CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

CLASSIFICAÇÃO E RECONHECIMENTO DE PADRÕES: NOVOS ALGORITMOS E APLICAÇÕES

EMMANUEL TAVARES FERREIRA AFFONSO

Orientador: Prof. Alisson Marques da Silva
CEFET-MG

Coorientador: Prof. Gray Farias Moita
CEFET-MG

BELO HORIZONTE
FEVEREIRO DE 2023

EMMANUEL TAVARES FERREIRA AFFONSO

CLASSIFICAÇÃO E RECONHECIMENTO DE PADRÕES: NOVOS ALGORITMOS E APLICAÇÕES

Tese apresentada ao Programa de Pós-graduação em Modelagem Matemática e Computacional do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Doutor em Modelagem Matemática e Computacional.

Área de concentração: Modelagem Matemática e Computacional

Linha de pesquisa: Sistemas Inteligentes

Orientador: Prof. Alisson Marques da Silva
CEFET-MG

Coorientador: Prof. Gray Farias Moita
CEFET-MG

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL
BELO HORIZONTE
FEVEREIRO DE 2023

Affonso, Emmanuel Tavares Ferreira
A257c Classificação e reconhecimento de padrões: novos algoritmos e aplicações / Emmanuel Tavares Ferreira Affonso. – 2023.
94 f.

Tese de doutorado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional.
Orientador: Alisson Marques da Silva.
Coorientador: Gray Farias Moita.
Tese (doutorado) – Centro Federal de Educação Tecnológica de Minas Gerais.

1. Reconhecimento de padrões – Classificação – Teses. 2. Sistemas difusos – Teses. 3. Algoritmos – Teses. 4. Computação evolutiva – Teses. 5. Inteligência computacional – Teses. I. Silva, Alisson Marques da. II. Moita, Gray Farias. III. Centro Federal de Educação Tecnológica de Minas Gerais. IV. Título.

CDD 004.0151

Elaboração da ficha catalográfica pela bibliotecária Jane Marangon Duarte, CRB 6º 1592 / Cefet/MG

SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
COORDENAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

"CLASSIFICAÇÃO E RECONHECIMENTO DE PADRÕES: NOVOS ALGORITMOS E APLICAÇÕES".

Tese de Doutorado apresentada por **Emmanuel Tavares Ferreira Affonso**, em 28 de fevereiro de 2022, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:

Prof. Dr. Alisson Marques da Silva
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Gray Farias Moita (Coorientador)
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Maurílio José Inácio
Universidade Estadual de Montes Claros

Prof. Dr. Daniel Furtado Leite
Adolfo Ibanez University

Profª. Drª. Elizabeth Fialho Wanner
Centro Federal de Educação Tecnológica de Minas Gerais

Profª. Drª. Elisângela Martins de Sá
Centro Federal de Educação Tecnológica de Minas Gerais

Visto e permitida a impressão,

Prof. Dr. Thiago de Souza Rodrigues
Coordenador do Programa de Pós-Graduação Stricto Sensu em
Modelagem Matemática e Computacional

Dedico esta tese primeiramente a Deus, por ter guiado meu destino e permitir que tudo isso seja possível. Minha esposa e minha mãe que me deram todo apoio e ajuda necessária para conquistar meus objetivos.

Agradecimentos

Agradeço a Deus por ter me dado saúde e forças para superar todas as dificuldades, ter permitido que tudo acontecesse na hora certa em minha vida. Agradeço minha esposa que esteve sempre presente, encorajando, entendendo e sempre do meu lado. Meus pais por sempre terem me apoiado. Minha família pela força e incentivo nesta luta. Meus amigos que nunca saíram do meu lado. Agradeço também aos meus orientadores pelos valiosos ensinamentos e dedicação, ao CEFET-MG pelo apoio financeiro e oportunidade de estudo.

Uma mente que se abre a uma nova ideia jamais volta ao seu tamanho original. (Albert Einstein)

Resumo

Este trabalho visa o desenvolvimento de novos algoritmos para tarefas de classificação e reconhecimento de padrões. Inicialmente é proposto um método para seleção de atributos (FS, do inglês *feature selection*) para classificadores com treinamento *offline*. A abordagem de FS proposta, chamada de *Mean Ratio for Feature Selection* (MRFS), é baseada na razão das médias dos atributos em cada classe. MRFS possui baixo custo computacional pois utiliza apenas operações matemáticas básicas como adição, divisão e comparação e realiza apenas uma passagem nos dados para ranquear os atributos. O MRFS foi implementado e avaliado como método *filter* (MRFS-F) e como *wrapper* (MRFS-W). O desempenho dos métodos foram avaliados e comparados com o estado da arte. As comparações realizadas sugerem que os métodos propostos possuem um desempenho comparável ou superior aos métodos alternativos. Além dos métodos de FS foram propostos três classificadores evolutivos com treinamento *online*. O primeiro chamado de *evolving Fuzzy Mean Classifier* (eFMC) é baseado em um algoritmo de agrupamento *fuzzy* que realiza a classificação com base no grau de pertinência dos grupos. Novos grupos são criados sempre que uma nova classe é descoberta e a atualização do centro dos grupos é através das médias amostrais calculadas de forma incremental. O segundo classificador introduzido é o *evolving Fuzzy Classifier* (eFC) que de maneira análoga utiliza as médias amostrais para atualização dos centros dos grupos. Seu diferencial está na capacidade de gerar mais de um grupo associado a uma mesma classe para mapear diferentes regiões do espaço dos dados, criação de novos grupos aplicando o conceito de procrastinação, união de grupos redundantes e exclusão de grupos obsoletos. Por fim, foi proposto um classificador evolutivo com seleção de atributos denominado *evolving Fuzzy Classifier with Feature Selection* (eFC-FS). Este classificador foi construído utilizando o mesmo algoritmo do eFC e incorporando o método MRFS de seleção de atributos. Os classificadores evolutivos propostos foram avaliados e comparados com três classificadores evolutivos alternativos. Os resultados experimentais e as comparações sugerem que os métodos baseados no MRFS para auxiliar modelos com treinamento *offline* e os 3 modelos evolutivos com treinamento *online* são promissores como alternativas para tarefas de classificação e reconhecimento de padrões, com boa acurácia e baixo custo computacional.

Palavras-chave: Seleção de Atributos; Sistemas *Fuzzy*; Sistemas *Fuzzy* Evolutivos; Classificação; Inteligência Computacional.

Abstract

This thesis aims the development of new algorithms for classification and pattern recognition tasks. Initially, a method for feature selection for classifiers with offline training is proposed. The proposed FS approach, called Mean Ratio for Feature Selection (MRFS), is based on each a per-class ratio considering feature averages. MRFS has a low computational cost due to the use of basic mathematical operations such as addition, division, and comparison. Moreover, it performs a single pass over the data to rank the attributes. MRFS was implemented and evaluated as a filter method (MRFS-F) and a wrapper (MRFS-W). The performance of the methods is evaluated and compared with that of state of the art methods. The comparisons suggest that the proposed methods have comparable or superior performance to alternative methods. In addition to the FS methods, three multiclass evolving classifiers with online training were proposed. The first, called the evolving Fuzzy Mean Classifier (eFMC), is a fuzzy clustering model that performs classification based on the degree of membership between the sample and the groups. New groups are created whenever a new class is discovered, and the center of the groups is updated through the sample means calculated incrementally. The second classifier introduced is the evolving Fuzzy Classifier (eFC). Similarly, eFC uses the sample means to update the centers of the groups. Its odds lie in the ability to generate more than one group associated with the same class to map different regions of the data space, create new groups applying the concept of procrastination, ability to merge redundant groups, and delete obsolete groups. Finally, an evolving classifier with feature selection called evolving Fuzzy Classifier with Feature Selection (eFC-FS) was proposed. This classifier was built using the same algorithm as the eFC and incorporating the MRFS feature selection method. The proposed classifiers were evaluated and compared with three alternative evolving classifiers. The experimental results and comparisons suggest that the three evolving models with online training are promising alternatives for classification and pattern recognition tasks. They have shown good accuracy and low computational cost.

Keywords: Feature Selection; *Fuzzy* System; Evolving *Fuzzy* System; Classification; Computational Intelligence.

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Abreviaturas e Siglas

ABC	*Artificial Bee Colony*
ACO	*Ant Colony Optimization*
ALMMo	*Autonomous Learning Multi-Model*
CFS	*Correlation-based Feature Selection*
DE	*Differential Evolution*
DENFIS	*Dynamic Evolving Neural-Fuzzy Inference System*
DISR	*Double Input Symmetrical Relevance*
DT	*Decision Tree*
EFS	*Evolving Fuzzy Systems*
eFuMo	*Evolving Fuzzy Model*
eGNN	*Evolving Granular Neural Network*
eHFIS	*Evolving Heterogeneous Fuzzy Inference System*
eFMC	*Evolving Fuzzy Mean Classifier*
eMG	*Evolving Multivariable Gaussian Fuzzy System*
eNFN	*Evolving Neo-Fuzzy Neuron*
eNFN-AFS	*Evolving Neo-Fuzzy Neural Network with Adaptive Feature Selection*
eT2Class	*Evolving Type-2 Classifier*
eT2FIS	*Evolving Type-2 Neural Fuzzy Inference System*
eT2RFNN	*Evolving Type-2 Fecurrent Fuzzy Neural Network*
eTS	*Evolving Takagi-Sugeno*
EUFSFC	*Unsupervised Feature Selection Method Through Feature Clustering*
eVQ	*Evolving Vector Quantization*
FA	*Firefly Algorithm*

FBeM *Fuzzy-set-Based evolving Modeling*

FCBF *Fast Correlation-Based Filter*

FCM *Fuzzy C-Means*

FLEXFIS *Flexible Evolving Fuzzy Inference Systems*

FS *Feature Selection*

F-ERA *Fuzzy Eigen-System Realization Algorithm*

GA Genetic Algorithms (algoritmos genéticos)

Gen-Smart-EFS *Generalized Smart Evolving Fuzzy Systems*

GK *Gustafson-Kessel clustering*

GMDH *Group Method of Data Handling*

GRASP *Greedy Randomized Adaptive Search Procedure*

GS-FIS *Group Sparse Fuzzy Inference Systems*

HS *Harmony Search*

IBeM *Interval-Based evolving Modeling*

IDFS *Interaction Dominance Based Feature Selection*

IT2NFS-SIFE *Interval Type-2 Neural Fuzzy System for Online System Identification and Feature Elimination*

JMI *Joint Mutual Information*

KNN *K-Nearest Neighbors*

LLCFS *Local Learning-based Clustering Feature Selection*

LSFS *Laplacian Score Feature Selection*

MCFS *Multi-Cluster Feature Selection*

MFFS *Multi-Filter Feature Selection Approach*

MIFS *Mutual Information Based Feature Selection*

MLP *Multi-Layer Perceptron*

mRmR *Minimum-Redundancy-Maximum-Relevance*

NB	*Naïve Bayes*
NDFS	*Nonnegative Discriminative Feature Selection*
NFN	*Neo-Fuzzy Neuron*
NFRS	*Heuristic Algorithm Based on Fitting Fuzzy Rough Sets*
OBSSO	*Opposition-based Social Spider Optimization*
OEWSC	*Online Entropy Weighting Subspace Clustering*
OFWSC	*Online Fuzzy Weighting Subspace Clustering*
OIA	*Online Identification Algorithm*
PAC	*Parsimonious Autonomous Controller*
PALM	*Parsimonious Autonomous Machine Learning*
PCA	*Principal Component Analysis*
PCM	*Possibilistic C-Means*
pENsemble	*Parsimonious Ensemble*
PSO	*Particle Swarm Optimization*
RF	*Random Forest*
RFS	*Robust Supervised Feature Selection*
RNA	Redes Neurais Artificiais
RUFS	*Robust Unsupervised Feature Selection*
SCA	*Sine-Cosine Algorithm*
SEFS	*Self-Evolving Fuzzy System*
SEIT2FNN	*Self-Evolving Interval Type-2 Fuzzy Neural Network*
SFC	*Subspace Feature Clustering*
SFR	*Stratified Feature Ranking*
SFS	*Sequential Forward Selection*
SMC	*Sliding Mode Controller*
SSSC-E	*Entropy Weighting Streaming Soft Subspace Clustering*

SSSC-F	*Fuzzy Weighting Streaming Soft Subspace Clustering*
SSO	*Social Spider Optimization*
SVM	*Support Vector Machine*
TS	*Takagi-Sugeno*
VFF-RLS	*Recursive Leastsquare Algorithm with Variable Forgetting Facto*
VQ	*Vector Quantization*
VSRF	*Variable Selection using Random Forest*
VTOL	*Vertical Takeoff and Landing*
xTS	*eXtended Takagi-Sugeno*

Lista de Símbolos

H	hipótese para um teste estatístico
u	média para um teste de hipótese
α	significância para um teste de hipótese
$\bar{X}$	média de um conjunto amostral de dados
$\S$	valor de aceitação do teste estatístico para o nível de significância
S	conjunto de médias de diferentes atributos
$\mathbf{T}$	número de amostras de um conjunto de dados
t	amostra atual
Γ	razão entre duas médias
R	regra *fuzzy*
i	índice dos grupos e regras ativas
l	número de grupos e regras ativas
x	vetor de atributos de entrada
C	grupos ativos
y_i	consequente da i-ésima regra
j	índice do atributo de entrada
m	número de atributos de entrada
c	matriz com os centros dos grupos ativos
s	matriz de somatório dos grupos ativos
N_i	número de amostras atribuídas ao i-ésimo grupo
$\mu_i(x^t)$	grau de pertinência da amostra x^t no i-ésimo grupo
d	distância Euclidiana
f	parâmetro de fuzzificação

K	número de classes
k	índice das classe
i_k^*	índice do grupo com o maior grau de compatibilidade da k-ésima classe
k^*	índice da classe com maior grau de compatibilidade
y^t	classe desejada
$\hat{y}^t$	classe de saída (prevista)
i^*	índice do grupo ativo com o maior grau de compatibilidade
G	número de grupos inativos
ci	matriz com os centros dos grupos inativos
si	matriz de somatório dos grupos inativos
z_p	classe da amostra que criou o p-ésimo grupo inativo
p	índice dos grupos inativos
p^*	índice do grupo inativo com o maior grau de compatibilidade
$i^\bullet$	índice do primeiro grupo a ser unido
$i^\diamond$	índice do segundo grupo a ser unido
i^+	índice do grupo mais velho
ω	limite de idade para exclusão de um grupo
n_k	número de grupos da k-ésima classe
$\bar{M}$	média resultante dos grupos ativos
$j^\diamond$	atributo candidato ao bloqueio
$j^\bullet$	atributo candidato ao desbloqueio
Φ	valor da razão entre os centros dos grupos
ϵ	vetor de índices dos atributos bloqueados

Sumário

Capítulo 1

Introdução

Neste capítulo inicial, serão apresentados alguns aspectos críticos encontrados na aplicação de algoritmos computacionais em tarefas de classificação e reconhecimento de padrões que motivam e mostram a relevância desta pesquisa. A Seção 1.1 apresenta a motivação para o desenvolvimento deste trabalho e a sua relevância. Em seguida, a Seção 1.2 ilustra os objetivos gerais e os específicos a serem alcançados e a Seção 1.3 apresenta, de forma sumarizada, as principais contribuições alcançadas por este estudo. A Seção 1.4 destaca os artigos publicados e os em desenvolvimento frutos desta pesquisa. Por fim, a Seção 1.5 descreve como o restante do trabalho está organizado.

1.1 Motivação e relevância

A aplicação de técnicas de inteligência computacional em tarefas de classificação e reconhecimento de padrões requer algoritmos que sejam capazes de processar diferentes tipos de dados, identificar as características inerentes a cada classe e realizar corretamente a classificação. Entretanto, alguns fatores podem impactar e, consequentemente, dificultar a atuação dos classificadores. Dentre esses fatores, um de grande relevância é a qualidade dos dados. Fatores associados à baixa qualidade dos dados incluem conjuntos de dados com grande número de atributos não representativos e altamente correlacionados, dados ruidosos e conjuntos de dados com classes desbalanceadas (Naik; Kuppili; Edla, 2019; Gauthama Raman et al., 2019).

A seleção de atributos (FS, do inglês *feature selection*) é uma das técnicas utilizadas para minimizar os problemas ocasionados pelo elevado número de atributos, e pela correlação entre os atributos. As técnicas de FS visam fornecer subconjuntos menores e mais representativos de atributos, isolando e extraindo apenas os atributos relevantes para o processo de treinamento do classificador. O objetivo é otimizar o processo de classificação, aumentando a acurácia dos classificadores e reduzindo seu tempo de execução (Blum; Langley, 1997; Guyon; Elisseeff, 2003; Vora; Yang, 2018).

Nos métodos de seleção de atributos, o processo para avaliar quais atributos apresentam maior representatividade a um classificador, são categorizados por métodos *filter*, *wrapper* e *embedded*. Nos métodos *filter* o processamento e ranqueamento dos atributos é realizado antes da etapa de treinamento. Por outro lado, nos métodos *wrapper* e *embedded* a seleção de atributos ocorre junto à etapa de treinamento. Os métodos de seleção de atributos serão melhores descritos nas próximas seções.

Em algumas tarefas de classificação, estão disponíveis conjuntos de dados completos para seleção dos atributos e para o treinamento dos classificadores. Nestas, os classificadores podem executar vários passos sobre os mesmos dados, chamados de épocas de treinamento, visando melhorar sua acurácia por meio do aprendizado de parâmetros mais apropriados. Neste contexto, diz-se que o aprendizado é *offline*.

Por outro lado, em algumas tarefas, as amostras são disponibilizadas uma a uma, na forma de um fluxo contínuo. Os dados geralmente provêm de distribuições de probabilidades variantes no tempo. Essa variação altera o conceito dos dados, variações graduais (*drift*) ou abruptas (*shift*) de distribuições de probabilidade são comuns em fluxo de dados, o que tende a confundir o sistema de classificação com treinamento *offline* (Leite et al., 2020). Nestes casos, é necessário realizar o retreinamento completo dos classificadores, o que nem sempre é viável, já que os dados são descartados após serem processados. Para suprir essa demanda, surgiu um campo de pesquisa emergente chamado de Sistemas Inteligentes Evolutivos, nos quais o treinamento dos parâmetros e também da estrutura do modelo é *online*, isto é, o treinamento é um processo incremental e contínuo. Os sistemas *fuzzy* com essas características são denominados sistemas *fuzzy* evolutivos (EFS, do inglês *Evolving Fuzzy Systems*) (Lughofer, Edwin, 2011; Edwin Lughofer, 2022).

Os EFS possuem estrutura flexível que se expande ou contrai à medida que novas informações são disponibilizadas. Sua estrutura normalmente é composta por grupos, neurônios, grânulos, funções de pertinência ou nuvem de dados (Skrjanc et al., 2019; Leite; Skrjanc; Gomide, 2020) e seus componentes podem ser incluídos, excluídos, divididos, unidos ou ter seus parâmetros alterados por um algoritmo de treinamento incremental. Estes algoritmos possuem um mecanismo de treinamento no qual a amostra é processada uma única vez, i.e., após o processamento e a obtenção das informações a amostra é descartada (Silva et al., 2013a).

Diante do exposto, o presente trabalho visa atuar em duas frentes. A primeira tendo como foco a proposição e a implementação de métodos de seleção de atributos para classificadores com treinamento *offline* e, na segunda, objetiva-se a introdução de novos classificadores com treinamento *online*, ou seja, classificadores evolutivos.

1.2 Objetivos gerais e específicos

O objetivo geral deste trabalho é propor e implementar métodos de seleção de atributos e classificadores evolutivos multiclasse, por meio de algoritmos eficientes, simples, com baixo custo computacional e com alta precisão.

Os objetivos específicos são:

- Desenvolver um método de seleção de atributos para classificadores *offline* baseado em equações matemáticas simples, com bom desempenho e com capacidade de selecionar atributos que possibilitem a melhora na acurácia dos classificadores. Deseja-se métodos flexíveis que possam ser utilizados como *filter* ou incorporados aos classificadores como métodos *wrapper*.
- Desenvolver classificadores evolutivos multiclasse que tenham baixo custo computacional, alto grau de autonomia e que sejam capazes de adaptar sua estrutura e manter uma boa classificação no advento de mudanças de comportamento e surgimento de novidades em um fluxo de dados contínuo.
- Desenvolver um classificador evolutivo multiclasse com seleção de atributos com baixo custo computacional e autonomia, no qual o processo de seleção de atributos seja incorporado ao algoritmo de treinamento sem causar descontinuidade ou danos no processo de aprendizagem.

1.3 Contribuições da pesquisa

As contribuições desta pesquisa estão no desenvolvimento de métodos de seleção de atributos e sistemas evolutivos para tarefas de classificação e reconhecimento de padrões. A seguir, as principais contribuições são sumarizadas.

- Proposição de um método de seleção de atributos utilizando a razão das médias. O método proposto, chamado *Mean Ratio for Feature Selection* (MRFS), é composto por equações simples e operações matemáticas básicas.
- Incorporação do MRFS no processo de treinamento de diferentes classificadores para seleção de atributos de maneira automática. Sem a necessidade de escolha prévia da quantidade de atributos a serem removidos.
- Proposição do *evolving Fuzzy Mean Classifier* (eFMC), que é um classificador evolutivo multiclasse com baixo custo computacional e que faz uso somente de equações matemáticas básicas como soma, adição, multiplicação e subtração. O eFMC usa a média amostral dos atributos para construção incremental de sua estrutura e cada classe é representada por um único grupo.
- Proposição de um classificador evolutivo multiclasse chamado *evolving Fuzzy Classi-*

fier (eFC). O eFC possui uma estrutura similar ao eFMC, no entanto neste classificador uma classe pode ser representada por vários grupos.
- Proposição do *evolving Fuzzy Classifier with Feature Selection* (eFC-FS), um classificador evolutivo multiclasse com o método de seleção de atributos MRFS incorporado ao seu algoritmo de treinamento incremental.

1.4 Publicações

Esta seção apresenta os artigos frutos desta pesquisa. Em resumo, foram publicados dois artigos em periódico e um artigo em evento nacional. Além disso, três artigos estão em desenvolvimento e serão submetidos para possível publicação em periódico.

1.4.1 Artigos publicados em periódico

O primeiro artigo ilustra a implementação e a avaliação do método de seleção de atributos MRFS *filter* e foi publicado no *Intelligent Decision Technologies*. O segundo artigo foi publicado no *Journal of Intelligent & Fuzzy Systems* e destaca a proposição e a avaliação do desempenho do classificador evolutivo eFMC.

- Tavares, E.; Silva, A. M.; Moita, G. F.; Cardoso, R. T. N. *A straightforward feature selection method based on mean ratio for classifiers. Intelligent Decision Technologies*, v. 15, n. 3, p. 421-432, 2021. Disponível em: <https://doi.org/10.3233/IDT-200186>.
- Tavares, E.; Silva, A. M.; Moita, G. F. *A fast clustering algorithm for evolving fuzzy classifier based on samples mean. Journal of Intelligent & Fuzzy Systems, IOS Press,* v. 43, n. 6, p. 6897–6908, 7 2022. Disponível em: <https://doi.org/10.3233/JIFS-212831>.

1.4.2 Artigo publicado em congresso

O artigo publicado no Congresso Brasileiro de Inteligência Computacional ilustra a implementação do MRFS como método *wrapper* incorporado ao *Fuzzy c-Means*.

- Tavares, E. F. A.; Santos, G. M. dos; Silva, A. M. D.; Moita, G. F. *Fuzzy c-Means* com método *wrapper* com baixo custo computacional de seleção de atributos. Anais do 15 Congresso Brasileiro de Inteligência Computacional. Joinville, SC: SBIC, 2021. p. 1–7. Disponível em: <https://doi.org/10.21528/CBIC2021-87>.

1.4.3 Artigos em desenvolvimento

O primeiro artigo em desenvolvimento trata da implementação do MRFS como método *wrapper* de seleção de atributos incorporado aos classificadores KNN, SVM, NB e RNA. No segundo é apresentado o eFC, modelo evolutivo capaz de mapear diversas regiões no

espaço dos dados, de maneira efetiva e com baixo custo computacional. No terceiro artigo é apresentado o eFC-FS, modelo evolutivo baseado no eFC com a capacidade de seleção de atributos do MRFS.

- Tavares, E.; Silva, A. M.; Moita, G. F. *A effective feature selection wrapper method for classification algorithms*.
- Tavares, E.; Silva, A. M.; Moita, G. F. *A simple and efficient evolving fuzzy system for binary and multiclass classification tasks in a data stream context*.
- Tavares, E.; Silva, A. M.; Moita, G. F.*evolving Fuzzy Classifier with Feature Selection applied to online environments*.

1.5 Organização do trabalho

O restante deste trabalho está organizado conforme:

- O Capítulo 2 apresenta os conceitos fundamentais de seleção de atributos e sistemas *fuzzy* evolutivos. Além disso, esse capítulo traz uma revisão da literatura sobre esses dois temas.
- O Capítulo 3 introduz o método de seleção de atributos *Mean Ratio for Feature Selection* (MRFS) e ilustra sua aplicação como método *filter* e *wrapper*.
- O Capítulo 4 descreve os experimentos computacionais e os resultados obtidos na avaliação do desempenho do MRFS. Foram realizados experimentos para avaliar o MRFS como método *filter* e *wrapper*.
- O Capítulo 5 destaca o primeiro classificador evolutivo proposto neste trabalho, denominado *evolving Fuzzy Mean Classifier* (eFMC).
- O Capítulo 6 detalha o segundo classificador introduzido nesta pesquisa e chamado *evolving Fuzzy Classifier* (eFC).
- O Capítulo 7 ilustra a incorporação do método de seleção de atributos MRFS no classificador eFC. Esse novo classificador recebe o nome de *evolving Fuzzy Classifier with Feature Selection* (eFC-FS).
- O Capítulo 8 trata dos experimentos computacionais, resultados e comparações realizados para avaliar o desempenho dos classificadores evolutivos propostos. Os classificadores são avaliados e comparados pela acurácia e pelo tempo de processamento. Além disso, este capítulo avalia a escalabilidade dos classificadores propostos.
- O Capítulo 9 sumariza as principais contribuições e indica propostas de continuidade desta pesquisa.
- O Apêndice A lista os atributos utilizados nos experimentos do MRFS como método *filter*.
- O Apêndice B mostra os resultados dos experimentos do MRFS como método *wrapper*.

Capítulo 2

Fundamentos e revisão da literatura

Neste capítulo apresenta-se a conceituação e uma breve revisão da literatura sobre os métodos de seleção de atributos e os sistemas *fuzzy* evolutivos, que são as duas temáticas principais deste trabalho. A Seção 2.1 introduz os conceitos fundamentais sobre os métodos de seleção de atributos e, em seguida, apresenta alguns dos principais métodos. Os conceitos e as características dos sistemas evolutivos são descritos na Seção 2.2, bem como os modelos evolutivos. A Seção 2.3 traz o estado da arte em modelos evolutivos com seleção de atributos. Por fim, a Seção 2.4 apresenta um fechamento caracterizando o que foi concluído a partir da revisão realizada das três diferentes metodologias apresentadas.

2.1 Seleção de atributos

Um dos grandes desafios na aplicação de técnicas de inteligência computacional em tarefas de classificação e reconhecimento de padrões é a identificação dos melhores atributos de entrada, ou seja, encontrar um conjunto de atributos que melhor represente as classes e possibilite uma maior acurácia na classificação (Guyon; Elisseeff, 2003). Os métodos de extração de atributos (FE, do inglês *Feature Extraction)* e seleção de atributos (FS, do inglês *Feature Selection*) são os principais meios para identificar um conjunto de atributos de maior representatividade.

Os métodos de FE têm como objetivo reduzir a dimensão do espaço de entrada por meio da combinação de atributos, i.e., combina atributos buscando representações em menores dimensões. Os atributos de entrada são combinados por funções lineares ou não lineares para gerar novos atributos. Análise de Componentes Principais (PCA, do inglês *Principal Component Analysis*) é um exemplo de método linear. No espaço dos atributos de entrada, o PCA encontra os autovetores de uma matriz de covariância relacionados aos mais altos autovalores e esses autovalores são utilizados para projetar matrizes de menores dimensões (Kang; Tian, 2018). Essa técnica é muito empregada para redução de dimensionalidade em tarefas de análise de imagem e em conjuntos de dados com elevado

número de atributos (Hsieh et al., 2015; Li et al., 2018). Dentre os métodos não lineares, destacam-se o Mapeamento de Atributos Isométricos (ISOMAP, do inglês *Isometric Feature Mapping*) e o Incorporação Localmente Linear (LLE, do inglês *Locally Linear Embedding*). O ISOMAP cria uma matriz de distância entre todos os pontos e calcula a posição de cada ponto. Em um espaço de entrada, a distância pode ser assumida pela distância direta entre esses pontos; o ISOMAP usa a técnica *Multi-Dimension Scaling* (MDS) para calcular a posição dimensional reduzida dos pontos (Li; Cai; He, 2017; Han; Cheng; Hou, 2018). O LLE, de modo semelhante ao ISOMAP, busca os vizinhos mais próximos de cada ponto e assume a matriz de pesos dos seus vizinhos como uma combinação linear para descrever o ponto, de modo que o ponto ainda seja descrito por essa combinação linear dos vizinhos (Meng et al., 2016; Li et al., 2018).

Por outro lado, os métodos de seleção de atributos visam ranquear e selecionar atributos de acordo com sua relevância e geralmente são baseados em métodos estatísticos ou heurísticos (Wang; Luan, 2019). Os métodos baseados em estatística utilizam análise de correlações, variância, redundância ou distância entre amostras para estabelecer o *ranking* de relevância dos atributos (Kang; Tian, 2018; Vora; Yang, 2018). Entre os métodos estatísticos, pode-se citar *Chi-Square*, Teste T, *Gini Index*, *Information Gain*, Kruskal-Wallis e *Fisher Score* (Anitha; Sherly, 2021). Nos métodos heurísticos de FS, o objetivo é encontrar um subconjunto de atributos que apresente um melhor desempenho por meio da avaliação de diferentes combinações de atributos. Os métodos heurísticos comumente são baseados em computação evolucionária (Umbarkar; Shukla, 2018; Suchetha; Nikhil; Hrudya, 2019) como, por exemplo, Algoritmos Genéticos (GA, do inglês *Genetic Algorithm*), Otimização por Enxame de Partículas (PSO, do inglês *Particle Swarm Optimization*), Otimização da Colônia de Formigas (ACO, do inglês *Ant Colony Optimization*) e Evolução Diferencial (DE, do inglês *Differential Evolution*).

Em geral, entre estatísticos e heurísticos, os métodos de FS podem ser classificados em:

- **Filtro (*filter*):** nos métodos *filter*, a seleção de atributos é realizada *a priori*, por meio de heurística ou análise estatística, e visa ranquear atributos de entrada de acordo com sua relevância. É importante destacar que nos métodos *filter*, a seleção de atributos é executada sem dependência do classificador.
- **Empacotamento (*wrapper*):** os métodos *wrapper* constroem diversos subconjuntos de atributos e os testam em um classificador, escolhendo o subconjunto que apresenta o melhor desempenho no classificador. Em outras palavras, nos métodos *wrapper*, os atributos são selecionados de acordo com seu desempenho em um determinado classificador.
- **Embarcados (*embedded*):** nos métodos *embedded* o processo de seleção é parte interna e natural do algoritmo de aprendizado do classificador. Nestes métodos, os

conjuntos de atributos são avaliados à medida que novas amostras são apresentadas.

Após essa breve conceituação sobre extração e seleção de atributos, apresenta-se uma breve revisão da literatura sobre os métodos de seleção de atributos. Nesta revisão destacam-se métodos heurísticos e estatísticos, tendo como foco principal os métodos estatísticos. Os métodos são apresentados em ordem cronológica, iniciando nos mais antigos e seguindo para os mais recentes.

Wang et al. (2017) propôs um método de FS heurístico para tarefas de classificação chamado *Heuristic Algorithm based on Fitting Fuzzy Rough Sets* (NFRS). No NFRS os atributos são avaliados pela sua significância em relação à vizinhança. Para isso, utiliza-se uma tabela de decisão para calcular o incremento da capacidade de classificação de cada atributo. A medida da significância proposta é introduzida no algoritmo NFRS que indica os atributos de maior relevância. Esse processo se repete enquanto houver adição de novos atributos.

Em Vora e Yang (2018) foi apresentado um *software* que possibilita a execução de onze métodos *filter* de seleção de atributos. Destes, 10 são métodos estatísticos: *Chi-Square*, *Information Gain*, *Fisher Score*, *Gini Index*, *Kruskal-Wallis*, *Laplacian Score*, *ReliefF*, *Fast Correlation-Based Filter* (FCBF), *Correlation-based Feature Selection* (CFS), e *Minimum-Redundancy-Maximum-Relevance* (mRMR); e um método é heurístico: VSRF (*Variable Selection using Random Forest*). Os classificadores empregados foram *Support Vector Machine* (SVM), *K-Nearest Neighbors* (KNN), *Random Forest* (RF), *Naïve Bayes* (NB) e *Decision Tree* (DT).

Chen et al. (2018) introduz o *Stratified Feature Ranking* (SFR). O SFR, é um método *filter* no qual inicialmente executa-se o algoritmo de agrupamento *Subspace Feature Clustering* (SFC). No SFC, os atributos são agrupados de acordo com sua relevância em cada classe. Em seguida, o SFR analisa os grupos e classifica os atributos de acordo com o peso produzido pelo SFC. Os experimentos foram executados comparando o SFR com os métodos ReliefF, *Robust Supervised Feature Selection* (RFS), mRmR, *Fisher Score*, *SVM-Recursive Feature Elimination-Correlation Bias Reduction* (SVM-RFE-CBR) e *Uncorrelated Group Lasso* (UGL). Os atributos selecionados pelos métodos foram avaliados no classificador SVM.

No trabalho de Ibrahim et al. (2019) é proposto um modelo heurístico chamado *Opposition-based Social Spider Optimization* (OBSSO). Este método foi implementado no KNN e na RF. Os resultados obtidos pelo OBSSO foram comparados com seis outros métodos heurísticos, como *Particle Swarm Optimization* (PSO), *Harmony Search* (HS), *Artificial Bee Colony* (ABC), *Firefly Algorithm* (FA), *Social Spider Optimization* (SSO) e *Sine-Cosine Algorithm* (SCA).

O *Interaction Dominance based Feature Selection* (IDFS) (Zeng; Heng, 2019) é um método *filter* que utilizada *Mutual Information* (MI) e *Three-way Interaction Information* (TWII) para o ranqueamento dos atributos. MI é utilizada para medir a relevância dos atributos associados às classes e TWII para calcular a redundância dos atributos. Após o cálculo da relevância e redundância dos atributos, o IDFS utiliza *Greedy Search* (GS) para ranquear os atributos. No referido trabalho, o IDFS é comparado com *Mutual Information based Feature Selection* (MIFS), mRmR, *Joint Mutual Information* (JMI) e *Double Input Symmetrical Relevance* (DISR).

Em Haq et al. (2019) são propostas duas abordagens *filter* com o método nomeado *Multi-Filter Feature Selection Approach* (MFFS). Para ambos os métodos *filter*, inicialmente são utilizados algoritmos de inteligência computacional como *Logistic Regression*, SVM e RF para filtrar entradas mais significativas e computar sua relevância. Em seguida, são utilizados os métodos MFFS-lr e MFFS-IG nas entradas restantes para medir o nível de associação entre elas e então selecionar o conjunto dos melhores atributos de entrada. Os métodos com MFFS-IG e MFFS-lr tem seu desempenho comparados com as técnicas *Information Gain*, *ReliefF*, *SVM-Genetic Algorithm* (SVM-GA), *SVM-Particle Swarm Optimization* (SVM-PSO), *Sequential Forward Selection* (SFS), *Sequential Floating Forward Selection* (SFFS), *Principal Component Analysis* (PCA) e *Clustering of Variable/Variables Selected using Random Forests* (CoV/VSURF).

O *Unsupervised Feature Selection Method through Feature Clustering* (EUFSFC) (Yan et al., 2020) é um método de FS que avalia a redundância e a significância dos atributos. A significância é obtida por um algoritmo de agrupamento e os grupos são utilizados para calcular a redundância. Por fim, o algoritmo elimina automaticamente os atributos menos relevantes. O EUFSFC foi comparado com *Laplacian Score Feature Selection* (LSFS), *Multi-Cluster Meature Selection* (MCFS), *Local Learning-based Clustering Feature Selection* (LLCFS), *Nonnegative Discriminative Feature Selection* (NDFS) e *Robust Unsupervised Feature Selection* (RUFS). Os métodos de seleção de atributos foram executados nos classificadores *C4.5 Decision Tree* (C4.5-DT), KNN e RNA do tipo *Multi-Layer Perceptron* (MLP).

Maximum Dual Interaction and Maximum Feature Relevance (MDIMFR) (Anitha; Sherly, 2021) é um método *filter* baseado em Informação Mútua que visa alcançar boa relevância e baixa redundância nos atributos selecionados. O desempenho do MDIMFR foi avaliado e comparado com mRmR, *Joint Mutual Information Maximization* (JMIM) e *Conditional Mutual Information Maximization* (CMIM). Os atributos selecionados pelos métodos foram avaliados no classificador NB.

Em Saranya e Pravin (2021) é introduzido um novo método *wrapper* que realiza a análise de sensibilidade nos atributos para ranquear e selecionar os melhores. A abordagem proposta,

chamada *Variance based Sensitivity* (VSA), foi incorporada ao RF e seu desempenho foi avaliado e comparado com outros métodos *wrapper* disponíveis em Shah et al. (2020).

2.2 Sistemas *fuzzy* evolutivos

Os Sistemas *Fuzzy* Evolutivos (EFS, do inglês *Evolving Fuzzy Systems*) podem ser considerados como uma sinergia entre os sistemas *fuzzy* com estrutura expansível (evolutiva) e os métodos recursivos de aprendizado de máquina. Introduzidos inicialmente em Kasabov e Song (1999), Angelov e Buswell (2001), Kasabov (2001), Kasabov e Song (2002), Angelov e Filev (2004), os EFS surgiram para cobrir uma lacuna metodológica existente na modelagem adaptativa e no processamento *online*, eventualmente em tempo real, de dados não estacionários (Silva et al., 2013a). A estrutura dos EFS é equipada com mecanismos capazes de lidar com a ocorrência de mudanças observadas no fluxo de dados, possibilitando a construção incremental de sua estrutura concomitantemente com a atualização dos seus parâmetros.

Os EFS se diferem dos sistemas adaptativos e evolucionários. Sistemas adaptativos possuem uma estrutura fixa e são capazes de ajustar apenas seus parâmetros. Já os evolucionários são naturalmente relacionados a simulações inspiradas na genética, i.e., suas características são passadas de pais para os filhos por meio de recombinação, mutação, seleção e outras operações. Nos EFS, o termo evolutivo vem da evolução incremental de sua estrutura, que se expande ou contrai conforme os dados de entrada são apresentados. A evolução da estrutura é baseada no aprendizado da natureza humana, i.e., aprendendo a partir de experiências e observações de mudanças nos fluxos de dados mensurados (Angelov; Zhou, 2006).

Os EFS possuem um aprendizado dinâmico, incremental e com apenas uma única passagem do fluxo de dados, ou seja, o sistema utiliza apenas a amostra atual ou uma janela de amostras recentes. As amostras uma vez processadas podem ser descartadas. Portanto, a característica principal dos EFS é ter sua estrutura flexível, na qual a base de regras é adaptável e evolui de acordo com informações obtidas de um fluxo de dados de entrada (Bouchachia; Lughofer; Sayed-Mouchaweh, 2014). Destaca-se que nos EFS não há necessidade de retreinamento do sistema, uma vez que seu treinamento é *online* e contínuo (Lughofer; Pratama; Skrjanc, 2018a).

Os EFS são amplamente aplicados em campos industriais, onde o envelhecimento, o desgaste e até as falhas das máquinas podem gerar variações nos dados; em sistemas de comunicações, nos quais as condições de transmissão geram contínuas variações; na economia, onde os índices de bolsa variam com o tempo, dentre outras aplicações (Rong et al., 2018; Angelov, 2010). Destaca-se que atualmente existe uma crescente demanda

pelo desenvolvimento e aplicação de sistemas *fuzzy* evolutivos em diversas áreas (Skrjanc et al., 2019; Rodrigues; Silva; Lemos, 2022). Exemplos de aplicações de EFS incluem previsão e reconhecimento de padrões com dados ausentes (Garcia et al., 2019; Garcia; Leite; Škrjanc, 2020; Silva-Junior; Silva, 2022), previsão e regressão (Jahandari; Kalhor; Araabi, 2020), classificação (Shah; Wang, 2022; Tavares; Silva; Moita, 2022), detecção e diagnóstico de falhas (Dovzan; Logar; Skrjanc, 2015; Dovžan, 2017), monitoramento e detecção de anomalias (Decker et al., 2020), controle (Kandath et al., 2019), classificação de imagens (Gu; Angelov, 2018b), entre outros.

Os EFS encontrados na literatura possuem uma estrutura que evolui incrementalmente baseada em um fluxo de dados contínuo. Estes se diferenciam pela estrutura, pelo mecanismo de construção do conhecimento e pelo método de atualização de seus parâmetros. Em alguns EFS, como por exemplo, em Angelov e Filev (2004), Lughofer e Klement (2005), Angelov e Zhou (2006), Lemos, Caminhas e Gomide (2011), a geração da base de conhecimento utiliza a organização espacial dos atributos de entrada e um algoritmo de agrupamento evolutivo, que processa amostras recursivamente, associando os grupos às funções de pertinência. EFS como os propostos em Hore, Hall e Goldgof (2007b), Hore, Hall e Goldgof (2007a), Lughofer (2008), Zhu et al. (2014) também utilizam algoritmos de agrupamento e informações espaciais dos atributos de entrada. No entanto, esses EFS não utilizam regras ou funções de pertinência e as amostras são avaliadas com base no grau de pertinência entre a amostra e os grupos existentes. Por fim, em EFS como os introduzidos Leite et al. (2012), Leite, Costa e Gomide (2012), Silva et al. (2013a) a estrutura evolui baseado no erro de modelagem computado recursivamente. O consequente das regras *fuzzy* dos EFS, comumente, tem seus parâmetros atualizados pelo método do gradiente, mínimos quadrados ou variações destes, todos em suas versões recursivas. A seguir uma revisão da literatura descreve de forma sumarizada as características de alguns dos principais EFS. Os EFS são apresentados em ordem cronológica, do mais antigo para o mais recente.

O *evolving Takagi-Sugeno* (eTS) (Angelov; Filev, 2004) foi um marco no desenvolvimento dos EFS. O eTS utiliza regras do tipo Takagi-Sugeno construídas por um algoritmo incremental de agrupamento subtrativo. A atualização dos parâmetros e regras é feita com uma combinação de aprendizado supervisionado e não supervisionado. A cada nova amostra, é possível a criação ou modificação das regras por uma avaliação recursiva do potencial da amostra. No eTS cada grupo representa o antecedente de uma regra com funções Gaussianas. Seu modelo evolutivo estendido, o *eXtended Takagi-Sugeno* (xTS) (Angelov; Zhou, 2006), apresenta uma nova abordagem para avaliar a influência dos grupos e uma medida de idade. A medida da influência dos grupos é utilizada para avaliar recursivamente a qualidade das regras, enquanto a medida de idade é empregada para avaliar a exclusão de regras.

Introduzido em Lughofer e Klement (2005) o *Flexible Evolving Fuzzy Inference Systems* (FLEXFIS) é um EFS com regras do tipo Takagi-Sugeno e funções de pertinência Gaussianas. A estrutura do FLEXFIS é construída incrementalmente por um algoritmo de agrupamento recursivo derivado de uma modificação do *Vector Quantization* (VQ) (Gray, 1984). Os parâmetros do consequente são atualizados usando o erro quadrático médio. Em Lughofer, Angelov e Zhou (2007) é proposto o FLEXFIS-CLASS, um classificador evolutivo baseado no algoritmo do FLEXFIS original.

O *evolving Granular Neural Network* (eGNN) (Leite; Costa; Gomide, 2010) possui um aprendizado incremental que não necessita de um treinamento prévio para geração de seu conhecimento. Neste modelo é utilizado um sistema do tipo híbrido Mamdani-Takagi-Sugeno para gerar a base de regras, que são criadas e atualizadas por granularidades. As granularidades têm por objetivo extrair a incerteza dos dados e ajustar as informações provenientes das entradas e saídas. Basicamente, a granularidade é responsável por mapear um limite entre conjuntos similares de informações. O aprendizado, adaptação e refinamento deste modelo é executado pela camada neural baseada nas informações dos grânulos. Em Leite et al. (2012), Leite, Costa e Gomide (2012) são apresentados dois modelos: o *Fuzzy-Set-Based Evolving Modeling* (FBeM) e *Interval-Based Evolving Modeling* (IBeM). Esses modelos são similares ao eGNN e sua granularidade para geração de regras pode ser associada a valores linguísticos, onde uma amostra "x é pequena" entre um determinado intervalo de valores. Esse intervalo forma uma granularidade de amostras x para saídas y, que são mapeadas em conjuntos de regras Se-Então. Essas regras são desenvolvidas incrementalmente a partir dos dados de entradas e saídas.

O *evolving Neo-Fuzzy Neuron* (eNFN) (Silva et al., 2013a) é uma versão evolutiva do *Neo-Fuzzy Neuron* (NFN) cuja estrutura é composta por um conjunto de modelos desacoplados do tipo *fuzzy* Takagi-Sugeno de Ordem Zero com funções de pertinência triangulares e complementares, um por variável de entrada e cada um contendo um conjunto de regras. A estrutura do eNFN evolui pela inclusão, exclusão e atualização de regras e funções de pertinência. O eNFN utiliza o valor médio do erro local, o valor médio do erro global e a variância do erro global para decidir pela criação e inserção de regras e funções de pertinência. A exclusão de regras é baseada no tempo de inatividade da regra. Os parâmetros do consequente são atualizados utilizando um algoritmo do gradiente com taxa de aprendizagem ótima (Caminhas; Gomide, 2000).

A estrutura do *evolving Type-2 Classifier* (eT2Class) proposto em Pratama, Jie Lu e Guangquan Zhang (2015) é composta por um conjunto de regras de Takagi-Sugeno de Ordem 2. O eT2Class pode iniciar seu conhecimento do zero ou ser parcialmente treinado. As regras são sustentadas por funções de pertinências Gaussianas e são geradas com intervalos de desvios padrões, calculados por uma matriz de covariância diagonal. O consequente é

calculado por meio de um polinômio Chebyshev não linear. A geração de novas regras é realizada através dos métodos denominados *Type-2 Datum Significance* (T2DS) e *Type-2 Data Quality* (T2DQ). Para exclusão é utilizado *Type-2 Extended Rule Significance* (T2ERS) e *Type-2 Potential+* (T2P). O método de união de regras do eT2Class utiliza o conceito de similaridade. No eT2Class a similaridade é calculada pela distância Euclidiana.

Um EFS baseado em regras tipo Takagi-Sugeno é apresentado em Precup et al. (2017). Neste modelo a base de regras é construída por um algoritmo chamado *Online Identification Algorithm* (OIA). O OIA inicia a base de regras a partir da primeira amostra e a atualização das regras é realizada por uma medida entre o potencial dos novos dados e o potencial das regras existentes. Os parâmetros do consequente são atualizados utilizando mínimos quadrados recursivos.

O *Autonomous Learning Multi-Model* (ALMMo), proposto em Angelov, Gu e Príncipe (2017), utiliza uma metodologia de agrupamento de nuvens de dados. O conceito de agrupamento de nuvem de dados se baseia em grupos que não apresentam um formato geométrico definido, como grupos circulares ou elipsoidais. O ALMMo possui regras *fuzzy* do tipo AnYa (Angelov; Yager, 2012) de Ordem Zero, em sua metodologia de aprendizado, o modelo extrai automaticamente nuvens de dados para cada classe e constrói um subclassificador para cada classe. No ALMMo as regras dos subclassificadores são sintetizadas em vetores que incluem pontos focais, nos quais as regras são armazenadas. Estes pontos indicam propriedades comuns entre dados similares e a classificação é feita de acordo com o grau de confiança de uma amostra com cada subclassificador.

Proposto em Bodyanskiy et al. (2018) o *Group Method of Data Handling* (GMDH) é composto por 5 camadas: a primeira é composta por conjuntos *fuzzy* com funções de pertinência Gaussianas que realiza a fuzzificação dos atributos de entrada; na segunda camada é realizada a agregação dos graus de ativação da primeira camada; na terceira camada, estão os pesos sinápticos, que são multiplicadores dos valores de saída da camada anterior; a quarta camada é formada por duas unidades de soma, em que uma calcula a soma da segunda camada e a outra da terceira camada; a quinta e última camada, é chamada de neurônio de normalização. Este processa os valores da quarta camada, produzindo a saída. O aprendizado é realizado apenas por um processo de atualização dos pesos sinápticos da terceira camada, aplicando o método de mínimos quadrados recursivos entre os pesos da terceira camada e o sinal de saída produzido.

O *Self-Evolving Fuzzy System* (SEFS) (Ge; Zeng, 2019) é um modelo evolutivo que utiliza o erro de modelagem para avaliar a qualidade das regras e, consequentemente, definir a geração de novas regras. A atualização das funções de pertinências é realizada por uma medida de similaridade geométrica baseada na distância entre as funções de pertinência Gaussianas. Os parâmetros do consequente das regras são atualizados pelo *Recursive*

Least Square Algorithm with Variable Forgetting Factor (VFF-RLS). O VFF-RLS é uma versão aprimorada do método de mínimos quadrados recursivos com melhor desempenho no rastreamento de mudanças repentinas do sistema. O VFF-RLS possui um fator ótimo de esquecimento, obtido a partir da minimização do erro médio quadrático.

Torres e Serra (2018) propõem um algoritmo de agrupamento *fuzzy* evolutivo baseado na estimativa de densidade que é utilizada na construção *online* do antecedente das regras. A densidade indica a capacidade de generalização e representação de uma amostra. Assim, a cada amostra apresentada, a densidade é atualizada e essa amostra pode se tornar um ponto focal. Para calcular e atualizar o consequente das regras *fuzzy*, emprega-se uma metodologia recursiva baseada no *Fuzzy Eigen-System Realization Algorithm* (F-ERA) que utiliza os parâmetros de Markov obtidos recursivamente a partir de dados experimentais.

O *Parsimonious Autonomous Controller* (PAC) é um modelo *neuro-fuzzy* evolutivo proposto por Kandath et al. (2019). O PAC foi desenvolvido a partir do modelo *Parsimonious Autonomous Machine Learning* (PALM) (Ferdaus et al., 2019). A arquitetura *neuro-fuzzy* do PAC utiliza regras *fuzzy* geradas por grupos formados de hiperplanos, onde cada regra está associada aos centros dos hiperplanos, e são representados nós de uma camada neural do modelo. A criação de novas regras é realizada pelo método chamado *Self-Constructive Clustering* (SSC), que calcula a correlação das novas amostras com os grupos existentes. A união de grupos e regras é realizada com uma medida de similaridade dos hiperplanos calculada pelo ângulo de cada grupo. A atualização dos parâmetros do consequente ocorre por meio do *Fuzzily Weighted Generalised Recursive Least Square* (FWGRLS), que é uma extensão do método de mínimos quadrados recursivos.

O *Multi-Layer Ensemble Evolving Fuzzy Inference System* (MEEFIS) (Gu, 2021) possui uma estrutura com vários sistemas de inferência *fuzzy* evolutivos de primeira ordem, organizados em uma arquitetura multicamada. Em cada camada, conjuntos de inferência *fuzzy* são gerados de maneira independente, ou seja, cada um possui suas regras e processam os mesmos dados simultaneamente. Após o processamento, os valores de saída são repassados para camada posterior, onde outros conjuntos de inferência *fuzzy* recebem os dados e os processam. A última camada é responsável por gerar a saída. A atualização dos conjuntos de inferência *fuzzy* em todas as camadas é realizada dinamicamente, amostra por amostra, utilizando os mesmos princípios do *evolving Takagi-Sugeno* (eTS) (Angelov; Filev, 2004). Porém, a atualização só ocorre nas regras mais ativas de cada camada.

Em Ahwiadi e Wang (2021) é apresentado um modelo de agrupamento evolutivo nomeado *Adaptive Evolving Fuzzy* (AEF). A proposta é apresentar um modelo evolutivo com um processamento de alta velocidade na geração dos grupos e regras. Gerar uma base de conhecimento para processamento das amostras de curto e de longo prazo. O EAF apresenta dois aspectos para resolver os problemas apresentados. Na primeira etapa, é

empregado um método de avaliação para monitorar a tendência do erro de treinamento e controlar a atualização dos grupos/regras. Neste método é calculado a diferença do erro entre a saída real com a saída do modelo, e então, é avaliado o potencial dessa amostra para gerar novos grupos e atualizar os grupos existentes. Na segunda etapa, é apresentado um algoritmo chamado *adaptive Particle Filter* (aPF) para ajustar o centro dos grupos visando melhorar a flexibilidade e a capacidade adaptativa do modelo.

Um classificador evolutivo é apresentado em Shah e Wang (2022). Neste classificador, a primeira amostra é utilizada para determinar os parâmetros iniciais. Os conjuntos *fuzzy* são representados por funções de pertinência Gaussianas associadas aos centros dos grupos. Para cada conjunto *fuzzy*, uma regra é gerada na qual sua saída representa uma classe de saída. Cada conjunto *fuzzy*, associado a uma regra, está conectado a uma nova camada na qual é aplicado um peso que representa a contribuição daquela regra. Este peso é atualizado recursivamente por uma função Adadelta.

2.3 Sistemas *fuzzy* evolutivos com seleção de atributos

Uma característica importante dos EFS é a capacidade de adaptar parâmetros e estrutura a partir dos dados de entrada. A capacidade adaptativa dos EFS possibilita que métodos *wrapper* ou *embedded* de seleção de atributos sejam incorporados ao seu algoritmo de aprendizado. É desejável que a seleção de atributos nos EFS sejam realizadas automaticamente e sem causar perdas ou descontinuidades no processo de aprendizado.

Os métodos de seleção de atributos *wrapper* e *embedded* podem ser implementados de duas maneiras, *forward selection* (seleção para frente) e *backward selection* (seleção para trás). No método de seleção para frente o modelo inicia com um ou mais atributos de entrada e novos atributos são adicionados durante a aprendizagem. No método de seleção para trás, o modelo inicia com todos os atributos e estes podem ser descartados à medida que se tornam menos relevantes (Nugroho; Fanani; Shidik, 2021; Riasi; Delrobaei, 2021).

Em Andreu e Angelov (2010) é proposta uma versão do eTS (Angelov; Filev, 2004) com seleção de atributos. O método de seleção de atributos utiliza como base as saídas internas do eTS, que são normalizadas de maneira *online*. Cada atributo é avaliado em relação à sua importância pela soma dos parâmetros dos consequentes para uma característica específica em relação a todos os outros atributos. O valor da soma é utilizado para comparar os atributos e excluir os que não fornecem contribuição ao modelo.

Em Lughofer (2011) foi proposta uma versão do FLEXIFS-Class com FS na qual são associados valores entre 0 e 1 para os atributos de entrada, sendo que, quanto mais próximo de 1, maior é a sua relevância. Quando um atributo se aproxima de 0, ele é excluído do conjunto. A abordagem de seleção de atributos do FLEXIFS-Class é explorada

de duas maneiras: na primeira proposta os pesos [0,1] para cada atributo de entrada é dado pela razão da variância entre as classes e pela variância de cada atributo para todas as classes. Essa abordagem possui um custo computacional alto, pois necessita atualizar constantemente as matrizes de variância. A segunda abordagem é gulosa e se dá por meio do critério de separabilidade de Dy-Brodley, visando reduzir o número de atualizações e encontrar os atributos que possuem o menor peso.

Uma versão do eNFN (Silva et al., 2013a) com seleção de atributos é introduzida em Silva et al. (2013b) e chamada de *evolving Neo-Fuzzy Neural Network with Adaptive Feature Selection* (eNFN-AFS). O eNFN-AFS pode ser inicializado com qualquer número de atributos e o seu desempenho é comparado com suas versões com mais atributos (modelos candidatos à inclusão) e com menos atributos (modelos candidatos à exclusão). O teste F (Allen, 1997) foi adaptado para avaliar o desempenho do eNFN-AFS e dos modelos candidatos e decidir quando e se, efetivamente, um modelo candidato deve substituir o eNFN-AFS.

Proposto em Lughofer et al. (2015), o *Generalized Smart Evolving Fuzzy Systems* (Gen-Smart-EFS) tem o aprendizado baseado em uma versão estendida de *evolving Vector Quantization* (eVQ) (Lughofer, 2008). A estrutura do Gen-Smart-EFS é composta por regras de Takagi-Sugeno construída por um algoritmo de agrupamento que gera grupos de formato elipsoidal. Os parâmetros do consequente são atualizados por mínimos quadrados recursivos ponderados. A seleção dos atributos é realizada atribuindo pesos entre 0 e 1 para os atributos de entrada, com 1 para totalmente importante e 0 para sem importância. Nessa relação de importância é usado o cálculo da distância de Mahalanobis entre os grupos existentes, nos quais atributos não importantes e com pesos baixos devem levar a pequenas distâncias. Neste modelo, a seleção de atributos de entrada ocorre de forma dinâmica, ou seja, os atributos podem ser ignorados em um determinado ponto do tempo, mas podem ser reativados em um estágio posterior do processo de fluxo de dados *online*.

O *Interval Type-2 Neural Fuzzy System for Online System Identification and Feature Elimination* (IT2NFS-SIFE) (Chin-Teng et al., 2015) é um modelo *neuro-fuzzy* Tipo-2 construído baseado na estrutura de aprendizagem do modelo SEIT2FNN (Chia-Feng; Yu-Wei, 2008). No IT2NFS-SIFE a atualização dos parâmetros dos antecedentes é realizada de maneira recursiva pelo algoritmo de gradiente descendente. Os parâmetros dos consequentes são atualizados pelo algoritmo de filtro de Kalman ordenado. A seleção de atributos é realizada utilizando o conceito de modulador de associação. Este modulador atua na camada do antecedente verificando quais atributos e funções de pertinências são mais úteis e ativam melhor o consequente das regras. Atributos e funções de pertinências considerados ruins são automaticamente descartados.

O *evolving Type-2 Recurrent Fuzzy Neural Network* (eT2RFNN) (Pratama et al., 2016b) é um

EFS com arquitetura de redes recorrentes. Sua estrutura emprega funções de pertinência Gaussianas multivariadas do Tipo-2, que podem ser geradas e atualizadas automaticamente de acordo com o grau de não-linearidade dos dados de entrada. A saída utiliza uma função Wavelet não linear para atualização dos parâmetros. Essa função possui a propriedade de resolução múltipla, que é capaz de extrair informações importantes em vários níveis de resolução, potencializando a aproximação da saída desejada. A seleção dos atributos de entrada desse modelo é baseada no conceito de Markov blanket (Yu; Liu, 2004).

Proposto por Alizadeh et al. (2016), o *evolving Heterogeneous Fuzzy Inference System* (eHFIS) é um EFS com seleção de atributos que utiliza um algoritmo de agrupamento incremental não supervisionado para particionar o espaço dos dados e poder gerar novos grupos e regras. No eHFIS o consequente é atualizado pelo *Recursive Fuzzily Weighted Least Square* (RFWLS). A seleção de atributos é baseada no método de migração para o vizinho. Este método verifica a qualidade de um atributo em todas as regras, caso um atributo apresente não ser representativo, um modelo vizinho é gerado sem o atributo indesejado. Durante um intervalo de tempo, ambos os modelos serão avaliados e caso o modelo vizinho apresente melhor desempenho, ele substitui o modelo principal.

Parsimonious Ensemble (pENsemble) é um classificador *neuro-fuzzy* evolutivo com regras de Takagi-Sugeno de Ordem 1 proposto em Pratama, Pedrycz e Lughofer (2018). O pENsemble utiliza funções de pertinência Gaussianas multivariáveis permitindo grupos elipsoidais e não paralelos. A inclusão de regras é baseada na contribuição estatística das amostras, e a exclusão é realizada pela sua significância obtida a partir de um método que avalia a capacidade de generalização local da regra, com base na saída do consequente para essa amostra. Os parâmetros do consequente das regras são adaptados pelo método de mínimos quadrados recursivos. O pENsemble utiliza um método de FS baseado na proposta apresentada em Lughofer (2011), Wang et al. (2014), na qual cada atributo recebe pesos entre 0 e 1, que permitem a ativação dinâmica e a desativação.

Lughofer e Pratama (2018) introduz um EFS com seleção de atributos que utiliza 3 critérios para seleção: no primeiro calcula-se o grau de não linearidade entre os atributos, avaliando a homogeneidade dos valores adjacentes; no segundo avalia-se os intervalos de confiança dos erros, usando um estimador proposto chamado de *local error bars*; no terceiro utiliza-se a otimilidade-A, que é baseada na matriz de Fisher. O aprendizado incremental e evolutivo desse modelo é baseado no Gen-Smart-EFS (Lughofer et al., 2015).

Em Jahandari, Kalhor e Araabi (2020) é apresentado um novo EFS que faz uma combinação de conjuntos de pequenas arquiteturas de processamento denominados *Evolving Linear Models* (ELMs). Cada ELM é uma estrutura *fuzzy* independente capaz de processar os atributos de entrada e produzir uma saída. Inicialmente diferentes ELMs são geradas com base no conhecimento de um especialista, cada uma com conjuntos de atributos

diferentes a serem processados. Na estrutura de cada ELM, os atributos selecionados passam por um método chamado *evolving correlation-based subset selection* (eCSS) que é responsável por avaliar a qualidade de seus atributos e, de acordo com as características dos dados, adicionar ou remover um novo atributo. Posteriormente, a saída de cada ELM é estimada pelo método *evolving adaptative linear regression* (eALR). Por fim, é realizada uma combinação das saídas estimadas das diferentes ELMs em um sistema *fuzzy* de Takagi-Sugeno que realiza a classificação da amostra. O número de atributos de cada ELM gerado pelo conhecimento especialista não é fixo, podendo evoluir pelo eCSS em cada ELM. Essa característica permite o modelo evoluir de maneira incremental e se adaptar melhor à mudança dos dados no tempo.

Em Decker et al. (2020) é proposto um classificador evolutivo multiclasse denominado *evolving Gaussian Fuzzy Classifier* (eGFC). O modelo eGFC utiliza funções de pertinência Gaussianas para cobrir regiões no espaço de dados representados por grânulos *fuzzy* e para associar essas regiões a uma determinada classe. Os grânulos são intervalos mapeados no espaço de dimensões dos dados gerados para representar informações locais. Essas regiões mapeadas pelos intervalos locais são criadas para representar um novo conhecimento. A classificação é realizada com base na proximidade da amostra processada dentre regiões de cada classe. O eGFC apresenta um método híbrido para extração de atributos. A extração de atributos combina um método *filter* Hodrick-Prescott (HP) (Hodrick; Prescott, 1997) e uma Transformada Discreta de Fourier (DFT) (Lathi; Sedra, 2005) para extração de atributos durante o aprendizado.

O *Deep Evolving Fuzzy Neural Networks* (DEVFNN) proposto em Pratama, Pedrycz e Webb (2020) apresenta uma arquitetura flexível capaz de evoluir de maneira autônoma suas regras e a profundidade de suas camadas neurais. Todas as camadas ocultas do DEVFNN são construídas pelo classificador gClass (Pratama et al., 2016a). Este trabalha cooperativamente entre as camadas para produzir a saída final. O gClass é estruturado sob um sistema *fuzzy* generalizado de Takagi-Sugeno com funções de pertinência Gaussianas multivariada e seu consequente adota o conceito de *functional link neural network* (FLANN) (Pao; Takefuji, 1992). No DEVFNN é incorporado um método de seleção de atributos que atuam em todas as camadas ocultas do modelo. A seleção é realizada aplicando pesos binários (0 ou 1) determinados a partir da correlação entre relevância e redundância dos atributos de entrada, baseado na proposta de Yu e Liu (2004).

2.4 Considerações do capítulo

Neste capítulo, apresentou-se os conceitos fundamentais de seleção de atributos e sistemas evolutivos, seguidos de uma revisão da literatura abordando as principais técnicas e modelos. Essa revisão permitiu observar aspectos ainda não explorados nos campos de pesquisa

deste trabalho.

Em relação às técnicas de seleção de atributos, não foi encontrado nenhum método que possa ser utilizado como *filter* ou *wrapper*, que seja executado com uma única passagem nos dados, i.e., - com processamento incremental - e que possa ser empregado em algoritmos de aprendizado *online* e *offline*. Por isso, uma contribuição importante nesta área de pesquisa é o desenvolvimento de métodos de seleção de atributos que possuam essas características. Além disso, é desejável métodos que possuam baixo custo computacional, utilize somente equações simples e possam ser utilizados em conjuntos de dados com muitas amostras e em ambientes nos quais o número total de amostras é desconhecido.

Nos últimos anos, vários modelos evolutivos com novas abordagens foram propostos, gerando grandes avanços neste campo de pesquisa. No entanto, ainda existe uma crescente demanda para o desenvolvimento desses sistemas, tendo como foco novos métodos de aprendizado, algoritmos com baixo custo computacional, autônoma e flexibilidade e, principalmente, classificadores evolutivos com métodos de seleção de atributos embutidos no seu algoritmo de aprendizado.

Capítulo 3

Razão das médias como método de seleção de atributos

Este capítulo introduz a proposta de uso da razão das médias (*mean ratio*) como um método simples e eficiente para seleção de atributos para tarefas de classificação e reconhecimento de padrões. O método proposto, chamado de *Mean Ratio for Feature Selection* (MRFS), foi implementado como um método *filter* e como um método *wrapper*[1]. A Seção 3.1 apresenta a fundamentação para desenvolvimento do MRFS e a Seção 3.2 o detalha. Em seguida, na Seção 3.3, é descrito o MRFS como método *filter*, e na Seção 3.4 como *wrapper*. Por fim, a Seção 3.5 sumariza os pontos de destaque do capítulo.

3.1 Introdução

O método de razão das médias para seleção de atributos (MRFS, do inglês *Mean Ratio for Feature Selection*) utiliza a diferença entre as médias amostrais associadas às classes para selecionar os atributos mais relevantes. O método proposto visa explorar uma solução direta, eficiente e de fácil aplicação em ambientes que requerem processamento rápido. Em outras palavras, objetiva um método de seleção de atributos com equações simples, operações matemáticas básicas, como adição, divisão e comparação e, consequentemente, baixo custo computacional.

Para o bom entendimento do texto, é definido que população é o conjunto total de dados, amostra é uma informação do conjunto total de dados, atributo é uma unidade de um valor que representa uma uma informação de uma amostra e classe é a categoria ou uma saída que uma amostra representa.

Para o bom entendimento do texto define-se aqui alguns conceitos básicos:

[1] Optou-se neste trabalho por manter os termos em inglês para os tipos de métodos - *filter* e *wrapper*.

- População: é o conjunto completo de todas as amostras que compõem um conjunto de dados disponível para a análise.
- Amostra: é uma instância do conjunto de dados, i.e., é uma única observação ou registro que contém valores para cada atributo.
- Atributos: são as características ou variáveis que são medidas para cada indivíduo ou objeto na população ou na amostra.
- Classe: é a categoria ou a saída que uma amostra representa.

Para o desenvolvimento do método, inicialmente testes estatísticos como Análise de Variância (ANOVA, do inglês *Analysis of Variance*) e Intervalo de Confiança (Runger, 2003; Montgomery, 2013) foram utilizados, visando provar que a exclusão de atributos estatisticamente iguais melhora a acurácia dos classificadores. A Figura 1 mostra as etapas da análise estatística realizada.

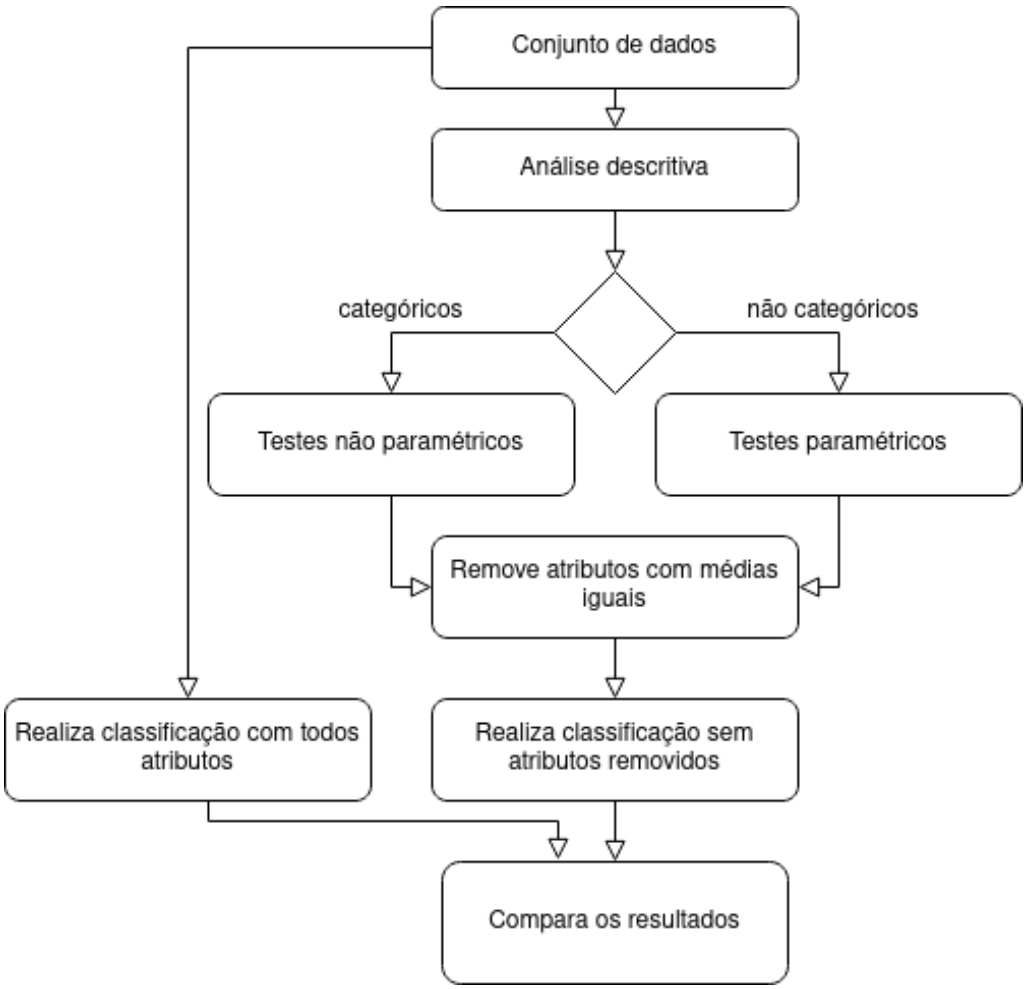

Figura 1 – Fluxograma de pré-desenvolvimento do MRFS.

Na primeira etapa, é realizada uma análise descritiva dos dados. O objetivo é extrair informações para definir o tipo de dado (categórico ou não categórico) e determinar um melhor método estatístico para seu processamento. O teste de hipótese foi aplicado para

verificar a igualdade estatística (ou não) entre as amostras de uma população com um nível de α. A significância está associada ao percentual de aceitação do teste, verificando se há igualdade nas médias das amostras examinadas; as hipóteses de igualdade podem ser confirmadas na Eq. (1).

$$\begin{cases} H_0 : \mu_0 - \mu_1 = 0 \\ H_1 : \mu_0 - \mu_1 \neq 0, \end{cases} \tag{1}$$

em que H_0 assume uma significância de risco para que as médias não apresentem igualdade entre si, ou seja, o resultado deste teste tem chance de não estar em uma região de aceitação e estar na região crítica. A hipótese alternativa, ou H_1, é a rejeição ou região crítica da hipótese de igualdade. μ_0 e μ_1 são as médias avaliadas. Os experimentos foram baseados na hipótese nula H_0: μ_0 - μ_1= 0, avaliando se os diferentes subconjuntos tinham médias estatisticamente iguais. Os atributos que apresentaram subconjuntos com média estatisticamente igual são removidos do conjunto de dados.

Na segunda etapa, os dados categóricos são processados usando método estatístico não paramétrico, enquanto os dados não categóricos são processados usando métodos paramétricos. Os testes realizados foram Chi-Quadrado, Kruskal-Wallis, Teste T, Teste de Wilcoxon, Intervalo de Confiança e Análise de Variância. Nos resultados desses experimentos avalia-se o p-valor. O nível de confiança adotado é de 95%, que define a região de aceitação com nível de significância α de 5% para médias iguais. Assim, com base no p-valor é possível determinar se os conjuntos amostrais de cada atributo possuem média estatisticamente diferente ou igual à média de uma população. A associação com as classes caracteriza cada conjunto amostral de uma população.

Na terceira etapa, os atributos que apresentaram igualdade estatística são excluídos, reduzindo o número de atributos para o classificador. Na quarta etapa - a etapa de teste -, o desempenho dos algoritmos de classificação é comparado usando o conjunto de dados com todos os atributos de entrada e o mesmo conjunto de dados sem atributos que apresentaram igualdade estatística (excluídos na terceira etapa). Os resultados obtidos sugerem que atributos com média amostral estatisticamente semelhante diminuem a capacidade preditiva dos classificadores.

Foi possível observar com os experimentos que independentemente do valor do nível de significância α, observa-se que quanto maior o p-valor, mais ele pertence à hipótese de igualdade para a média amostral associada às classes. Com isso, verifica-se que a razão das médias é semelhante ao p-valor, independentemente da distribuição dos dados e de sua variância. A razão das médias segue um padrão "quanto maior o valor da razão, mais similares são as médias". Outro conceito adotado que motivou o uso da razão das médias,

desconsiderando medidas como variância e distribuição de dados, é o teorema do limite central (CLT) (Runger, 2003; Montgomery, 2013). O CLT afirma que grandes conjuntos de amostras tendem a ser iguais à média da população, apoiando ainda mais o uso da razão das médias.

Com base nos resultados, o MRFS foi desenvolvido para avaliar os atributos do melhor ao pior, dada a razão das médias dos conjuntos de atributos. O MRFS utiliza apenas uma ideia central dos métodos estatísticos, que são as médias amostrais, e ignora outras equações matemáticas que necessitam de processamento posterior para cálculo das variáveis tais como variância, desvio padrão, médias residuais, Torricelli, entre outras. O MRFS é detalhado na próxima seção.

3.2 Mean Ratio for Feature Selection - MRFS

O MRFS é um método de seleção de atributos que utiliza as médias amostrais dos atributos para identificar os mais relevantes. Sugere-se no MRFS que os atributos com os menores valores da razão entre suas médias nas classes são os mais relevantes para a classificação. Em outras palavras, os atributos que possuem valores de médias menos similares entre as classes, são os mais representativos para os algoritmos de classificação. O MRFS possui 3 etapas que são detalhadas a seguir:

(i) Cálculo da Média Amostral: no MRFS, cada população é o atributo de entrada de um conjunto de dados e as médias amostrais são os conjuntos associados às classes. Assim, dado um conjunto de dados com duas classes e quatro atributos, para cada atributos é calculada duas médias amostrais, uma para cada classe. Esta afirmação pode ser representada pela Eq. (2):

$$S_{ij} = \frac{\sum_{t=1}^{T} x_j^t}{N} \in k_i, \tag{2}$$

na qual S é o conjunto da médias de cada classe, x é a amostra atual, N é a quantidade de amostra da i-ésima classe k, t indexa a amostra e T é o número de amostras.

(ii) Cálculo da Razão das Médias: a razão é uma medida que expressa uma fração ou porcentagem entre dois valores e é utilizada para comparar duas quantidades. Nesta etapa é calculada a razão Γ_i das médias do atributo i em cada classe j. A Eq. (3) apresenta o cálculo da razão das médias para duas ou mais classes.

$$\Gamma_j = \sum_{i=2}^{k} (min(\frac{S_{ij}}{S_{(i-1)j}}, \frac{S_{(i-1)j}}{S_{ij}})), \tag{3}$$

na qual k é o número de classes e Γ é a consequência direta do teorema do valor médio (Lang, 1986; Sahoo; Riedel, 1998). Este teorema é dado pela densidade de uma função de probabilidade contínua para os intervalos médios de cada classe i de cada atributo j, com base nos intervalos entre o valor mínimo da razão da média i por $i-1$ ou $i-1$ por i.

(iii) Ranqueamento dos Atributos: no contexto do MRFS, os atributos que possuem médias mais dispersas são mais relevantes para os classificadores. Portanto, os atributos são ranqueados em ordem crescente do valor da razão das médias Γ, dados que os com menores valores são os mais relevantes.

3.3 Mean Ratio for Feature Selection - *Filter* - MRFS-F

Nesta seção o MRFS é implementado como método *filter* e chamado de MRFS-F *(Mean Ratio for Feature Selection - Filter)*. Nos métodos *filter*, os atributos são selecionados *a priori* em uma etapa de pré-processamento. Em seguida, os atributos mais relevantes são selecionados para compor o conjunto de dados que será utilizado pelos classificadores. Nos métodos *filter* existe independência entre o método de seleção de atributos e o classificador. É importante destacar que os métodos *filter* ranqueiam os atributos de acordo com sua relevância, mas não informam quantos atributos devem ser utilizados para compor o conjunto de dados de entrada do classificador.

O MRFS-F é um método simples e eficiente. Suas equações são simples e utilizam somente operações matemáticas básicas e comparações. Além disso, o MRFS-F realiza somente uma passada em cada amostra da base de dados. Este processo agiliza o processamento e reduz o custo computacional e o consumo de memória. A Figura 2 ilustra o fluxograma do MRFS-F que é detalhado no Algoritmo 1.

3.4 Mean Ratio for Feature Selection - *Wrapper* - MRFS-W

Esta seção detalha a implementação do MRFS-W *(Mean Ration for Feature Selection - Wrapper)*. Diferentemente dos métodos *filter*, os métodos *wrapper* são executados em conjunto com o classificador. Conforme descrito na Seção 2.3, os métodos *wrapper* podem ser utilizados com seleção para frente ou seleção para trás. Neste trabalho, optou-se por utilizar a seleção para trás. Salienta-se que o MRFS-W pode ser adaptado para trabalhar com a seleção para frente.

O MRFS-W pode ser resumido em 7 etapas, como pode ser visto no fluxograma da Figura 3. As 3 primeiras etapas são, *mutatis mutandis*, as mesmas descritas na Seção 3.2 - (i) Cálculo da Média Amostral; (ii) Cálculo da Razão das Médias; e, (iii) Ranqueamento dos

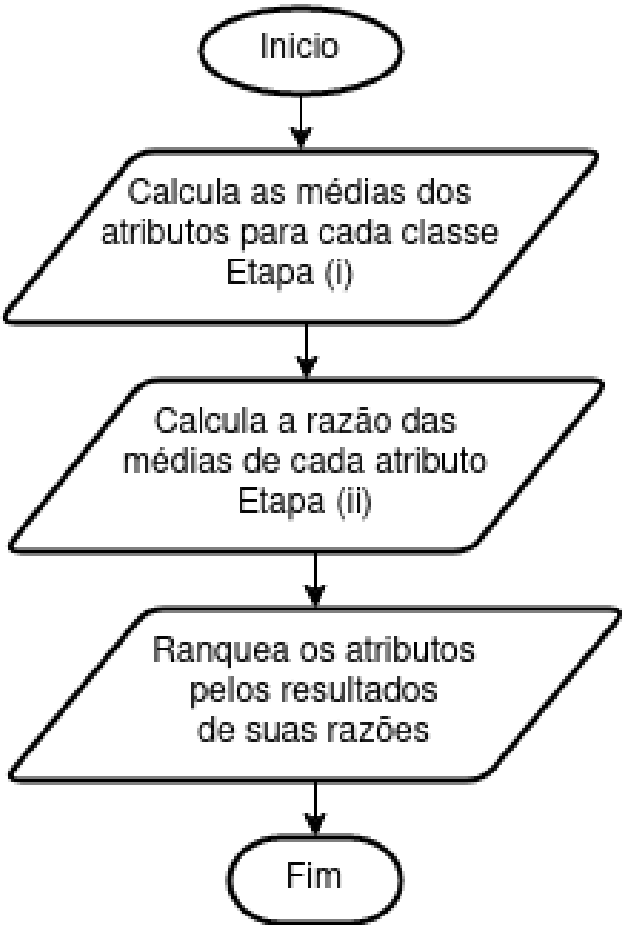

Figura 2 – Fluxograma do MRFS-F.

Algoritmo 1: Método de seleção de atributos MRFS-F.

Entrada: X, Y;
Saída: rankAtrib;
for *t=1 to T* **do**
  `// m é o número de atributos`
  for *j=1 to m* **do**
    S_{ij} = Calcula a média das amostras - Eq. (2);
  end
end
`// K é o número de classes`
for *i=1 to K* **do**
  Γ_i = Calcula a razão das médias - Eq. (3);
end
rankAtrib = Ordena atributos em ordem crescente do Γ_i

Atributos - e, portanto, não serão detalhadas novamente. As etapas 4-7 são detalhadas a seguir.

(iv) Criação dos Modelos: de acordo com a ordenação proveniente da razão das médias

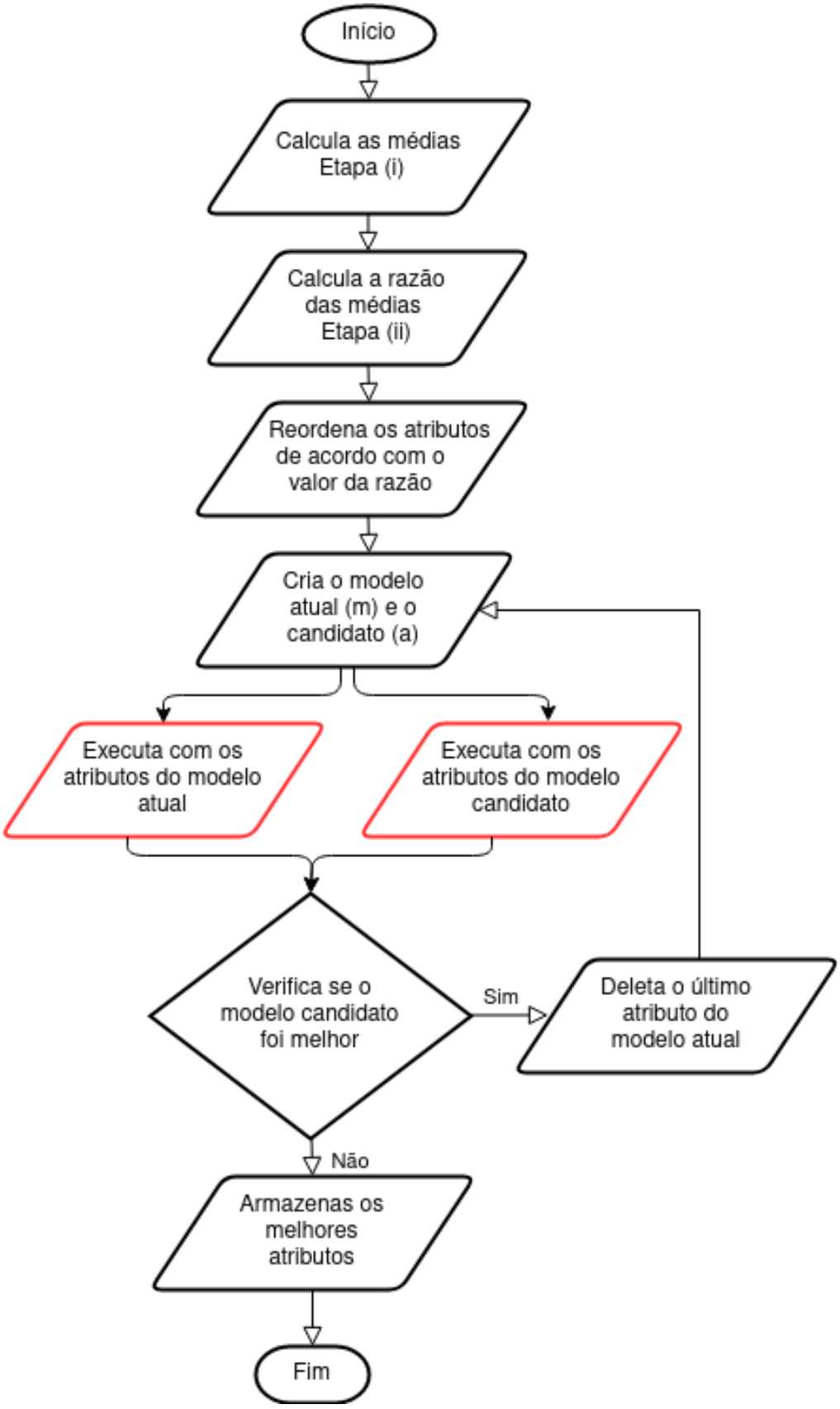

Figura 3 – Fluxograma do MRFS-W.

(Etapa iii) são criados dois modelos: o modelo atual com o número atual de atributos, e que, na primeira iteração do treinamento do classificador, inicia com todos os atributos do conjunto de dados; e o modelo candidato que é construído excluindo o atributo menos relevante do modelo atual.

(v) Execução dos Modelos: nesta etapa, é executada a classificação com os dois modelos (candidato e atual). A ideia é verificar se vale a pena substituir o modelo atual pelo modelo candidato, reduzindo assim a quantidade de atributos até um ponto que o modelo atual apresente melhor acurácia que o modelo candidato.

(vi) Avaliação dos Modelos: a avaliação é realizada para comparar o desempenho dos dois modelos. Caso o modelo candidato tenha um desempenho melhor, ele se torna o novo modelo atual e é criado um novo modelo candidato, excluindo o atributo menos relevante. Isto é, caso o modelo candidato apresente melhores resultados volta-se a Etapa (iv). Esse processo se repete enquanto o modelo candidato apresentar melhores resultados que o modelo atual e, caso contrário, encerra o processo de seleção de atributos e avança para última etapa.

(vii) Armazenamento dos atributos: essa etapa é encerrada junto com o final do processo de treinamento do classificador. Os atributos que apresentaram a melhor acurácia durante o treinamento são utilizados no classificador.

Algoritmo 2 detalha o processo de seleção de atributos e classificação do MRFS-W. Conforme pode ser visto, MRFS-W pode ser aplicado em qualquer algoritmo de classificação.

3.5 Considerações do capítulo

Este capítulo introduziu um novo método para seleção de atributos baseado na razão das médias chamado MRFS. No método proposto a razão entre as médias dos atributos é utilizada para atribuir relevância e, consequentemente, selecionar os atributos mais relevantes. Foram propostas duas abordagens para o MRFS, uma como método *filter*, e outra como *wrapper*. Destaca-se que o MRFS utiliza somente operações matemáticas básicas e realiza somente uma passagem nos dados para ranquear os atributos.

Algoritmo 2: Método de seleção de atributos MRFS-W.

Entrada: X, Y;
Saída: m;
`// Início`
Ordena atributos com MRFS, Algoritmo 1;
Teste = verdadeiro;
`// Seleção dos atributos`
a=m-1;
while *Teste* **do**
    Cria o modelo atual com m atributos;
    Cria o modelo candidato com a atributos;
    Executa o modelo atual no classificador;
    Executa o modelo candidato no classificador;
    if *Acurácia Modelo Atual* $<$ *Acurácia Modelo Candidato* **then**
        m = a;
        a = a-1 (remove último atributo);
    end
    else
        Teste = falso;
    end
end
Retorna m atributos com melhor acurácia

Capítulo 4

Experimentos computacionais com métodos de seleção de atributos

Este capítulo apresenta experimentos computacionais realizados para avaliar e comparar o desempenho das variações de aplicação do método MRFS. Nos experimentos, foram exploradas duas abordagens para utilização do MRFS, uma utilizando-o como método *filter* (MRFS-F) e outra como *wrapper* (MRFS-W). Os conjuntos de dados e a medida de desempenho utilizadas nos experimentos são apresentados na Seção 4.1. A Seção 4.2 detalha os resultados experimentais do MRFS-F e avalia seu desempenho em relação a métodos *filter* do estado da arte. A Seção 4.3 ilustra experimentos computacionais com MRFS-W em diferentes classificadores. Os classificadores com o MRFS-W foram comparados com suas versões originais sem o método de seleção de atributos incorporado. Por fim, a Seção 4.4 sumariza o capítulo e discute o desempenho do MRFS.

4.1 Conjuntos de dados e medida de desempenho

Os conjuntos de dados utilizados nos experimentos são detalhados na Tabela 1. Ela ilustra o nome do conjunto, o número de amostras, o número de atributos, o número de classes e a fonte na qual o conjunto foi obtido.

Tabela 1 – Características dos conjuntos de dados.

Nome	Amostras	Atributos	Classes	Fonte
Cervical Cancer Risk	858	35	2	(KAGGLE, 2023)
Glass Identification Dataset	214	10	6	(UCI, 2023)
Heart Disease	303	13	2	(KAGGLE, 2023)
Mobile Price Classification	2000	20	4	(KAGGLE, 2023)
Page Blocks Classification	5473	10	5	(UCI, 2023)
QSAR Biodegradation	1054	41	2	(UCI, 2023)
South African Health	461	9	2	(KEEL, 2019)
Wine	178	13	3	(UCI, 2023)

Para os experimentos, os conjuntos de dados foram divididos em dois subconjuntos: um

com 80% das amostras para treinamento e o outro com os 20% restantes para avaliação e comparação dos resultados. O desempenho é avaliado pela acurácia média de 10 execuções de cada experimento. A acurácia (Acc) é obtida pelo número de classificações corretas dividido pelo número de amostras, conforme:

$$Acc(y, \hat{y}) = \frac{1}{\mathbf{T}} \sum_{i=1}^{\mathbf{T}} 1(\hat{y}_i = y_i), \tag{4}$$

na qual $\hat{y}_i$ é a classe prevista pelo classificador da $i-ésima$ amostra, y_i é a classe correta, e $\mathbf{T}$ é o número total de amostras.

4.2 Experimentos com os métodos *filter*

O MRFS-F foi avaliado utilizando quatro classificadores e os oito conjuntos de dados descritos na Tabela 1 . Os resultados obtidos foram comparados com três métodos *filter* do estado da arte: Kruskal Wallis, ReliefF e Chi-Square. Os classificadores utilizados para avaliar o desempenho dos métodos de seleção de atributos foram: Rede Neural Artificial (RNA) do tipo *Multilayer Perceptron*, *K-Nearest Neighbor* (KNN), Naïve Bayes (NB) e *Support Vector Machine* (SVM). Os classificadores foram implementados empregando a biblioteca sklearn (Scikit-learn, 2020) da linguagem Python. A metodologia empregada nos experimentos pode ser sumarizada em 4 etapas:

- **Etapa 1 - Configuração dos Classificadores:** inicialmente os classificadores foram avaliados utilizando todos os atributos dos conjuntos de dados objetivando encontrar a melhor configuração de parâmetros e estrutura. A melhor configuração para cada classificador foi obtida por meio de busca exaustiva e a que obteve o melhor desempenho foi utilizada para avaliar os métodos de seleção de atributos. As configurações e os parâmetros utilizados foram:
  - **RNA:** camadas ocultas = 2; neurônios por camada oculta = 10; função de ativação = tangente hiperbólica, linear, sigmóide logístico, linear retificado; algoritmo de aprendizado = quasi-newton, gradiente descendente estocástico, otimizador baseado em gradiente estocástico; todos com 700 épocas de treinamento.
  - **KNN:** distância = Euclidiana, Manhattan; número de vizinhos = 2 até 25 - intervalo de 1.
  - **NB:** distribuição = Gaussiana, Bernoulli, Categórica, Multinomial.
  - **SVM:** kernel = linear; C = 0.1 até 1 - intervalo de 0.1.
- **Etapa 2 - Ranqueamento de Atributos:** nesta etapa, os quatro métodos de seleção de atributos foram executados a fim de ranquear os atributos em ordem decrescente de acordo com sua relevância. O Apêndice (A) ilustra, para cada conjunto de dados e

para cada método, a ordem de ranqueamento dos atributos.

- **Etapa 3 - Construção dos Subconjuntos de Dados:** os métodos de seleção de atributos ranqueiam os atributos de acordo com a relevância de cada atributo. Porém, os métodos *filter* não identificam quantos atributos devem ser utilizados nos classificadores para obter o melhor desempenho. Para avaliar os métodos de seleção de atributos, foram criados, a partir do conjunto de dados com todos os atributos, quatro subconjuntos considerando uma redução gradativa de 20% no número de atributos. Tabela 2 ilustra o número de atributos utilizado em cada um dos subconjuntos de dados.

Tabela 2 – Número de atributos utilizados - *filter*.

Nome	Total de Atributos	N.º de Atrib. Selec.
Cervical Cancer	35	26, 20, 14, 8
Glass Identification	10	8, 6, 4, 3
Heart Disease	13	11, 9, 7, 5
Mobile Price	20	16, 12, 8, 4
Page Blocks	10	8, 6, 4, 3
QSAR Biodegradation	41	32, 24, 16, 8
South African Health	09	8, 7, 6, 5
Wine	13	11, 9, 7, 5

- **Etapa 4 - Experimentos Computacionais:** os experimentos foram realizados utilizando a melhor configuração de parâmetros e estrutura dos classificadores obtidos na Etapa 1 e os subconjuntos de dados de cada método *filter* criados na Etapa 3.

Para permitir a replicação dos experimentos, o MRFS-F foi disponibilizado no link URL: http://shorturl.at/ilJPZ. Os resultados experimentais são apresentados nas seções subsequentes. A Seção 4.2.1 destaca os experimentos com RNA, enquanto a Seção 4.2.2 ilustra os com KNN. Os experimentos com o classificador NB são apresentados na Seção 4.2.3. Por fim, a Seção 4.2.4 mostra os experimentos com SVM.

4.2.1 Resultados experimentais com RNA - *Filter*

Nesta seção, os métodos de seleção de atributos foram avaliados utilizando uma Rede Neural Artificial (RNA) como classificador. Inicialmente, foram realizados experimentos utilizando os conjuntos de dados com todos os atributos para encontrar, para cada um, a melhor configuração de rede (função de ativação e algoritmo de treinamento). Em seguida, as configurações que obtiveram o melhor desempenho foram utilizadas para avaliar os métodos de seleção de atributos. A Tabela 3 mostra, para cada conjunto de dados, a função de ativação e o algoritmo de aprendizado que obtiveram o melhor resultado.

A Tabela 4 mostra a acurácia e o número de atributos do melhor resultado obtido por cada método nos experimentos com RNA. Avaliando os resultados, o MRFS-F obteve a melhor acurácia em quatro dos oito experimentos, Chi-Square em dois e Kruskal-Wallis em um.

Tabela 3 – Melhores configurações de RNA em conjuntos de dados com todos os atributos.

Nome	Função de Ativação	Aprendizado
Heart Disease	Linear	Quasi-Newton
Cervical Cancer	Sigmoide Logístico	Gradiente Descendente Estoc.
Mobile Price	Linear	Gradiente Descendente Estoc.
Page Blocks	Tangente Hiperbólica	Gradiente Descendente Estoc.
QSAR Biodegradation	Sigmoide Logístico	Quasi-Newton
South African Health	Linear Retificado	Gradiente Descendente Estoc.
Glass Identification	Linear Retificador	Quasi-Newton
Wine	Sigmóide Logístico	Quasi-Newton

No experimento no conjunto de dados QSAR, o MRFS alcançou a mesma acurácia do Chi-Square, porém com menos atributos.

Tabela 4 – Desempenho dos métodos *filter* com RNA.

Conj. Dados	MRFS-F Acc / Atrib.	Kruskal-Wallis Acc / Atrib.	ReliefF Acc / Atrib.	Chi-Square Acc / Atrib.
H. Disease	**94,39** / **09**	89,77 / 07	87,46 / 05	80,53 / 07
C. Cancer	97,75 / 26	97,16 / 20	97,46 / 08	**97,90** / **14**
Mobile Price	68,25 / 16	62,40 / 16	70,50 / 16	**72,90** / **16**
Pg. Blocks	**97,08** / **04**	95,91 / 06	95,83 / 06	95,59 / 08
QSAR	**89,18** / **24**	88,99 / 32	88,80 / 32	**89,18** / **32**
S.A. Health	**73,97** / **04**	73,10 / 08	69,63 / 07	70,07 / 03
Glass Iden.	**87,85** / **06**	68,22 / 08	68,22 / 06	68,22 / 08
Wine	40,11 / 09	**88,70** / **07**	40,11 / 09	40,11 / 09

4.2.2 Resultados experimentais com KNN - *Filter*

Esta seção avalia os métodos de seleção de atributos com o KNN. Inicialmente, foram encontradas as melhores configurações do KNN, considerando os conjuntos de dados com todos os atributos. A Tabela 5 mostra as configurações que alcançaram os melhores resultados para cada conjunto de dados. As configurações que obtiveram o melhor desempenho foram utilizadas para avaliar o desempenho dos métodos de seleção de atributos.

Tabela 5 – Melhores configurações do KNN em conjuntos de dados com todos os atributos.

Nome	Distância	Vizinhos
Heart Disease	Manhattan	2
Cervical Cancer	Manhattan	3
Mobile Price	Manhattan	3
Page Blocks	Manhattan	2
QSAR Biodegradation	Manhattan	2
South African Health	Manhattan	3
Glass Identification	Euclidiana	4
Wine	Manhattan	3

Os resultados experimentais da avaliação dos métodos de seleção de atributos com o KNN são sintetizados na Tabela 6. O MRFS-F obteve o melhor desempenho em cinco dos oito experimentos e o Kruskal-Wallis em dois. MRFS-F e Kruskal-Wallis obtiveram a maior acurácia nos experimentos com Heart Disease, no entanto, o MRFS-F alcançou o resultado com sete atributos e o Kruskal-Wallis com nove.

Tabela 6 – Desempenho dos métodos *filter* com KNN.

Conj. Dados	MRFS-F Acc / Atrib.	Kruskal-Wallis Acc / Atrib.	ReliefF Acc / Atrib.	Chi-Square Acc / Atrib.
Heart Disease	**89,11** / **07**	**89,11** / **09**	88,45 / 05	87,79 / 05
Cervical Cancer	**96,56** / **26**	96,26 / 08	95,81 / 08	96,41 / 08
Mobile Price	**96,70** / **16**	96,55 / 16	96,60 / 16	96,60 / 12
Pg. Blocks	97,42 / 08	**97,61** / **06**	97,50 / 06	97,42 / 08
QSAR	**92,13** / **24**	91,84 / 24	91,27 / 16	91,37 / 24
S.A. Health	82,00 / 07	**84,38** / **06**	80,48 / 07	82,21 / 04
Glass Iden.	**100,00** / **06**	99,53 / 06	99,53 / 08	99,53 / 06
Wine	**90,96** / **07**	90,40 / 11	90,40 / 11	90,40 / 07

4.2.3 Resultados experimentais com NB - *Filter*

Os métodos de seleção de atributos são avaliados nesta seção utilizando o classificador Naïve Bayes (NB). Experimentos computacionais foram realizados para encontrar a melhor configuração do classificador para cada conjunto de dados e a melhor configuração foi utilizada para avaliar o desempenho dos métodos de seleção de atributos. As configurações do classificador Naïve Bayes que obtiveram a maior acurácia para cada conjunto de dados são apresentadas na Tabela 7.

Tabela 7 – Melhores configurações do NB em conjuntos de dados com todos os atributos.

Conj. Dados	Método
Heart Disease	Gaussiano
Cervical Cancer	Bernoulli
Mobile Price	Gaussiano
Page Blocks	Bernoulli
QSAR Biodegradation	Bernoulli
South African Health	Bernoulli
Glass Identification	Gaussiano
Wine	Gaussiano

A Tabela 8 apresenta os resultados experimentais da avaliação dos métodos de seleção de atributos com o Naïve Bayes. O método MRFS-F obteve o melhor desempenho em quatro dos oito experimentos, Chi-Square em apenas um. Nos experimentos com o conjunto Cervical Cancer, os melhores resultados foram obtidos pelo MRFS-F e ReliefF. Com o conjunto Page Blocks, todos os métodos de seleção de atributos tiveram a mesma acurácia. Com o conjunto Wine, os métodos MRFS-F, ReliefF e Chi-Square tiveram a mesma acurácia.

Tabela 8 – Desempenho dos métodos *filter* com NB.

Conj. Dados	MRFS-F Acc / Atrib.	Kruskal-Wallis Acc / Atrib.	ReliefF Acc / Atrib.	Chi-Square Acc / Atrib.
Heart Disease	**83,83** / **11**	81,52 / 11	83,50 / 11	82,84 / 11
Cervical Cancer	**96,11** / **14**	95,21 / 08	**96,11** / **08**	95,06 / 14
Mobile Price	81,95 / 16	80,40 / 16	81,75 / 16	**82,25** / **16**
Page Blocks	**89,78** / **03**	**89,78** / **03**	**89,78** / **03**	**89,78** / **03**
QSAR	**80,36** / **16**	79,32 / 32	70,87 / 24	77,61 / 32
S.A. Health	**72,67** / **04**	72,23 / 04	70,72 / 08	71,37 / 08
Glass Iden.	**88,79** / **06**	91,59 / 04	92,52 / 08	89,25 / 06
Wine	**97,74** / **11**	97,18 / 11	**97,74** / **07**	**97,74** / **07**

4.2.4 Resultados experimentais com SVM - *Filter*

Esta seção detalha os experimentos realizados para avaliar os métodos de seleção de atributos utilizando o classificador SVM. Assim como nos demais experimentos, inicialmente foram realizados experimentos para encontrar a melhor configuração do SVM para cada um dos conjuntos de dados. As configurações que alcançaram a maior acurácia (Tabela 9) foram utilizadas para avaliar os métodos de seleção de atributos.

Tabela 9 – Melhores configurações de SVM em conjuntos de dados com todos os atributos.

Conj. Dados	Parâmetro C
Heart Disease	0,2
Cervical Cancer	0,5
Mobile Price	0,7
Page Blocks	1,0
QSAR Biodegradation	0,8
South African Health	0,1
Glass Identification	0,2
Wine	0,5

A Tabela 10 ilustra os resultados experimentais obtidos na avaliação dos métodos de seleção de atributos utilizando o SVM. O MRFS-F obteve, em conjunto com outros métodos, o melhor desempenho em cinco dos oito experimentos. Nos outros três experimentos os melhores resultados foram obtidos pelo Kruskal-Wallis (QSAR), ReliefF (Page Blocks) e Chi-Square (Mobile Price). Destaca-se que nos conjuntos Cervical Cancer e Wine, o MRFS-F conseguiu o melhor resultado com um menor conjunto de atributos de entrada.

Tabela 10 – Desempenho dos métodos *filter* com SVM.

Conj. Dados	MRFS-F Acc / Atrib.	Kruskal-Wallis Acc / Atrib.	ReliefF Acc / Atrib.	Chi-Square Acc / Atrib.
Heart Disease	**85,81** / **12**	84,16 / 09	**85,81** / **12**	84,72 / 12
Cervical Cancer	**96,11** / **14**	95,81 / 26	**96,11** / **20**	**96,11** / **26**
Mobile Price	98,55 / 16	96,45 / 16	98,75 / 16	**98,80** / **16**
Page Blocks	96,56 / 08	96,62 / 08	**97,00** / **08**	96,36 / 08
QSAR	87,19 / 32	**88,43** / **32**	87,76 / 32	**88,43** / **32**
S.A. Health	**75,05** / **08**	73,97 / 08	72,02 / 08	**75,05** / **08**
Glass Iden.	**100,00** / **06**	**100,00** / **06**	**100,00** / **04**	**100,00** / **04**
Wine	**98,31** / **07**	97,74 / 11	97,74 / 11	**98,31** / **11**

4.3 Experimentos com o método *wrapper*

Esta seção apresenta experimentos computacionais realizados para avaliar o MRFS-W. O objetivo do experimento é avaliar o desempenho do MRFS-W incorporado a quatro classificadores (RNA, KNN, NB e SVM) comparando seus resultados com os obtidos pelos classificadores sem o método de seleção de atributos. A melhor configuração dos classificadores com e sem o MRFS-W foi obtida por busca exaustiva, considerando:

- RNA: camadas ocultas = 2; neurônios por camada oculta = 10; algoritmo de aprendizado = quasi-newton, gradiente descendente estocástico, otimizador baseado em gradiente estocástico; épocas de treinamento = 700.
- KNN: distância = Euclidiana, Manhattan; número de vizinhos = 2, 5, 30 e 50.
- NB: Gaussiana e Bernoulli.
- SVM: kernel = linear e RBF; C = 0,1, 0,5 e 1.

Os classificadores foram implementados utilizando a biblioteca sklearn do Python e os experimentos foram conduzidos considerando os conjuntos de dados e a medida de desempenho descritos na Seção 4.1. Conforme proposto na Seção 4.1, cada experimento foi executado 10 vezes e o desempenho avaliado pela acurácia média. O Apêndice (B) ilustra a acurácia média para cada conjunto de dados e configuração dos classificadores. Os conjuntos de dados e códigos-fonte utilizados nos experimentos estão disponíveis em http://shorturl.at/ilJPZ.

Nas próximas seções são apresentados os melhores resultados obtidos para cada classificador/conjunto de dados. A Seção 4.3.1 detalha os experimentos com RNA e a Seção 4.3.2 apresenta os experimentos utilizando o KNN. Os experimentos com Naïve Bayes e SVM são apresentados, respectivamente, nas seções 4.3.3 e 4.3.4.

4.3.1 Resultados experimentais com RNA - *Wrapper*

Nesta seção, avalia-se o desempenho do MRFS-W incorporado à RNA, nomeado de RNA-MRFS-W. Os experimentos foram realizados considerando as configurações da RNA apresentadas na Seção 4.3. A configuração que obteve o melhor desempenho foi utilizada para fins de comparação. A acurácia média de cada configuração da RNA pode ser obtida no Apêndice (B.1).

A Tabela 11 ilustra o desempenho da RNA e da RNA-MRFS-W. A tabela mostra, para cada conjunto de dados, a acurácia dos classificadores, o número de atributos selecionado pela RNA-MRFS-W e o número de atributos do conjunto de dados. Os resultados experimentais sugerem que a RNA-MRFS-W obteve a maior acurácia em 7 dos 8 conjuntos de dados quando comparada com a RNA sem MRFS. Destaca-se que na média o desempenho da RNA-MRFS-W foi mais de 10% superior ao da RNA.

4.3.2 Resultados experimentais com KNN - *Wrapper*

Esta seção apresenta experimentos computacionais com o MRFS-W incorporado ao KNN e nomeado de KNN-MRFS-W. As configurações de parâmetros utilizadas nos experimentos com o KNN estão descritas na Seção 4.3. No Apêndice (B.2) são apresentados todos os resultados obtidos com cada variação dos parâmetros do KNN-MRFS-W e do KNN. A

Tabela 11 – Desempenho da RNA e da RNA-MRFS-W.

Conj. Dados	RNA	RNA-MRFS-W	Atrib. Selec.	Atrib.
Wine	40,00	**76,95**	4 ± 2	13
Glass Identification Dataset	86,97	**90,69**	6 ± 3	10
Heart Disease	75,25	**84,43**	9 ± 1	13
South African Health	67,20	**72,68**	6 ± 3	9
Cervical Cancer Risk	93,28	**95,00**	29 ± 3	35
QSAR Biodegradation	82,94	**87,82**	38 ± 1	41
Mobile Price Classification	57,27	**79,60**	18 ± 1	20
Page Blocks Classification	**95,21**	91,25	8 ± 1	10

configuração que obteve o melhor desempenho é apresentada nesta seção para avaliar o desempenho do método de seleção de atributos.

Tabela 12 ilustra para cada conjunto de dados a melhor acurácia obtida pelo KNN-MRFS-W e pelo KNN, o número de atributos selecionados pelo KNN-MRFS-W e o número total de atributos. Em todos os conjuntos de dados, o melhor desempenho é alcançado pelo KNN-MRFS-W que obteve uma acurácia média 6% superior ao KNN.

Tabela 12 – Desempenho do KNN e do KNN-MRFS-W.

Conj. Dados	KNN	KNN-MRFS-W	Atrib. Selec.	Atrib.
Wine	71,38	**92,22**	2 ± 1	13
Glass Identification Dataset	96,97	**98,13**	6 ± 2	10
Heart Disease	65,91	**83,61**	9 ± 1	13
South African Health	67,85	**71,39**	6 ± 2	9
Cervical Cancer Risk	93,58	**94,10**	29 ± 1	35
QSAR Biodegradation	81,13	**84,88**	25 ± 5	41
Mobile Price Classification	92,47	**93,00**	8 ± 6	20
Page Blocks Classification	95,65	**96,09**	6 ± 2	10

4.3.3 Resultados experimentais com Naïve Bayes - *Wrapper*

Nesta seção o MRFS-W é avaliado incorporado ao NB, NB-MRFS-W. Os experimentos foram conduzidos empregando as configurações do NB apresentadas na Seção 4.3 e, para cada conjunto de dados e classificador, a configuração que obteve melhor desempenho foi utilizada para fins de comparação. O Apêndice (B.3) ilustra os resultados obtidos com cada variação da configuração dos classificadores.

Os melhores resultados obtidos pelo NB e NB-MRFS-W são sumarizados na Tabela 13. Os resultados obtidos pelo NB-MRFS-W são melhores do que os obtidos pelo NB em todos os experimentos. Na acurácia média, o NB-MRFS-W supera o NB em 2%.

4.3.4 Resultados experimentais com SVM - *Wrapper*

Esta seção detalha os experimentos computacionais para avaliar o MRFS-W incorporado ao classificador SVM, nomeado de SVM-MRFS-W. Os experimentos foram realizados considerando as configurações de parâmetros descritas Seção 4.3. O Apêndice (B.4)

Tabela 13 – Desempenho do NB e do NB-MRFS-W.

Conj. Dados	NB	NB-MRFS-W	Atrib. Selec.	Atrib.
Wine	95,28	**96,67**	8 ± 4	13
Glass Identification Dataset	81,39	**88,37**	5 ± 4	10
Heart Disease	80,33	**83,77**	6 ± 2	13
South African Health	69,68	**71,82**	5 ± 3	9
Cervical Cancer Risk	93,43	**95,67**	30 ± 1	35
QSAR Biodegradation	77,16	**77,87**	34 ± 4	41
Mobile Price Classification	79,92	**81,40**	16 ± 3	20
Page Blocks Classification	89,87	**90,08**	7 ± 2	10

mostra os resultados obtidos com cada conjunto de parâmetros dos algoritmos SVM-MRFS-W e SVM. Os classificadores foram comparados com a configuração que obteve o melhor desempenho.

Os resultados dos experimentos com SVM-MRFS-W e SVM são apresentados na Tabela 14. SVM-MRFS-W obteve a melhor acurácia em seis de oito experimentos. No conjunto *South African Health* o SVM sem o método de seleção de atributos teve o melhor resultado. No conjunto Cervical Cancer, ambos os modelos alcançaram a mesma acurácia média. A acurácia média mostra que o SVM-MRFS superou o SVM em mais de 1%.

Tabela 14 – Desempenho do SVM e do SVM-MRFS-W.

Conj. Dados	SVM	SVM-MRFS	Atrib. Selec.	Atrib.
Wine	95,55	**98,06**	6 ± 3	13
Glass Identification Dataset	98,60	**100,00**	5 ± 2	10
Heart Disease	83,28	**84,59**	8 ± 3	13
South African Health	**74,94**	74,62	7 ± 1	9
Cervical Cancer Risk	**95,75**	**95,75**	10 ± 1	35
QSAR Biodegradation	86,78	**87,25**	38 ± 1	41
Mobile Price Classification	97,32	**97,70**	17 ± 1	20
Page Blocks Classification	96,67	**96,72**	8 ± 1	10

4.4 Considerações do capítulo

Esta seção apresenta a síntese dos experimentos computacionais para avaliar e comparar o método de seleção de atributos MRFS que foi avaliado como um método *filter* (MRFS-F) e incorporado aos classificadores como um método *wrapper* (MRFS-W).

O MRFS-F foi avaliado e comparado com três métodos de seleção de atributos do estado da arte, em quatro classificadores e oito conjuntos de dados. Os experimentos computacionais mostram que o MRFS-F obteve o melhor desempenho (individual ou em conjunto com um ou mais métodos) em mais 70% dos experimentos.

Nos experimentos do MRFS-W, avaliou-se o ganho de desempenho do método incorporado aos classificadores em relação ao classificador sem a utilização do método. O MRFS-W obteve a maior acurácia em mais de 90% dos experimentos. Mais especificamente, nos experimentos com o KNN e NB, o MRFS-W obteve o melhor desempenho em todos os

conjuntos de dados. Nos experimentos com a RNA, o MRFS-W alcançou a maior a acurácia em 7 dos 8 conjuntos de dados e nos experimentos com SVM em 6 dos 8. Destaca-se que todos os conjuntos de dados utilizados são disponibilizados por plataformas e comunidades *online* para testes de classificação, no qual algumas bases de dados podem não possuir todos seus atributos, ou seja, tiveram alguns dos atributos considerados irrelevantes removidos antes de serem disponibilizados. Mesmo assim, o MRFS-W conseguiu reduzir o número de atributos e aumentar a assertividade na classificação.

Os resultados experimentais e as comparações sugerem o MRFS como uma abordagem promissora e eficiente para seleção de atributos, tanto como método o *filter* ou método *wrapper* de seleção de atributos. Destaca-se que o MRFS realiza apenas operações matemáticas básicas, possui baixo custo computacional e pode ser incorporado como método *wrapper* em, praticamente, qualquer classificador.

Capítulo 5

Classificador eFMC

Este capítulo detalha o classificador evolutivo eFMC (*evolving Fuzzy Mean Classifier*). A estrutura evolutiva do eFMC é baseada em grupos de atributos gerados dinamicamente por meio das médias amostrais dos atributos. O eFMC pode iniciar seu processamento sem conhecimento prévio e não necessita que o número de grupos seja definido previamente.

As próximas seções deste capítulo detalham as etapas do eFMC: a Seção 5.1 traz uma introdução aos conceitos do eMFC; a Seção 5.2 descreve o mecanismo de inicialização do classificador; a Seção 5.3 detalha o cálculo de pertinência entre a amostra e os grupos; a Seção 5.4 trata do processo de classificação; a Seção 5.5 descreve como os grupos são criados; a Seção 5.6 apresenta a abordagem para atualização dos grupos; e, a última seção (5.7) ilustra os parâmetros do eFMC, o seu fluxograma e algoritmo.

5.1 Introdução

O eFMC é um classificador evolutivo multiclasses que utiliza um algoritmo incremental de agrupamento supervisionado baseado nas médias amostrais. No algoritmo de agrupamento proposto para o eFMC cada classe está associada diretamente a um grupo e somente a um grupo. Os grupos são iniciados baseados no valor de uma amostra e têm seus centros atualizados pelo valor médio das amostras (média amostral). O classificador eFMC pode ser sumarizado nas seguintes 4 etapas:

(i) Inicialização dos Grupos: o eFMC inicia seu processamento com dois grupos, sendo um centrado na primeira amostra e o outro inicializado aleatoriamente. Esta etapa é realizada somente para a primeira amostra.

(ii) Cálculo do Grau de Pertinência: a cada amostra é calculado seu grau de pertinência em cada um dos grupos existentes.

(iii) Classificação: encontra-se o grupo com maior grau de pertinência e a amostra é classificada como pertencente a este grupo. Em seguida, é verificado se a amostra está classificada corretamente e decide pela criação de um novo grupo (Etapa v) ou pela atualização de um grupo existente (Etapa vi).

(iv) Criação de Grupos: um grupo é criado quando a amostra é classificada incorretamente e não existe nenhum grupo que represente a classe da amostra.

(v) Atualização de Grupos: se a amostra foi classificada corretamente, atualiza-se o grupo dessa amostra. Caso a amostra seja classificada incorretamente, a atualização é realizada no grupo correto referente a classe da amostra.

5.2 Inicialização do classificador

No algoritmo de agrupamento do eFMC, cada grupo representa uma única classe, e uma classe é representada por somente um grupo. Os grupos são criados à medida que amostras pertencentes a novas classes são apresentadas ao eFMC.

O eFMC inicia seu processamento com dois grupos. O centro do primeiro grupo (c_m^1) é definido pelo valor da primeira amostra (x_m^1) conforme Eq. (5).

$$c_m^1 = x_m^1. \tag{5}$$

O segundo centro pode ser inicializado de diferentes formas, como, por exemplo: com zeros em cada dimensão do espaço de entrada; ou com uma posição aleatoriamente diferente dos valores do centro do primeiro grupo. Destaca-se que esta etapa é realizada apenas uma vez. Os centros dos grupos c são representados por uma matriz:

$$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ & \cdots\cdots\cdots & & \\ c_{l1} & c_{l2} & \dots & c_{lm} \end{bmatrix} \tag{6}$$

na qual l é o número de classes, ou seja, o número de grupos; e m é a dimensão do espaço de entrada, isto é, o número de atributos de entrada.

Os centros dos grupos são atualizados com base no valor médio das amostras. Para isso, utiliza-se uma matriz de soma s com as mesmas dimensões da matriz c, que calcula incrementalmente a soma dos valores das amostras associadas a cada grupo como pode ser visto na Eq. (7).

$$s_m^1 = x_m^1. \tag{7}$$

Para controlar a quantidade de amostras na matriz de somatório e calcular a média, é iniciado também um contator $N_1 = 1$. O procedimento para atualizar os centros será melhor abordado nas próximas seções. Salienta-se que no eFMC o número de classes não precisa ser conhecido *a priori*.

5.3 Cálculo do grau de pertinência

O cálculo da pertinência de uma amostra com os grupos existentes é baseado na teoria dos conjuntos *fuzzy* (Zadeh, 1965), na qual uma amostra pode pertencer a diferentes grupos com diferentes graus de pertinência entre eles. A soma dos graus de pertinência de uma amostra em todos os grupos é sempre igual a 1. Assim, para cada amostra x_m^t, é calculado o grau de pertinência $\mu_i(x_m^t)$ para cada grupo i, em que x_m^t é a amostra de dados descrita como $[x_1^t \cdots x_j^t \cdots \ x_m^t]^T$, t é o passo atual, j é o índice dos atributos de entrada, m é o número de atributos de entrada, $i \in \{1, \cdots, l\}$ indexa os grupos e l é o número de grupos. O grau de pertinência $\mu_i(x_m^t)$ tem valor entre 0 e 1.

O grau de pertinência $\mu_i(x^t)$ é calculado usando a Distância Euclidiana como uma medida de dissimilaridade entre a amostra atual x^t e o centro dos grupos c. O uso da Distância Euclidiana se justifica devido ao seu baixo custo computacional e aos resultados obtidos em Nizam e Hassan (2020). Nos experimentos apresentados em Nizam e Hassan (2020) diferentes medidas de distância foram avaliadas nos algoritmos *K-Means* e *Fuzzy C-Means* e a Distância Euclidiana apresentou o menor tempo de processamento e uma boa precisão. Assim, usando a Distância Euclidiana, Eq. (8), como medida de dissimilaridade, o grau de pertinência $\mu_i(x^t)$ pode ser obtido pela Eq. (9).

$$d_i(x^t; c_i) = \sqrt{\sum_{j=1}^{m} |x_j^t - c_{ij}|^2}, \tag{8}$$

em que i indexa os grupos, j indexa os atributos de entrada, m é o número de atributos.

$$\mu_i(x^t) = \frac{\left(\frac{1}{d_i(x^t;c_i)}\right)^{\frac{1}{1-f}}}{\sum_{i=1}^{l}\left(\frac{1}{d_i(x^t;c_i)}\right)^{\frac{1}{1-f}}}, \tag{9}$$

na qual l é o número de grupos e f é um coeficiente de fuzzificação (Cox, 2005).

5.4 Classificação

A saída do eFMC, ou seja, a classificação, é obtida pelo grau de pertinência da amostra em relação aos grupos existentes. Para isso, encontra-se o índice i^* do grupo com maior grau de pertinência $\mu_i(x^t)$ habilitado por x^t. A classe associada ao grupo indexado por i^* é a classe prevista pelo eFMC.

Uma vez realizada a classificação, verifica se a classe prevista está correta. Em outras palavras, se a classe prevista pelo eFMC $\hat{y}^t$ é igual à classe desejada y^t. Então:

- Se $\hat{y}^t \neq y^t$, verifica se a classe desejada y^t é conhecida e:
  - caso a classe desejada não seja conhecida, crie um novo grupo centrado na amostra x^t para representar a classe (Seção 5.5); ou
  - caso a classe desejada seja conhecida, atualize o centro do grupo da classe desejada (Seção 5.6).
- Se $\hat{y}^t = y^t$, a classificação está correta e o centro do grupo indexado por i^* (grupo mais ativo) é atualizado conforme descrito na Seção 5.6.

5.5 Criação de novo grupo

No eFMC um grupo é criado sempre que uma nova classe é descoberta. Em outras palavras, um grupo é criado se a amostra atual for classificada incorretamente e não existir nenhum grupo que represente a classe da referida amostra.

Primeiramente, para criar um novo grupo, um novo centro c é adicionado com a projeção dos dados da amostra atual. Em seguida, o número de grupos l é atualizado. Finalmente, a matriz de soma s é ajustada com os valores da amostra atual.

A Figura 4 mostra amostras geradas aleatoriamente para ilustrar o comportamento do eFMC e os respectivos centros dos grupos. É possível observar, no canto inferior direito da Figura 4 (A), uma nova amostra que possivelmente não pertence a nenhum grupo existente. Isso sugere a descoberta de uma nova classe e a necessidade de criação de um novo grupo. A Figura 4 (B) ilustra o centro do grupo criado e as amostras pertencentes a este grupo.

5.6 Atualização dos grupos

Os centros dos grupos são atualizados com base na média das amostras. Para cada nova amostra x^t apresentada ao eFMC, identifica-se o grupo ao qual esta amostra pertence (indexado por $i^\diamond$) e soma-se o valor da amostra na matriz de soma s para cada um dos atributos m Eq. (10)

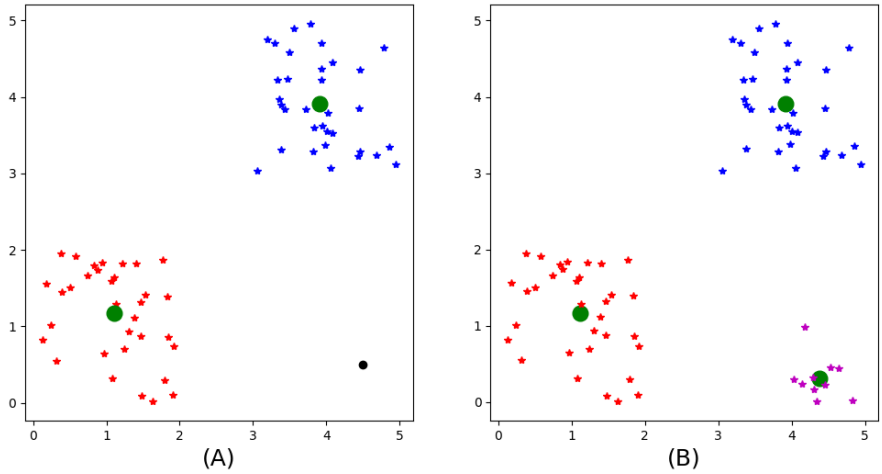

Figura 4 – Criação de grupos.

$$s_{i^\diamond j} = s_{i^\diamond j} + x_j^t. \tag{10}$$

Então, o centro $c_{i^\diamond}$ é atualizado para a média das amostras associadas ao grupo $i^\diamond$ Eq. (11)

$$c_{i^\diamond j} = \frac{s_{i^\diamond j}}{N_i^\diamond}, \tag{11}$$

em que $i^\diamond$ indexa o grupo, j indexa os atributos, m é o número de atributos, x é a amostra, t é o passo atual e $N_i^\diamond$ é o número de amostras associadas ao grupo $i^\diamond$.

Note que o centro dos grupos são atualizados a cada nova amostra apresentada ao eFMC. Os centros somente não são atualizados quando a amostra atual provoca a criação de um novo grupo. A Figura 5 ilustra um exemplo de atualização dos centros. Como pode ser observado na Figura 5 (A), os grupos possuem poucas amostras e estão centrados respectivamente em $[0,7; 0,18]$ e $[4,5; 4,4]$. A Figura 5 (B) ilustra os mesmos grupos após o processamento de novas amostras. Como pode ser visualizado, os centros foram atualizados para $[1; 1,1]$ e $[3,9; 3,8]$.

5.7 Parâmetros e algoritmo do eFMC

O eFMC é classificador evolutivo que utiliza um algoritmo incremental de agrupamento supervisionado baseado na razão da médias amostrais com uma única passada nos dados *(single pass)*. A cada amostra, o eFMC identifica a classe da amostra, cria um novo grupo ou atualiza um grupo existente. O eFMC possui um único parâmetro definido pelo usuário que é o f de fuzzificação, cujo valor normalmente é definido entre 1,25 e 2,00 (Cox, 2005).

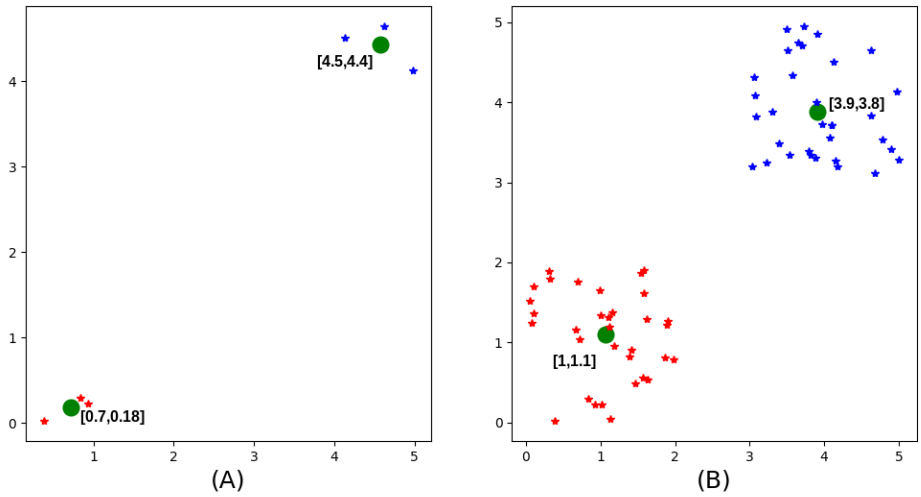

Figura 5 – Atualização do centro dos grupos.

A Figura 6 ilustra o fluxograma do eFMC enquanto o Algoritmo 3 detalha o método de aprendizado e classificação.

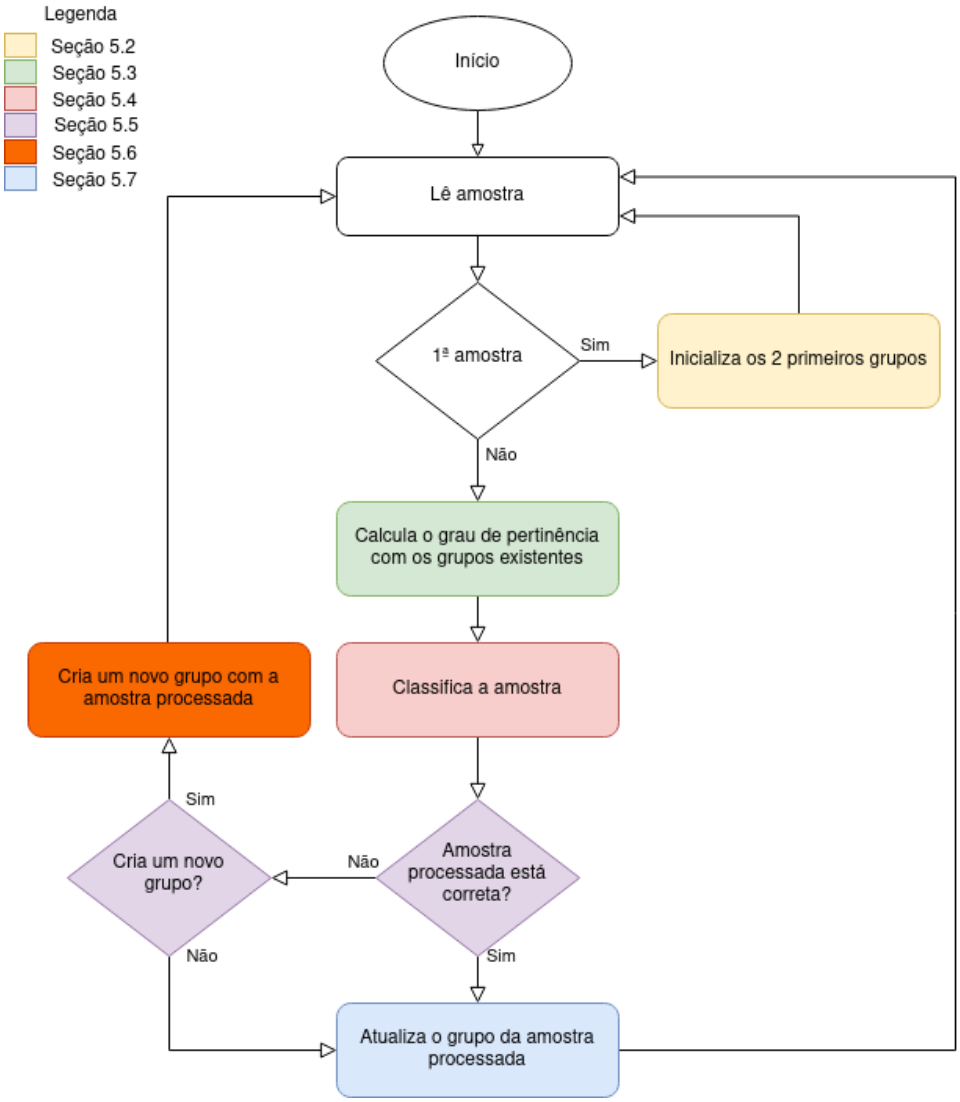

Figura 6 – Fluxograma do eFMC.

Algoritmo 3: Algoritmo do eFMC.

Entradas: x^t, y^t, f;
Saída: $\hat{y}^t$;
// x indexado por t do total T
for *t=1,2,...* **do**
    // Verifica se x^t é a primeira amostra
    if *($t == 1$)* **then**
        c = inicializa centros com 2 grupos;
        s = inicializa somatórios para 2 grupos;
        k = inicializa 2 grupos;
        c_1, s_1 = atualiza com os valores de x^1;
        c_2, s_2 = valores aleatórios ou zerados ;
        N_1 = 1, N_2 = 0 ;
    else
        Calcula distância Euclidiana - Eq. (8);
        Calcula o grau de pertinência - Eq. (9);
        Encontra i^*;
        $\hat{y}_t$ = *classe indexada por* i^*;
        if $\hat{y}^t \neq y^t$ **then**
            if *($\hat{y}^t ==$ nova classe)* **then**
                Atualiza nº de grupos $l = l + 1$;
                Cria novo grupo $c_{nj} = x_j^t$;
                Inicializa nº de amostras $N_l = 1$;
                Atualiza matriz de somatório $s_{nj} = x_j^t$;
            else
                $i^\diamond = y^t$;
                Atualiza matriz de somatório $s_{i^\diamond j}$ - Eq. (10);
                Atualiza nº de amostras $N_{i^\diamond} = N_{i^\diamond} + 1$;
                Atualiza centro dos grupo $c_{\diamond j}$ - Eq. (11);
            end
        else
            $i^\diamond = i^*$;
            Atualiza a matriz de somatório $s_{i^\diamond j}$ - Eq. (10);
            Atualiza o nº de amostras $N_{i^\diamond} = N_{i^\diamond} + 1$;
            Atualiza o centro dos grupos $c_{\diamond j}$ - Eq. (11);
        end
    end
end

Capítulo 6

Classificador eFC

O eFC (*evolving Fuzzy Classifier*) é um classificador evolutivo com alto grau de adaptabilidade e autonomia em seu aprendizado. A geração e atualização de centros para os diferentes grupos é realizada por meio da média das amostras, de maneira semelhante ao eFMC. No entanto, diferentemente do eFCM, no qual uma classe é representada somente por um grupo, no eFC uma classe pode ser representada por um ou mais grupos.

O algoritmo do eFC é detalhado nas seções seguintes. A Seção 6.1 apresenta uma breve introdução ao classificador. Em seguida, a Seção 6.2 ilustra as regras *fuzzy* e o processo de inicialização. A Seção 6.3 detalha o método utilizado para calcular o grau de pertinência entre a amostra e os grupos existentes. A etapa de classificação é o assunto da Seção 6.4. Posteriormente, a Seção 6.5 detalha o mecanismo proposto para atualização do centro dos grupos e a Seção 6.6 introduz a abordagem para criação e ativação de grupos e regras. A Seção 6.7 apresenta a metodologia para união de grupos e a Seção 6.8 detalha o processo para exclusão de grupos baseado no conceito de idade. Fechando o capítulo, a Seção 6.9 descreve os hiperparâmetros e apresenta o fluxograma e o algoritmo do eFC.

6.1 Introdução

O eFC é um classificador *fuzzy* evolutivo multiclasses que utiliza um algoritmo incremental de agrupamento baseado nas médias amostrais. Este classificador possui um alto grau de adaptabilidade e autonomia e inicia sua base de conhecimento do zero. A cada amostra apresentada é realizada a classificação paralelamente com a evolução da estrutura (criação e exclusão de grupos) e a atualização dos grupos. O eFC é composto por 3 camadas conforme ilustrado na Figura 7 e sumarizado como se segue:

- **Camada 1 - Cálculo do grau de pertinência:** cada grupo dessa camada gera graus de pertinência e especifica o quanto a amostra satisfaz cada um dos grupos. Os grupos representam o antecedente das regras *fuzzy* e, portanto, a saída dessa

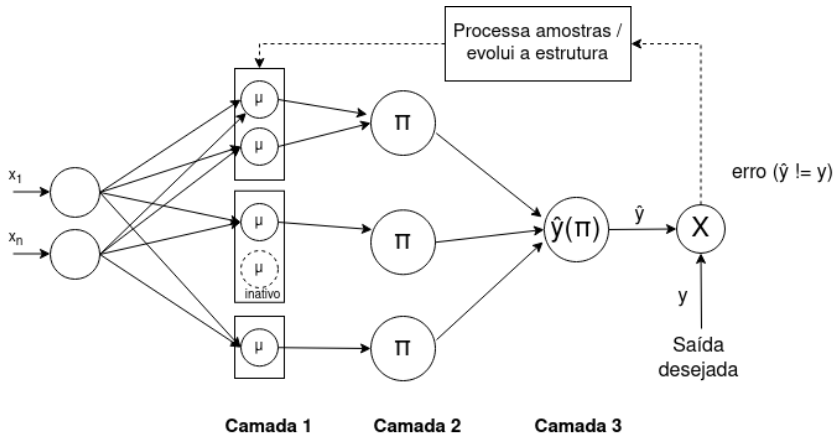

Figura 7 – Arquitetura do eFC.

camada é o grau de ativação das regras. O algoritmo de aprendizado do eFC pode incluir, excluir ou ativar os grupos à medida que novas informações são apresentadas ao classificador. É importante salientar que somente os grupos ativos serão utilizados na Camada 2.

- **Camada 2 - Agregação:** nesta camada é realizada a agregação das regras que possuem o mesmo consequente, i.e., regras associadas à mesma classe. Portanto, a quantidade de elementos nessa camada é igual ao número de classes. Note que o número de classes não precisa ser definido *a priori* e este é obtido automaticamente à medida que novas classes surgem. Para cada classe, encontra-se a regra que possui o maior grau de compatibilidade com a amostra e a saída desta camada é o grau de compatibilidade da amostra com cada uma das classes.
- **Camada 3 - Classificação:** esta camada identifica a classe que possui maior grau de compatibilidade com a amostra. Assim, a saída dessa camada é a classe prevista pelo classificador.

Como visto na Figura 7, o eFC é um classificador *fuzzy* no qual vários grupos podem representar uma classe. Essa característica permite mapear várias regiões do espaço multidimensional para uma mesma classe. Na abordagem proposta, os grupos são inicializados centrados na amostra e podem ter seu centro atualizado com base no valor médio das amostras. O algoritmo utiliza um mecanismo robusto para evitar *outliers*, baseado no conceito de procrastinação (Lughofer; Pratama; Skrjanc, 2018b), no qual os grupos são criados inativos, sem associação a uma regra *fuzzy* e, portanto, não contribuem para a classificação. Para evitar a criação excessiva de grupos são utilizados dois mecanismos de poda. O primeiro é a união que visa fundir grupos/regras similares que representam a mesma classe. Por outro lado, a exclusão de grupos usa o conceito de idade para excluir os grupos obsoletos e pouco representativos. Destaca-se que o algoritmo do eFC realiza todos os cálculos recursivamente e utiliza apenas equações simples e com baixo custo

computacional.

O eFC utiliza algumas equações que são utilizadas pelo eFMC e foram apresentadas no Capítulo 5. No entanto, visando facilitar a leitura e a compreensão do eFC optou-se por descrever novamente essas equações.

6.2 Regras *fuzzy* e inicialização do modelo

O algoritmo do classificador evolutivo multiclasses proposto neste trabalho utiliza o valor médio das amostras para atualizar o centro dos grupos (Tavares; Silva; Moita, 2022) e pode, a cada amostra, criar um grupo, excluir, unir ou atualizar um grupo existente. Os grupos representam o antecedente das regras *fuzzy* cujo consequente é uma das classes. Destaca-se que, cada grupo está associado a uma e somente uma classe, e uma classe está associada a um ou mais grupos. As regras *fuzzy* são:

$$\begin{cases} R_1: & \textit{Se}\ x^t\ \textit{é}\ C_1\ \textit{Então}\ y_1 = Classe_1 \\ R_2: & \textit{Se}\ x^t\ \textit{é}\ C_2\ \textit{Então}\ y_2 = Classe_2 \\ R_3: & \textit{Se}\ x^t\ \textit{é}\ C_3\ \textit{Então}\ y_3 = Classe_1 \\ & \vdots \\ R_i: & \textit{Se}\ x^t\ \textit{é}\ C_i\ \textit{Então}\ y_i = Classe_1 \\ & \vdots \\ R_l: & \textit{Se}\ x^t\ \textit{é}\ C_l\ \textit{Então}\ y_l = Classe_k, \end{cases}$$

em que x^t é a amostra de dados em t descrita como $[x_1^t \cdots x_j^t \cdots x_m^t]^T$, j é o índice dos atributos de entrada, m é o número de atributos de entrada, C representa os grupos, i indexa os grupos, l é o número de grupos/regras e y_i é o consequente da *i*-ésima regra e define a qual classe *i*-ésimo grupo pertence, k indexa a classe.

No eFC, a primeira amostra é usada para criar o primeiro grupo, cujo centro é definido pelos valores da amostra, isto é:

$$c_m^1 = x_m^1. \tag{12}$$

Ao iniciar o primeiro grupo, inicia-se, também, uma matriz de somatório s. A matriz de somatório s é atualizada a cada amostra e armazena a soma de todas as amostras processadas por cada um dos grupos. A matriz de somatório s, assim como o centro do primeiro grupo, é iniciada com o valor da primeira amostra, como em:

$$s_m^1 = x_m^1. \tag{13}$$

Após a criação da matriz de somatório inicializa-se o número de amostras do primeiro grupo com 1 ($N_1 = 1$) e em seguida cria-se a primeira regra *fuzzy*.

6.3 Cálculo do grau de pertinência

A cada amostra x^t, é calculado o seu grau de pertinência $\mu_i(x^t)$ para cada um dos l grupos. Para o cálculo do grau de pertinência utiliza-se a Distância Euclidiana como uma medida de dissimilaridade entre a amostra atual x^t e os centros dos grupos c_i^t, Eq. (14)

$$d_i(x^t; c_i^t) = \sqrt{\sum_{j=1}^{m} |x_j^t - c_{ij}^t|^2}, \tag{14}$$

na qual i indexa os centros dos grupos, j indexa os atributos de entrada e m é o número de atributos de entrada.

A medida do grau de pertinência para os grupos utilizada neste trabalho é, *mutatis mutandis*, a mesma empregada no *Fuzzy C-Means* (Dunn, 1973; Bezdek, 1981). O grau de pertinência $\mu_i(x^t)$ pode ser obtido pela Eq. (15)

$$\mu_i(x^t) = \frac{(\frac{1}{d_i(x^t;c_i^t)})^{\frac{1}{1-f}}}{\sum_{i=1}^{l}(\frac{1}{d_i(x^t;c_i^t)})^{\frac{1}{1-f}}}, \tag{15}$$

em que l é o número grupos e f é um parâmetro de fuzzificação (Cox, 2005), cujo o valor normalmente utilizado é 1,25 ou 2. O processo do cálculo do grau de pertinência é ilustrado no Algoritmo 4.

Algoritmo 4: Procedimento para cálculo do grau de pertinência.

```
// Para cada grupo
for i=1,2,...,l do
    Calcule a distância Euclidiana d_i(x^t; c_i^t) - Eq. (14) ;
    Calcule o grau de pertinência μ_i(x^t) - Eq. (15) ;
end
```

6.4 Agregação e classificação

Após o cálculo do grau de ativação para cada um dos l grupos, é realizada a agregação das regras que possuem o mesmo consequente. Em outras palavras, encontra-se, para

cada uma das K classes, o índice do grupo com maior grau de compatibilidade i_k^*, em que $i_k^* = arg\ max_k(\mu_i(x^t))$, com $k = 1, ..., K$, sendo K o número de classes.

A etapa de classificação é responsável por identificar a classe que possui a maior similaridade com a amostra x^t. Assim, a classificação é realizada encontrando, entre os índices i_k^*, o índice da classe com maior grau de compatibilidade k^*, isto é, $k^* = arg\ max(i_k^*)$. A saída do classificador $\hat{y}^t$ será representada pela classe indexada por k^* ($\hat{y}^t = k^*$). Após a classificação é verificado se a amostra foi classificada corretamente. Assim:

- se $\hat{y}^t = y^t$, a amostra foi classificada corretamente e o grupo com maior grau de pertinência (indexado por i^*) será atualizado conforme descrito na Seção 6.5;
- se $\hat{y}^t \neq y^t$, a amostra foi classificada incorretamente e será utilizada para ativar um grupo ou criar um grupo inativo conforme apresentado na Seção 6.6.

O Algoritmo 5 detalha o método proposto para agregação das regras e classificação.

Algoritmo 5: Procedimento para agregação das regras e classificação.

```
// Agregação das regras
Encontre i_k^* ;
// Classe com maior compatibilidade
Encontre k^* ;
Classifique a amostra ŷ^t = k^* ;
// Verifica classificação
if (ŷ^t = y^t) then
    Atualiza o grupo com maior compatibilidade (indexado por i^*) - Algoritmo 6 ;
else
    Ativa um grupo ou cria um grupo inativo - Algoritmo 7;
end
```

6.5 Atualização de grupos

Na abordagem proposta os centros dos grupos são atualizados com base no valor médio das amostras, conforme Tavares, Silva e Moita (2022). Assim, caso a amostra seja classificada corretamente, o centro do grupo mais ativo (grupo indexado por i^*) é atualizado. Para isso, inicialmente atualiza-se a matriz de somatório $s_m^{i^*}$ por:

$$s_m^{i^*} = \sum_{j=1}^{m} s_j^{i^*} + x_j^t. \tag{16}$$

Após a atualização da matriz de somatório, atualiza-se o número de amostras do grupo ($N_{i^*} = N_{i^*} + 1$). Em seguida, o centro $c_m^{i^*}$ é atualizado pela média de todas as amostras do grupo i^* pela Eq. (17). O Algoritmo 6 apresenta o procedimento para atualização do centro do grupo i^*.

$$c_m^{i^*} = \frac{s_m^{i^*}}{N_i^*}. \tag{17}$$

Algoritmo 6: Procedimento para atualização do centro dos grupos.

Atualiza a matriz de somatório $s_m^{i^*}$ - Eq. (16) ;
Atualiza o número de amostras $N_{i^*} = N_{i^*} + 1$;
Atualiza o centro $c_m^{i^*}$ - Eq. (17) ;

6.6 Criação ou ativação de grupos

Esta seção descreve o método para criação de grupos que utiliza o conceito de procrastinação (Lughofer; Pratama; Skrjanc, 2018b). A ideia central da procrastinação é que os grupos sejam criados inativos e sem associação a uma regra até sua ativação ou exclusão, sua exclusão segue os mesmos critérios de exclusão dos grupos ativos descrito na Seção 6.8. Portanto, os grupos recém-criados não contribuem para a saída do classificador. Na abordagem proposta, um grupo é criado ou ativado sempre que uma amostra é classificada incorretamente ($\hat{y}^t \neq y^t$). Assim, após uma amostra ser classificada incorretamente é verificado se existe algum grupo inativo, ou seja, se o número de grupos inativos $G > 0$:

- Caso $G = 0$, um novo grupo inativo será criado e associado a classe y^t da amostra x^t. O centro do grupo inativo recém-criado recebe os valores da amostra conforme:

$$ci_m^1 = x_m^t. \tag{18}$$

  Em seguida, a matriz de somatório de grupos inativos si é iniciada por:

$$si_m^1 = x_m^t. \tag{19}$$

  Por fim, z_1 recebe a classe y^t da amostra x^t e o número de grupos inativos G é definido com 1.
- Caso $G > 0$, calcula-se a distância Euclidiana entre a amostra x^t e o centro de todos os G grupos inativos, ou seja, calcula-se $d_p(x^t; ci_p^t)$, $\forall p = 1, ..., G$. Em seguida, encontra-se o grupo inativo com a maior compatibilidade (menor distância) com a amostra x^t (indexado por p^*) e a classe deste p^*-ésimo grupo inativo.
  - se a classe do grupo indexado por p^* for diferente classe da amostra x^t ($z_{p^*} \neq y^t$), um novo grupo é criado e o número de grupos inativos é atualizado ($G = G + 1$). Em seguida, z_G recebe a classe da amostra atual x^t e o centro do novo grupo inativo e a matriz de somatório são inicializados conforme:

$$ci_m^G = x_m^t. \tag{20}$$

$$si_m^G = x_m^t. \tag{21}$$

– se a classe do grupo indexado por p^* for igual classe da amostra x^t ($z_{p^*} = y^t$), o grupo inativo indexado por p^* é ativado. Inicialmente, atualiza-se o número de grupos ($l = l + 1$). Em seguida, define-se o centro do novo grupo ativo como em:

$$c_m^l = \frac{x_m^t + ci_m^{p^*}}{2}. \tag{22}$$

Posteriormente, a matriz de somatório para o grupo recém ativado é obtida por:

$$s_m^l = x_m^t + si_m^{p^*}, \tag{23}$$

e o número de amostras do grupo é atualizado por $N_l = 2$. Depois, cria-se uma nova regra na qual o antecedente é o grupo recém ativado e o consequente é a classe da amostra atual. Por fim, atualiza-se o número de grupos inativos ($G = G - 1$), os índices e as classes, se necessário. Após a ativação de um grupo é realizada a checagem por grupos redundantes. O Algoritmo 7 sumariza o método para criação e ativação de grupos.

Algoritmo 7: Procedimento para criar ou ativar grupos.

```
//
if (G = 0) then
    Inicialize o centro do grupo inativo - Eq. (18) ;
    Inicialize a matriz de somatório do grupo inativo - Eq. (19) ;
    Armazene em z_1 a classe do grupo inativo - z_1 = y^t;
    Atualize o número de grupos inativos - G = 1;
else
    Calcule a distância Euclidiana - d_p(x^t; ci_p^t), ∀p = 1, ..., G;
    Encontre o grupo com maior compatibilidade (grupo indexado por p*);
    Encontre a classe do grupo com maior compatibilidade- z_p^*;
    if (z_{p*} ≠ y^t) then
        Atualize o número de grupos inativos - G = G + 1;
        Inicialize o centro do grupo inativo - Eq. (20) ;
        Inicialize a matriz de somatório do grupo inativo - Eq. (21) ;
        Armazene em z_G a classe do grupo inativo - z_G = y^t;
    else
        Atualiza número de grupos ativos - l = l + 1 ;
        Inicialize o centro do grupo ativo - Eq. (22) ;
        Inicialize a matriz de somatório do grupo ativo - Eq. (23) ;
        Atualiza número de amostras do grupo ativo - N_l = 2 ;
        Crie a l-ésima regra fuzzy;
        Atualize o número de grupos inativos - G = G − 1;
        Atualize os índices;
        Checagem de grupos redundantes - Algoritmo 8;
    end
end
```

6.7 União de grupos

O método proposto para união de grupos visa eliminar grupos redundantes que representam uma mesma classe, isto é, que possuem o mesmo consequente. Mais formalmente, se dois grupos possuem a menor distância Euclidiana entre todos os pares de grupos de uma mesma classe, eles poderão ser unidos.

Inicialmente, calcula-se a distância Euclidiana entre os pares de grupos. Em seguida, encontra-se o par de grupo (indexado por $i^{\bullet}$ e $i^{\diamond}$) que possui a menor distância Euclidiana. Os grupos indexados por $i^{\bullet}$ e $i^{\diamond}$ serão unidos se possuírem o mesmo consequente, i.e., se $y_{i^{\bullet}} = y_{i^{\diamond}}$.

O centro do novo grupo $c_{i^{\bullet} \cup i^{\diamond}}$ será definido com base no conceito matemático do ponto médio e é obtido por:

$$c_{i^{\bullet} \cup i^{\diamond}} = \frac{c_{i^{\bullet} j} + c_{i^{\diamond} j} \; \forall \; j}{2}. \tag{24}$$

Em seguida, a matriz de somatório é atualizada por:

$$s_{i^{\bullet} \cup i^{\diamond}} = s_{i^{\bullet} j} + s_{i^{\diamond} j} \; \forall \; j. \tag{25}$$

O número de amostras do novo grupo é obtido pela soma do número de amostras dos dois grupos unidos, isto é:

$$N_{i^{\bullet} \cup i^{\diamond}} = N_{i^{\bullet}} + N_{i^{\diamond}}. \tag{26}$$

Após a união dos grupos, o número de grupos, a regra *fuzzy* e os índices são atualizados. Destaca-se que neste trabalho a análise para união de grupos é realizada somente após a ativação de um grupo. O processo de união de grupos é detalhado no Algoritmo 8.

Algoritmo 8: Procedimento para união de grupos.

```
Calcule a distância entre os pares de grupos;
Encontre o par de grupo com a menor distância (indexados por i• e i◇) ;
if (y_i• == y_i◇) then
    Crie o centro - Eq. (24) ;
    Crie a matriz de somatório - Eq. (25) ;
    Inicialize o número de amostras - Eq. (26) ;
    Atualize índices, grupos e regras;
end
```

6.8 Exclusão de grupos

O eFC utiliza um método baseado no conceito de idade para exclusão de grupos. Este conceito foi introduzido em modelos evolutivos no trabalho de Angelov e Filev (2005), estendido em Lughofer e Angelov (2011) e utilizado em diversos trabalhos como em Silva et al. (2013a), Rodrigues, Silva e Lemos (2021), Rodrigues, Silva e Lemos (2022). Neste trabalho a idade de um grupo é obtida como se segue:

- se a amostra foi classificada corretamente, o grupo indexado por i^* tem sua idade definida como $idade_{i^*}^t = 0$ e para os demais grupos a idade é incrementada em 1, isto é, $idade_i^t = idade_i^{t-1} + 1, \forall\, i\ com\ i = 1...l\ e\ i \neq i^*$.
- se a amostra foi classificada incorretamente todos os grupos têm sua idade incrementada em 1, ou seja, $idade_i^t = idade_i^{t-1} + 1, \forall\, i\ com\ i = 1...l$.

A cada nova amostra x^t encontra-se o grupo com maior idade (indexado por i^+). O grupo indexado por i^+ será excluído se sua idade for maior que um limiar w e número de grupos associados a classe do grupo i^+ for igual ou maior que dois. Mais especificamente, o grupo indexado por i^+ será excluído se:

$$idade_{i^+}^t \geq w\ e\ n_{k_{i^+}} \geq 2, \tag{27}$$

em que $n_{k_{i^+}}$ é o número de grupos associados a classe do grupo indexado por i^+. Após eliminar um grupo, os índices e regras são atualizados. O Algoritmo 9 detalha o processo de exclusão de um grupo.

Algoritmo 9: Procedimento para exclusão de grupos.

```
// Atualiza idade dos grupos
if (ŷ^t = y^t) then
    age_{i*}^t = 0;
    age_i^t = age_i^{t-1} + 1, ∀ i com i = 1...l e i ≠ i*;
else
    age_i^t = idade_i^{t-1} + 1, ∀ i com i = 1...l;
end
Encontre o grupo com maior idade (indexado por i+);
// Exclusão do grupo
if (age_{i+}^t ≥ w e n_{k_{i+}} ≥ 2) then
    Exclui o centro do grupo - c_{i+}^t ;
    Exclui o somatório do grupo - s_{i+}^t;
    Exclui o número de amostra - N_{i+};
    Exclui a regra fuzzy - R_{i+};
    Atualiza número de grupos da classe - n_{k_{i+}};
    Atualiza índices;
end
```

6.9 Parâmetros e algoritmo do eFC

O algoritmo de aprendizado do eFC possui somente dois parâmetros:

- f é um parâmetro de fuzzificação, cujo valor normalmente é 1,25 ou 2,00 (Cox, 2005).
- ω é o limite de idade para exclusão de grupos e está associado a capacidade do modelo esquecer conhecimento antigo sempre que este se torna inútil. O valor de ω é tipicamente escolhido avaliando a aplicação do sistema em conjuntos de dados de maior periodicidade, frequência que as informações chegam e até memória disponível para o processamento.

O Algoritmo 10 detalha o procedimento de aprendizado do eFC. A Figura 8 ilustra o fluxograma da abordagem proposta.

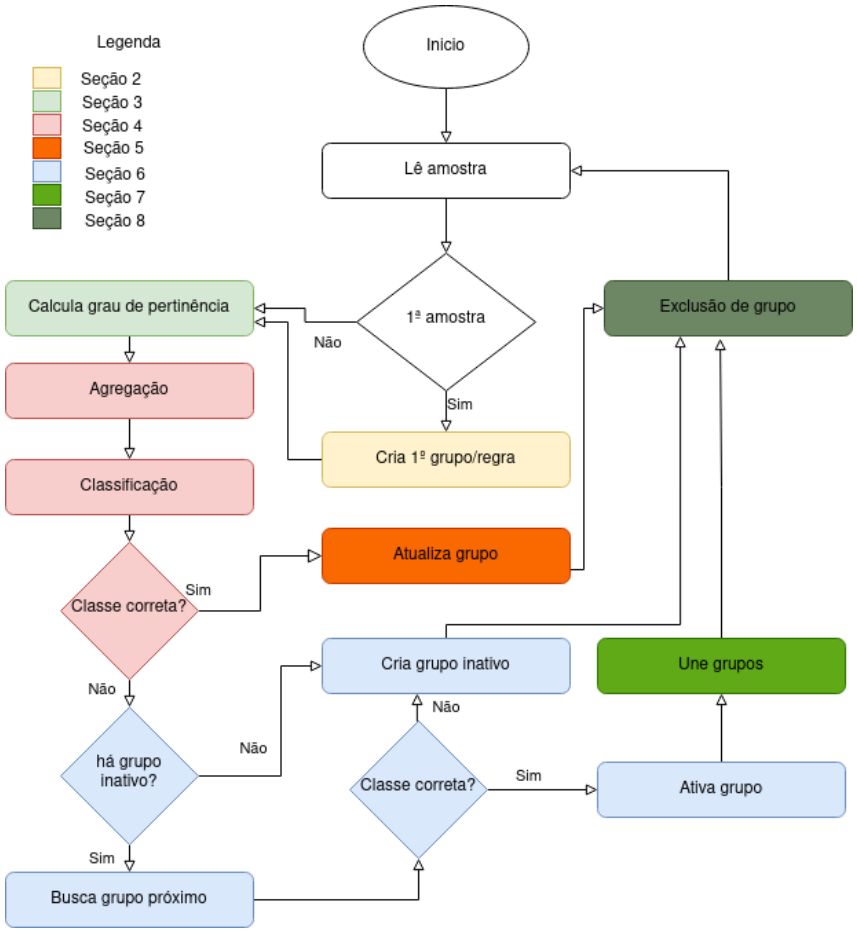

Figura 8 – Fluxograma do eFC.

Algoritmo 10: Algoritmo do eFC.

Entradas: x^t, y^t, ω, f;
Saída: $\hat{y}^t$;
for *t=1,2,...* **do**
    if $(t == 1)$ **then**
        Inicializa o centro do grupo - Eq. (12);
        Inicializa a matriz de somatório - Eq. (13);
        Inicializa número de amostras do grupo - $N_1 = 1$;
        Atualiza idade do grupo - $idade_1 = 0$;
        Atualiza índices;
    else
        Proced. Cálculo do grau de pertinência - Alg. 4;
        Proced. Agregação e Classificação - Alg. 5;
        `// Executado se` $\hat{y}^t = y^t$
        Proced. Atualização - Alg. 6 ;
        `// Executado se` $\hat{y}^t \neq y^t$
        Proced. Criação ou ativação - Alg. 7 ;
        `// Executado se` $\hat{y}^t \neq y^t$
        Proced. União de grupos - Alg. 8 ;
        Proced. Exclusão de grupos - Alg. 9
    end
end

Capítulo 7

Modelo evolutivo eFC-FS

Este capítulo introduz o eFC-FS (*evolving Fuzzy Classifier with Features Selection*). O eFC-FS é um classificador evolutivo multiclasse construído sob a estrutura do eFC que incorpora o método de seleção de atributos MRFS (*Mean Ratio for Feature Selection*) em seu processo de aprendizado incremental.

As seções do capítulo estão organizadas como se segue. A Seção 7.1 apresenta uma breve introdução ao classificador. Em seguida, a Seção 7.2 detalha a metodologia de seleção de atributos. Por fim, a Seção 7.3 discorre sobre os parâmetros e apresenta o fluxograma e o algoritmo do eFC-FS.

7.1 Introdução

O eFC-FS é um classificador evolutivo com seleção adaptativa de atributos incorporada ao seu processo de aprendizado. O mecanismo de seleção de atributos do eFC-FS é ativado sempre que uma amostra é classificada incorretamente. A Figura 9 ilustra o diagrama esquemático da estrutura de aprendizado e seleção de atributos do eFC-FS e que pode ser sumarizada como segue:

- **(i) Classificação:** o eFC-FS adota o mesmo procedimento de inicialização e classificação do eFC (vide seções 6.2, 6.3 e 6.4). Portanto, estes não serão detalhados neste capítulo.
- **(ii) Check Classificação:** verifica se a amostra foi classificada corretamente: **caso positivo**, atualiza o grupo (procedimento descrito na Seção 6.5) e segue para a etapa (i); **caso negativo**, i.e., a amostra foi classificada incorretamente, inicia o procedimento de seleção de atributos - etapa (iii).
- **(iii) Check Seleção de Atributos:** verifica se o número de grupos associados às classes é maior do que dois: **caso positivo**, realiza o ranqueamento dos atributos proposto pelo MRFS (descrito no Capítulo 3) e segue para a etapa (iv); **caso negativo**,

não haverá um mínimo de dois grupos distintos para se calcular a razão entre as médias, seguindo para a Seção 6.6;

- **(iv) Exclusão de Atributos:** gera um modelo um modelo candidato com $m - j^{\diamond}$ atributos de entrada e verifica a classificação da amostra. Se a classificação for correta, armazena os atributos bloqueados $\epsilon = \epsilon + j^{\diamond}$ e executa a etapa (vi). Caso contrário, executa a etapa (v);
- **(v) Inclusão de Atributos:** gera um modelo um modelo candidato com $m + j^{\bullet}$ atributos de entrada e verifica a classificação. Se for correta, desbloqueia o atributo $j^{\bullet}$, atualiza o conjunto de atributos bloqueados $\epsilon = \epsilon - j^{\bullet}$ e segue para etapa (vi);
- **(vi) Aprendizado e Evolução da Estrutura:** realiza as etapas de atualização, ativação, geração, união ou exclusão dos grupos descritas nas seções 6.5, 6.6, 6.7 e 6.8.

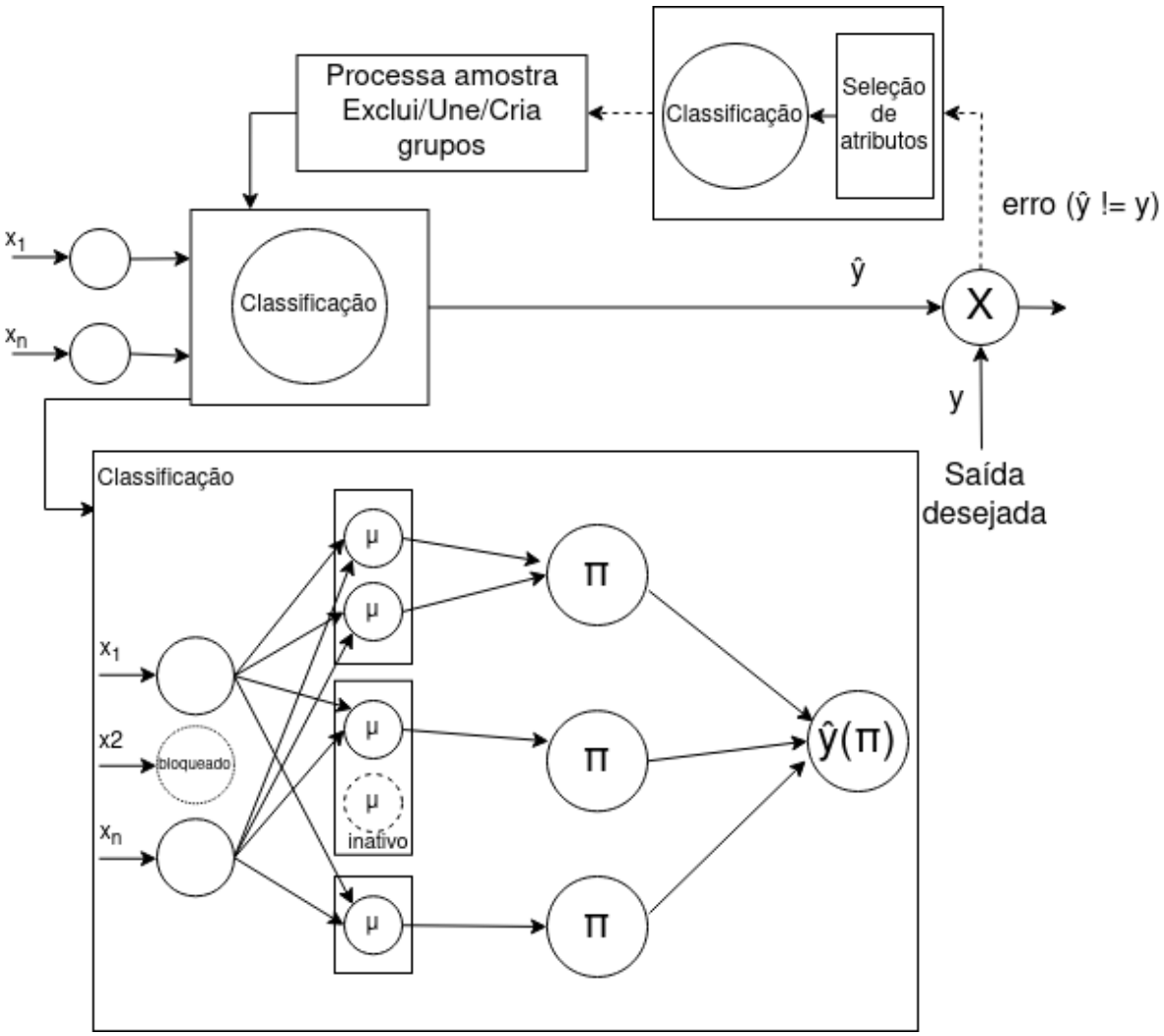

Figura 9 – Arquitetura do eFC-FS.

7.2 Seleção de atributos

Na metodologia para seleção de atributos de entrada do eFC-FS, inicialmente todos os atributos são processados e classificados. Novos grupos são gerados e atualizados, seguindo os mesmos procedimentos do eFC (Capítulo 6). O processo para se obter a razão das médias e ranquear os atributos de entrada é iniciado quando há dois ou mais grupos c associados a mais de uma classe de saída, $K > 1$.

A seleção de atributos ocorre sempre que uma amostra é classificada incorretamente, i.e., $\hat{y}_t \neq y_t$. Assim, quando uma amostra é classificada errada e há mais de duas classes de saída, primeiro eFC-FS calcula a média resultante $\bar{M}$ de todos os centros c pertencentes a uma mesma classe k. A média resultante $\bar{M}$ é calculada para todos atributos de entrada que estão associados aos centros de mesma classe de saída por:

$$\bar{M}_m^k = \sum_{i=1}^{l} (\frac{c_{im}}{N_i} \in k), \tag{28}$$

na qual $\bar{M}$ armazena as médias de cada classe k para cada m atributos de entrada, c são os centros dos grupos, l é a quantidade de grupos ativos indexados por i e N o contador de atributos de cada centro.

Para o ranqueamento dos atributos, a razão entre as médias resultantes $\bar{M}$ é aplicada de acordo com o método MRFS. Assim, a razão entre as médias resultantes pode ser representada por:

$$\Phi_m = \sum_{k=2}^{K} (min(\frac{\bar{M}_m^k}{\bar{M}_m^{k-1}}, \frac{\bar{M}_m^{k-1}}{\bar{M}_m^k})), \tag{29}$$

em que Φ é uma consequência direta da razão entre dois valores, onde a fração ou porcentagem é dada pelo menor valor dividido pelo maior valor, ou seja, a média resultante da classe k por $k-1$ ou $k-1$ por k.

Por fim, os valores obtidos pela razão das médias resultantes Φ serão utilizados para ranquear os atributos de entrada. O ranqueamento é realizado seguindo os princípios do MRFS descritos no Capítulo 3. Na abordagem utilizada pelo eFC-FS, um atributo de entrada não precisa ser excluído permanentemente. Eles podem ser bloqueados ou desbloqueados de acordo com o fluxo de dados processados.

Desta forma, após ranqueamento dos atributos de entrada, o eFC-FS executa uma nova classificação da amostra atual sem o atributo $j^\diamond$ considerado menos representativo e candidato ao seu bloqueio. Se a amostra sem o atributo candidato ao bloqueio for classificada

corretamente, o atributo menos representativo é armazenado, $\epsilon = \epsilon + j^{\diamond}$. Somente os $m = m - j^{\diamond}$ atributos considerados relevantes serão utilizados na classificação de novas amostras.

Caso a classificação sem o atributo candidato ao bloqueio continuar errada, $\hat{y}_t \neq y_t$. O eFC-FS irá realizar uma nova classificação e verificar se um atributo bloqueado deve voltar ao conjunto de atributos processados, indexado por $j^{\bullet}$. Se após a inclusão de um atributo a amostra for classificada corretamente, este atributo voltará a fazer parte dos atributos processados pelo eFC-FS, atualizando o conjunto de atributos $m = m + j^{\bullet}$ e a variável que armazena os atributos bloqueados $\epsilon = \epsilon - j^{\bullet}$.

O processo de verificação se um atributo é candidato ao bloqueio, só ocorre quando a classificação é incorreta. Consequentemente, a verificação se um atributo bloqueado é candidato a fazer parte do conjunto de atributos processados, ocorre somente quando a classificação é errada na verificação de um atributo candidato ao bloqueio. Em outras palavras: se o eFC-FS realiza uma classificação incorreta, primeiro o modelo verifica se a classificação da amostra atual seria correta com o bloqueio de um atributo; caso o bloqueio não tenha sucesso, o modelo tenta desbloquear um atributo bloqueado. O fluxo para o processo de bloqueio ou desbloqueio de atributos é apresentado na Figura 10.

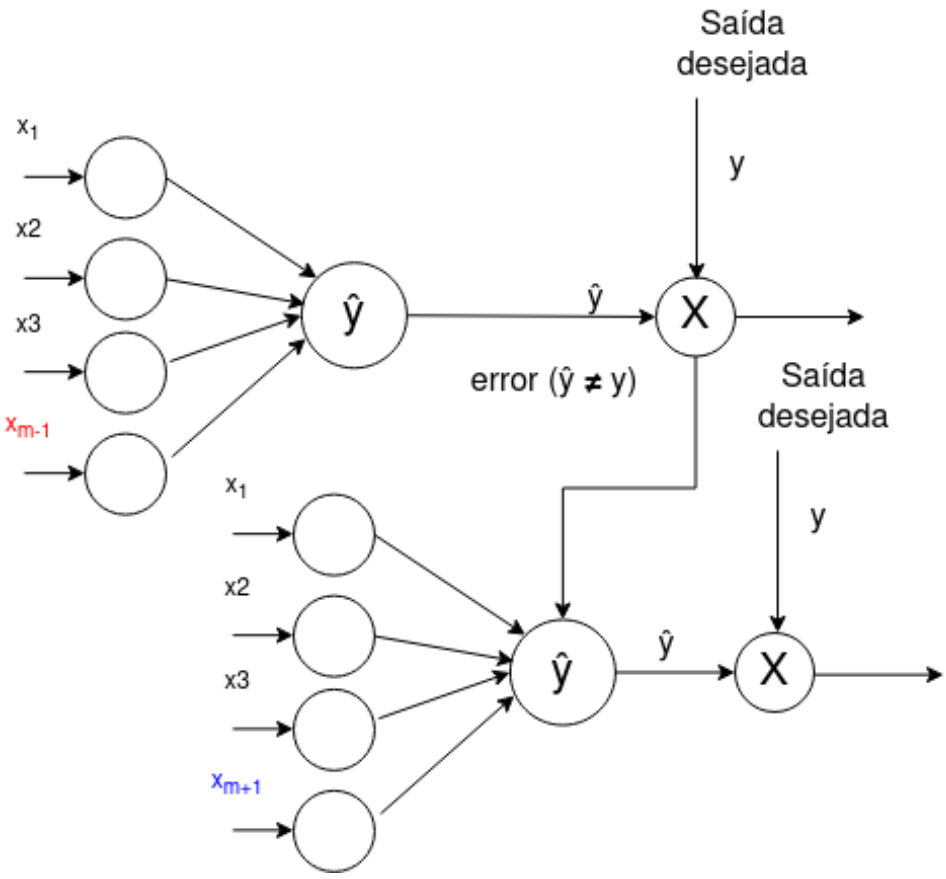

Figura 10 – Fluxo para o bloqueio ou desbloqueio de atributos de entrada no eFC-FS.

7.3 Parâmetros e algoritmo do eFC-FS

O algoritmo do eFC-FS possui somente dois parâmetros:

- f é um parâmetro de fuzzificação, cujo valor normalmente é definido entre 1,25 e 2,00 (Cox, 2005).
- ω é o limite de idade para exclusão de grupos e está associado a capacidade do modelo esquecer conhecimento antigo sempre que este se torna inútil.

O Algoritmo 11 detalha o procedimento de aprendizado e seleção de atributos do eFC-FS que é ilustrado graficamente pelo fluxograma apresentado na Figura 11.

Algoritmo 11: Algoritmo do eFC-FS.

Entradas: x^t, y^t, ω, f;
Saída: $\hat{y}^t$;
for *t=1,2,...* **do**
  if $(t == 1)$ **then**
    Inicializa o centro do grupo - Eq. (12) ;
    Inicializa a matriz de somatório - Eq. (13);
    Inicializa número de amostras do grupo - $N_1 = 1$;
    Atualiza idade do grupo - $idade_1 = 0$;
    Atualiza índices;
  else
    Proced. Cálculo do grau de pertinência - Alg. 4;
    Proced. Agregação e Classificação - Alg. 5;
    `// Executado se` $\hat{y}^t = y^t$
    Proced. Atualização - Alg. 6 ;
    `// Executado se` $\hat{y}^t \neq y^t$
    Proced. de ranqueamento dos atributos - Alg. 1;
    Proced. classificação com $m - 1$ atributo - Algo. 4 e 5 ;
    Proced. classificação com $m + 1$ atributo - Algo. 4 e 5 ;
    Proced. Criação ou ativação - Alg. 7 ;
    `// Executado se` $\hat{y}^t \neq y^t$
    Proced. União de grupos - Alg. 8 ;
    Proced. Exclusão de grupos - Alg. 9
  end
end

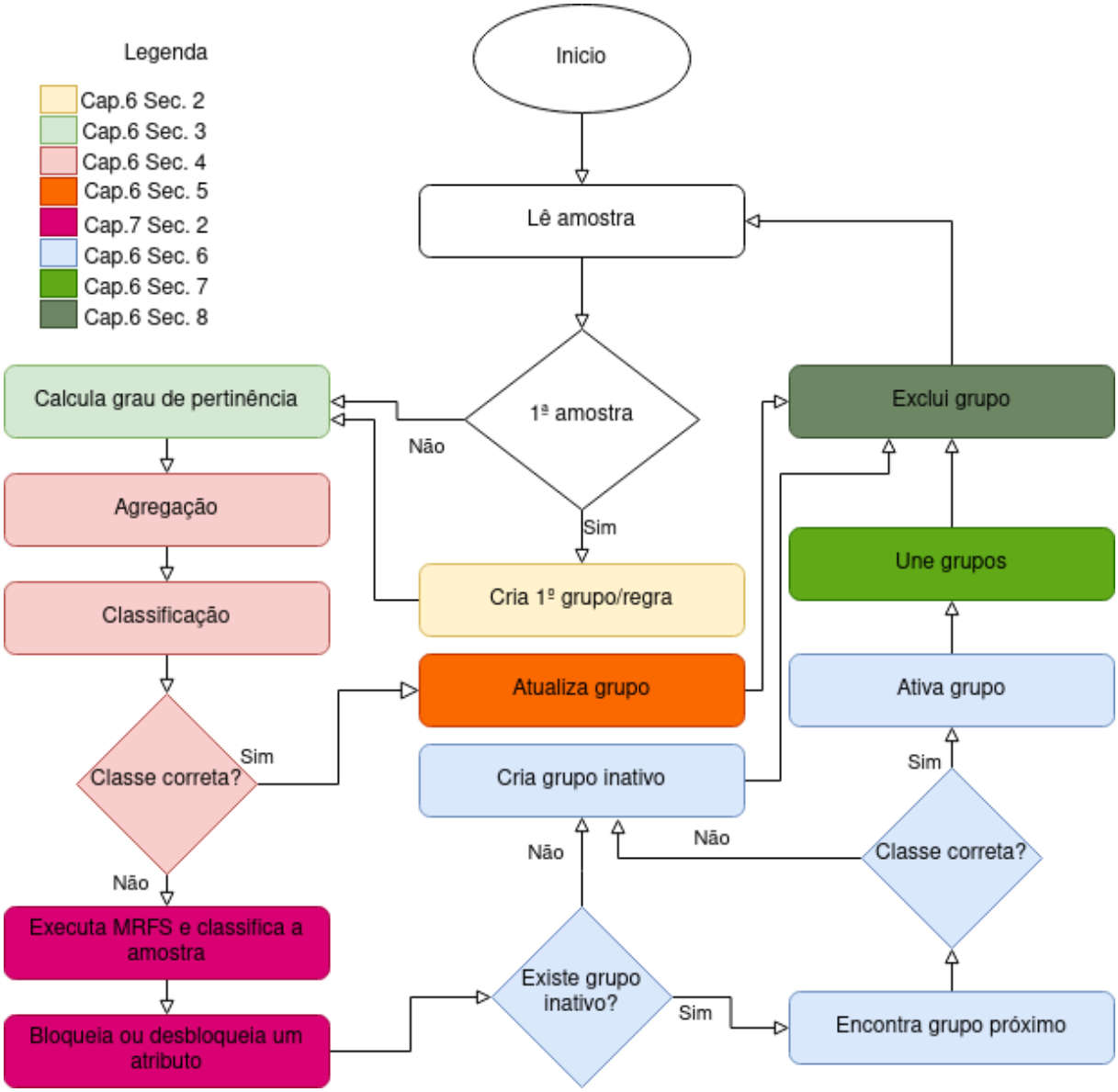

Figura 11 – Fluxograma do eFC-FS.

Capítulo 8

Experimentos computacionais com classificadores evolutivos

Este capítulo apresenta experimentos computacionais realizados para avaliar e comparar o desempenho dos classificadores evolutivos propostos. A metodologia adotada nos experimentos é descrita na Seção 8.1. Em seguida, a Seção 8.2 apresenta os resultados obtidos na avaliação da acurácia e a Seção 8.3 avalia o tempo de processamento dos classificadores. A escalabilidade dos classificadores propostos é avaliada na Seção 8.4. Concluindo o capítulo, a Seção 8.5 apresenta uma sumarização e discute os resultados experimentais.

8.1 Metologia dos experimentos

Os experimentos computacionais foram realizados para avaliar o desempenho dos três classificadores propostos (eFCM, eFC, eFC-FS) e comparar os resultados obtidos com os de três classificadores do estado da arte (eGNN (Leite; Costa; Gomide, 2010), eT2Class (Pratama; Jie Lu; Guangquan Zhang, 2015) e FBeM (Leite et al., 2012)). O código em Matlab dos classificadores alternativos foi disponibilizado por seus respectivos autores. O código em Matlab do eFMC e do eFC, além do código em Python do eFC-FS estão disponíveis em http://shorturl.at/ipEP6.

Ressalta-se que nenhum ajuste foi realizado nos classificadores para os diferentes conjuntos de dados, os parâmetros dos classificadores do estado da arte foram ajustados de acordo com os parâmetros reportados nos experimentos apresentados em sua literatura. O eFMC foi executado com parâmetro $f = 1,25$, o eFC e eFC-FS com $\omega = 30$ e $f = 1,25$, o restante dos modelos usaram:

- eT2Class: executado com ρ_3 = 0,5 , de acordo com Pratama et al. (2016b);
- eGNN: sua execução utilizou ρ = 0,25, η = 2, β = 0,9, ζ = 0,9, χ = 0,2 e h = 100 em

Leite, Costa e Gomide (2010). Excepcionalmente nos conjuntos de dados *Wine* e *Optical R. Handwriteen D.* utilizou-se $\rho = 15, 25$.

- FBeM: os parâmetros foram ρ = 0,7, η = 2 e h = 48 como em Leite et al. (2012);

Os experimentos computacionais foram realizados em 8 conjuntos de dados simulando processamento *online*, ou seja, a estrutura e parâmetros dos classificadores evoluem para todas as amostras. Destaca-se que estes 8 conjuntos de dados são amplamente utilizados literatura em teste de classificadores, inclusive sendo empregados na avaliação de classificadores evolutivos como em Shihabudheen e Pillai (2017), Gu e Angelov (2018a), Wang et al. (2023) e Souza e Lughofer (2021), para citar somente alguns. Além disso, salienta-se que existe uma grande diversidade entre o número de amostras, atributos e classes entre os conjuntos de dados. A Tabela 15 apresenta as principais características dos conjuntos de dados. Nos experimentos, os dados foram utilizados sem normalização.

Tabela 15 – Descrição dos conjuntos de dados.

Nome	# Amostras	# Atributos	# Classes
Iris Dataset (UCI, 2023)	150	4	3
Wine (UCI, 2023)	178	13	3
Glass Identification (UCI, 2023)	214	10	6
Breast Cancer W. (KAGGLE, 2023)	569	31	2
Australian C. Approval (UCI, 2023)	609	14	2
Pima Indians D. (KAGGLE, 2023)	768	8	2
Banknote Authentication (UCI, 2023)	1372	5	2
Optical R. Handwritten D. (UCI, 2023)	5620	64	10

A cada execução dos experimentos, os dados foram embaralhados para simular um ambiente real em que os dados não seguem uma tendência/organização em sua disponibilização. A quantidade de execuções dos experimentos foi de 10 vezes e os classificadores foram avaliados pela média da Acurácia Eq. (30).

$$Acc(y, \hat{y}) = \frac{1}{\mathbf{T}} \sum_{i=1}^{\mathbf{T}} 1(\hat{y}_i = y_i), \tag{30}$$

na qual $\hat{y}_i$ é a classe prevista pelo classificador da $i-ésima$ amostra, y_i é a classe correta e $\mathbf{T}$ é o número total de amostras.

8.2 Resultados experimentais com classificadores evolutivos

A Tabela 16 apresenta o desvio padrão e a acurácia média obtida pelos classificadores. Como pode ser visto, o eFC-FS obteve melhor desempenho em 5 conjuntos de dados e o eFC em dois. No experimento com o conjunto de dados *Wine*, a maior acurácia foi

obtida pelo FBeM, seguido pelo eFC-FS. É possível observar também que o eFMC ficou com a terceira melhor acurácia em 5 de 8 conjuntos de dados. Na média geral o melhor desempenho foi alcançado pelo eFC-FS, seguido pelo eFC, FBeM [1], eFMC, eGNN e eT2Class.

Tabela 16 – Acurácia e desvio padrão dos classificadores evolutivos.

	eFC-FS	eFC	eFMC	eT2Class	eGNN	FBeM
Iris Dataset	**96,26** / **0,59**	93,56 / 1,19	81,21 / 2,80	65,00 / 9,10	79,80 / 2,35	91,93 / 1,28
Wine	83,92 / 1,55	79,78 / 1,67	63,05 / 2,59	33,11 / 1,87	52,90 / 4,41	**91,74** / **1,80**
Glass Identification	**93,69** / **0,56**	91,45 / 0,80	67,85 / 2,33	33,36 / 2,36	48,40 / 2,39	53,08 / 4,95
Breast Cancer Wisconsin	**94,21** / **1,23**	93,98 / 0,99	88,91 / 0,43	37,36 / 1,69	52,63 / 2,18	-
Australian C. Approval	**79,29** / **1,06**	77,09 / 1,49	64,68 / 0,87	55,74 / 1,48	50,56 / 1,60	-
Pima Indians Diabetes	**83,04** / **0,24**	81,46 / 1,57	63,61 / 0,78	63,36 / 4,92	54,28 / 1,82	-
Banknote Authentication	87,17 / 0,30	**91,05** / **4,28**	67,30 / 1,81	55,86 / 1,47	49,92 / 3,83	84,96 / 2,62
Optical R. Handwritten D.	86,33 / 0,28	**92,48** / **0,28**	89,43 / 0,14	54,01 / 8,28	75,33 / 8,52	77,59 / 0,66
Média	**87,99** / **0,49**	87,61 / 1,20	73,26 / 1,04	49,73 / 3,17	57,98 / 2,22	79,86 / 1,67

8.3 Tempo de processamento dos classificadores evolutivos

Os experimentos foram realizados utilizando um computador com processador Intel Core I5 2600MHz e 8 GB de memória RAM. O tempo de execução dos classificadores é apresentado em milissegundos (ms) e referem-se ao tempo médio de processamento por amostra, considerando 10 repetições para cada conjunto de dados.

No tempo de execução, é possível observar pela Tabela 17 que o eFMC foi melhor em 6 de 8 conjuntos de dados. O eFC teve melhor tempo de execução em 2 experimentos e o FBeM apresentou o terceiro melhor desempenho em todos os conjuntos de dados.

Tabela 17 – Tempo de processamento por amostra em milissegundos dos classificadores.

	eFC-FS	eFC	eFMC	eT2Class	eGNN	FBeM
Iris Dataset	0,7000	0,5208	**0,4616**	3,5163	0,7008	0,6842
Wine Dataset	4,1000	0,6267	**0,3781**	5,3901	0,7437	0,7638
Glass Identification	2,6000	**0,5458**	0,5736	4,8070	0,7359	0,6772
Breast Cancer Wisconsin	1,8000	0,2154	**0,1422**	4,6091	0,6440	-
Australian C. Approval	3,6000	0,4605	**0,1899**	3,6969	0,5987	-
Pima Indians Diabetes	2,0000	0,3926	**0,2044**	3,7844	0,5023	-
Banknote Authentication	1,2000	0,2252	**0,1128**	2,3062	3,4932	0,2786
Optical R. Handwritten D.	35,6000	**0,4164**	0,5484	1,7551	1,6179	0,5135
Média	6,4500	0,4254	0,3264	3,7331	1,0046	0,5835

[1] O FBeM apresentou falha no processamento de 3 conjuntos. Assim, a média geral desse classificador foi calculada considerando somente 5 conjuntos.

8.4 Escalabilidade dos classificadores evolutivos propostos

Esta seção avalia a escalabilidade dos classificadores propostos. Inicialmente avalia-se o número de grupos e de atributos de entrada. A Tabela 18 apresenta a quantidade final de grupos do eFC e eFC-FS, assim como a quantidade de atributos deletados pelo classificador com seleção de atributos[2]. O eFC-FS conseguiu excluir quantidades significativas de atributos de entrada e consequentemente melhorar a classificação. Em alguns conjuntos de dados, devido à redução de atributos de entrada, a quantidade de grupos necessários para mapear as classes de saída ainda foi menor que o eFC.

Nas Figuras de 12 até 19 mostra a evolução da estrutura dos classificadores propostos, na qual é possível visualizar ao longo do tempo o processo de inclusão e exclusão de grupos. Em todas as figuras é possível observar que o eFMC iniciou com dois grupos e estabilizou em uma quantidade igual ao total de classes de cada base de dados. Já nos modelos eFC e eFC-FS, sua estrutura evolui excluindo e incluindo novos grupos.

Tabela 18 – Número de grupos e de atributos de entrada dos classificadores eFC e eFC-FS.

	eFC-FS	eFC	Atr. Del.	Atr. Totais	Classes
Iris Dataset	3	3	2	4	3
Wine	5	5	2	13	3
Glass Identification	6	7	1	10	6
Breast Cancer Wisconsin	3	3	7	31	2
Australian C. Approval	4	4	8	14	2
Pima Indians Diabetes	3	5	5	8	2
Banknote Authentication	4	5	2	5	2
Optical R. Handwritten D.	10	10	27	64	10

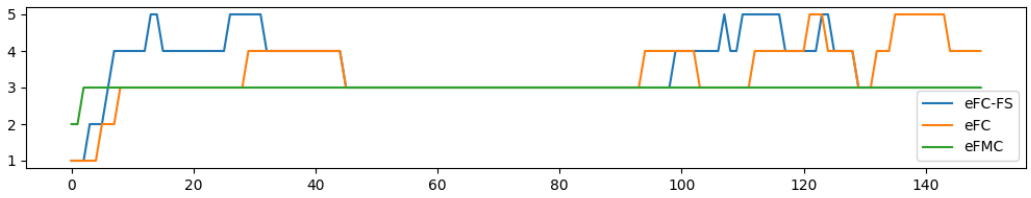

Figura 12 – Evolução da estrutura dos classificadores no conjunto de dados *Iris Dataset*.

A escalabilidade dos classificadores propostos também foi avaliada considerando a relação entre o número de atributos, o número de classes e o tempo de execução. Em outras palavras, busca-se avaliar o quanto o número de atributos e o número de classes influenciam no tempo de processamento dos classificadores. Para tanto, conjuntos de dados foram

[2]Optou por não apresentar quantidade de grupos do eFMC pelo fato de ser sempre igual ao número de classes de saída

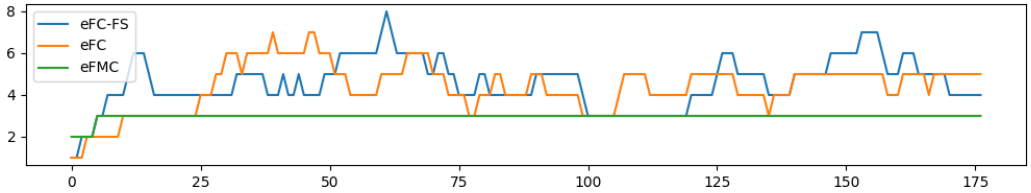

Figura 13 – Evolução da estrutura dos classificadores no conjunto de dados *Wine Dataset*.

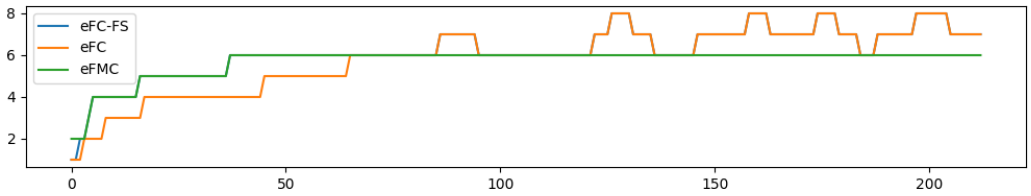

Figura 14 – Evolução da estrutura dos classificadores no conjunto de dados *Glass Identification*.

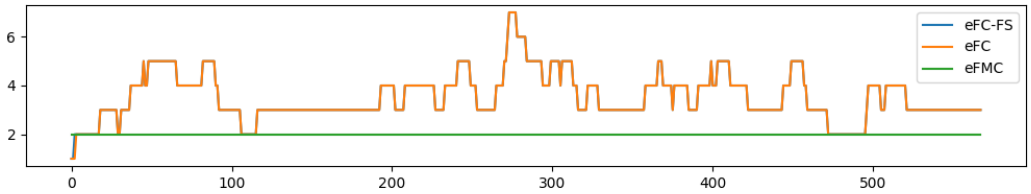

Figura 15 – Evolução da estrutura dos classificadores no conjunto de dados *Breast Cancer Wisconsin*.

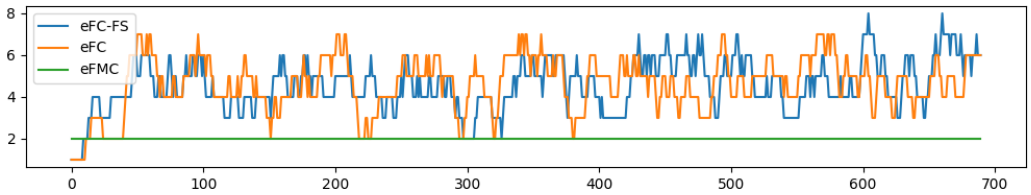

Figura 16 – Evolução da estrutura dos classificadores no conjunto de dados *Australian C. Approval*.

gerados com 10.000 amostras utilizando a biblioteca Python Sklearn (Scikit-learn, 2020), com função de geração Gaussiana isotrópica e amostragem por quantil.

Para avaliar a escalabilidade dos modelos em termos da quantidade dos atributos de entrada, foram gerados conjuntos de dados até 2^{11} variando de 2 até 2048 o número de atributos de entrada e 2 classes. A Figura 20 ilustra o tempo de execução dos classificadores. Nesta comparação, é possível notar que o eFMC não sofre uma perda grande de desempenho

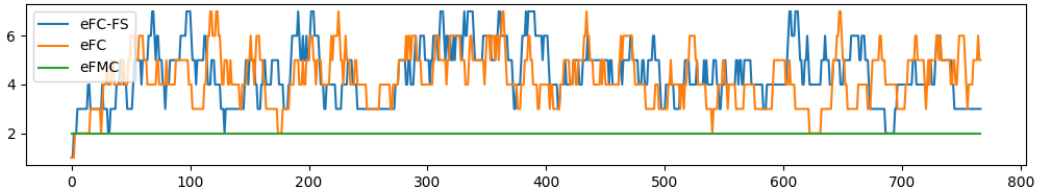

Figura 17 – Evolução da estrutura dos classificadores no conjunto de dados *Pima Indians Diabetes*.

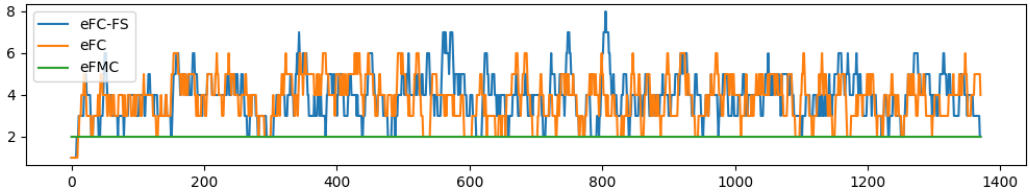

Figura 18 – Evolução da estrutura dos classificadores no conjunto de dados *Banknote Authentication*.

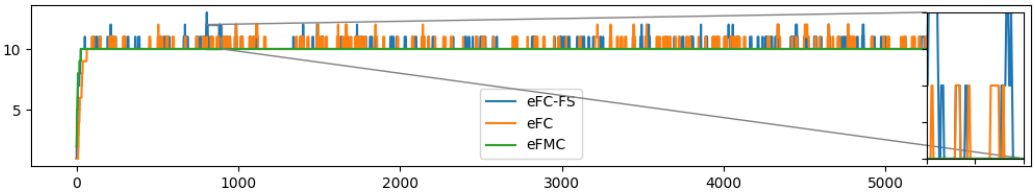

Figura 19 – Evolução da estrutura dos classificadores no conjunto de dados *Optical R. Handwritten D.*.

com o aumento de atributos de entrada. As linhas pontinhadas representam o tempo de processamento de cada modelo, já as linhas contínuas apresentam uma tendência linear estimada para o tempo de processamento. É possível observar uma tendência linear análoga do tempo de processamento obtido com o tempo estimado.

Na análise da escalabilidade de acordo com as classes foram gerados conjuntos de dados com 2, 4, 8, 16 e 32 classes e 10 atributos de entrada. O tempo de execução dos classificadores são apresentados na Figura 21. Todos os classificadores apresentaram um comportamento semelhante com o aumento de classes, sendo o eFC-FS o que mais sofreu com o aumento do número de classes e o menor aumento no tempo de processamento foi do eFC.

Na análise da escalabilidade em função do número de atributos, os classificadores apresentaram um aumento linear com o incremento do número de atributos. No entanto, o mesmo não aconteceu com relação ao aumento do número de classes no qual o comportamento

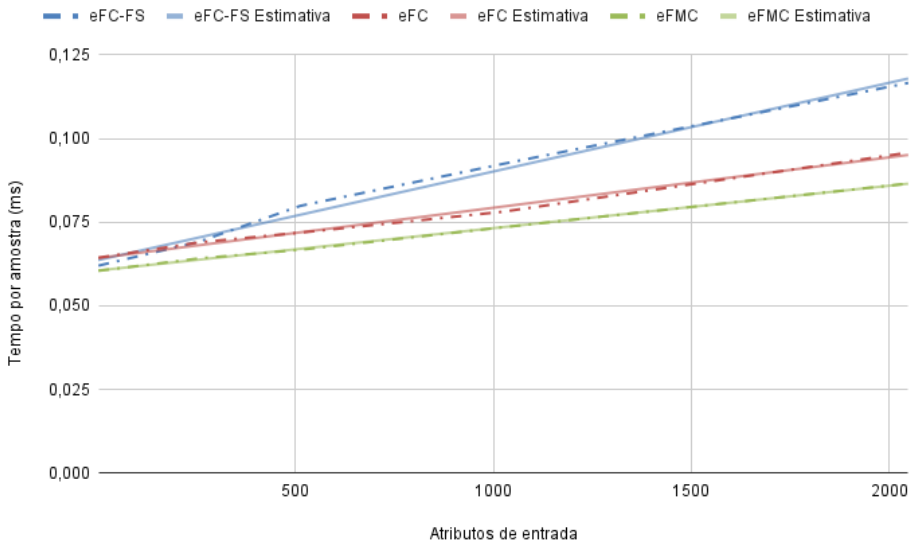

Figura 20 – Escalabilidade pelo incremento no número de atributos de entrada (ms).

tende para o exponencial. Isso mostra que os classificadores são mais sensíveis em tempo computacional com o incremento do número de classes do que com o do número de atributos.

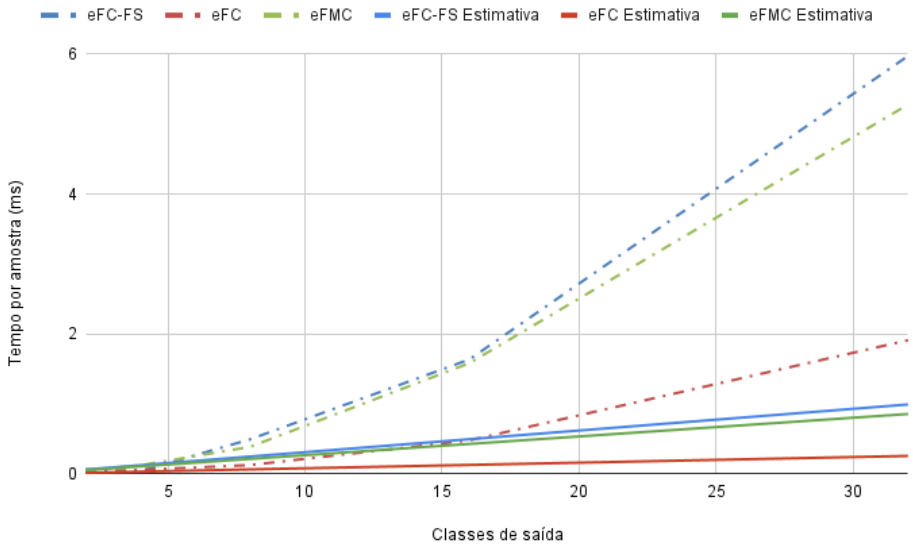

Figura 21 – Escalabilidade pelo incremento no número de classes de saída (ms).

8.5 Considerações do capítulo

Neste capítulo os classificadores evolutivos propostos foram testados e comparados com o estado da arte. Além disso, avaliou-se a escalabilidade dos classificadores propostos.

Na avaliação da acurácia entre os classificadores propostos e os do estado da arte, o eFC-FS apresentou a melhor acurácia e ainda reduziu a quantidade de atributos de entrada. O eFC, apresentou o segundo melhor desempenho quando comparado com os demais classificadores e foi sucedido pelo eFMC. Na avaliação de tempo de processamento, o eFMC foi o mais rápido, posteriormente o eFC, e como esperado o eFC-FS possui um maior custo computacional devido ao seu método de FS.

A análise de escalabilidade dos classificadores propostos mostrou que o eFMC apresenta o melhor tempo de processamento com aumento de atributos de entrada e poucas classes. Por outro lado, o eFC apresenta melhor desempenho com um número maior de classes. Dessa forma, é possível afirmar que o eFMC apresenta boa acurácia e mais agilidade no processamento em conjuntos de dados com muitos atributos e poucas classes. O eFC possui uma melhor acurácia que o eFMC e um tempo superior de processamento. O eFC-FS, possui a melhor acurácia entre os classificadores propostos, mas é inferior no tempo de processamento.

Capítulo 9

Conclusão

Um grande desafio em inteligência computacional aplicado a problemas de classificação e reconhecimento de padrões é identificar bons atributos de entrada, ou seja, encontrar um conjunto de atributos de entrada que melhor represente as classes e, consequentemente, ofereça maior acurácia para o classificador. Assim, este trabalho introduziu uma nova abordagem para seleção de atributos tendo como fundamentos a simplicidade, o baixo custo computacional e a flexibilidade, em outra palavras, que o seletor de atributo possa ser usado como método *filter* ou incorporado aos classificadores como método *wrapper* sem um grande aumento no tempo de treinamento.

Inicialmente foi proposto o método *Mean Ratio for Feature Selection* (MRFS) que utiliza a razão das médias para ranquear os atributos. O MRFS tem como principal característica a utilização de operações matemáticas básicas, como adição, divisão e comparação. O objetivo é explorar uma solução mais direta, eficiente e de fácil aplicação em ambientes que requerem processamento rápido. O MRFS proposto foi aplicado como um método de seleção de atributos de entrada como *filter* (MRFS-F) e *wrapper* (MRFS-W).

Na implementação do MRFS como *filter* (MRFS-F) são calculadas as médias dos dados associados às diferentes classes para cada atributo. Em seguida, calcula-se a razão das médias. Por fim, os atributos são ordenados pelos resultados da razão, do menor para o maior valor. Os resultados da razão permitem verificar quais dos atributos de menor valor possuem as médias mais discrepantes. No método, os atributos com médias diferentes são mais relevantes para a classificação. Por outro lado, na implementação do MRFS como *wrapper* (MRFS-W) são construídos classificadores com m e $m-1$ atributos, nomeados de modelos atual e candidatos. Durante o treinamento, a metodologia do MRFS é aplicada, ranqueando os atributos de entrada e avaliando o desempenho dos modelos candidatos e atual. Enquanto o modelo candidato apresentar melhor desempenho, o modelo atual substitui seus atributos pelos atributos do modelo candidato, e o modelo candidato exclui um novo atributo menos relevante.

O MRFS-F foi comparado com outros 3 métodos de seleção de atributos e o ranqueamento dos atributos de cada um dos métodos foram aplicados em 4 diferentes classificadores. O MRFS-W foi agregado em 4 classificadores, comparando os seus resultados com o respectivo classificador sem seleção de atributos. Os experimentos computacionais sugerem o MRFS como uma abordagem promissora para ser empregada tanto como método *filter*, quanto como método *wrapper* de seleção de atributos. Destaca-se, além da boa acurácia obtida na utilização do MRFS, a simplicidade das suas equações e seu baixo custo computacional.

A segunda etapa da atual pesquisa trata-se do desenvolvimento de novos modelos de classificação evolutivos. Estes modelos possuem uma característica de um aprendizado incremental e que pode ser amplamente utilizado nos ambientes em que os dados são disponibilizados em fluxo contínuo e que são variantes no tempo. Os modelos propostos foram nomeados de *evolving Fuzzy Mean Classifier* (eFMC) e *evolving Fuzzy Classifier* (eFC). Por fim, as propostas de seleção de atributos e modelo evolutivo de classificação foram exploradas no desenvolvimento do *evolving Fuzzy Classifier with Features Selection* (eFC-FS).

No eFMC, a estrutura do classificador é construída por um novo algoritmo de agrupamento no qual o grau de pertinência medido dos grupos existentes é usado para classificar as amostras processadas. Cada amostra processada é utilizada para atualizar os centros dos grupos por meio de médias amostrais calculadas de forma incremental a partir dos dados já processados. O eFMC possui a capacidade de geração de um novo grupo sempre que uma amostra de uma nova classe é identificada. A criação e atualização dos grupos ocorre de forma dinâmica, sem a necessidade de interromper o processamento para treinar o algoritmo.

O eFC foi o segundo classificador evolutivo introduzido neste trabalho. No eFC, os centros dos grupos são atualizados por meio da média amostral, com base na metodologia do eFMC. Neste classificador, mais de um grupo pode ser gerado associado a uma mesma classe de saída, permitindo assim, mapear diferentes regiões nos espaços dos dados. No eFC, os grupos podem ser unidos, excluídos e criados. O algoritmo para criação de novos grupos provê um mecanismo robusto e automático para decidir se uma nova amostra será representante de um novo grupo ou apenas um *outlier*. Assim, a criação de grupos acontece em duas etapas, na qual um novo grupo gerado só passa a contribuir na classificação após um processo de verificação e ativação. A união é realizada por um mecanismo que verifica se dois grupos próximos possuem regras que estão associadas a um mesmo grupo. Já a exclusão, é realizada com base na verificação do tempo de ociosidade do grupo.

Por fim, foi desenvolvido um classificador evolutivo multiclasses com seleção de atributos, o eFC-FS. No eFC-FS foi utilizado a capacidade de aprendizado evolutivo do eFC com

o método MRFS. Considerando a natureza dos cálculos utilizados do MRFS, que são baseados nas médias para atributos associados às diferentes classes e a atualização dos centros dos grupos do eFC que também utilizam as médias das amostras processadas, o método MRFS foi perfeitamente adaptado no eFC. A seleção de atributos ocorre sempre que uma amostra é classificada errada, o eFC-FS utiliza os centros dos grupos que são as médias das amostras já processadas de cada grupo e calcula a razão entre eles. Desta forma, o modelo apenas realiza a ordenação do resultado da razão e pode tomar uma decisão de bloquear um novo atributo ou desbloquear um atributo bloqueado em iterações passadas.

O desempenho dos classificadores evolutivos propostos - eFMC, eFC e eFC-FS - foi avaliado e comparado com classificadores alternativos. Os resultados computacionais sugerem que os classificadores introduzidos neste trabalho possuem um desempenho superior ou comparável com modelos do estado da arte. Além de sua precisão, os modelos eFMC e eFC têm um custo computacional pelo menos uma ordem de grandeza menor que outros modelos. Já o eFC-FS, apesar de apresentar a melhor acurácia, possui um tempo de processamento igual ou inferior a outros modelos. O baixo tempo de processamento e simplicidade sugerem todos os modelos propostos como abordagens adequadas em ambientes de fluxo de dados *online* e em tempo real. Ressaltando o eFMC como um modelo de boa acurácia e alta velocidade, o eFC como um modelo com melhor acurácia e com velocidade um pouco inferior ao eFMC, já o eFC-FS como modelo de melhor acurácia.

Como proposta de continuidade da pesquisa apresentada neste trabalho sugere-se:

- a utilização do método MRFS como *wrapper* em outros algoritmos de classificação;
- a implementação do MRFS-W como seleção para frente ou *embedded* na tentativa de redução no tempo de treinamento;
- a implementação dos classificadores propostos com outras medidas de distância para o cálculo da grau de pertinência, como por exemplo a distância de Mahalanobis;
- a implementação do modelo eFC com outros operadores de agregação, assim como desenvolvimento e melhoria dos métodos de exclusão e união de grupos;
- a utilização do modelo eFC-FS como seleção para frente ou alguma outra variação, como empregar uma metodologia de janelas deslizantes antes de realizar a seleção de atributos.

Referências

Ahwiadi, M.; Wang, W. An Adaptive Evolving Fuzzy Technique for Prognosis of Dynamic Systems. **IEEE Transactions on Fuzzy Systems**, Institute of Electrical and Electronics Engineers Inc., 2021. ISSN 19410034. Citado na página 14.

Alizadeh, S.; Kalhor, A.; Jamalabadi, H.; Araabi, B. N.; Ahmadabadi, M. N. Online Local Input Selection Through Evolving Heterogeneous Fuzzy Inference System. **IEEE Transactions on Fuzzy Systems**, v. 24, n. 6, p. 1364–1377, dec 2016. ISSN 1063-6706. Disponível em: <http://ieeexplore.ieee.org/document/7378479/>. Citado na página 17.

Allen, M. **Understanding Regression Analysis**. 1. ed. [S.l.]: Springer, 1997. Citado na página 16.

Andreu, J.; Angelov, P. Forecasting time-series for NN GC1 using Evolving Takagi-Sugeno (eTS) Fuzzy Systems with on-line inputs selection. In: **International Conference on Fuzzy Systems**. IEEE, 2010. p. 1–5. ISBN 978-1-4244-6919-2. Disponível em: <http://ieeexplore.ieee.org/document/5584130/>. Citado na página 15.

Angelov, P. Evolving Takagi-Sugeno Fuzzy Systems from Streaming Data (eTS+). In: **Evolving Intelligent Systems**. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2010. p. 21–50. Disponível em: <http://doi.wiley.com/10.1002/9780470569962.ch2>. Citado na página 10.

Angelov, P.; Buswell, R. Evolving Rule-based Models: A Tool for Intelligent Adaptation. In: **Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference**. [S.l.: s.n.], 2001. v. 2, p. 1062–1067. Citado na página 10.

Angelov, P.; Filev, D. Simpl_eTS: A Simplified Method for Learning Evolving Takagi-Sugeno Fuzzy Models. In: **Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE '05**. [S.l.: s.n.], 2005. p. 1068–1073. Citado na página 54.

Angelov, P.; Yager, R. A new type of simplified fuzzy rule-based system. **International Journal of General Systems**, Taylor & Francis, v. 41, n. 2, p. 163–185, 2012. Disponível em: <https://doi.org/10.1080/03081079.2011.634807>. Citado na página 13.

Angelov, P.; Zhou, X. Evolving Fuzzy Systems from Data Streams in Real-Time. In: **2006 International Symposium on Evolving Fuzzy Systems**. [S.l.: s.n.], 2006. p. 29–35. Citado 2 vezes nas páginas 10 e 11.

Angelov, P. P.; Filev, D. P. An approach to online identification of Takagi-Sugeno fuzzy models. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, v. 34, n. 1, p. 484–498, feb 2004. ISSN 1083-4419. Citado 4 vezes nas páginas 10, 11, 14 e 15.

Angelov, P. P.; Gu, X.; Príncipe, J. Autonomous Learning Multi-Model Systems from Data Streams. **IEEE Transactions on Fuzzy Systems**, n. 99, p. 1, 2017. ISSN 1063-6706. Citado na página 13.

Anitha, M. A.; Sherly, K. K. A Novel Forward Filter Feature Selection Algorithm Based on Maximum Dual Interaction and Maximum Feature Relevance(MDIMFR) for Machine Learning. **10th International Conference on Advances in Computing and Communications,**

ICACC 2021, Institute of Electrical and Electronics Engineers Inc., 2021. Citado 2 vezes nas páginas 7 e 9.

Bezdek, J. C. **Pattern Recognition with Fuzzy Objective Function Algorithms**. [S.l.]: Springer US, 1981. Citado na página 50.

Blum, A. L.; Langley, P. Selection of relevant features and examples in machine learning. **Artificial Intelligence**, Elsevier, v. 97, n. 1-2, p. 245–271, dec 1997. ISSN 0004-3702. Citado na página 1.

Bodyanskiy, Y.; Boiko, O.; Zaychenko, Y.; Hamidov, G. Evolving hybrid GMDH-Neuro-fuzzy network and its applications. In: **2018 IEEE 1st International Conference on System Analysis and Intelligent Computing, SAIC 2018 - Proceedings**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. ISBN 9781538671955. Citado na página 13.

Bouchachia, A.; Lughofer, E.; Sayed-Mouchaweh, M. Special Issue: Evolving Soft Computing Techniques and Applications. **Applied Soft Computing**, Applied Soft Computing, v. 14, p. 141–143, jan 2014. ISSN 1568-4946. Citado na página 10.

Caminhas, W.; Gomide, F. A fast learning algorithm for neofuzzy networks. In: **Proceedings of the Information Processing and Management of Uncertainty in Knowledge Based Systems, IPMU '00**. [S.l.: s.n.], 2000. v. 1, n. 1, p. 1784–1790. Citado na página 12.

Chen, R.; Sun, N.; Chen, X.; Yang, M.; Wu, Q. Supervised Feature Selection with a Stratified Feature Weighting Method. **IEEE Access**, Institute of Electrical and Electronics Engineers Inc., v. 6, p. 15087–15098, mar 2018. ISSN 21693536. Citado na página 8.

Chia-Feng, J.; Yu-Wei, T. A Self-Evolving Interval Type-2 Fuzzy Neural Network With Online Structure and Parameter Learning. **IEEE Transactions on Fuzzy Systems**, v. 16, n. 6, p. 1411–1424, dec 2008. ISSN 1063-6706. Disponível em: <http://ieeexplore.ieee.org/document/4529083/>. Citado na página 16.

Chin-Teng, L.; Pal, N. R.; Shang-Lin, W.; Yu-Ting, L.; Yang-Yin, L. An Interval Type-2 Neural Fuzzy System for Online System Identification and Feature Elimination. **IEEE Transactions on Neural Networks and Learning Systems**, v. 26, n. 7, p. 1442–1455, jul 2015. ISSN 2162-237X. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6881716>. Citado na página 16.

Cox, E. **Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration**. Elsevier Science, 2005. (The Morgan Kaufmann Series in Data Management Systems). ISBN 9780121942755. Disponível em: <https://books.google.com.br/books?id=H93cJpnbwpcC>. Citado 5 vezes nas páginas 41, 43, 50, 55 e 62.

Decker, L.; Leite, D.; Giommi, L.; Bonacorsi, D. Real-time anomaly detection in data centers for log-based predictive maintenance using an evolving fuzzy-rule-based approach. In: **2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. [S.l.: s.n.], 2020. p. 1–8. Citado 2 vezes nas páginas 11 e 18.

Dovzan, D.; Logar, V.; Skrjanc, I. Implementation of an Evolving Fuzzy Model (eFuMo) in a Monitoring System for a Waste-Water Treatment Process. **IEEE Transactions on Fuzzy Systems**, v. 23, n. 5, p. 1761–1776, oct 2015. ISSN 1063-6706. Disponível em: <http://ieeexplore.ieee.org/document/6980077/>. Citado na página 11.

Dovžan, D. Evolving fuzzy model in fault detection system. In: **2017 Evolving and Adaptive Intelligent Systems (EAIS)**. [S.l.: s.n.], 2017. p. 1–8. Citado na página 11.

Dunn, J. C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. **Journal of Cybernetics**, v. 3, n. 3, p. 32–57, jan 1973. ISSN 00220280. Citado na página 50.

Edwin Lughofer. Evolving multi-label fuzzy classifier. **Information Sciences**, v. 597, p. 1–23, 2022. ISSN 0020-0255. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020025522002560>. Citado na página 2.

Ferdaus, M. M.; Pratama, M.; Anavatti, S. G.; Garratt, M. A. PALM: An Incremental Construction of Hyperplanes for Data Stream Regression. **IEEE Transactions on Fuzzy Systems**, Institute of Electrical and Electronics Engineers Inc., v. 27, n. 11, p. 2115–2129, nov 2019. ISSN 19410034. Citado na página 14.

Garcia, C.; Esmin, A.; Leite, D.; Škrjanc, I. Evolvable fuzzy systems from data streams with missing values: With application to temporal pattern recognition and cryptocurrency prediction. **Pattern Recognition Letters**, v. 128, p. 278–282, December 2019. Citado na página 11.

Garcia, C.; Leite, D.; Škrjanc, I. Incremental missing-data imputation for evolving fuzzy granular prediction. **IEEE Transactions on Fuzzy Systems**, v. 28, n. 10, p. 2348–2362, 2020. Citado na página 11.

Gauthama Raman, M. R.; Nivethitha, S.; Kannan, K.; Shankar, S. V. S. A hybrid approach using rough set theory and hypergraph for feature selection on high-dimensional medical datasets. **Soft Computing**, Springer Berlin Heidelberg, v. 23, n. 23, p. 12655–12672, dec 2019. ISSN 1432-7643. Disponível em: <http://link.springer.com/10.1007/s00500-019-03818-6>. Citado na página 1.

Ge, D.; Zeng, X.-J. A self-evolving fuzzy system which learns dynamic threshold parameter by itself. **IEEE Transactions on Fuzzy Systems**, v. 27, n. 8, p. 1625–1637, 2019. ISSN 1063-6706. Disponível em: <https://ieeexplore.ieee.org/document/8571270/>. Citado na página 13.

Gray, R. Vector quantizagion. **IEEE ASSP Magazine**, p. 4–29, 1984. Citado na página 12.

Gu, X. Multilayer Ensemble Evolving Fuzzy Inference System. **IEEE Transactions on Fuzzy Systems**, Institute of Electrical and Electronics Engineers Inc., v. 29, n. 8, p. 2425–2431, aug 2021. ISSN 19410034. Citado na página 14.

Gu, X.; Angelov, P. P. Self-organising fuzzy logic classifier. **Information Sciences**, v. 447, p. 36–51, 2018. ISSN 0020-0255. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020025518301737>. Citado na página 65.

Gu, X.; Angelov, P. P. Semi-supervised deep rule-based approach for image classification. **Applied Soft Computing**, v. 68, p. 53–68, 2018. ISSN 1568-4946. Citado na página 11.

Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. **The Journal of Machine Learning Research**, JMLR.org, v. 3, p. 1157–1182, mar 2003. Citado 2 vezes nas páginas 1 e 6.

Han; Cheng; Hou. Fault Detection Method Based on Improved Isomap and SVM in Noise-Containing Nonlinear Process. In: **2018 International Conference on Control, Automation and Information Sciences (ICCAIS)**. [S.l.]: IEEE, 2018. p. 461–466. ISBN 978-1-5386-6020-1. Citado na página 7.

Haq, A. U.; Zhang, D.; Peng, H.; Rahman, S. U. Combining Multiple Feature-Ranking Techniques and Clustering of Variables for Feature Selection. **IEEE Access**, Institute of Electrical and Electronics Engineers Inc., v. 7, p. 151482–151492, 2019. ISSN 21693536. Citado na página 9.

Hodrick, R.; Prescott, E. Post-war us business cycles: An empirical investigation. **Journal of Money, Credit and Banking**, v. 29, p. 1–16, 1997. Citado na página 18.

Hore, P.; Hall, L. O.; Goldgof, D. B. Creating streaming iterative soft clustering algorithms. In: **Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS**. [S.l.: s.n.], 2007. p. 484–488. ISBN 1424412145. Citado na página 11.

Hore, P.; Hall, L. O.; Goldgof, D. B. Single pass fuzzy c means. In: **IEEE International Conference on Fuzzy Systems**. [S.l.: s.n.], 2007. ISBN 1424412102. ISSN 10987584. Citado na página 11.

Hsieh, P.-J.; Li, C.-H.; Kuo, B.-C.; Tsai, P.-L. An automatic kernel parameter selection method for kernel nonparametric weighted feature extraction with the RBF kernel for hyperspectral image classification. In: **2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)**. IEEE, 2015. p. 1706–1709. ISBN 978-1-4799-7929-5. Disponível em: <http://ieeexplore.ieee.org/document/7326116/>. Citado na página 7.

Ibrahim, R. A.; Elaziz, M. A.; Oliva, D.; Cuevas, E.; Lu, S. An opposition-based social spider optimization for feature selection. **Soft Computing**, Springer Berlin Heidelberg, v. 23, n. 24, p. 13547–13567, dec 2019. ISSN 1432-7643. Disponível em: <http://link.springer.com/10.1007/s00500-019-03891-x>. Citado na página 8.

Jahandari, S.; Kalhor, A.; Araabi, B. N. Online forecasting of synchronous time series based on evolving linear models. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, v. 50, n. 5, p. 1865–1876, 2020. Citado 2 vezes nas páginas 11 e 17.

KAGGLE. **The Home of Data Science & Machine Learning**. 2023. Https://www.kaggle.com/. Citado 2 vezes nas páginas 29 e 65.

Kandath, H.; Hady, M. A.; Pratama, M.; Feng, N. B. Robust Evolving Neuro-Fuzzy Control of a Novel Tilt-rotor Vertical Takeoff and Landing Aircraft. In: **IEEE International Conference on Fuzzy Systems**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2019. v. 2019-June. ISBN 9781538617281. ISSN 10987584. Citado 2 vezes nas páginas 11 e 14.

Kang, M.; Tian, J. Machine Learning: Data Pre-processing. In: **Prognostics and Health Management of Electronics**. Chichester, UK: John Wiley and Sons Ltd, 2018. p. 111–130. Disponível em: <http://doi.wiley.com/10.1002/9781119515326.ch5>. Citado 2 vezes nas páginas 6 e 7.

Kasabov, N. Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. **IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)**, v. 31, n. 6, p. 902–918, 2001. ISSN 10834419. Disponível em: <http://ieeexplore.ieee.org/document/969494/>. Citado na página 10.

Kasabov, N.; Song, Q. **Dynamic Evolving Fuzzy Neural Networks with "m-out-of-n"\,\, Activation Nodes for On-line Adaptive Systems**. [S.l.], 1999. 1–29 p. (Information Science Discussion Papers Series). Citado na página 10.

Kasabov, N.; Song, Q. Dynamic Evolving Neuro-Fuzzy Inference System (DENFIS): On-line Learning and Application for Time-Series Prediction. **IEEE Transactions of Fuzzy Systems**, v. 10, n. 2, p. 144–154, apr 2002. Citado na página 10.

KEEL. **Knowledge Extraction based on Evolutionary Learning**. 2019. Https://sci2s.ugr.es/keel/category.php?cat=clas. Citado na página 29.

Lang, S. **The Mean Value Theorem**. New York, NY: Springer New York, 1986. 159-180 p. ISBN 9781441985323. Citado na página 24.

Lathi, B.; Sedra, A. **Linear Systems and Signals**. Oxford University Press, 2005. (Oxford series in electrical and computer engineering). ISBN 9780195158335. Disponível em: <https://books.google.com.br/books?id=7resQgAACAAJ>. Citado na página 18.

Leite, D.; Ballini, R.; Costa, P.; Gomide, F. Evolving fuzzy granular modeling from nonstationary fuzzy data streams. **Evolving Systems**, v. 3, n. 2, p. 65–79, jun 2012. ISSN 1868-6478. Disponível em: <http://link.springer.com/10.1007/s12530-012-9050-9>. Citado 4 vezes nas páginas 11, 12, 64 e 65.

Leite, D.; Costa, P.; Gomide, F. Evolving granular neural network for semi-supervised data stream classification. In: **The 2010 International Joint Conference on Neural Networks (IJCNN)**. IEEE, 2010. p. 1–8. ISBN 978-1-4244-6916-1. Disponível em: <http://ieeexplore.ieee.org/document/5596303/>. Citado 3 vezes nas páginas 12, 64 e 65.

Leite, D.; Costa, P.; Gomide, F. Interval approach for evolving granular system modeling. In: **Learning in Non-Stationary Environments**. New York, NY: Springer New York, 2012. p. 271–300. Citado 2 vezes nas páginas 11 e 12.

Leite, D.; Decker, L.; Santana, M.; Souza, P. EGFC: Evolving Gaussian Fuzzy Classifier from Never-Ending Semi-Supervised Data Streams – With Application to Power Quality Disturbance Detection and Classification. In: **2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. IEEE, 2020. p. 1–9. ISBN 978-1-7281-6932-3. Disponível em: <https://ieeexplore.ieee.org/document/9177847/>. Citado na página 2.

Leite, D.; Skrjanc, I.; Gomide, F. An overview on evolving systems and learning from stream data. **Evolving Systems 2020**, v. 11, n. 2, p. 181–198, mar 2020. ISSN 1868-6486. Citado na página 2.

Lemos, A.; Caminhas, W.; Gomide, F. Multivariable Gaussian Evolving Fuzzy Modeling System. **IEEE Transactions on Fuzzy Systems**, v. 19, n. 1, p. 91–104, feb 2011. ISSN 1063-6706. Citado na página 11.

Li, X.; Cai, C.; He, J. Density-based multi-manifold ISOMAP for data classification. In: **2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)**. IEEE, 2017. p. 897–903. ISBN 978-1-5386-1542-3. Disponível em: <http://ieeexplore.ieee.org/document/8282172/>. Citado na página 7.

Li, X.; Jia, G.; Li, J.; Zhuo, L. A Face Hallucination Algorithm via an LLE Coefficients Prior Model. **Chinese Journal of Electronics**, v. 27, n. 6, p. 1234–1240, nov 2018. ISSN 1022-4653. Citado na página 7.

Lughofer, E. Evolving vector quantization for classification of on-line data streams. In: **2008 International Conference on Computational Intelligence for Modelling Control and Automation, CIMCA 2008**. [S.l.: s.n.], 2008. p. 779–784. ISBN 9780769535142. Citado 2 vezes nas páginas 11 e 16.

Lughofer, E. On-line incremental feature weighting in evolving fuzzy classifiers. **Fuzzy Sets and Systems**, North-Holland, v. 163, n. 1, p. 1–23, jan 2011. ISSN 0165-0114. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0165011410003519>. Citado 2 vezes nas páginas 15 e 17.

Lughofer, E.; Angelov, P. Handling drifts and shifts in on-line data streams with evolving fuzzy systems. **Applied Soft Computing**, Elsevier Science, v. 11, n. 2, p. 2057–2068, 2011. ISSN 1568-4946. Citado na página 54.

Lughofer, E.; Angelov, P.; Zhou, X. Evolving Single- And Multi-Model Fuzzy Classifiers with FLEXFIS-Class. In: **2007 IEEE International Fuzzy Systems Conference**. IEEE, 2007. p. 1–6. ISBN 1-4244-1209-9. ISSN 1098-7584. Disponível em: <http://ieeexplore.ieee.org/document/4295393/>. Citado na página 12.

Lughofer, E.; Cernuda, C.; Kindermann, S.; Pratama, M. Generalized smart evolving fuzzy systems. **Evolving Systems**, Springer Berlin Heidelberg, v. 6, n. 4, p. 269–292, dec 2015. ISSN 1868-6478. Disponível em: <http://link.springer.com/10.1007/s12530-015-9132-6>. Citado 2 vezes nas páginas 16 e 17.

Lughofer, E.; Klement, E. FLEXFIS: A Variant for Incremental Learning of Takagi-Sugeno Fuzzy Systems. In: **The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.** IEEE, 2005. p. 915–920. ISBN 0-7803-9159-4. Disponível em: <http://ieeexplore.ieee.org/document/1452516/>. Citado 2 vezes nas páginas 11 e 12.

Lughofer, E.; Pratama, M. Online Active Learning in Data Stream Regression Using Uncertainty Sampling Based on Evolving Generalized Fuzzy Models. **IEEE Transactions on Fuzzy Systems**, v. 26, n. 1, 2018. ISSN 10636706. Citado na página 17.

Lughofer, E.; Pratama, M.; Skrjanc, I. Incremental Rule Splitting in Generalized Evolving Fuzzy Systems for Autonomous Drift Compensation. **IEEE Transactions on Fuzzy Systems**, Institute of Electrical and Electronics Engineers Inc., v. 26, n. 4, p. 1854–1865, aug 2018. ISSN 1063-6706. Disponível em: <https://ieeexplore.ieee.org/document/8039219/>. Citado na página 10.

Lughofer, E.; Pratama, M.; Skrjanc, I. Incremental rule splitting in generalized evolving fuzzy systems for autonomous drift compensation. **IEEE Transactions on Fuzzy Systems**, v. 26, n. 4, p. 1854–1865, 2018. Citado 2 vezes nas páginas 48 e 52.

Lughofer, Edwin. **Evolving fuzzy systems-methodologies, advanced concepts and applications**. [S.l.]: Springer, 2011. v. 53. Citado na página 2.

Meng, D.; Cao, G.; Cao, W.; He, Z. Supervised Feature Learning Network Based on the Improved LLE for face recognition. In: **2016 International Conference on Audio, Language and Image Processing (ICALIP)**. IEEE, 2016. p. 306–311. ISBN 978-1-5090-0654-0. Disponível em: <http://ieeexplore.ieee.org/document/7846591/>. Citado na página 7.

Montgomery, D. **Design and Analysis of Experiments, 8th Edition**. [S.l.]: John Wiley & Sons, Incorporated, 2013. ISBN 9781118214718. Citado 2 vezes nas páginas 21 e 23.

Naik, A. K.; Kuppili, V.; Edla, D. R. Efficient feature selection using one-pass generalized classifier neural network and binary bat algorithm with a novel fitness function. **Soft Computing**, Springer Berlin Heidelberg, p. 1–13, jul 2019. ISSN 1432-7643. Disponível em: <http://link.springer.com/10.1007/s00500-019-04218-6>. Citado na página 1.

Nizam, T.; Hassan, S. I. Exemplifying the effects of distance metrics on clustering techniques: F-measure, accuracy and efficiency. In: **2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)**. [S.l.: s.n.], 2020. p. 39–44. Citado na página 41.

Nugroho, A.; Fanani, A. Z.; Shidik, G. F. Evaluation of feature selection using wrapper for numeric dataset with random forest algorithm. In: **2021 International Seminar on Application for Technology of Information and Communication (iSemantic)**. [S.l.: s.n.], 2021. p. 179–183. Citado na página 15.

Pao, Y.-H.; Takefuji, Y. Functional-link net computing: theory, system architecture, and functionalities. **Computer**, v. 25, p. 76–79, 1992. Citado na página 18.

Pratama, M.; Jie Lu; Guangquan Zhang. An incremental interval Type-2 neural fuzzy Classifier. In: **2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. IEEE, 2015. p. 1–8. ISBN 978-1-4673-7428-6. Disponível em: <http://ieeexplore.ieee.org/document/7337801/>. Citado 2 vezes nas páginas 12 e 64.

Pratama, M.; Lu, J.; Anavatti, S.; Lughofer, E.; Lim, C. P. An incremental meta-cognitive-based scaffolding fuzzy neural network. **Neurocomputing**, v. 171, 2016. ISSN 18728286. Citado na página 18.

Pratama, M.; Lughofer, E.; Er, M. J.; Rahayu, W.; Dillon, T. Evolving type-2 recurrent fuzzy neural network. In: **2016 International Joint Conference on Neural Networks (IJCNN)**. IEEE, 2016. p. 1841–1848. ISBN 978-1-5090-0620-5. Disponível em: <http://ieeexplore.ieee.org/document/7727423/>. Citado 2 vezes nas páginas 16 e 64.

Pratama, M.; Pedrycz, W.; Lughofer, E. Evolving Ensemble Fuzzy Classifier. **IEEE Transactions on Fuzzy Systems**, v. 26, n. 5, p. 2552–2567, oct 2018. ISSN 1063-6706. Disponível em: <https://ieeexplore.ieee.org/document/8265083/>. Citado na página 17.

Pratama, M.; Pedrycz, W.; Webb, G. I. An incremental construction of deep neuro fuzzy system for continual learning of nonstationary data streams. **IEEE Transactions on Fuzzy Systems**, v. 28, n. 7, p. 1315–1328, 2020. Citado na página 18.

Precup, R.-E.; Bojan-Dragos, C.-A.; Hedrea, E.-L.; Rarinca, M.-D.; Petriu, E. M. Evolving fuzzy models for the position control of magnetic levitation systems. In: **2017 Evolving and Adaptive Intelligent Systems (EAIS)**. IEEE, 2017. p. 1–6. ISBN 978-1-5090-6444-1. Disponível em: <http://ieeexplore.ieee.org/document/7954839/>. Citado na página 13.

Riasi, A.; Delrobaei, M. Automatic classification of parkinson's disease using best parameters of forward and backward walking. In: **2021 29th Iranian Conference on Electrical Engineering (ICEE)**. [S.l.: s.n.], 2021. p. 939–942. Citado na página 15.

Rodrigues, F.; Silva, A.; Lemos, A. Evolving fuzzy predictor with multivariable gaussian participatory learning and multi-innovations recursive weighted least squares: efmi. **Evolving Systems**, p. 1–20, 2022. Citado 2 vezes nas páginas 11 e 54.

Rodrigues, F. P. S.; Silva, A. M.; Lemos, A. P. Evolving fuzzy system with multivariable gaussian participatory learning and recursive maximum correntropy - efce. In: **2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. [S.l.: s.n.], 2021. p. 1–6. Citado na página 54.

Rong, H. J.; Angelov, P. P.; Gu, X.; Bai, J. M. Stability of Evolving Fuzzy Systems Based on Data Clouds. **IEEE Transactions on Fuzzy Systems**, Institute of Electrical and Electronics Engineers Inc., v. 26, n. 5, p. 2774–2784, oct 2018. ISSN 10636706. Citado na página 10.

Runger, D. **Applied Statistics and Probability for Engineers**. [S.l.]: Wiley, 2003. ISBN 9780471204541. Citado 2 vezes nas páginas 21 e 23.

Sahoo, P.; Riedel, T. **Mean Value Theorems and Functional Equations**. World Scientific, 1998. ISBN 9789810235444. Disponível em: <https://books.google.com.br/books?id=My0wvbZgoF4C>. Citado na página 24.

Saranya, G.; Pravin, A. An Efficient Feature Selection Approach using Sensitivity Analysis for Machine Learning based Heart Desease Classification. **Proceedings - 2021 IEEE 10th International Conference on Communication Systems and Network Technologies, CSNT 2021**, Institute of Electrical and Electronics Engineers Inc., p. 539–542, 2021. Citado na página 9.

Scikit-learn. 2020. Disponível em: <https://scikit-learn.org/>. Citado 2 vezes nas páginas 30 e 68.

Shah, J.; Wang, W. An evolving neuro-fuzzy classifier for fault diagnosis of gear systems. **ISA Transactions**, v. 123, p. 372–380, 2022. ISSN 0019-0578. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0019057821002743>. Citado 2 vezes nas páginas 11 e 15.

Shah, S. M. S.; Shah, F. A.; Hussain, S. A.; Batool, S. Support Vector Machines-based Heart Disease Diagnosis using Feature Subset, Wrapping Selection and Extraction Methods. **Computers & Electrical Engineering**, v. 84, p. 106628, 2020. ISSN 0045-7906. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0045790620304833>. Citado na página 10.

Shihabudheen, K.; Pillai, G. N. Regularized extreme learning adaptive neuro-fuzzy algorithm for regression and classification. **Knowledge-Based Systems**, v. 127, p. 100–113, 2017. ISSN 0950-7051. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950705117301831>. Citado na página 65.

Silva, A. M.; Caminhas, W.; Lemos, A.; Gomide, F. A fast learning algorithm for evolving neo-fuzzy neuron. **Applied Soft Computing**, v. 14, p. 194–209, 2013. ISSN 1568-4946. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1568494613001373>. Citado 6 vezes nas páginas 2, 10, 11, 12, 16 e 54.

Silva, A. M.; Caminhas, W. M.; Lemos, A. P.; Gomide, F. Evolving Neo-fuzzy Neural Network with Adaptive Feature Selection. In: **2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence**. IEEE, 2013. p. 341–349. ISBN 978-1-4799-3194-1. Disponível em: <http://ieeexplore.ieee.org/document/6855873/>. Citado na página 16.

Silva-Junior, G. A.; Silva, A. M. A simple and efficient incremental missing data imputation method for evolving neo-fuzzy network. **Evolving Systems**, v. 13, n. 2, p. 201–220, 2022. Citado na página 11.

Skrjanc, I.; Iglesias, J. A.; Sanchis, A.; Leite, D.; Lughofer, E.; Gomide, F. Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey. **Information Sciences**, v. 490, p. 344–368, 2019. ISSN 0020-0255. Citado 2 vezes nas páginas 2 e 11.

Souza, P. V.; Lughofer, E. An evolving neuro-fuzzy system based on uni-nullneurons with advanced interpretability capabilities. **Neurocomputing**, v. 451, p. 231–251, 2021. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S092523122100607X>. Citado na página 65.

Suchetha, N. K.; Nikhil, A.; Hrudya, P. Comparing the Wrapper Feature Selection Evaluators on Twitter Sentiment Classification. In: **2019 International Conference on Computational Intelligence in Data Science (ICCIDS)**. IEEE, 2019. p. 1–6. ISBN 978-1-5386-9471-8. Disponível em: <https://ieeexplore.ieee.org/document/8862033/>. Citado na página 7.

Tavares, E.; Silva, A. M.; Moita, G. F. A fast clustering algorithm for evolving fuzzy classifier based on samples mean. **Journal of Intelligent & Fuzzy Systems**, IOS Press, v. 43, n. 6, p. 6897–6908, 7 2022. ISSN 1064-1246. Citado 3 vezes nas páginas 11, 49 e 51.

Torres, L. M. M.; Serra, G. L. O. A Novel Approach for Online Multivariable Evolving Fuzzy Modeling from Experimental Data. In: **2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. IEEE, 2018. p. 1–6. ISBN 978-1-5090-6020-7. Disponível em: <https://ieeexplore.ieee.org/document/8491470/>. Citado na página 14.

UCI. **UC Irvine Machine Learning Repository**. 2023. Https://archive.ics.uci.edu/ml/index.php. Citado 2 vezes nas páginas 29 e 65.

Umbarkar, S.; Shukla, S. Analysis of Heuristic based Feature Reduction method in Intrusion Detection System. In: **2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)**. IEEE, 2018. p. 717–720. ISBN 978-1-5386-3045-7. Disponível em: <https://ieeexplore.ieee.org/document/8474283/>. Citado na página 7.

Vora, S.; Yang, H. A comprehensive study of eleven feature selection algorithms and their impact on text classification. In: **Proceedings of Computing Conference 2017**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. v. 2018-January, p. 440–449. ISBN 9781509054435. Citado 3 vezes nas páginas 1, 7 e 8.

Wang, C.; Qi, Y.; Shao, M.; Hu, Q.; Chen, D.; Qian, Y.; Lin, Y. A Fitting Model for Feature Selection With Fuzzy Rough Sets. **IEEE Transactions on Fuzzy Systems**, v. 25, n. 4, p. 741–753, aug 2017. ISSN 1063-6706. Disponível em: <http://ieeexplore.ieee.org/document/7482828/>. Citado na página 8.

Wang, J.; Zhao, P.; Hoi, S. C.; Jin, R. Online feature selection and its applications. **IEEE Transactions on Knowledge and Data Engineering**, v. 26, n. 3, p. 698–710, mar 2014. ISSN 10414347. Citado na página 17.

Wang, K.; Lu, J.; Liu, A.; Zhang, G.; Xiong, L. Evolving gradient boost: A pruning scheme based on loss improvement ratio for learning under concept drift. **IEEE Transactions on Cybernetics**, v. 53, n. 4, p. 2110–2123, 2023. Citado na página 65.

Wang, X.-T.; Luan, X.-Z. Bayesian Penalized Method for Streaming Feature Selection. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 7, p. 103815–103822, jul 2019. ISSN 2169-3536. Citado na página 7.

Yan, X.; Nazmi, S.; Erol, B. A.; Homaifar, A.; Gebru, B.; Tunstel, E. An efficient unsupervised feature selection procedure through feature clustering. **Pattern Recognition Letters**, Elsevier B.V., v. 131, p. 277–284, mar 2020. ISSN 01678655. Citado na página 9.

Yu, L.; Liu, H. Efficient Feature Selection via Analysis of Relevance and Redundancy. **J. Mach. Learn. Res.**, JMLR.org, v. 5, p. 1205–1224, dec 2004. ISSN 1532-4435. Citado na página 17.

YU, L.; LIU, H. Efficient feature selection via analysis of relevance and redundancy. **J. Mach. Learn. Res.**, JMLR.org, v. 5, p. 1205–1224, dec 2004. ISSN 1532-4435. Citado na página 18.

Zadeh, L. A. Fuzzy sets. **Information and Control**, Academic Press, v. 8, n. 3, p. 338–353, jun 1965. ISSN 00199958. Citado na página 41.

Zeng, Z.; Heng, X. Feature selection and visualization based on interaction dominance. In: **Proceedings - 2019 IEEE 4th International Conference on Data Science in Cyberspace, DSC 2019**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2019. v. 2019-Janua, p. 668–673. ISBN 9781728145280. Citado na página 9.

Zhu, L.; Cao, L.; Yang, J.; Lei, J. Evolving soft subspace clustering. **Applied Soft Computing Journal**, Elsevier, v. 14, p. 210–228, jan 2014. ISSN 15684946. Citado na página 11.

Apêndices

APÊNDICE A – Atributos utilizados no método *filter*

Nesta seção, a Tabela 19 apresenta as quantidade de atributos utilizados nos experimentos; os valores são de uma redução em etapas de 20% do total de atributos; são apresentados o conjunto de dados, o número de atributos presentes no conjunto de dados e o número de atributos selecionados para os experimentos. As tabelas 20-27 traz a ordem dos atributos ranqueados pelos métodos de FS.

Tabela 19 – Número de atributos selecionados.

Bases	Atributos	Atributos Selecionados
Heart Disease	13	11, 9, 7, 5
Cervical Cancer	33	26, 20, 14, 8
Mobile Price	20	16, 12, 8, 4
Page Blocks	10	8, 6, 4, 3
QSAR Biodegradation	41	32, 24, 16, 8
South African Health	9	8, 7, 6, 5
Glass Identification	10	8, 6, 4, 3
Wine	13	11, 9, 7, 5

Tabela 20 – Ranqueamento dos atributos - *Heart Disease*.

Método	Ordem
MRFS	9, 12, 3, 10, 2, 11, 7, 13, 6, 8, 1, 4, 5
Kruskal-Wallis	3, 7, 11, 6, 2, 10, 12, 9, 1, 4, 5, 8, 13
ReliefF	12, 13, 3, 10, 11, 8, 7, 1, 4, 2, 9, 6, 5
Chi-Square	8, 10, 12, 3, 9, 5, 1, 4, 11, 2, 13, 7, 6

Tabela 21 – Ranqueamento dos atributos - *Cervical Cancer*.

Método	Ordem
MRFS	15, 18, 19, 20, 25, 24, 16, 22, 21, 31, 32, 33, 29, 27, 28, 30, 23, 17, 14, 13, 12, 26, 10, 11, 9, 6, 5, 7, 4, 1, 8, 3, 2
Kruskal-Wallis	32, 31, 33, 29, 27, 30, 17, 14, 12, 13, 26, 10, 11, 23, 7, 20, 5, 28, 6, 15, 22, 24, 21, 19, 25, 16, 18, 9, 8, 2, 4, 3, 1
ReliefF	32, 33, 31, 8, 9, 4, 5, 1, 29, 20, 30, 3, 23, 27, 18, 28, 10, 26, 6, 12, 13, 11, 2, 19, 14, 16, 7, 17, 15, 22, 25, 24, 21
Chi-Square	32, 31, 33, 9, 29, 27, 13, 30, 20, 6, 11, 17, 14, 12, 26, 1, 23, 28, 10, 4, 18, 7, 5, 16, 3, 25, 19, 21, 24, 8, 2, 15, 22

Tabela 22 – Ranqueamento dos atributos - *Mobile Price*.

Método	Ordem
MRFS	14, 12, 1, 13, 6, 7, 19, 16, 10, 4, 2, 5, 8, 9, 15, 20, 17, 11, 18, 3
Kruskal-Wallis	3, 5, 18, 8, 6, 20, 4, 2, 19, 14, 17, 16, 15, 1, 13, 12, 10, 9, 7, 11
ReliefF	14, 1, 13, 12, 17, 5, 7, 10, 9, 15, 11, 3, 16, 18, 8, 2, 6, 4, 19, 20
Chi-Square	14, 12, 1, 13, 9, 7, 16, 17, 5, 15, 11, 10, 19, 6, 8, 2, 3, 4, 20, 18

Tabela 23 – Ranqueamento dos atributos - *Page Blocks*.

Método	Ordem
MRFS	9, 4, 7, 10, 2, 8, 3, 1, 5, 6
Kruskal-Wallis	1, 7, 5, 10, 4, 6, 2, 3, 9, 8
ReliefF	1, 6, 2, 8, 5, 9, 3, 10, 4, 7
Chi-Square	3, 8, 9, 10, 4, 7, 1, 2, 5, 6

Tabela 24 – Ranqueamento dos atributos - *QSAR Biodegradation*.

Método	Ordem
MRFS	28, 21, 19, 41, 20, 6, 4, 3, 25, 26, 40, 24, 29, 5, 33, 32, 11, 34, 7, 38, 23, 10, 14, 31, 9, 36, 30, 13, 39, 8, 1, 27, 35, 15, 22, 37, 17, 18, 16, 2, 12
Kruskal-Wallis	30, 9, 40, 35, 10, 28, 16, 32, 19, 29, 26, 24, 21, 4, 20, 14, 6, 34, 25, 11, 38, 5, 41, 23, 22, 3, 39, 7, 37, 36, 8, 12, 33, 13, 31, 15, 17, 18, 27, 2, 1
ReliefF	38, 19, 22, 1, 37, 28, 35, 40, 39, 8, 36, 4, 2, 27, 24, 14, 16, 12, 15, 18, 29, 32, 10, 31, 23, 34, 13, 30, 11, 9, 20, 5, 26, 17, 6, 25, 41, 21, 3, 7, 33
Chi-Square	8, 11, 7, 34, 5, 23, 41, 33, 30, 3, 31, 39, 10, 38, 36, 15, 14, 6, 1, 13, 9, 12, 25, 27, 32, 37, 20, 40, 22, 35, 4, 24, 21, 26, 29, 16, 19, 28, 17, 18, 2

Tabela 25 – Ranqueamento dos atributos - *South African Health*.

Método	Ordem
MRFS	2, 9, 3, 5, 4, 8, 1, 6, 7
Kruskal-Wallis	2, 8, 1, 3, 4, 5, 6, 7, 9
ReliefF	2, 1, 8, 6, 9, 4, 3, 7, 5
Chi-Square	9, 2, 4, 1, 8, 3, 6, 5, 7

Tabela 26 – Ranqueamento dos atributos - *Glass Identification*.

Método	Ordem
MRFS	9, 7, 10, 4, 5, 1, 8, 3, 6, 2
Kruskal-Wallis	5, 2, 1, 9, 3, 4, 6, 7, 8, 10
ReliefF	1, 7, 2, 8, 5, 6, 9, 3, 4, 10
Chi-Square	1, 9, 4, 7, 5, 3, 8, 10, 6, 2

Tabela 27 – Ranqueamento dos atributos - *Wine*.

Método	Ordem
MRFS	10, 7, 13, 12, 6, 2, 9, 8, 11, 4, 1, 3, 5
Kruskal-Wallis	8, 2, 11, 9, 4, 1, 3, 5, 6, 7, 10, 12
ReliefF	13, 1, 4, 5, 7, 9, 3, 10, 11, 12, 2, 6, 8
Chi-Square	13, 10, 7, 5, 4, 2, 12, 6, 9, 1, 11, 8, 3

APÊNDICE B – Resultados gerais aplicando o método *wrapper* com MRFS

Neste apêndice, são mostrados os resultados experimentais considerando todos os conjuntos de parâmetros usando os classificadores RNA (B.1), KNN (B.2), NB (B.3) e SVM (B.4). As tabelas de 28-42 apresentam a acurácia média, o número de atributos selecionados pelo classificador com MRFS e o número total de atributos de cada base de dados. Os resultados foram calculados após 10 repetições de cada experimento.

B.1 Resultados com RNA

As tabelas 28-30 apresentam os resultados com RNA-MRFS e RNA. Nestes experimentos, RNA foi executada com 3 diferentes algoritmos de aprendizado: quasi-Newton; gradiente descendente estocástico e; otimizador baseado em gradiente estocástico.

Tabela 28 – Desempenho da RNA com método quasi-Newton.

Bases	RNA	RNA-MRFS	# Atrib. Selec.	# Atrib.
Wine	40,00	**76,95**	4 ± 2	13
Glass Identification Dataset	86,97	**90,69**	6 ± 3	10
Heart Disease	**75,25**	74,27	9 ± 1	13
South African Health	64,30	**71,83**	6 ± 1	9
Cervical Cancer Risk	93,06	**94,55**	29 ± 1	35
QSAR Biodegradation	65,88	**85,07**	38 ± 1	41
Mobile Price Classification	31,25	**32,15**	13 ± 5	20
Page Blocks Classification	88,44	**89,17**	8 ± 1	10
Médias	70,91	**82,23**		

Tabela 29 – Desempenho da RNA com método gradiente descendente.

Bases	RNA	RNA-MRFS	# Atrib. Selec.	# Atrib.
Wine	32,78	**45,00**	9 ± 2	13
Glass Identification Dataset	**53,48**	38,60	8 ± 1	10
Heart Disease	61,97	**66,56**	6 ± 3	13
South African Health	62,90	**66,45**	7 ± 1	9
Cervical Cancer Risk	92,98	**94,93**	27 ± 3	35
QSAR Biodegradation	82,94	**83,27**	38 ± 1	41
Mobile Price Classification	23,07	**26,40**	17 ± 2	20
Page Blocks Classification	89,88	**91,76**	3 ± 3	10
Médias	64,51	**65,80**		

Tabela 30 – Desempenho da RNA com método otimizador baseado em gradiente estocástico.

Bases	RNA	RNA-MRFS	# Atrib. Selec.	# Atrib.
Wine	35,56	**43,89**	9 ± 2	13
Glass Identification Dataset	81,16	**81,39**	8 ± 1	10
Heart Disease	67,54	**84,43**	9 ± 1	13
South African Health	67,20	**72,68**	6 ± 3	9
Cervical Cancer Risk	93,28	**95,00**	29 ± 3	35
QSAR Biodegradation	65,83	**87,82**	38 ± 1	41
Mobile Price Classification	57,27	**79,60**	18 ± 1	20
Page Blocks Classification	95,21	**91,25**	8 ± 1	10
Médias	68,43	**77,54**		

B.2 Resultados com KNN

As tabelas 31, 32, 33 e 34 mostram os resultados obtidos com KNN-MRFS e KNN. Os experimentos com KNN-MRFS e KNN usaram a distância euclidiana e número de vizinhos 2, 5, 30 e 50.

Tabela 31 – Desempenho do KNN com 2 vizinhos.

Bases	KNN	KNN-MRFS	# Atrib. Selec.	# Atrib.
Wine	63,33	**92,22**	2 ± 1	13
Glass Identification Dataset	96,97	**98,13**	6 ± 2	10
Heart Disease	56,23	**77,37**	9 ± 2	13
South African Health	60,86	**67,74**	7 ± 1	9
Cervical Cancer Risk	92,39	**93,36**	28 ± 2	35
QSAR Biodegradation	81,13	**83,45**	27 ± 3	41
Mobile Price Classification	89,37	**90,45**	12 ± 4	20
Page Blocks Classification	94,96	**95,32**	6 ± 2	10
Médias	75,15	**85,38**		

Tabela 32 – Desempenho do KNN com 5 vizinhos.

Bases	KNN	KNN-MRFS	# Atrib. Selec.	# Atrib.
Wine	68,33	**89,72**	2 ± 1	13
Glass Identification Dataset	96,97	**97,20**	6 ± 2	10
Heart Disease	59,56	**83,61**	9 ± 1	13
South African Health	59,89	**65,80**	6 ± 2	9
Cervical Cancer Risk	92,01	**93,51**	28 ± 1	35
QSAR Biodegradation	80,56	**84,88**	25 ± 5	41
Mobile Price Classification	91,52	**92,45**	11 ± 4	20
Page Blocks Classification	95,65	**96,09**	6 ± 2	10
Médias	76,22	**85,79**		

Tabela 33 – Desempenho do KNN com 30 vizinhos.

Bases	KNN	KNN-MRFS	# Atrib. Selec.	# Atrib.
Wine	71,38	**76,94**	3 ± 1	13
Glass Identification Dataset	88,14	**93,48**	5 ± 3	10
Heart Disease	65,91	**79,01**	9 ± 1	13
South African Health	65,91	**69,67**	7 ± 1	9
Cervical Cancer Risk	93,58	**94,10**	29 ± 1	35
QSAR Biodegradation	77,53	**81,27**	27 ± 3	41
Mobile Price Classification	92,47	**93,00**	8 ± 6	20
Page Blocks Classification	93,89	**94,24**	6 ± 3	10
Médias	77,08	**82,41**		

Tabela 34 – Desempenho do KNN com 50 vizinhos.

Bases	KNN	KNN-MRFS	# Atrib. Selec.	# Atrib.
Wine	67,49	**87,50**	2 ± 1	13
Glass Identification Dataset	80,23	**83,25**	5 ± 4	10
Heart Disease	61,69	**80,01**	8 ± 1	13
South African Health	67,85	**71,39**	6 ± 2	9
Cervical Cancer Risk	92,09	**93,06**	29 ± 1	35
QSAR Biodegradation	76,33	**81,18**	29 ± 2	41
Mobile Price Classification	91,02	**92,75**	8 ± 8	20
Page Blocks Classification	92,86	**93,15**	7 ± 1	10
Médias	74,28	**82,73**		

B.3 Resultados com NB

As tabelas 35 e 36 apresentam os resultados com NB-MRFS e NB. Nestes experimentos, NB-MRFS e NB utilizaram os modelos com distribuição Gaussiana e Bernoulli.

Tabela 35 – Desempenho do NB com modelo Gaussiano.

Bases	NB	NB-MRFS	# Atrib. Selec.	# Atrib.
Wine	95,28	**96,67**	8 ± 4	13
Glass Identification Dataset	81,39	**88,37**	5 ± 4	10
Heart Disease	80,33	**80,82**	10 ± 2	13
South African Health	69,68	**71,82**	5 ± 3	9
Cervical Cancer Risk	25,23	**88,58**	29 ± 1	35
QSAR Biodegradation	70,81	**72,18**	33 ± 7	41
Mobile Price Classification	79,92	**81,40**	16 ± 3	20
Page Blocks Classification	88,67	**90,08**	7 ± 2	10
Médias	70,45	**83,07**		

Tabela 36 – Desempenho do NB com modelo Bernoulli.

Bases	NB	NB-MRFS	# Atrib. Selec.	# Atrib.
Wine	36,39	**37,22**	11 $\pm$ 1	13
Glass Identification Dataset	41,16	**51,16**	4 $\pm$ 3	10
Heart Disease	79,18	**83,77**	6 $\pm$ 2	13
South African Health	64,62	**68,38**	3 $\pm$ 3	9
Cervical Cancer Risk	93,43	**95,67**	30 $\pm$ 1	35
QSAR Biodegradation	77,16	**77,87**	34 $\pm$ 4	41
Mobile Price Classification	22,70	**26,50**	17 $\pm$ 2	20
Page Blocks Classification	89,87	**90,04**	3 $\pm$ 1	10
Médias	65,32	**69,01**		

B.4 Resultados com SVM

As tabelas de 37 a 42 mostram os resultados com SVM-MRFS e SVM. Nestes experimentos, SVM foi utilizado os parâmetros C com valores de 0,1, 0,5 e 1, também com os *Kernels* Linear e RBF.

Tabela 37 – Desempenho do SVM com *kernel* linear e C 0,1.

Bases	SVM	SVM-MRFS	# Atrib. Selec.	# Atrib.
Wine	95,55	**97,22**	5 $\pm$ 3	13
Glass Identification Dataset	97,90	**100,00**	5 $\pm$ 2	10
Heart Disease	83,12	**84,26**	7 $\pm$ 3	13
South African Health	73,25	**73,76**	6 $\pm$ 1	9
Cervical Cancer Risk	94,40	**95,52**	10 $\pm$ 2	35
QSAR Biodegradation	86,78	**87,06**	35 $\pm$ 3	41
Mobile Price Classification	97,32	**97,60**	16 $\pm$ 1	20
Page Blocks Classification	**96,52**	96,37	7 $\pm$ 1	10
Médias	88,50	**89,64**		

Tabela 38 – Desempenho do SVM com *kernel* RBF e C 0,1.

Bases	SVM	SVM-MRFS	# Atrib. Selec.	# Atrib.
Wine	37,78	**90,56**	2 $\pm$ 1	13
Glass Identification Dataset	79,33	**80,00**	5 $\pm$ 1	10
Heart Disease	56,56	**84,10**	7 $\pm$ 2	13
South African Health	61,29	**62,37**	4 $\pm$ 3	9
Cervical Cancer Risk	**93,28**	92,01	2 $\pm$ 1	35
QSAR Biodegradation	73,74	**75,83**	37 $\pm$ 2	41
Mobile Price Classification	90,28	**90,85**	11 $\pm$ 3	20
Page Blocks Classification	**89,57**	89,53	8 $\pm$ 1	10
Médias	67,00	**80,81**		

Tabela 39 – Desempenho do SVM com *kernel* linear e C 0,5.

Bases	SVM	SVM-MRFS	# Atrib. Selec.	# Atrib.
Wine	94,72	**98,06**	6 ± 3	13
Glass Identification Dataset	98,60	**99,53**	5 ± 1	10
Heart Disease	82,79	**83,77**	9 ± 2	13
South African Health	**74,94**	70,75	5 ± 2	9
Cervical Cancer Risk	**95,75**	**95,75**	10 ± 1	35
QSAR Biodegradation	86,54	**86,92**	35 ± 5	41
Mobile Price Classification	96,84	**97,05**	17 ± 1	20
Page Blocks Classification	96,63	**96,72**	8 ± 1	10
Médias	88,89	**89,13**		

Tabela 40 – Desempenho do SVM com *kernel* RBF e C 0,5.

Bases	SVM	SVM-MRFS	# Atrib. Selec.	# Atrib.
Wine	37,78	**59,72**	8 ± 2	13
Glass Identification Dataset	81,15	**83,72**	5 ± 1	10
Heart Disease	56,56	**84,59**	8 ± 3	13
South African Health	61,94	**67,82**	3 ± 2	9
Cervical Cancer Risk	93,28	**93,81**	13 ± 5	35
QSAR Biodegradation	83,51	**84,31**	38 ± 1	41
Mobile Price Classification	94,67	**94,80**	12 ± 5	20
Page Blocks Classification	89,75	**89,85**	3 ± 1	10
Médias	69,04	**79,00**		

Tabela 41 – Desempenho do SVM com *kernel* linear e C 1.

Bases	SVM	SVM-MRFS	# Atrib. Selec.	# Atrib.
Wine	94,72	**97,78**	7 ± 2	13
Glass Identification Dataset	98,60	**99,53**	5 ± 1	10
Heart Disease	83,28	**83,77**	9 ± 2	13
South African Health	74,07	**74,62**	7 ± 1	9
Cervical Cancer Risk	**95,75**	95,67	20 ± 5	35
QSAR Biodegradation	86,78	**87,25**	38 ± 1	41
Mobile Price Classification	97,17	**97,70**	17 ± 1	20
Page Blocks Classification	**96,66**	**96,66**	8 ± 1	10
Médias	88,87	**89,77**		

Tabela 42 – Desempenho do SVM com *kernel* RBF e C 1.

Bases	SVM	SVM-MRFS	# Atrib. Selec.	# Atrib.
Wine	40,56	**58,34**	7 ± 3	13
Glass Identification Dataset	81,39	**83,72**	5 ± 1	10
Heart Disease	56,72	**75,58**	8 ± 3	13
South African Health	63,01	**71,18**	6 ± 1	9
Cervical Cancer Risk	93,28	**95,00**	24 ± 4	35
QSAR Biodegradation	84,64	**85,69**	34 ± 3	41
Mobile Price Classification	94,93	**94,90**	16 ± 2	20
Page Blocks Classification	90,06	**90,21**	4 ± 1	10
Médias	69,93	**78,25**		