de desvios de conceitos, os abruptos (através do HDDM_A) e os graduais (através do HDDM_W), o que possibilita que dados variados possam ser usados para analisar a melhora do SCBRF. A partir de testes foi determinado o uso do parâmetro *confidence for the drift* de 0,005 para o HDDM_A e de 0,01 para o HDDM_W. Inclusive, os detectores propostos são testados em [Frias-Blanco et al. 2014] com uma variedade de conjuntos de dados, tanto artificiais quanto reais, e que são minuciosamente descritos, possibilitando a replicação desse uso para a avaliação do SCBRF, o que será explicado posteriormente.

3.2. Datasets e simulação do fluxo de dados

Baseado nos experimentos feitos por [Frias-Blanco et al. 2014], foram selecionados 15 *datasets*, sendo 12 deles com dados artificiais gerados a partir do *framework Massive Online Analysis* (MOA), e três *datasets* reais obtidos também com o MOA. É válido ressaltar que, embora não se possa ter certeza acerca da posição ou até mesmo da presença de *concept drift* em *datasets* que envolvam dados reais [Gonçalves Júnior 2013], ainda assim é importante testar o melhoramento de modelos usando dados reais, simulando o que seria a condição de execução normal do modelo.

Os *datasets* artificiais foram gerados a partir dos geradores STAGGER, LED e AGRAWAL. Os dados gerados pelo STAGGER englobam três atributos que codificam um objeto e o modelo precisa classificar se os atributos se referem ou não às características do objeto que está sendo mapeado pela função da vez. O *drift* é feito ao alternar entre as funções que estão definindo quais são as características que definem a classe do objeto como verdadeira. Já o LED, o objetivo é prever qual dígito será mostrado em um display de LED de sete segmentos, com cada atributo tendo 10% de chance de estar invertido, o *drift* é gerado ao alternar quais atributos são relevantes para determinar o número que será exibido. Por fim, o AGRAWAL produz dados com nove atributos, sendo seis numéricos e três categóricos, simulando a avaliação de uma pessoa para um empréstimo bancário. O *drift* é feito ao alterar a função que vai determinar se aquela pessoa é ou não habilitada a receber o empréstimo. Cada um dos geradores produziram quatro *datasets*, sendo

dois com *concepts drifts* abruptos, com 100 e 1000 instâncias de distância entre cada conceito, e os outros dois com *concepts drifts* graduais, que também apresentam 100 e 1000 instâncias de distâncias entre conceito, mas com a diferença de que o conceito começa a mudar 50 e 500 instâncias antes, respectivamente.

Da mesma forma, os conjuntos de dados reais também foram obtidos do MOA, sendo eles o Electricity, Covertype e o PokerHand, mas que não precisaram ser gerados. Dos três datasets utilizados, somente o "PokerHand" não foi usado em [Frias-Blanco et al. 2014], embora seja amplamente usado em outros trabalhos que tratam do tema [Bifet et al. 2009] [Brzezinski and Stefanowski 2013] [Sun et al. 2018]. O *Electricity* [University of Waikato 2025] é um *dataset* amplamente utilizado e que trata da tarefa de prever a mudança no preço da eletricidade no *New South Wales Electricity Market*, na Austrália, no período de 7 de Maio de 1996 até 5 de Dezembro de 1998. Cada exemplo é descrito por oito atributos e a classe representa se o preço subiu ou desceu. Já o Covertype [University of Waikato 2025] descreve características observacionais de uma área de floresta, a partir de 51 variáveis cartográficas, usando seis classes para determinar o tipo de cobertura vegetal. Finalmente, o PokerHand [University of Waikato 2025] consiste na descrição de 1 milhão de instâncias com 11 atributos. Cada instância representa uma mão de poker e cada carta é descrita pelo seu número e seu naipe. Há 10 classes possíveis de previsão, a versão utilizada foi a normalizada, na qual as cartas são ordenadas pelo ranque e naipe e as duplicatas são removidas.

A Tabela 1 detalha os *datasets* utilizados, apresentando seus nomes, tipos, tipo de *drift* e quantidade de instâncias. Destaque para os 12 datasets gerados a partir dos geradores artificiais, que foram nomeados com base nos geradores e no número de distâncias entre um conceito e o começo da mudança do próximo. Para treinar e testar o SCBRF, cada dataset foi organizado em janelas de 50 instâncias cada, com isso, a cada 50 instâncias o SCBRF é testado com os novos dados, e ajustado, caso tenha ocorrido um *concept drift*.

Tabela 1. Configuração dos *datasets* utilizados.

Nome do dataset	Tipo	Tipo de drift	Nº Instâncias
PokerHand	Real	Indeterminado	829201
Electricity	Real	Indeterminado	45312
CoverType	Real	Indeterminado	581012
Agrawal-100-Abrupto	Artificial	Abrupto	3100
Agrawal-1000-Abrupto	Artificial	Abrupto	31000
LED-100-Abrupto	Artificial	Abrupto	3100
LED-1000-Abrupto	Artificial	Abrupto	31000
STAGGER-100-Abrupto	Artificial	Abrupto	3100
STAGGER-1000-Abrupto	Artificial	Abrupto	31000
Agrawal-50-Gradual	Artificial	Gradual	3100
Agrawal-500-Gradual	Artificial	Gradual	31000
LED-50-Gradual	Artificial	Gradual	3100
LED-500-Gradual	Artificial	Gradual	31000
STAGGER-50-Gradual	Artificial	Gradual	3100
STAGGER-500-Gradual	Artificial	Gradual	31000

3.3. Construção da base de regras

Para construir a base de regras do SCBRF foi utilizado o algoritmo de Wang e Mendel [Wang and Mendel 1992]. Para cada base de regras gerada é considerada uma janela de dados, a qual será utilizada para classificar os dados das janelas seguintes. Além disso, para cada regra é criada uma métrica de utilidade, conforme proposto por [Angelov and Zhou 2008] e definido pela Equação 1. A utilidade η da i -ésima regra é a soma das suas forças de ativação (λ) desde a sua criação no instante $l=1$ até o instante atual k , dividido pelo seu tempo de vida (instante atual k menos o instante de criação (t_i) da i -ésima regra).

$$\eta_i(k) = \frac{\sum_{l=1}^k \lambda_l}{k - t_i}; i = [1, R] \quad (1)$$

3.4. Cenários propostos

Neste trabalho consideramos três possíveis cenários para o ajuste da base de regras do SC-BRF quando um *concept drift* é detectado. O "Cenário 1" corresponde ao caso em que não é aplicado nenhum tipo de aprendizado incremental na base de regras, ou seja, as regras são criadas na primeira janela de dados, e sempre que um desvio de conceito é detectado, elas são recriadas do zero considerando somente os dados da janela atual. Desta forma, o modelo está ajustado para os conceitos mais atuais dos dados, sem nenhuma preocupação com os anteriores.

O "Cenário 2" propõe um aprendizado incremental baseado na utilidade de cada uma das regras. Quando é detectado um *concept drift*, a base de regras não é mais totalmente recriada. Um valor de utilidade médio é calculado para as regras, e as que tiverem utilidade abaixo dessa média são descartadas. Após esse processo, novas regras são criadas utilizando os dados atuais. Eventuais conflitos na base de regras, como por exemplo, regras redundantes e/ou conflitantes, são resolvidos seguindo o algoritmo de Wang e Mendel. Esse cenário busca avaliar se o conhecimento criado a partir dos conceitos antigos ainda tem alguma utilidade em um novo contexto, e evita que um número excessivo de regras seja criado de forma desnecessária.

Já o último cenário também se propõe a ser um aprendizado incremental com base na utilidade das regras, mas evitando a criação de novas regras que podem ser inúteis. No "Cenário 3", a cada desvio de conceito, novas regras são geradas, e em seguida, a utilidade média é calculada. As regras, novas e antigas, que tenham utilidade menor do que a média são descartadas. Dessa forma, embora haja um incremento da base de regras, esse incremento é mais gradual, pois regras com baixa força de ativação são descartadas logo após a sua criação.

Com esses três cenários pretendemos avaliar os impactos de ajustes de um SCBRF em um contexto de fluxo de dados e *concepts drifts*. Devido à natureza desbalanceada e a presença de *datasets* com múltiplas classes, a métrica escolhida para avaliar o desempenho do sistema durante a sua evolução foi o F1-Score [Fawcett 2006].

4. Resultados e discussões

Esta seção descreve os resultados obtidos a partir dos três cenários descritos na Sessão 3. Os algoritmos foram implementados utilizando a linguagem Java através do OpenJDK

22.0.1, e o framework *Massive Online Analysis* (MOA) para a geração dos *datasets* artificiais e implementação dos detectores de *concept drift*. O framework *jFuzzyLogic* também foi usado para a implementação do SCBRF.

Tabela 2. Resultados dos cenários considerando dados artificiais e detecção gradual de *concept drift*.

Algoritmo de detecção	Cenário	100 exemplos por conceito			1000 exemplos por conceito		
		Led	Stagger	Agrawal	Led	Stagger	Agrawal
HDDM_W	1	\bar{x}	0,51833	0,819688	0,615276	0,442677	0,837599
		s	0,115144	0,177091	0,190356	0,107062	0,183713
	2	\bar{x}	0,487139	0,715109	0,651625	0,534488	0,812128
		s	0,155839	0,123829	0,13711	0,085977	0,200887
	3	\bar{x}	0,400497	0,729596	0,27843	0,356196	0,685461
		s	0,146377	0,105726	0,345828	0,092789	0,180194

Tabela 3. Resultados dos cenários considerando dados artificiais e detecção abrupta de *concept drift*.

Algoritmo de detecção	Cenário	100 exemplos por conceito			1000 exemplos por conceito		
		Led	Stagger	Agrawal	Led	Stagger	Agrawal
HDDM_A	1	\bar{x}	0,516201	0,644731	0,254401	0,455544	0,848314
		s	0,085688	0,190641	0,314377	0,097013	0,185051
	2	\bar{x}	0,516884	0,629707	0,453142	0,547091	0,71221
		s	0,084982	0,167922	0,293848	0,084977	0,184696
	3	\bar{x}	0,511907	0,542635	0,601526	0,359993	0,683298
		s	0,096278	0,149778	0,206757	0,094118	0,188213

Na Tabela 2 e na Tabela 3 são mostrados os valores do F1-score médio (\bar{x}) e do seu desvio padrão (s) para cada *dataset*, algoritmo de detecção de *concept drift*, tamanho da janela de dados (número de instâncias entre os *drifts*) e o cenário correspondente.

Considerando os *datasets* artificiais com desvios graduais e tamanho da janela igual a 100 exemplos (Tabela 2), é possível perceber que no Cenário 1, o SCBRF obteve melhor desempenho para os *datasets* Led e Stagger, enquanto que o Cenário 2 proporcionou um melhor ajuste para o SCBRF usando o *dataset* Agrawal. Por outro lado, quando o tamanho da janela é igual a 1000 exemplos, os resultados se invertem, ou seja, o Cenário 2 ajustou melhor o sistema para dois *datasets*, Led e Agrawal, enquanto o Cenário 1 ajustou melhor o sistema usando o *dataset* Stagger. Fazendo uma simples contagem, o SBRC obteve melhor desempenho em três *datasets* no Cenário 1 e 2, enquanto que o Cenário 3 em nenhum.

Um detalhe interessante é que, para os *datasets* Stagger e Agrawal, o sistema obteve melhor desempenho no Cenário 1 e Cenário 2, respectivamente, ou seja, o tamanho da janela de dados não influenciou na forma de ajuste do sistema. Porém, para o *dataset* Led, o SCBRF foi melhor no Cenário 1 quando o tamanho da janela é igual a 100, e foi melhor no Cenário 2 quando o tamanho é igual a 1000.

Tabela 4. Resultados dos cenários considerando dados reais.

Algoritmo de detecção	Cenário	PokerHand	CovType	Electricity
HDDM_W	1	\bar{x}	0,389401	0,405618
		s	0,1835867	0,293539
	2	\bar{x}	0,3450943	0,518195 0,3593684
		s	0,2060719	0,30127
	3	\bar{x}	0,349716	0,312741
		s	0,1931338	0,328775
HDDM_A	1	\bar{x}	0,3881609	0,406921
		s	0,187667	0,300594
	2	\bar{x}	0,3710257	0,564878 0,2944679
		s	0,1985836	0,273665
	3	\bar{x}	0,3572965	0,30267
		s	0,1978069	0,327809

Analizando a Tabela 3, que descreve os resultados considerando os *datasets* artificiais com desvios abruptos e tamanho da janela com 100 exemplos, constatamos que em cada cenário o sistema obteve melhor desempenho com um *dataset* diferente. No Cenário 1 com o Stagger, no Cenário 2 com o Led, e no Cenário 3, com o Agrawal. Considerando o tamanho da janela com 1000 exemplos, o Cenário 1 ajustou melhor o SCBRF usando os *datasets* Stagger e Agrawal, enquanto que no Cenário 2, o sistema foi melhor com o *dataset* Led. Um fato importante a destacar é que, independente do tamanho da janela, dois *datasets* não influenciaram no ajuste do SCBRF considerando detecção abrupta de *concept drift*. O Cenário 2 foi a melhor forma de ajuste do sistema usando o *dataset* Led, e o Cenário 1 ajustou melhor usando o *dataset* Stagger.

Novamente, fazendo uma simples contagem, o SBRC obteve melhor desempenho em três *datasets* no Cenário 1, em dois *datasets* no Cenário 2, e apenas um *dataset* no Cenário 3. Portanto, considerando dados artificiais com detecção de conceito gradual e abrupta, os Cenários 1 e 2 foram os que proporcionaram a melhor forma de ajuste para o SCBRF.

Considerando dados reais, na Tabela 4 são mostrados os valores de média e desvio padrão do F1-score. Para dados reais não é possível saber se há ou não, e qual o tipo de *concept drift* está presente. Sendo assim, as duas abordagens do HDDM foram aplicadas. Em ambos os casos, os cenários que tiveram os melhores desempenhos para cada *dataset* são os mesmos. O Cenário 1 para o PokerHand e o Cenário 2 para CovType e Electricity. Mais uma vez, o SCBRF obteve melhor ajuste nos Cenários 1 e 2, enquanto que o Cenário 3 não apresentou bons resultados em nenhum *dataset*.

Para verificar se há diferença estatisticamente significativa nos resultados dos cenários comparados, foi aplicado o teste estatístico de Friedman. A Tabela 5 exibe os resultados dos *p*_valores para cada *dataset*. Estão destacados em negrito apenas os *p*_valores que obtiveram valor maior que o nível de confiança de 0,05, o que significa

que a hipótese nula não foi rejeitada. Em outras palavras, não há evidências estatísticas suficientes para afirmar que estes resultados são significativos. Em apenas dois casos os resultados são não significativos, e para ambos os casos, o tamanho da janela é 100 com detecção abrupta. Esses casos são com os *datasets* Led e Stagger.

Tabela 5. P_valores do Teste de Friedman entre os três cenários.

Dataset-janela-tipo	p_valor	Dataset-janela-tipo	p_valor
PokerHand-Abrupto	3,8E-33	Led-1000-Abrupto	2,6E-153
PokerHand-Gradual	6,73E-69	Stagger-100-Abrupto	0,184637
Electricity-Abrupto	1,1E-208	Stagger-1000-Abrupto	3,16E-86
Electricity-Gradual	1,4E-247	Agrawal-50-Gradual	6,48E-05
CoverType-Abrupto	0	Agrawal-500-Gradual	2,28E-78
CoverType-Gradual	0	Led-50-Gradual	9,33E-08
Agrawal-100-Abrupto	4,2E-07	Led-500-Gradual	1,5E-139
Agrawal-1000-Abrupto	0,000848	Stagger-50-Gradual	7,49E-06
Led-100-Abrupto	0,22313	Stagger-500-Gradual	2,4E-108

5. Conclusões

Neste trabalho foram analisados três cenários de ajuste de um Sistema de Classificação Baseado em Regras Fuzzy considerando fluxo de dados com *concept drift*. Quando um *concept drift* é detectado, a base de regras do sistema de classificação deve ser ajustada, para que o sistema possa continuar tendo um bom desempenho de classificação. Para o ajuste da base de regras foi adotada a métrica utilidade, a qual foi usada como critério de escolha das regras a serem excluídas.

Os experimentos foram executados com dados artificiais e reais, sendo que quando são considerados dados reais, não é possível saber se há ou não a presença de *concept drift* e qual a sua frequência, gradual ou abrupta. Os resultados mostraram que os Cenários 1 e 2 foram os que conseguiram ajustar melhor o sistema diante das mudanças de conceito, tanto para os dados artificiais, quanto para os reais. Além disso, a maioria dos resultados são estatisticamente significativos, exceto em dois casos.

Em trabalhos futuros pretendemos avaliar a junção da utilidade das regras com outras características, como por exemplo a idade, para definir um critério mais preciso na escolha de regras com alto potencial de representação de conhecimento significativo, mesmo após uma mudança de conceito.

Referências

- Agrahari, S. and Singh, A. K. (2022). Concept drift detection in data stream mining: A literature review. *Journal of King Saud University-Computer and Information Sciences*, 34(10):9523–9540.
- Alshalali, T. and Josyula, D. (2020). Evaluating extreme learning machine models in the presence of concept drift in streaming data. In *Proceedings of the 2020 the 4th International Conference on Information System and Data Mining*, pages 107–112.

- Angelov, P. and Zhou, X. (2008). On line learning fuzzy rule-based system structure from data streams. In *2008 IEEE international conference on fuzzy systems (IEEE world congress on computational intelligence)*, pages 915–922. IEEE.
- Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., and Gavalda, R. (2009). New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 139–148.
- Brzezinski, D. and Stefanowski, J. (2013). Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE transactions on neural networks and learning systems*, 25(1):81–94.
- Chagas, P., Souza, L., Araújo, I., Aldeman, N., Duarte, A., Angelo, M., dos Santos, W. L., and Oliveira, L. (2020). Classification of glomerular hypercellularity using convolutional features and support vector machine. *Artificial Intelligence in Medicine*, 103:101808.
- Charizanos, G., Demirhan, H., and İcen, D. (2024). An online fuzzy fraud detection framework for credit card transactions. *Expert Systems with Applications*, 252:124127.
- Emam, M. M., Houssein, E. H., Samee, N. A., Alohal, M. A., and Hosney, M. E. (2024). Breast cancer diagnosis using optimized deep convolutional neural network based on transfer learning technique and improved coati optimization algorithm. *Expert Systems with Applications*, 255:124581.
- Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874. ROC Analysis in Pattern Recognition.
- Frias-Blanco, I., del Campo-Ávila, J., Ramos-Jimenez, G., Morales-Bueno, R., Ortiz-Diaz, A., and Caballero-Mota, Y. (2014). Online and non-parametric drift detection methods based on hoeffding’s bounds. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):810–823.
- Gama, J., Źliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37.
- Gonçalves Júnior, P. M. (2013). *Multivariate Non-Parametric Statistical Tests to Reuse Classifiers in Recurring Concept Drifting Environments*. PhD thesis, Universidade Federal de Pernambuco, Recife, Brasil.
- Lu, J., Ma, G., and Zhang, G. (2024). Fuzzy machine learning: A comprehensive framework and systematic review. *IEEE Transactions on Fuzzy Systems*, 32(7):3861–3878.
- Shahparast, H., Jahromi, M. Z., Taheri, M., and Hamzeloo, S. (2014). A novel weight adjustment method for handling concept-drift in data stream classification. *Arabian Journal for Science and Engineering*, 39:799–807.
- Sun, Y., Tang, K., Zhu, Z., and Yao, X. (2018). Concept drift adaptation by exploiting historical knowledge. *IEEE transactions on neural networks and learning systems*, 29(10):4822–4832.
- Tan, Q., Fu, M., Wang, Z., Yuan, H., and Sun, J. (2024). A real-time early warning classification method for natural gas leakage based on random forest. *Reliability Engineering & System Safety*, 251:110372.

- University of Waikato (2025). MOA Datasets. <https://moa.cms.waikato.ac.nz/datasets/>.
- Varshney, A. K. and Torra, V. (2023). Literature review of the recent trends and applications in various fuzzy rule-based systems. *International Journal of Fuzzy Systems*, 25(6):2163–2186.
- Wang, L.-X. and Mendel, J. (1992). Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6):1414–1427.