

**UNIVERSIDADE FEDERAL DE MINAS GERAIS**  
**Escola de Engenharia**  
**Programa de Pós-Graduação em Engenharia Elétrica**

Rafael Ramos Celestino Silva

**PREVISÃO DE SÉRIES TEMPORAIS MULTIVARIADAS UTILIZANDO MODELO  
HÍBRIDO INTERPRETÁVEL COM ÁRVORES DE DECISÃO FUZZY**

Belo Horizonte

2021

Rafael Ramos Celestino Silva

**PREVISÃO DE SÉRIES TEMPORAIS MULTIVARIADAS UTILIZANDO MODELO  
HÍBRIDO INTERPRETÁVEL COM ÁRVORES DE DECISÃO FUZZY**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais em cumprimento dos requisitos para o grau de Mestre em Engenharia Elétrica.

Orientador: Frederico Gadelha Guimarães

Belo Horizonte

2021

S586p

Silva, Rafael Ramos Celestino.

Previsão de séries temporais multivariadas utilizando modelo híbrido interpretável com árvores de decisão fuzzy [recurso eletrônico] / Rafael Ramos Celestino Silva. - 2021.

1 recurso online (136 f. : il., color.) : pdf.

Orientador: Frederico Gadelha Guimarães.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f.127-136.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia Elétrica - Teses. 2. Análise multivariada – Teses.  
3. Sistemas difusos – Teses. I. Guimarães, Frederico Gadelha.  
II. Universidade Federal de Minas Gerais. Escola de Engenharia.  
III. Título.

CDU: 621.3(043)

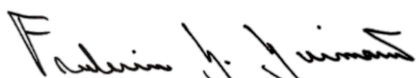
# "Previsão de Séries Temporais Multivariadas Utilizando Modelo Híbrido Interpretável Com Árvores de Decisão Fuzzy"

Rafael Ramos Celestino Silva

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 14 de dezembro de 2021.

Por:



---

**Prof. Dr. Frederico Gadelha Guimarães**  
DEE (UFMG) - Orientador



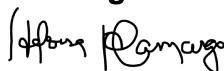
---

**Prof. Dr. Walmir Matos Caminhas**  
DELT (UFMG)



---

**Prof. Dr. Gustavo Pessin**  
(Instituto Tecnológico Vale (ITV-MI))



---

**Prof. Dr. Heloisa de Arruda Camargo**  
Departamento de Computação (UFSCAR)

*Dedico esse trabalho a minha família, meu pai José e minha mãe Letícia, a minha irmã Camila, aos meus avôs Francisco e Lourival, as minhas avós Marlene e Judite, a todos os amigos e familiares que tornaram a minha vida num caminho feliz de realizações e gratidão. Dedico, também, a Deus que mesmo com a minha fé errante, direciona meus passos aos caminhos do meu coração e que por algum motivo me manteve vivo, otimista e em boas companhias mesmo em tempos tão difíceis e aleatórios.*

# Agradecimentos

Agradeço a minha família pelo apoio dado durante toda a minha vida.

Ao meu pai José e minha mãe Letícia por serem sempre compreensivos e presentes.

À minha irmã Camila pelas piadas e por ser minha advogada "free of charge".

Aos meus avôs Francisco e Lourival e as minhas avós Judite e Marlene pelo carinho e amor mesmo que distantes.

Aos meus tios e tias pelos momentos divertidos em família.

Aos meus familiares distantes que guardo no coração.

À minha namorada Thati pelo apoio, suporte e sempre me mandando comida quando estou triste.

Aos meus amigos da Engenharia da UFMG pelas festas malucas, pelos estudos e pela alegria.

Aos meus amigos do Santo Agostinho pela felicidade das vezes que os vejo.

Aos meus amigos do intercâmbio por me despertarem o sentimento de saudade.

Aos orientadores e professores que tive ao longo da vida pelos ensinamentos.

Aos colegas do MINDS pela boa companhia e ajuda.

Ao meu orientador Prof. Dr. Frederico pela paciência, bom humor e calma em transmitir sua experiência e conhecimento.

Aos meus gatos "Jadirzinho" e "Mel Gibson" por serem engraçados, espertos, por me darem alguma sorte, por conversarem comigo e por sempre trazerem felicidade quando estão por perto.

A UFMG por possibilitar minha educação superior.

Ao antigo legislador brasileiro que positivou a educação pública na constituição brasileira e que possibilitou toda minha formação superior, senso de civilidade e a compreensão de que o conhecimento deve ser encarado como um projeto de vida.

Obrigado, pessoal!

*“A vida das pessoas não termina quando elas morrem. Termina quando elas perdem a fé.”*

*Uchiha Itachi*

# Resumo

O presente trabalho tem como objetivo combinar a ideia dos sistemas fuzzy com os algoritmos de árvore decisão já existentes a fim de conceber um método de previsão para séries temporais multivariadas baseado num banco de regras. Dessa forma, obtém-se um método que possui explicabilidade e interpretabilidade a partir dos parâmetros de entrada do modelo, propriedade essa que é vantajosa comparado aos outros diversos métodos que produzem expressões matemáticas ou algoritmos caixa preta (alta complexidade), como é o caso das redes neurais. Para realizar tal projeto, utilizou-se a linguagem python, além de bibliotecas e frameworks já concebidos nessa mesma linguagem, a fim de criar funções que possam combinar árvores de decisão com atributos fuzzificados. Em adição, para realizar tal projeto, utilizou-se de bibliotecas como a “scikit learn”, a qual possui árvores de decisão já implementadas. Além disso, usou-se a biblioteca pyFTS a fim de possibilitar a transformação das séries temporais numéricas (variável de saída) em séries fuzzy, e da mesma forma transformar os atributos tanto numéricos quanto categóricos em séries fuzzy. Ao realizar tal procedimento utilizou-se árvores de decisão de classificação para gerar regras do tipo “se-então” com atributos fuzzy tanto em seu antecedente quanto em seu consequente. Após o treinamento de uma árvore e a obtenção das regras ou o grafo da própria árvore como base de conhecimento, usa-se funções de defuzzificação a fim de transformar as saídas fuzzy em saídas numéricas e então prever o próximo valor da série. Ademais, os resultados desse método se apresentaram competitivos com outros modelos de previsão de séries multivariadas, e apresenta flexibilidade para a aplicação em variadas séries como as seguintes: consumo de energia elétrica, demanda do serviço de metrô, contrato futuro do índice bovespa(B3), QoS (Qualidade de Serviço) e regime fluvial. Em adição, foi possível verificar que a produção de regras como base de conhecimento possibilitou explicação para os movimentos das séries bem como entender quais atributos são determinantes aos movimentos de cada série, e como esses influenciam a série no tempo. Por fim, vale destacar que o método proposto obteve bons resultados no que tange as métricas de regressão utilizadas como MAE, RMSE, MAPE, SMAPE, entre outras. Com isso, além da boa acurácia nas previsões, como o modelo é composto por regras é possível gerar conhecimento e entender como os valores das séries são gerados no tempo.

Palavras-chave: séries temporais; lógica fuzzy; sistemas fuzzy; séries temporais fuzzy; árvores de decisão; árvores de decisão fuzzy; métricas de regressão.



# Abstract

The present work aims to combine the idea of fuzzy systems with existing decision trees algorithms in order to design a multivariate forecasting method based on a rule set. In this way, a method which has explainability and interpretability based on the model's input parameters is obtained, a property that is advantageous when compared to other various methods that provide mathematical expressions or black box algorithms (high complex algorithms), such as the neural networks models. The project was conceived in python language, and some other libraries and frameworks already designed in python, which allowed the creation of functions that can combine decision trees with fuzzyfied attributes. For instance, it was used the scikit-learn librarie, which has decision trees algorithms already implemented and the pyFTS library, which enables the transformation of numerical time series (output variable) into a fuzzy time series, and likewise transforms numerical and categorical input attributes into fuzzy time series. When performing this procedure, classification decision trees were used to generate "if-then" rules with fuzzy attributes both in their antecedent and in their consequent. After training a tree and obtaining the rules or graph from the tree itself as a knowledge base, defuzzification functions are used in order to transform fuzzy values into numerical ones and then predict the next time series value. In addition, the results of the method are competitive with other multivariate time series forecasting models, and it presents the flexibility to be applied in several series such as the following: electricity consumption, metro service demand, bovespa future contracts (B3), QoS (Quality of Service) and river regime. In addition, it was possible to verify that the production of rules as a knowledge base allowed an explanation for the movements of the series, as well as understanding which attributes are determinant to the movements of each series, and how these influence the series over time. Finally, it is worth noting that the obtained method has good results when considering the regression metrics used as MAE, RMSE, MAPE, SMAPE, among others. Beyond the good accuracy, as the model is composed of rules, it is possible to generate knowledge and understand how the values of the series are generated over time.

*Keywords: time series; fuzzy logic; fuzzy systems; fuzzy time series; decision trees; fuzzy decision trees; regression metrics.*

# Lista de Ilustrações

|   |    |
|---|----|
| Figura 1 – Exemplo das operações de união (a), interseção (b) e negação (c). . . .  | 29 |
| Figura 2 – Função de pertinência triangular. . . . .  | 30 |
| Figura 3 – Função de pertinência trapezoidal. . . . .   | 31 |
| Figura 4 – Função de pertinência gaussiana. . . . .   | 31 |
| Figura 5 – Exemplo da variável linguística “Distância”. . . . .   | 32 |
| Figura 6 – Arquitetura de um sistema de inferência fuzzy. . . . .   | 33 |
| Figura 7 – Exemplo de homocedasticidade. . . . .  | 37 |
| Figura 8 – Exemplo de heterocedasticidade. . . . .  | 37 |
| Figura 9 – Série Febre Tifoide - Série Original vs Séries Prevista. . . . .   | 43 |
| Figura 10 – Resultado do MAPE em relação ao SVM para as doenças listadas. . . .   | 43 |
| Figura 11 – Série Original vs Séries Previstas para a Carga Agregada. . . . .   | 44 |
| Figura 12 – Série numérica dos alunos matriculados na Universidade do Alabama. .  | 46 |
| Figura 13 – FTS dos alunos matriculados na Universidade do Alabama. . . . .   | 46 |
| Figura 14 – Interpretabilidade vs Predição. . . . .   | 48 |
| Figura 15 – Imagens criadas por algoritmos genéticos irreconhecíveis por seres hu-<br>manos, no entanto as redes neurais profundas, treinadas com o banco de<br>dados ImageNet, acreditam com mais 99,6% de certeza ser um objeto<br>familiar . . . . . | 50 |
| Figura 16 – Exemplo de árvore de decisão . . . . .  | 52 |
| Figura 17 – Erro da Regressão versus Taxa de Erro da Classificação. . . . .   | 53 |
| Figura 18 – Exemplo dos particionamento dos dados realizado por uma estrutura<br>de árvore de decisão. . . . .  | 56 |
| Figura 19 – Arquitetura da aplicação de árvores de decisão fuzzy ( <i>fuzzy decision<br/>trees</i> ) para previsão de séries temporais. . . . .   | 66 |
| Figura 20 – Exemplo de fuzzificação da Série de um Ativo $Y(R\$)$ no $Tempo(1h)$ . .  | 68 |
| Figura 21 – Discriminação das regras por conjunto de valores da série. . . . .  | 75 |
| Figura 22 – Série original do tempo de resposta e tempo de resposta utilizando um<br>filtro de média móvel exponencial com $p=8$ . . . . .  | 80 |
| Figura 23 – <i>Boxplot</i> da série original de tempo de resposta e da série filtrada. . . .  | 80 |
| Figura 24 – Intervalos de treino e teste utilizados no experimento. . . . .   | 85 |
| Figura 25 – Série original da máxima do rio Xingu. . . . .  | 89 |

|   |     |
|---|-----|
| Figura 26 – Histograma da série original da máxima do rio Xingu. . . . .  | 90  |
| Figura 27 – Autocorrelação da série original do rio Xingu. . . . .  | 90  |
| Figura 28 – Intervalos de treino e teste para os experimentos da série de máxima do Rio Xingu. . . . .                                    | 95  |
| Figura 29 – Série da potência ativa e da série filtrada. . . . .  | 98  |
| Figura 30 – Autocorrelação da série da demanda de potência ativa em uma casa. . .   | 98  |
| Figura 31 – Intervalos de treino e teste utilizados no experimento da demanda de energia. . . . .   | 99  |
| Figura 32 – Série original do fluxo de passageiros no metrô e a série do filtro de média móvel exponencial com $p=4$ . . . . .            | 105 |
| Figura 33 – Autocorrelação da série filtrada do volume de passageiros do metrô. . .   | 105 |
| Figura 34 – Intervalos de treino e teste utilizados no experimento de fluxo de passageiros em um Metrô. . . . .                           | 110 |
| Figura 35 – Série do preço de fechamento do minicontrato futuro do IBOVESPA. . .  | 113 |
| Figura 36 – Série da primeira diferença do preço de fechamento do minicontrato do IBOVESPA. . . . .                                       | 113 |
| Figura 37 – Autocorrelação da série de primeira diferença. . . . .  | 114 |
| Figura 38 – Intervalos de treino e teste utilizados no experimento de previsão da série temporal do contrato futuro do IBOVESPA. . . . .  | 114 |
| Figura 39 – Matriz de confusão do algoritmo RF Fuzzy para o método FDT-FTS simples referente ao problema do fluxo de passageiros. . . . . | 120 |
| Figura 40 – Previsão da série fuzzy pelo ID3 Fuzzy gerado pelo método FDT-FTS simples para o problema de fluxo de passageiros. . . . .    | 121 |

# Lista de Tabelas

|   |    |
|---|----|
| Tabela 1 – Variáveis utilizadas no problema de previsão do tempo de resposta para o serviço de <i>QoS</i> . . . . .   | 79 |
| Tabela 2 – Análise da autocorrelação da série de <i>QoS</i> com e sem o uso de filtro. . . . .  | 81 |
| Tabela 3 – Gráficos da série original temporal e as previsões dos métodos de árvore utilizando o método FDT-FTS composto para a série de <i>QoS</i> . . . . .             | 82 |
| Tabela 4 – Comparação da importância entre as variáveis de entrada do experimento da série de <i>QoS</i> . . . . .  | 83 |
| Tabela 5 – Gráficos da série original e as previsões intervalares do método de FDT-FTS composto para a série de <i>QoS</i> . . . . .                                      | 84 |
| Tabela 6 – Métricas utilizadas na série de <i>QoS</i> com o filtro de média móvel exponencial aplicando-se o método FDT-FTS simples. . . . .                              | 85 |
| Tabela 7 – Métricas utilizadas na série de <i>QoS</i> com o filtro de média móvel exponencial aplicando-se o método FDT-FTS composto. . . . .                             | 85 |
| Tabela 8 – Métricas utilizadas na série de <i>QoS</i> com o filtro de média móvel exponencial aplicando-se o método DTR. . . . .  | 85 |
| Tabela 9 – Métricas expressas em <i>boxplots</i> para o método FDT-FTS simples e composto, além do método de comparação DTR para a série de <i>QoS</i> . . . . .          | 86 |
| Tabela 10 – Variáveis utilizadas para a previsão da série temporal do Rio Xingu. . . . .  | 88 |
| Tabela 11 – Gráficos da série de máxima e as previsões dos métodos de árvore utilizando a FDT-FTS composta para a série de máxima do Rio Xingu. . . . .                   | 91 |
| Tabela 12 – Comparação da importância entre as variáveis de entrada do experimento da série de máxima do Rio Xingu. . . . .   | 92 |
| Tabela 13 – Gráficos da série de máxima do Rio Xingu e as previsões intervalares da FDT-FTS composta. . . . .   | 93 |
| Tabela 14 – Métricas expressas em <i>boxplots</i> para o método FDT-FTS simples e composto, além do método de comparação DTR para a série de máxima do rio Xingu. . . . . | 94 |
| Tabela 15 – Métricas utilizadas na série de máxima do Rio Xingu aplicando-se o método FDT-FTS simples. . . . .  | 95 |
| Tabela 16 – Métricas utilizadas na série de máxima do Rio Xingu aplicando-se o método FDT-FTS composto. . . . .   | 95 |

|   |     |
|---|-----|
| Tabela 17 – Métricas utilizadas na série de máxima do Rio Xingu aplicando-se o método DTR. . . . .  | 95  |
| Tabela 18 – Valores de RMSE, MAE e $R^2$ comparados entre os algoritmos RF Fuzzy Composto e GA com Regressão Linear (Vieira et al., 2021). . . . .                                    | 96  |
| Tabela 19 – Significado das variáveis presentes no conjunto de dados da demanda de energia. . . . .   | 97  |
| Tabela 20 – Gráficos da série de potência ativa global e as previsões dos métodos de árvore utilizando o FDT-FTS composto. . . . .  | 99  |
| Tabela 21 – Comparação da importância entre as variáveis de entrada que os métodos atribuem para o experimento da série de Energia. . . . .   | 100 |
| Tabela 22 – Gráficos da série de potência ativa global e as previsões intervalares do método FDT-FTS composto. . . . .  | 101 |
| Tabela 23 – Métricas expressas em <i>boxplots</i> para o método FDT-FTS simples e composto, além dos métodos de comparação DTR para a série de potência ativa global. . . . .         | 102 |
| Tabela 24 – Métricas utilizadas na série previsão da potência ativa global com o filtro de média móvel exponencial aplicando-se o método FDT-FTS simples. . . . .                     | 103 |
| Tabela 25 – Métricas utilizadas na série de potência ativa global com o filtro de média móvel exponencial aplicando-se o método FDT-FTS composto. . . . .                             | 103 |
| Tabela 26 – Métricas utilizadas na série de potência ativa global com o filtro de média móvel exponencial aplicando-se o método DTR. . . . .  | 103 |
| Tabela 27 – Descrição das variáveis de entrada e saída do modelo referente ao problema de fluxo de passageiro no metrô situado entre <i>Minneapolis</i> e <i>Saint Paul</i> . . . . . | 104 |
| Tabela 28 – Gráficos da série de fluxo de passageiros e as previsões dos métodos de árvore utilizando o FDT-FTS composto. . . . .   | 106 |
| Tabela 29 – Gráficos da série de fluxo de passageiros e as previsões intervalares do FDT-FTS composto. . . . .  | 107 |
| Tabela 30 – Comparação da importância entre as variáveis de entrada que os métodos atribuem para o experimento da série de fluxo de passageiros. . . . .                              | 108 |
| Tabela 31 – Métricas expressas em <i>boxplots</i> para o método FDT-FTS simples e composto, além do método DTR para a série de fluxo de passageiros. . . . .                          | 109 |
| Tabela 32 – Métricas utilizadas na série de fluxo de passageiros com o filtro de média móvel exponencial aplicando-se o método FDT-FTS simples. . . . .                               | 110 |
| Tabela 33 – Métricas utilizadas na série de fluxo de passageiros com o filtro de média móvel exponencial aplicando-se o método FDT-FTS composto. . . . .                              | 110 |
| Tabela 34 – Métricas utilizadas na série de fluxo de passageiros com o filtro de média móvel exponencial aplicando-se o método DTR. . . . .   | 110 |

|             |   |     |
|-------------|---|-----|
| Tabela 35 – | Varáveis utilizadas para a previsão do preço de fechamento do IBOVESPA.   | 112 |
| Tabela 36 – | Gráficos da série original temporal e as previsões dos métodos de árvore utilizando a FDT-FTS composta. . . . .                                 | 115 |
| Tabela 37 – | Gráficos da série da primeira diferença e as previsões intervalares da FDT-FTS composta. . . . .  | 116 |
| Tabela 38 – | Comparação da importância entre as variáveis de entrada que os métodos atribuem para o experimento da série de primeira diferença. . . . .      | 117 |
| Tabela 39 – | Métricas para a série do preço de fechamento expressas em <i>boxplots</i> para o método FDT-FTS simples e composto, além do método DTR. . . . . | 118 |
| Tabela 40 – | Métricas utilizadas para medição da acurácia na série do preço de fechamento para o método FDT-FTS simples. . . . .                             | 119 |
| Tabela 41 – | Métricas utilizadas para medição da acurácia na série do preço de fechamento para o método FDT-FTS composto. . . . .                            | 119 |
| Tabela 42 – | Métricas utilizadas na série de preço de fechamento para o método DTR.  | 119 |
| Tabela 43 – | Regras extraídas da árvore de decisão ID3 fuzzy gerada pelo método FDT-FTS simples relativa a série da Fig. 40. . . . .                         | 122 |
| Tabela 44 – | Comparação dos métodos em face aos resultados das medianas proveniente das métricas utilizadas. . . . .   | 123 |

# Lista de Abreviaturas e Siglas

|        |  |
|--------|--|
| AR     | <i>Autoregressive Model</i>  |
| ARMA   | <i>Autoregressive Moving Average Model</i>                             |
| ARIMA  | <i>Autoregressive Integrated Moving Average Model</i>                  |
| VAR    | <i>Vector Autoregressive Model</i>                                     |
| VARMA  | <i>Vector Autoregressive Moving Average Model</i>                      |
| VARIMA | <i>Vector Autoregressive Integrated Moving Average Model</i>           |
| SVM    | <i>Support Vector Machine</i>  |
| SVR    | <i>Support Vector Regression</i>                                       |
| SARIMA | <i>Seasonal Autoregressive Integrated Moving Average Modelo</i>        |
| ARCH   | <i>Autoregressive Conditional Heteroscedasticity Model</i>             |
| GARCH  | <i>Generalized Autoregressive Conditional Heteroscedasticity Model</i> |
| ML     | <i>Machine Learning</i>  |
| MAPE   | <i>Mean Average Percentage Error</i>                                   |
| SMAPE  | <i>Symmetric Mean Absolute percentage Error</i>                        |
| RMSE   | <i>Root Mean Squared Error</i>   |
| NRMSE  | <i>Normalized Root Mean Squared Error</i>                              |
| MAE    | <i>Mean Absolute Error</i>   |
| $R^2$  | <i>Determination Coefficient</i>                                       |
| MLP    | <i>Multi Layer Perceptron</i>  |
| LSTM   | <i>Long short-term memory</i>  |
| GA     | <i>Genetic Algorithm</i>   |

|         |  |
|---------|--|
| CART    | <i>Classification and Regression Trees</i>           |
| ID3     | <i>Iterative Dichotomiser 3</i>                      |
| RF      | <i>Random Forest</i>                                 |
| FTS     | <i>Fuzzy Time Series</i>                             |
| FDT     | <i>Fuzzy Decision Tree</i>                           |
| FDT-FTS | <i>Fuzzy Decision Tree - Fuzzy Time Series Model</i> |
| DTR     | <i>Decision Tree Regressor</i>                       |



# Sumário

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introdução</b>                                       | <b>23</b> |
| 1.1      | Motivação   | 23        |
| 1.2      | Objetivos   | 24        |
| 1.3      | Contribuições   | 25        |
| 1.4      | Estrutura do Documento                                  | 25        |
| <b>2</b> | <b>Fundamentação Teórica</b>                            | <b>27</b> |
| 2.1      | Teoria de Conjuntos Fuzzy                               | 27        |
| 2.1.1    | Conjuntos Fuzzy   | 28        |
| 2.1.2    | Funções de Pertinência                                  | 29        |
| 2.1.3    | Variáveis Linguísticas                                  | 31        |
| 2.1.4    | Sistemas de Inferência Fuzzy                            | 33        |
| 2.2      | Séries Temporais - Time Series (TS)                     | 35        |
| 2.2.1    | Séries Temporais Univariadas e Multivariadas            | 38        |
| 2.2.2    | Modelos de previsão em Séries Temporais                 | 38        |
| 2.2.3    | Métricas de Acurácia                                    | 40        |
| 2.2.4    | Aplicações Provenientes dos Modelos de Séries Temporais | 42        |
| 2.3      | Séries Temporais Fuzzy - Fuzzy Time Series (FTS)        | 44        |
| 2.4      | Explicabilidade e Interpretabilidade                    | 45        |
| 2.5      | Utilizando métodos de classificação para regressão      | 51        |
| 2.6      | Considerações Finais - Capítulo 2                       | 54        |
| <b>3</b> | <b>Árvores de Decisão</b>                               | <b>55</b> |
| 3.1      | Árvores de Decisão Fuzzy - Fuzzy Decision Trees (FDT)   | 61        |
| 3.2      | Considerações Finais - Capítulo 3                       | 63        |
| <b>4</b> | <b>Métodos Propostos</b>                                | <b>65</b> |
| 4.1      | Introdução  | 65        |
| 4.1.1    | Leitura e Análise de Dados                              | 65        |
| 4.1.2    | Processo de fuzzificação                                | 66        |
| 4.1.3    | Treino da Árvore de Decisão e Defuzzificação            | 68        |
| 4.2      | FDT-FTS Método Simples                                  | 70        |
| 4.3      | FDT-FTS Método Composto                                 | 72        |
| 4.3.1    | Avaliação dos Resultados das Árvores Fuzzy              | 74        |
| 4.4      | Interpretabilidade e Explicabilidade do Modelo          | 75        |
| 4.5      | Considerações Finais - Capítulo 4                       | 76        |
| <b>5</b> | <b>Experimentos Computacionais</b>                      | <b>77</b> |
| 5.1      | QoS (Qualidade do Serviço)                              | 78        |
| 5.1.1    | Caracterização dos Dados                                | 80        |

|          |   |            |
|----------|---|------------|
| 5.1.2    | Resultados . . . . .                                | 81         |
| 5.1.3    | Discussão dos Resultados . . . . .                  | 87         |
| 5.2      | Previsão do Nível do Rio Xingu . . . . .            | 87         |
| 5.2.1    | Caracterização dos Dados . . . . .                  | 89         |
| 5.2.2    | Resultados . . . . .                                | 90         |
| 5.2.3    | Discussão dos Resultados . . . . .                  | 95         |
| 5.3      | Consumo de Energia . . . . .                        | 96         |
| 5.3.1    | Caracterização dos Dados . . . . .                  | 97         |
| 5.3.2    | Resultados . . . . .                                | 98         |
| 5.3.3    | Discussão dos Resultados . . . . .                  | 103        |
| 5.4      | Fluxo de Passageiros num Serviço de Metrô . . . . . | 104        |
| 5.4.1    | Caracterização dos Dados . . . . .                  | 105        |
| 5.4.2    | Resultados . . . . .                                | 106        |
| 5.4.3    | Discussão dos Resultados . . . . .                  | 110        |
| 5.5      | Séries Temporais Financeiras - IBOVESPA . . . . .   | 111        |
| 5.5.1    | Caracterização dos Dados . . . . .                  | 112        |
| 5.5.2    | Resultados . . . . .                                | 114        |
| 5.5.3    | Discussão dos Resultados . . . . .                  | 119        |
| 5.6      | Considerações Finais . . . . .                      | 120        |
| <b>6</b> | <b>Conclusão . . . . .</b>                          | <b>125</b> |
|          | <b>Referências . . . . .</b>                        | <b>127</b> |

# Capítulo 1

## Introdução

“Ser sábio não é contar todas as  
gostas de uma queda d’água.  
Ser sábio é aprender o porquê  
da água procurar a terra.”

---

Conselho dos Soratamis

### 1.1 Motivação

Uma série temporal se refere ao conjunto de dados que ocorrem em sequência no tempo. Dada essa situação, os modelos que tratam de séries temporais univariadas e multivariadas são abordados de formas diferentes dos problemas básicos da estatística clássica inferencial, em que não há dependência entre os valores da própria série, dependência temporal e influência de variáveis externas. Quando se trabalha com esses modelos busca-se estudar como os dados são gerados, quais circunstâncias levam às suas variações, à previsão dos seus valores e à tentativa de compreender determinadas propriedades da série como: sazonalidade, tendência, níveis, periodicidade e ocorrência de ruído(Montgomery et al., 2015; Morettin and Toloi, 2004; Montgomery and Runger, 2003).

Como exemplos que abordam problemas de séries temporais tem-se: a previsão do tráfego de pessoas em um determinado aeroporto ou estação de metrô, previsão do valor de uma ação na bolsa de valores, previsões climáticas relacionadas à quantidade de chuvas numa determinada região, previsão do aumento do consumo de energia num país, entre outras várias situações.

Todas essas séries podem possuir alto ou baixo grau de autocorrelação, e até mesmo alto ou baixo valor de correlação com séries externas. Assim, um primeiro desafio é identificar quais dos métodos utilizar e quais tipos de dados serão utilizados na entrada desses. Um segundo ponto, trata-se da verificação dos pressupostos para a aplicação

dos variados métodos que podemos empregar. Uma vez que não se verifica a presença desses pressupostos numa série temporal, a aplicação de um determinado modelo é invalidada. Como exemplo, para computar o intervalo de previsão do modelo ARIMA deve-se verificar a normalidade e a não correlação dos resíduos. Ao se aplicar determinados testes estatísticos e não se verificarem esses pressupostos, tem-se uma evidência de que esse intervalo possui valores incorretos e não poderá ser aplicado. Dessa maneira, boa parte dos modelos matemáticos apresentam restrições quanto ao seu uso, tendo em vista que descobrir um modelo adequado para determinadas séries pode representar um trabalho hercúleo (Montgomery et al., 2015; Morettin and Toloi, 2004; Montgomery and Runger, 2003; Maia, 2017).

Nesse sentido, utiliza-se algoritmos de aprendizado de máquina supervisionados para realizar regressões e prever valores de séries temporais tanto multivariadas como univariadas. Uma das vantagens do uso desses algoritmos é o fato de não possuírem pressupostos que devem ser validados a priori. Além disso, muitos aceitam como entrada variáveis tanto numéricas como categóricas, e podem ser otimizados ou regularizados a fim de personalizar sua aplicação a um problema específico (Geron, 2017; Luger, 2013). No entanto, mesmo tendo em vista a vasta gama de vantagens as quais os algoritmos de aprendizado de máquina possuem, é visível algumas desvantagens que podem impossibilitar o seu uso. Primeiramente, boa parte dos algoritmos de regressão e dos modelos matemáticos tradicionais não possuem uma boa capacidade de explicar de forma direta a previsão de um determinado valor para uma série temporal. Ademais, quanto mais complexa é a série temporal tratada, mais complexo, geralmente, são os algoritmos e modelos utilizados, levando à criação de pipelines de dados e processamento extensos e computacionalmente custosos (Geron, 2017; Luger, 2013; Du et al., 2019).

Tendo em vista o exposto, a motivação deste trabalho se traduz na necessidade de conceber um método de aprendizado de máquina que seja eficiente na previsão de séries temporais, computacionalmente simples e rápido em termos de treinamento e teste, além de possuir alta explicabilidade.

Para isso, a presente dissertação propõe um método em que serão combinadas árvores de decisão com séries temporais fuzzy, a fim de possibilitar um método explicável para previsão de séries temporais, em específico as séries multivariadas.

## 1.2 **Objetivos**

A presente dissertação apresenta os seguintes objetivos:

1. Combinar árvores de decisão com séries temporais fuzzy como principal motor para geração de regras as quais serão responsáveis por prever os valores futuros de uma

série;

2. Prover um conjunto de regras ou uma estrutura de dados que funcione como uma base de conhecimento que possua explicabilidade, não apenas no sentido das regras proverem uma explicação direta da ocorrência de um evento, mas também, fornecer evidências probabilísticas quanto a uma determinada hipótese extraída dos resultados das Árvores de Decisão Fuzzy (Fuzzy Decision Trees - FDTs);
3. Utilizar bases de dados relativas às séries temporais multivariadas com propriedades diversas a fim de demonstrar o uso dos métodos desenvolvidos.
4. Comparar o método de FDT concebido com um método tradicional de aprendizado de máquina para previsão de séries multivariadas a partir de métricas já conhecidas quanto a avaliação dos resultados para previsão de séries temporais.
5. Conceber um conjunto de funções na linguagem Python que possibilitem a qualquer árvore de decisão ser utilizada como uma FDT para prever valores de séries temporais.

## 1.3 Contribuições

A principal contribuição deste trabalho é a implementação de funções em Python, disponíveis ao público, para a aplicação do modelo proposto que combina árvores de decisão com lógica fuzzy a fim de possibilitar a previsão de séries temporais multivariadas. Esse método tem o foco de prover explicabilidade e interpretabilidade aos modelos de previsão de série temporal, assim como a geração de conhecimento. Por fim, outro produto desse trabalho foi a publicação no congresso FUZZ-IEEE 2021, com a seguinte citação:

- R. R. C. Silva, W. M. Caminhas, P. C. de Lima e Silva and F. G. Guimarães, “A C4.5 Fuzzy Decision Tree Method for Multivariate Time Series Forecasting,” 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2021, pp. 1-6, doi: 10.1109/FUZZ45933.2021.9494439. URL da Apresentação do Artigo no Congresso: <https://youtu.be/CSeoihvPq24>

## 1.4 Estrutura do Documento

Os capítulos seguintes deste trabalho abordarão:

- O Capítulo 2 abordará o referencial teórico em que será tratado as teorias sobre conjuntos fuzzy, séries temporais e séries temporais fuzzy, além de temas importantes como o uso de algoritmos de classificação para se resolver problemas de regressão, e como a explicabilidade e interpretabilidade tem se mostrado importante nas pesquisas correntes de aprendizado de máquina.

- 
- O capítulo 3 trará uma explicação mais profunda quanto aos métodos de árvore de decisão e as árvores de decisão fuzzy.
  - O capítulo 4 explicitará como o método proposto de árvores de decisão fuzzy pode ser utilizado para prever séries temporais multivariadas, e mostrará as duas variações do método.
  - O capítulo 5 tratará da aplicação do método por meio de experimentos computacionais concebidos em conjunto de dados que abordam variados problemas como as previsões de: consumo de energia elétrica, demanda do serviço de metrô, contrato futuro do índice bovespa (B3), QoS (Qualidade de Serviço) e o nível do rio Xingu.
  - O capítulo 6, por fim, concluirá explicitando as formas de utilização do método, além dos resultados que o diferenciam dos outros modelos de previsão em séries temporais multivariadas. Em adição, será tratado os pontos que podem ser melhorados em trabalhos futuros, por exemplo, os aspectos de otimização e regularização do modelo.

## Capítulo 2

# Fundamentação Teórica

“Em condições normais, o cientista não é um Inovador, mas um solucionador de quebra-cabeças, e os quebra-cabeças sobre os quais ele se concentra são apenas aqueles que ele acredita poderem ser declarados e resolvidos dentro da tradição científica existente.”

---

Thomas Kuhn

O presente capítulo discorrerá sobre conceitos importantes e necessários ao entendimento dos métodos de árvores fuzzy que trataremos em seguida. Neste capítulo será feita uma revisão quanto a lógica fuzzy, séries temporais, séries temporais fuzzy, bem como a necessidade atual da explicabilidade em problemas relativos aos métodos de aprendizado de máquina e o uso dos algoritmos de classificação para a resolução de problemas de séries.

### 2.1 Teoria de Conjuntos Fuzzy

A lógica fuzzy foi introduzida no meio científico por Lotfi A. Zadeh, professor na Universidade de Berkeley, no ano de 1965 (Zadeh, 1965). A teoria desenvolvida relativa aos conjuntos fuzzy apresenta uma lógica diferente da utilizada pela teoria de conjuntos ordinários, em que os valores estão ou não contidos em determinado grupo. Os conjuntos fuzzy trazem uma ideia de incerteza e por isso trazem uma inovação por serem capazes de lidar com fenômenos inerentemente imprecisos, como é o caso do raciocínio humano, que por muitas vezes é ambíguo e não definido em termos numéricos. Com isso, a lógica fuzzy possibilita a criação de sistemas fuzzy, que solucionam problemas complexos com níveis

de imprecisão consideráveis. Como exemplo, utiliza-se sistemas fuzzy para a resolução de problemas de classificação, regressão de séries, controle de processos, entre outros (Zadeh, 1965; Gomide et al., 1995; Jang et al., 1997).

### 2.1.1 Conjuntos Fuzzy

Dentre as principais distinções entre a lógica clássica e a lógica fuzzy tem-se que a lógica clássica é bivalente, ou seja, uma premissa pode possuir apenas dois valores, falso ou verdadeiro (comumente tratado com os valores numéricos 0 ou 1). Já a lógica fuzzy traz o conceito de multivaloração, em que uma variável ou premissa pode assumir diversos valores dentro de um intervalo. Nesse sentido, o valor de pertinência de um determinado elemento na teoria de conjuntos ordinários irá assumir valor de 0 ou 1 (pertence ou não a um conjunto), já nos conjuntos fuzzy o grau de pertinência pode obter um valor entre o intervalo de 0 ou 1 e pertencer a mais de um conjunto. Assim, na lógica difusa existe uma passagem gradual entre os elementos que estão ou não em um conjunto, além do grau de pertinência fornecer a intuição de que um elemento está contido em maior grau num conjunto do que em outro (Jang et al., 1997). Com isso, apresenta-se a definição de um conjunto fuzzy, que pode ser expresso por:

$$A = \{(x, \mu_A(x)) \mid x \in X\} \quad (2.1)$$

Na Eq. (2.1),  $X$  representa o universo de discurso,  $\mu_A(x)$  é a função de pertinência e  $x$  é um elemento de  $X$ .

A função de pertinência expressa a compatibilidade que um elemento  $x$  possui em relação a um conjunto  $A$ . Assim, quanto mais próximo do valor de 1, maior o grau de compatibilidade de  $x$  com  $A$ , e quanto mais próximo o valor com 0, menor é o grau de compatibilidade. Vale citar que dentre os vários operadores presentes na teoria de conjuntos fuzzy, tem-se a interseção, a união e a negação como operadores com alto grau de relevância (Jang et al., 1997; Tanscheit, 2004). Discorrendo brevemente sobre a aplicação desse operadores:

1. A interseção na teoria dos conjuntos fuzzy é semelhante ao operador de conjunção da lógica de primeira ordem, operando “E”, que significa interseção entre dois conjuntos. No entanto, o cálculo é computado de forma distinta da lógica tradicional, dada pela seguinte expressão:

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (2.2)$$

Na Eq. (2.2), tem-se que  $\mu_A$  e  $\mu_B$  são as funções de pertinência dos respectivos conjuntos  $A$  e  $B$  dado um elemento  $x$ , que se estiver em algum desses dois conjuntos terá valor maior que 0, caso contrário terá valor nulo. Logo, para o cálculo da



interseção basta escolher qual conjunto tem menor valor de pertinência. Como exemplo, caso  $\mu_A(x) > \mu_B(x)$ , logo:  $\mu_{A \cap B}(x) = \mu_B(x)$ .

2. Já a união de conjuntos fuzzy é semelhante ao operador de disjunção da lógica de primeira ordem, operando “OU”, que significa a união de dois conjuntos. Entretanto, a união, diferentemente do operador de disjunção, é calculada da seguinte maneira:

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (2.3)$$

Na Eq. (2.3), tem-se que  $\mu_A(x)$  e  $\mu_B(x)$  são as funções de pertinência dos respectivos conjuntos A e B, da mesma forma que na (2.2). Porém, para se calcular a união escolhe-se o maior valor de pertinência dados dois conjuntos fuzzy. Por exemplo, caso  $\mu_A(x) > \mu_B(x)$ , logo:  $\mu_{A \cup B}(x) = \mu_A(x)$ .

3. Por fim, o operador de negação usual, o qual é computado da seguinte forma:

$$\mu_{A'}(x) = 1 - \mu_A(x) \quad (2.4)$$

Assim, vale explicitar que  $\mu_A(x)$  é a função de pertinência para um dado valor  $x$  na Eq. (2.4).

Por fim, veja na Fig. 1 um exemplo das operações descritas anteriormente.

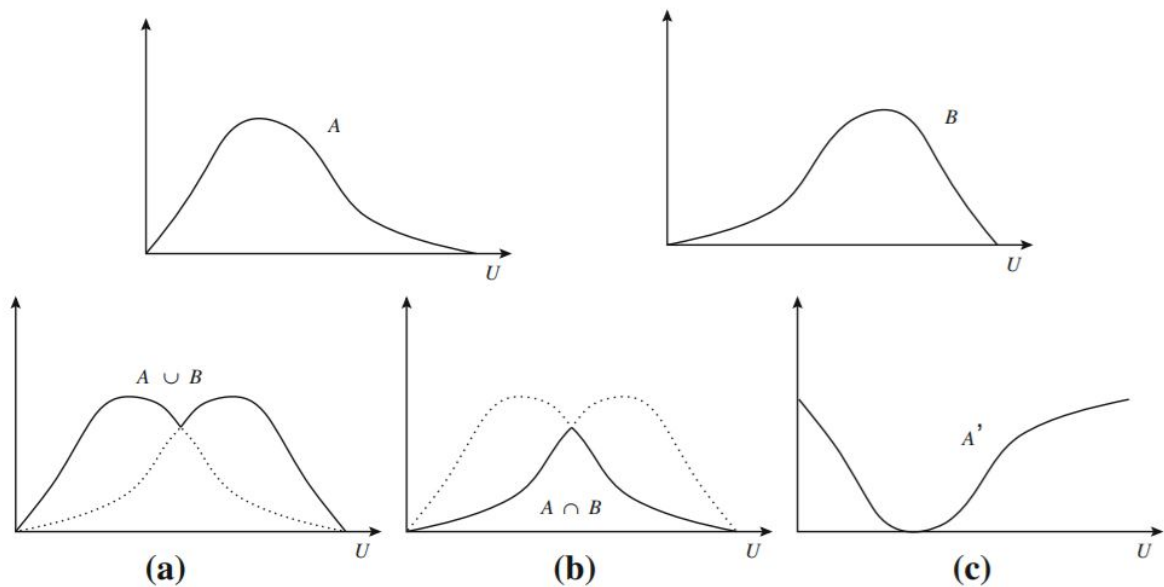


Figura 1 – Exemplo das operações de união (a), interseção (b) e negação (c).

Fonte: (Jang et al., 1997)

### 2.1.2 Funções de Pertinência

As funções de pertinência são funções numéricas que definem um conjunto nebuloso e são responsáveis por atribuir a certa variável linguística valores de pertinência dado

um universo de discurso. As funções de pertinência mais frequentes na literatura são as funções triangulares, trapezoidais e gaussianas (Jang et al., 1997; Lima, 2015; Tanscheit, 2004). Nesse sentido vale discorrer sobre as características de cada uma dessas funções, dessa maneira tem-se:

1. Função Triangular: Descreve-se a função de pertinência triangular em termos matemáticos na Eq. (2.5) (Lima, 2015). Veja a Fig. 2 como exemplo da representação gráfica da função de pertinência triangular.

$$\mu_A(x; a, m, b) = \begin{cases} 0, & \text{se } x \leq a \\ \frac{x-a}{m-a}, & \text{se } a < x \leq m \\ \frac{b-x}{b-m}, & \text{se } m < x < b \\ 0, & \text{se } x \geq b \end{cases} \quad (2.5)$$

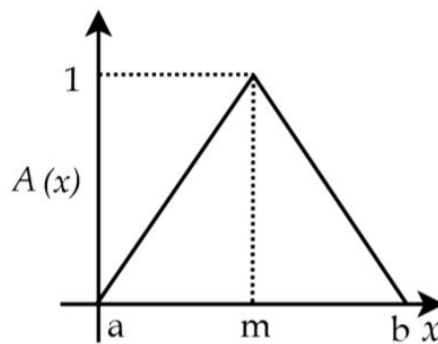


Figura 2 – Função de pertinência triangular.

Fonte: (Lima, 2015)

2. Função Trapezoidal: Descreve-se a função de pertinência trapezoidal em termos matemáticos na Eq. (2.6) (Lima, 2015). Veja a Fig. 3 como exemplo da representação gráfica da função de pertinência trapezoidal.

$$\mu_A(x; a, m, n, b) = \begin{cases} 0, & \text{se } x \leq a \\ \frac{x-a}{m-a}, & \text{se } a < x \leq m \\ 1, & \text{se } m < x \leq n \\ \frac{b-x}{b-n}, & \text{se } n < x < b \\ 0, & \text{se } x \geq b \end{cases} \quad (2.6)$$

3. Função Gaussiana: Descreve-se a função de pertinência gaussiana em termos matemáticos na Eq. (2.7) (Lima, 2015). Veja a Fig. 4 como exemplo da representação

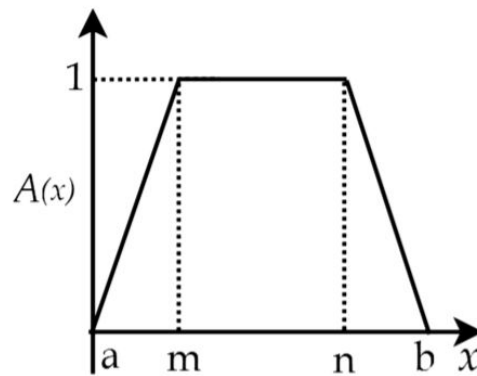


Figura 3 – Função de pertinência trapezoidal.

Fonte: (Lima, 2015)

gráfica da função de pertinência gaussiana.

$$\mu_A(x) = e^{-k(x-m)^2} \quad (2.7)$$

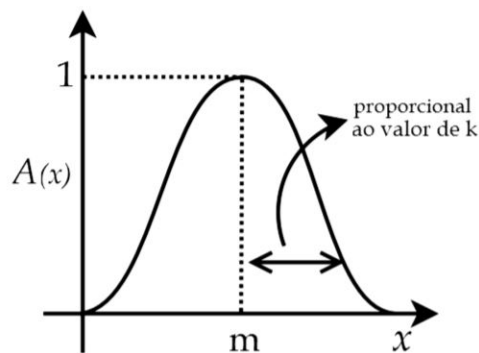


Figura 4 – Função de pertinência gaussiana.

Fonte: (Lima, 2015)

### 2.1.3 Variáveis Linguísticas

Para Tanscheit (2004), uma variável linguística é a representação de um conjunto fuzzy por meio de termos ou sentenças de uma linguagem (variáveis linguísticas não assumem valores numéricos). Esses termos podem ser primários (termos adjetivos, como: baixo, alto, grande, pequeno, etc), lógicos (termos conectivos como: “não”, “ou”, “e”), modificadores (em geral advérbios, como exemplo: muito, extremamente, suavemente, etc). Dessa maneira, uma variável linguística consegue expressar e lidar com uma situação imprecisa e inderteminada quando atribui à um conjunto fuzzy certas denominações, as quais são responsáveis por particionar o universo de discurso por meio de funções de pertinência fuzzy. Com isso, problemas que não possuem uma formulação objetiva e numérica podem ser tratados pela lógica fuzzy por variáveis linguísticas.

Em Tanscheit (2004), uma variável para ser considerada como linguística deve ter as seguintes propriedades bem definidas: nome da variável, universo de discurso, conjunto com  $N$  nomes o qual particiona o universo de discurso, regra sintática a qual gera os nomes baseados no particionamento do universo de discurso e uma regra semântica para associar os nomes dados a cada parte do universo de discurso a uma ou mais funções de pertinência.

Tem-se em Cichini (2009), um exemplo em que uma variável linguística com o nome “Distância” pode assumir os seguintes termos **Pequena**, **Média**, **Grande**. Adaptando melhor o exemplo para uma situação real, imagine que uma cidade A tenha trajetos para várias outras cidades, e que a cidade com a maior distância tenha 3000 Km. Assim, o nosso universo de discurso é um universo entre a distância 0 km até 3000 Km. Logo, podemos particionar esse universo de discurso usando regras sintáticas e semânticas em que podemos escolher, por exemplo, duas funções de pertinência trapezoidais para o início e o fim do universo de discurso, e os valores medianos do universo de discurso podem ser representados por uma função triangular. Além disso, a interseção entre as funções é algo em torno de 50%, e os termos que cada função de pertinência irá representar serão os termos que a variável linguística possui na mesma ordem mencionada anteriormente. Com isso, caracterizamos a distância entre a cidade A e outras cidades na forma de uma variável linguística que pode ser utilizada como entrada para um sistema fuzzy a fim de solucionar algum problema cuja imprecisão seja inerente. Por fim, veja a representação gráfica da variável linguística “Distância” particionando o universo de discurso na Fig. 5.

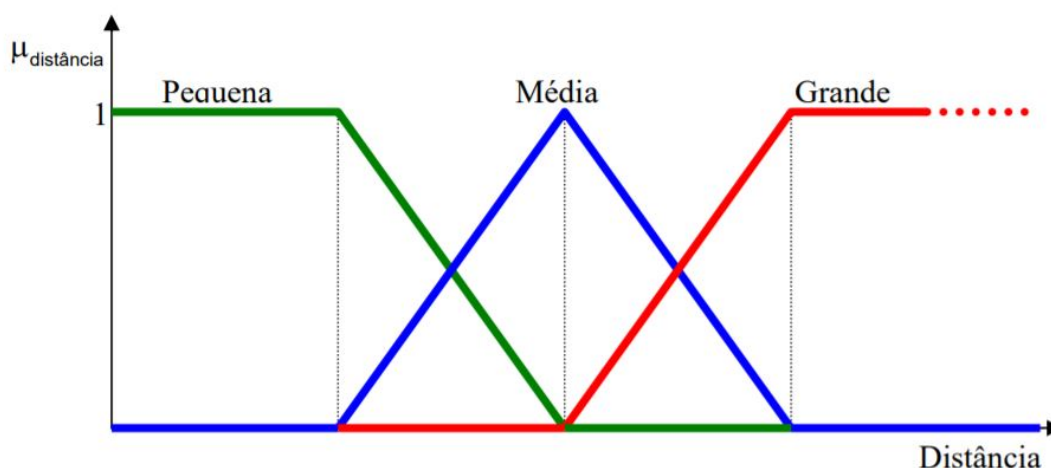


Figura 5 – Exemplo da variável linguística “Distância”.

Fonte: (Cichini, 2009)

### 2.1.4 Sistemas de Inferência Fuzzy

Um sistema de inferência fuzzy pode ser descrito nas seguintes etapas, veja a Fig. 6 como exemplo das etapas descritas a seguir:

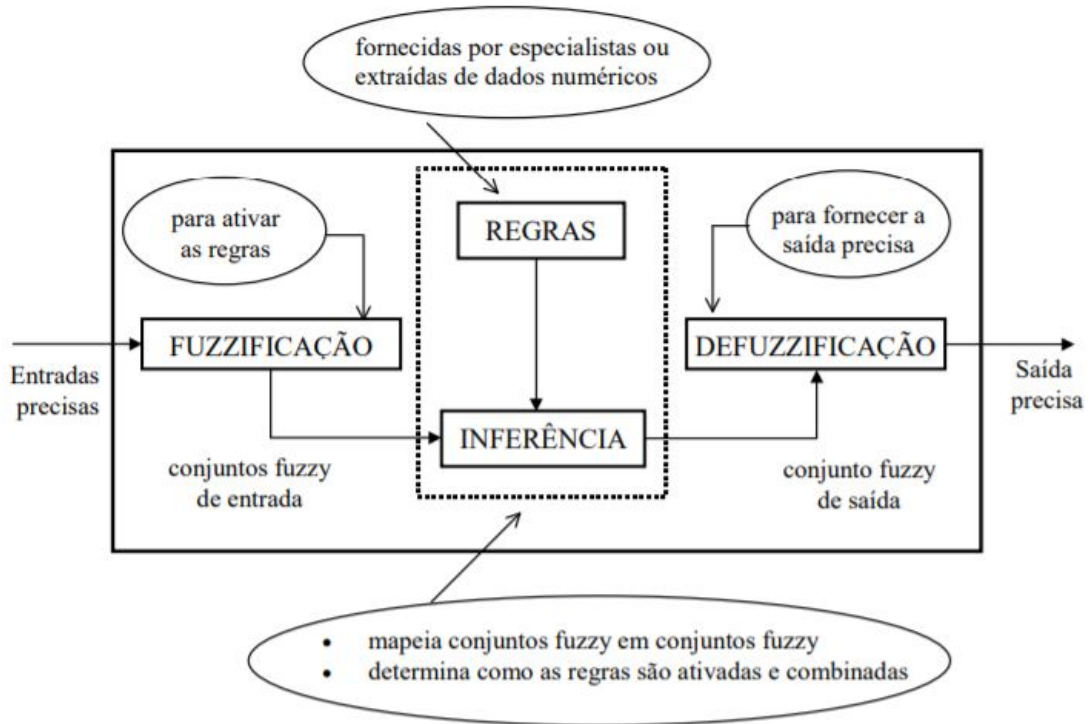


Figura 6 – Arquitetura de um sistema de inferência fuzzy.

Fonte: (Tanscheit, 2004)

1. Fuzzificação: É um sistema que recebe valores numéricos, sendo esses contínuos ou discretos, os quais sofrem um processo de fuzzificação. Esse processo de fuzzificação é o particionamento do universo de discurso das variáveis de entrada e saída, o qual resulta na criação de variáveis linguísticas dado um conjunto de funções de pertinência escolhidas como forma de mapear os valores precisos em valores fuzzy (Tanscheit, 2004; Jang et al., 1997).
2. Regras e Inferência: As regras de um sistema fuzzy são constituídas utilizando-se os valores fuzzy obtidos do processo de fuzzificação. Essas possuem tanto em seu antecedente como em seu conseqüente valores linguísticos e podem ser criadas por meio da experiência de um especialista no assunto do problema tratado ou por meio do treinamento de um algoritmo de inteligência computacional capaz de gerar regras (Tanscheit, 2004; Jang et al., 1997). Ademais, é válido salientar que a estrutura genérica de uma regra é dado em (2.8):

$$\text{Regra : Se } (\text{ANTECEDENTE}) \text{ logo } (\text{CONSEQUENTE}) \quad (2.8)$$

Assim, tendo em vista que a base de conhecimento de um sistema fuzzy se dá por meio de um banco regras, os mecanismos de inferência são concebidos a partir da interação da base de dados fuzzy com as regras fuzzy criadas. Podemos citar na literatura diversos métodos de inferência, dentre os mais comuns, Mamdani e o Takagi-Sugeno. O método Mamdani (Mamdani, 1974) utiliza relações condicionais no antecedente da regra para inferir, na parte consequente da regra, um valor fuzzy. Já o método Takagi-Sugeno (Takagi and Sugeno, 1985) tem como consequente da regra uma função polinomial.

Assim, como exemplo, veja o formato das regras de cada um dos métodos inferenciais:

$$R_{Mamdani} : X_1 \text{ é } A_1 \text{ e } X_2 \text{ é } A_2 \dots X_n \text{ é } A_n \text{ logo : } Y \text{ é } B_n \quad (2.9)$$

$$R_{TakagiSugeno} : X_1 \text{ é } A_1 \text{ e } X_2 \text{ é } A_2 \dots X_n \text{ é } A_n \text{ logo : } Y = f(X_1, X_2 \dots X_n) \quad (2.10)$$

- Defuzzificação: Ao final do processo da criação de um sistema fuzzy, tem-se que as conclusões geradas pela base de conhecimento são valores fuzzy e necessitam passar por um processo de defuzzificação para que tenhamos um resultado numérico, o qual possa ser usado por algum outro sistema do mundo real. Dessa maneira, esse processo abarca a criação de uma função ou método matemático para que as instâncias das variáveis linguísticas se transformem em valores reais. Vale ressaltar que o processo de defuzzificação não se trata de encontrar uma função inversa para a transformação da variável de saída ou que exista uma única função que realize esse processo, mas tão somente projetar uma função que transforme um valor fuzzy a fim de adequar a saída do sistema fuzzy ao problema proposto. Quanto aos tipos de métodos utilizados para a defuzzificação tem-se diversos, dentre eles os mais utilizados na literatura são: o “Centroide” (ou Centro de Massa), a “Média dos Máximos” e o “Critério do Máximo”. O centro de massa é obtido calculando-se a média aritmética dos centroides dos conjuntos fuzzy, ponderados pelos valores de pertinência, como mostrado na Eq. 2.11, no caso,  $\mu_n$  é o grau de pertinência do elemento que pertence ao  $n$ -ésimo conjunto fuzzy,  $C_n$  é o valor do centroide do  $n$ -ésimo conjunto fuzzy e  $k$  é numero de conjuntos fuzzy a qual o elemento pertence. Perceba que a saída do “Centroide” será uma saída discreta e que para o cálculo do centro de massa em problemas contínuos basta a utilização da função integral no lugar do somatório (Gomide et al., 1995; Watanabe, 2016; Jang et al., 1997).

$$Centroide = \frac{\sum_{n=1}^k \mu_n \cdot C_n}{\sum_{n=1}^k \mu_n} \quad (2.11)$$

Em [Watanabe \(2016\)](#), a “Média dos Máximos” é considerada uma variação do método centroide e esse garante o critério de seleção nuclear apenas para conjuntos convexos. A “Média dos Máximos” é calculada a partir da média aritmética de todos os valores centrais pertencentes aos conjuntos fuzzy selecionados no processo de inferência e caso exista mais de um valor máximo no conjunto fuzzy, recomenda-se utilizar o valor médio. Já o “Critério do Máximo” calcula por meio da saída fuzzy um valor discreto que representa o maior valor de pertinência do conjunto fuzzy.

## 2.2 Séries Temporais - Time Series (TS)

Para [Morettin and Tolo \(2004\)](#), uma série temporal é um conjunto de dados coletados em sequência no tempo. Essa coleta, geralmente, é feita a uma dada taxa de amostragem fazendo com que os pontos aferidos sejam equidistantes no tempo. Pode-se fazer uso de métodos paramétricos ou não paramétricos para construção de seus modelos. As séries paramétricas são definidas como tendo um número de parâmetros finitos e utilizam modelos no domínio do tempo. Já as séries não paramétricas, utilizam o domínio da frequência para constituição dos modelos, como é o caso da análise espectral.

É interessante salientar que uma série temporal, segundo [Morettin and Tolo \(2004\)](#), corresponde a uma observação dentro de várias outras que poderiam ocorrer a respeito de um mesmo fenômeno. Ademais, como visto anteriormente na introdução, os modelos de séries temporais possuem determinados objetivos como investigar o mecanismo gerador da série temporal, fazer previsões de valores futuros e procurar por padrões no comportamento das séries.

Dentre os objetivos citados, aquele mais incidente é o de previsão dos valores futuros de uma série. Para [Montgomery et al. \(2015\)](#), são três os tipos de previsão considerando janelas temporais, as de curto prazo, médio prazo e longo prazo. As janelas de curto prazo dizem respeito a dias, semanas e meses, já períodos médios são relativos a um ou dois anos, e janelas de longo prazo fazem referência a muitos anos.

Segundo [Montgomery et al. \(2015\)](#), para uma previsão quantitativa, é necessário utilizar dados históricos internos ou externos da série a fim de se captar padrões e comportamentos, uma vez que dada série pode possuir forte autocorrelação, ou correlação com séries externas. Assim, por meio de dados passados, torna-se possível capturar padrões e comportamentos por meio de modelos estocásticos para extrapolar situações futuras.

Dentre conceitos básicos relativos ao comportamento das séries temporais temos os principais:

1. Estacionariedade: Segundo [Barros \(2004\)](#), uma série temporal é estacionária quando suas propriedades como média, variância e autocorrelação são contantes ao longo

do tempo. Dentro dessa ideia básica temos a definição de estacionariedade estrita e a definição de estacionariedade de segunda ordem (fracamente estacionárias). Para [Montgomery et al. \(2015\)](#), uma série possui estacionariedade estrita se ao mudar o ponto de observação dessa no tempo as suas propriedades não apresentaram mudanças. No caso, as distribuições conjuntas de probabilidade no tempo não variam, como expresso na (2.12). Já uma série fracamente estacionária, refere-se a qualidade da série possuir uma média e variância constante (ex.:  $E(Y_t) = \mu$  e  $Var(Y_t) = \sigma^2$ ), independente do tempo, além de possuir uma covariância que depende apenas da distância temporal dos valores da série (ex.:  $Cov(Y_t, Y_{t+k}) = Cov(Y_t, Y_{t-k})$ ).

$$P(Y_{t+1}, Y_{t+2} \dots Y_{t+n}) = P(Y_{t+k+1}, Y_{t+k+2} \dots Y_{t+k+n}) \quad (2.12)$$

2. Tendência: Em [Barros \(2004\)](#), a tendência numa série temporal é expressa por um movimento de “longo prazo” em que a série apresenta um crescimento ou decréscimo em seus valores, causando uma modificação no valor de sua média ao longo do tempo. Alguns testes específicos para identificar a presença de tendências são os testes de hipótese de *Wald-Wolfowitz*, de *Cox-Stuart* e *Mann-Kendall* ([Morettin and Tolo, 2004](#); [Maia, 2017](#); [Montgomery and Runger, 2003](#)).
3. Sazonalidade: Segundo [Maia \(2017\)](#), é uma flutuação periódica que ocorre na série temporal, o que indica a ocorrência de um padrão de forma cíclica.
4. Homocedasticidade e Heterocedasticidade: Em [Maia \(2017\)](#), quando o erro de uma variável explanatória possui uma variância constante ao longo da série ou regressão diz-se que possui propriedade homocedástica. Já para variáveis heterocedásticas, a variância do erro muda ao longo da série. Tanto na Fig. 7 quanto na Fig. 8 podemos ver um exemplo quanto a definição desses conceitos.

O processo para se conceber modelos voltados a previsão de séries temporais é tratado em [Montgomery et al. \(2015\)](#), em que os passos descritos são os seguintes:

1. Definição do problema: Envolve a forma como a previsão irá contribuir para a solução do problema. Deve-se questionar nessa fase que tipo de métricas deverão ser usadas durante as previsões, o que se deseja extrair dos resultados das previsões e como o modelo irá agregar valor aos seus destinatários. É preciso ir fundo na lógica do problema tratado a fim de se avaliar, também, os riscos que envolvem a tomada de decisão baseada na previsão dos valores realizada ([Montgomery et al., 2015](#)).
2. Coleta dos Dados: Consiste em obter dados históricos relevantes e com certo grau de qualidade das variáveis que serão tratadas no problema de previsão. É necessário planejar como as coletas futuras vão ser feitas e guardadas, uma vez que durante a



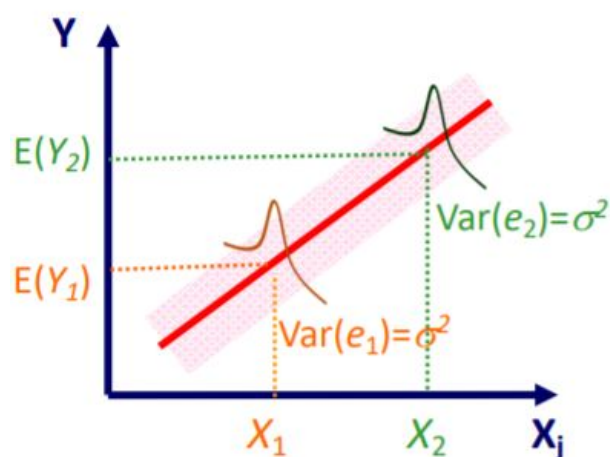


Figura 7 – Exemplo de homocedasticidade.

Fonte: (Maia, 2017)

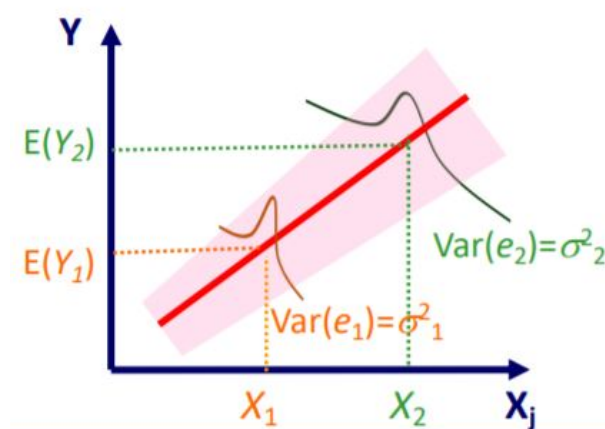


Figura 8 – Exemplo de heterocedasticidade.

Fonte: (Maia, 2017)

realização da coleta para a aplicação do modelo de previsão haverá dados faltantes, outliers, entre outros problemas (Montgomery et al., 2015).

3. Análise dos Dados: É uma fase importante, a qual será utilizada para definir possíveis modelos de previsão por meio da análise gráfica das séries, bem como de dados estatísticos. Durante essa fase é importante perceber padrões das séries como a presença de sazonalidade, tendência, níveis da série, entre outros (Montgomery et al., 2015).
4. Seleção do Modelo e Treino do Modelo: Escolhe-se um ou mais modelos de previsão e os treina com base nos dados coletados a fim de determinar os seus hiperparâmetros.
5. Validação do Modelo: Consiste em avaliar os modelos treinados com novos dados. Divide-se os conjuntos de dados para o treinamento e para o teste do algoritmo. O

teste do algoritmo num conjunto de dados novos permite a análise do desempenho desse com diferentes métricas utilizadas para análise de regressões (MAE, MAPE, RMSE, SMAPE, etc). Além disso, permite saber se o algoritmo treinado possui bom ou ruim grau de generalização e se é um modelo que atende a complexidade do problema (ex.: há ocorrência de *overfitting* ou *underfitting*) (Montgomery et al., 2015).

6. Implantação e monitoramento da performance do modelo: Trata-se da implementação do modelo para previsão em tempo real do problema tratado. Deve-se verificar se as métricas para a avaliação do modelo em tempo real permanecem com resultados adequados aos objetivos propostos (Montgomery et al., 2015).

Os passos descritos anteriormente para a concepção de modelos preditivos orientará a forma como os variados conjuntos de dados serão analisados e a forma como as séries desses serão modeladas.

### 2.2.1 Séries Temporais Univariadas e Multivariadas

Tratar séries como univariadas ou multivariadas constitui uma forma de abordagem do problema de análise e previsão de séries temporais. Ao visualizar um problema como uma série univariada, considera-se que estamos trabalhando com uma única variável no tempo (única série histórica) e há apenas a possibilidade de dependência entre os próprios valores da série a fim de possibilitar a previsão da mesma, como sugere os métodos AR, MA, ARIMA, Holt-Winters, etc . Os vários modelos de séries univariadas usam propriedades das séries como pressupostos para a aplicação desses modelos, como é o caso da: autocorrelação, estacionariedade, heterocedasticidade, sazonalidade, etc) para sua análise e previsão (Morettin and Toloi, 2004; Montgomery et al., 2015; Maia, 2017).

Já os modelos multivariados, não só consideram a autocorrelação da série principal, mas também valores de séries externas (séries independentes da série principal) que impliquem em ganhos na previsão e análise dessa, uma vez que essas séries podem fornecer evidências de causalidade ou correlação lineares ou não lineares a fim de esclarecer como os valores da série principal são gerados no tempo. Podemos citar como exemplo alguns modelos de previsão de séries multivariadas VAR (Wang, 2018), VARMA (Tsay, 2013), modelo GARCH multivariado (Stoyanov, 2010), entre outros.

### 2.2.2 Modelos de previsão em Séries Temporais

Apresenta-se alguns dos modelos de previsão de séries temporais tradicionais existentes. Discorreremos brevemente sobre modelos univariados, multivariados e sobre os algoritmos de aprendizado de máquina.

A ideia proposta por Box-Jenkins (Pankratz, 1983; Morettin and Toloï, 2004) sugere que apenas valores defasados da série mais algum coeficiente que expresse um erro (modelos univariados) seja necessário para cunhar expressões matemáticas capazes de descrever a série e seus valores futuros. Em relação aos modelos univariados os modelos AR, MA e ARIMA são modelos paramétricos tradicionais e funcionam conforme suas premissas são atendidas. Essas podem ser verificadas por meio de testes de hipótese estatísticos, os quais proveem evidência da adequação do conjunto de dados ao modelo que pretende ser usado. Primeiramente, discorre-se sobre o modelo AR (auto-regressivo), em que o valor predito em  $t + 1$  pode ser representado como uma combinação de valores lineares passados somado a uma variável que representa um ruído. Já a MA (Média Móvel) utiliza a composição de  $N$  valores de ruídos passados independentes e aleatórios para prever um valor. Por fim, o modelo ARIMA (modelo auto regressivo integrado de médias móveis) é usado em séries não estacionárias e possui as propriedades dos modelos AR (utiliza auto-regressão), MA (utiliza combinação linear dos erros passados) e I (parte integrada) que representa a diferenciação nos dados. Em adição, é interessante mostrar que esses modelos consideram que exista pressupostos nos dados a serem verificados, como é o caso da: dependência linear (autocorrelação linear), estacionariedade e ergodicidade (RESENDE, 2001; Morettin and Toloï, 2004).

Além disso, existem variações do modelo ARIMA como o SARFIMA e o ARFIMA utilizados, respectivamente, para dados sazonais e de longa dependência sazonal (Morettin and Toloï, 2004).

Vale citar que esses modelos tradicionais possuem também versões multivariadas, como é o caso do VAR (Vetor AR), VARMA (Vetor ARMA) e VARIMA (Vetor ARIMA) (De Gooijer and Hyndman, 2006; Tsay, 2013).

Além disso, vários são os modelos de aprendizado de máquina, que tradicionalmente, são usados em problemas de séries, como é o caso das árvores de decisão (Breiman et al., 1984; Loh, 2011; Geron, 2017), *Random Forest* para regressão (Breiman, 2001; Geron, 2017), *Support Vector Regression* (SVR) (Geron, 2017; Chang and Lin, 2011; Evgeniou and Pontil, 2001), variados tipos de redes neurais (Geron, 2017), entre outros.

As árvores de decisão, como a CART (Breiman et al., 1984), realizam regressão ao repartir o universo de discurso dos dados de uma série por meio de regras (caminho entre o nó pai e nó filho), em que o conseqüente dessas preveem valores, ou são capazes de representar uma função que gera os valores de uma dada regressão. Já os modelos de *Random Forest Regressor* são uma combinação desses estilos de árvore num método *ensemble*, o que significa que o valor final da *Random Forest* é ponderado por uma população de árvores de regressão.

Ademais, o SVR (Support Vector Regression) é um algoritmo baseado na SVM (Geron, 2017; Chang and Lin, 2011; Evgeniou and Pontil, 2001), só que voltado para

problemas de regressão.

Dentre os algoritmos de redes neurais, tem-se desde uma rede neural MLP usada para modelos de regressão até redes neurais recorrentes, como é o caso das LSTM (Hochreiter and Schmidhuber, 1997), ESN (Jaeger, 2001) e ELMAN (Elman, 1990; Braga et al., 2000). As rede neurais mais simples como a MLP, são algoritmos *feed-forward* que possuem uma estrutura em que a primeira camada de neurônios está ligada as variáveis de entrada, já as camadas mais internas (camadas ocultas) realizam operações sobre os dados de entrada a depender dos pesos de cada neurônio dessas e proveem esses dados a uma camada de saída. Já nas redes recorrentes, o fluxo de dados não ocorre de forma direta, pois existem nós interconectados de forma cíclica dentro das camadas ocultas das rede neurais que permitem que essas tenham informação quanto a memória da série temporal (Geron, 2017; Braga et al., 2000).

Em suma, é válido notar que se pode utilizar diferentes abordagens para os problemas de análise e previsão de séries temporais, como é o caso de se escolher entre os modelos de séries temporais paramétricos ou algoritmos de aprendizado de máquina, cada qual possuindo seus pontos fortes e fracos. Quando se tem conjunto de dados que obedecem as premissas de estacionariedade e autocorrelação linear é recomendado o uso de técnicas como o modelo AR, ARIMA, etc. Não obstante, quando nenhuma dessas premissas é verificada nos dados, ou quando não permanecem no tempo e tem-se grande influência de variáveis exógenas, é de boa acolhida os modelos de aprendizado de máquina.

### 2.2.3 Métricas de Acurácia

Diversas são as métricas utilizadas para verificar a acurácia dos modelos de regressão e previsão de séries. Para Makridakis and Hibon (1995), as variadas funções que avaliam os ajustes dos modelos têm vantagens e desvantagens, e frequentemente podem possuir algum tipo de *trade-off*, significando que ao otimizar o valor de uma das métricas, uma outra poderá ser prejudicada. Abordaremos algumas funções mais utilizadas na literatura dentre as dezenas existentes. Essas serão RMSE, NRMSE, MAPE, SMAPE e MAE. Utilizaremos a notação presente em Steurer et al. (2021) para apresentar as equações e explicar as propriedades dessas métricas. No caso, os valores  $p_n$  são os valores reais e os valores  $\hat{p}_n$  são os preditos na equações a seguir.

Primeiramente, o erro absoluto médio (MAE) é uma métrica simples e refere-se ao cálculo da média quanto a magnitude dos resíduos, como demonstrado na Eq. 2.13. Perceba que o MAE é uma métrica que trata os resíduos em seu valor absoluto e que quanto menor o valor do MAE, melhor os valores preditos se aproximaram dos valores reais, indicando de forma direta a qualidade da regressão feita. Como o MAE utiliza o módulo da diferença entre os valores reais e preditos não é possível saber se o modelo está acima ou abaixo dos valores reais, uma vez que não se sabe se há maior influência dos

valores negativos ou positivos. Por não possuir natureza quadrática, o MAE é robusto a outliers.

$$\frac{1}{N} \sum_{n=1}^N |p_n - \hat{p}_n| \quad (2.13)$$

Já o MAPE (Porcentagem Média do Erro Absoluto) é uma métrica semelhante ao MAE no que tange o uso do erro absoluto em sua equação, veja a fórmula do MAPE na Eq. 2.14. No caso, o MAPE possui a vantagem de expressar a medida dos erros como uma porcentagem dos dados reais. O erro dado em porcentagem contribui para um entendimento mais simples e intuitivo da distância entre o modelo concebido e os dados reais, uma vez que dados em porcentagem são comumente utilizados no dia-a-dia. Além disso, um mesmo valor absoluto do erro pode indicar 0.1% ou 10% a depender da escala e da série tratada. Assim, o MAPE possui a vantagem de ser uma métrica que é independente de escala e inclusive pode ser utilizada para comparar modelos de previsões em escalas diferentes. Vale salientar que uma desvantagem e limitação do MAPE é que não há simetria no erro. Explicitando melhor, em [Makridakis and Hibon \(1995\)](#), é citado o exemplo que para um valor real  $p_n = 100$ , e tendo em vista o valor predito  $\hat{p}_n = 150$ , o valor do MAPE será de 50%. Já para o caso em que  $p_n = 150$  e  $\hat{p}_n = 100$ , o valor do MAPE será de 33,3%. Em adição, o MAPE não lida bem com problemas de séries que possuem uma amplitude de valores muito alta (ex.: valores da série vão de 30 até 120.000).

$$\frac{1}{N} \sum_{n=1}^N \left| \left( \frac{p}{\hat{p}_n} \right) - 1 \right| \quad (2.14)$$

A fim de lidar com esse problema da assimetria no MAPE surge o SMAPE (MAPE simétrico), como mostrado na Eq. 2.15. Assim, como o MAPE, o SMAPE também gera resultados relativos.

$$\frac{1}{N} \sum_{n=1}^N \left( \frac{|p_n - \hat{p}_n|}{p_n + \hat{p}_n} \right) \quad (2.15)$$

Já o RMSE é definido como a raiz quadrada do erro médio e trata de uma métrica dependente da escala e, também, é baseado no erro absoluto. O RMSE é uma métrica que pode ser utilizada para comparar métodos diferentes dado um mesmo conjunto de dados de teste ([Steurer et al., 2021](#); [Makridakis and Hibon, 1995](#)). Além disso, o RMSE apresenta uma vantagem quanto ao MSE ([Steurer et al., 2021](#); [Makridakis and Hibon, 1995](#)), que é o fato do seu resultado possuir a mesma unidade da série em estudo.

$$\sqrt{\frac{\sum_{n=1}^N (p_n - \hat{p}_n)^2}{N}} \quad (2.16)$$

Por fim, o NRMSE é a versão normalizada do RMSE, o qual possui a vantagem de se estabelecer uma comparação geral dos modelos para diferentes escalas (Otto et al., 2018).

$$\frac{\sqrt{(1/N) \sum_{n=1}^N (p_n - \hat{p}_n)^2}}{p_{\max} - p_{\min}} \quad (2.17)$$

Em suma, apresentamos nessa subseção algumas métricas para a avaliação de problemas de séries temporais e regressão, a fim de mostrar que as diferentes métricas podem ser utilizadas com diferentes finalidades e cada qual possui suas vantagens e desvantagens. Ademais, essas podem ser combinadas para se extrair diversas análises de um mesmo problema, enriquecendo a comparação entre a performance dos algoritmos aplicados nos problemas de previsão (Steurer et al., 2021; Makridakis and Hibon, 1995).

## 2.2.4 Aplicações Provenientes dos Modelos de Séries Temporais

Muitas são as aplicações extraídas da análise e previsão de séries temporais, as quais são aplicadas em diversas áreas do conhecimento, como mercado financeiro, meio ambiente, saúde, entre outras. Nossa abordagem, tentará brevemente mostrar a importância e os ganhos do desenvolvimento e a aplicação desses métodos na atualidade. A seguir abordaremos, como exemplo, três aplicações relevantes dos modelos de análise e previsão de séries temporais.

Em Crabtree et al. (1990), temos uma compilação das várias aplicações relativas aos modelos de séries na área da saúde. Como exemplo, foi modelado em formato de série multivariada a relação entre o efeito de realizar uma caminhada e a queda da glicose no sangue períodos de tempo depois, estando o indivíduo em jejum. Por meio desse, conseguiu-se chegar numa equação que relaciona a queda da glicose sanguínea ao número de milhas percorridas, basicamente, para o paciente testado a cada 1 milha percorrida a glicose sanguínea diminuía em  $3.93385mg/dl$ . Em Zhang et al. (2014), mostra-se a importância das séries temporais para a previsão no avanço de doenças e como estes modelos podem auxiliar os órgãos públicos no monitoramento do contágio. Além disso, nesse trabalho foram utilizados quatro métodos: ARIMA, SVM (Support Vector Machine), Regressão e Suavização Exponencial. Em adição, esses foram modelados a partir das séries epidêmicas na China de várias doenças dentre os anos 2005 e 2011 como febre tifoide, Hepatite A e B, Febre Hemorrágica, entre outras. O ano de 2012 foi utilizado como teste para os modelos. Por fim, as métricas de avaliação do ajuste dos modelos utilizados foram o: MAPE (mean absolute percentage error), MAE (mean square error) e RMSE (root mean square error). Veja, como exemplo, nas Fig. 9 e Fig. 10 o comportamento dos modelos de predição para a série da febre tifoide em relação a série original, e os resultados do MAPE para o SVM.

Uma outra área em que os modelos de séries temporais são demandadas é a previsão de consumo energético nas redes de *smart grid*. Em Ryu et al. (2017), como

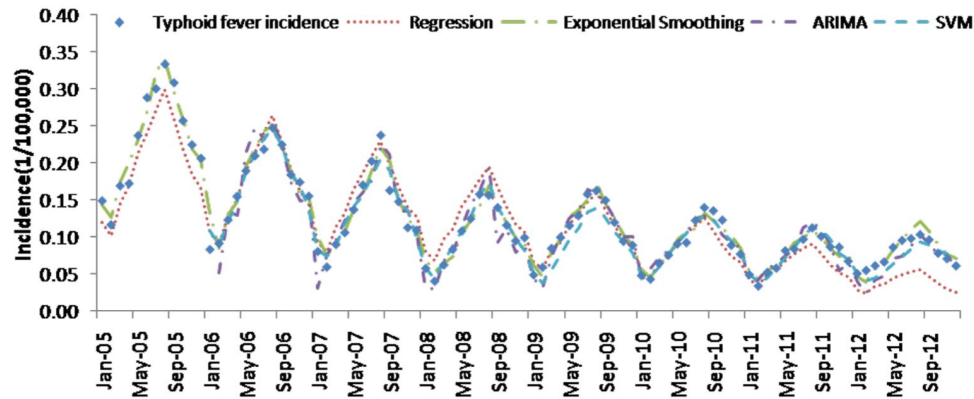


Figura 9 – Série Febre Tifoide - Série Original vs Séries Prevista.

Fonte: (Zhang et al., 2014)

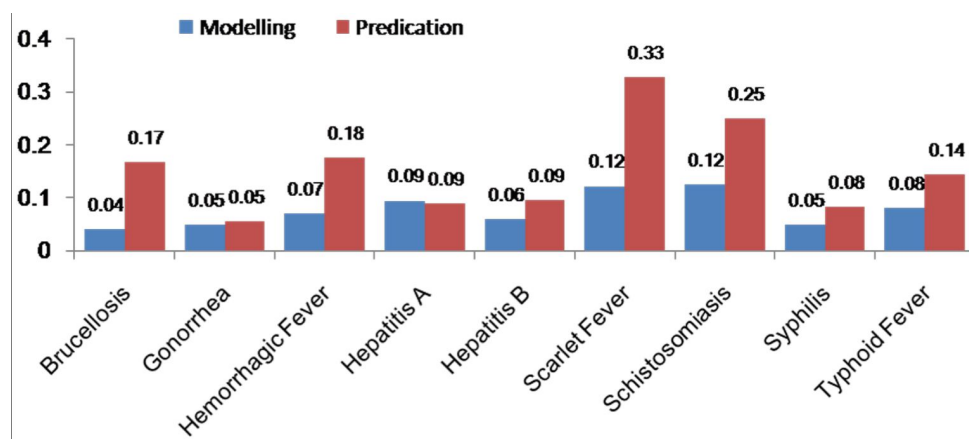


Figura 10 – Resultado do MAPE em relação ao SVM para as doenças listadas.

Fonte: (Zhang et al., 2014)

exemplo, realiza-se experimentos com redes neurais profundas em que se obtém bons resultados em comparação a outros métodos quando computadas as métricas de MAPE (mean absolute percentage error) e RRMSE (relative root mean square error). As duas redes neurais profundas (DNN) utilizadas são: uma máquina de Boltzmann restrita com pré-treinamento e uma outra máquina de Boltzmann com função linear retificada (ReLU) sem pré-treinamento. Já os parâmetros de entrada dos modelos são as séries de consumo individuais e dados meteorológicos regionais. Os modelos comparados são: ARIMA, Holt-Winter sazonal duplo e uma rede neural rasa (SNN). Veja na Fig. 11 um exemplo da previsão de carga agregada de 500 diferentes grandes instalações consumidoras de energia. Tanto a DNN com pré-treinamento como a DNN que utilizou a ReLU sem pré-treinamento obtiveram bons resultados com relação ao MAPE, respectivamente, 2,27% e 2,19%. Ademais, para o RRMSE obteve-se, respectivamente, 2,91% e 2,76%.

Por fim, uma terceira aplicação que popularmente usa modelos de séries temporais diz respeito a problemas financeiros e econômicos. Em Sen and Mehtab (2020), é feita



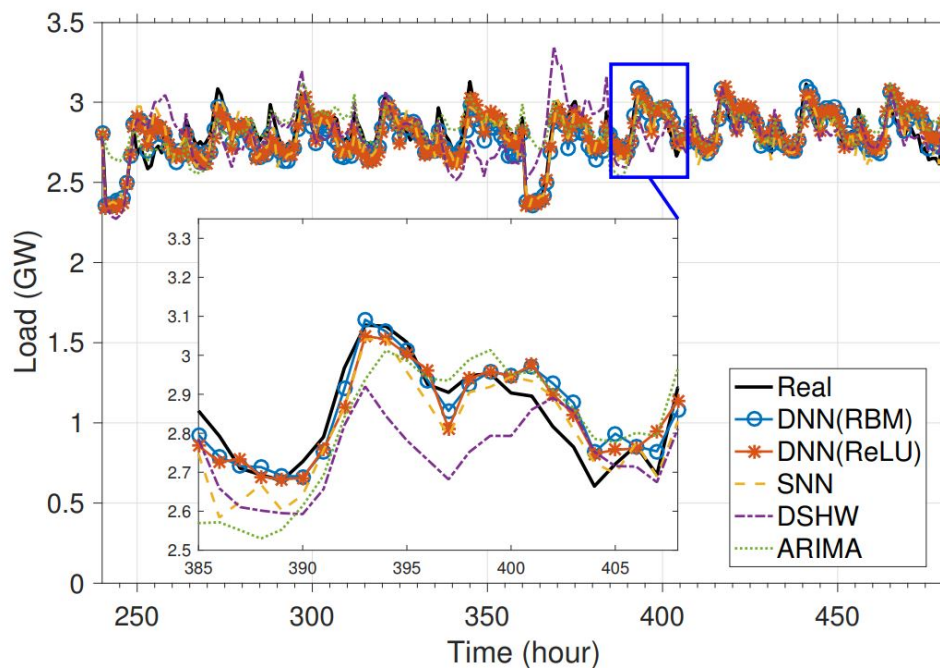


Figura 11 – Série Original vs Séries Previstas para a Carga Agregada.

Fonte: (Ryu et al., 2017)

uma aglomeração dos diversos modelos de aprendizado de máquina para previsão de séries temporais do preço de uma ação (*Godrej Consumer Products Ltd*) listada na *National Stock Exchange* (NSE) da Índia. Foram construídos oito modelos de classificação e oito modelos de regressão baseados em algoritmos tradicionais em aprendizado máquina, além de dois modelos de aprendizado profundo (uma LSTM e uma CNN). Os resultados de todos os modelos foram analisados e demonstraram que as redes de aprendizado profundo performaram melhor que os demais modelos, mesmo que utilizados em conjunto (*bagging* e *boosting*). Os bons resultados dos modelos demonstram a possibilidade de que os algoritmos podem prever e performar bem na tomada de decisão quanto a compra e venda de ações, contrariando algumas ideias da hipótese de mercado eficiente.

## 2.3 Séries Temporais Fuzzy - Fuzzy Time Series (FTS)

Em Song and Chissom (1993a), as séries temporais fuzzy (*fuzzy time series*) são tratadas como um método baseado na lógica fuzzy para analisar e prever eventos dinâmicos que podem ser ou não variantes no tempo. As *fuzzy time series* (FTS) são descritas em termos de variáveis linguísticas ao invés de variáveis numéricas e os diversos métodos que abordam FTS são considerados não paramétricos. Uma FTS pode ser ou não variante no tempo, o que diz respeito as propriedades da série, que podem se manter ou não constantes ao longo do tempo (estacionariedade, heterocedasticidade, etc). De forma geral, segundo Song and Chissom (1993a), as FTS não variantes no tempo são mais simples de serem



tratadas do que as FTS variantes no tempo, dado que não é possível considerar que a série será definida por um único conjunto de funções ou relações, nem que essas permanecerão constantes no tempo.

Uma FTS pode ser gerada naturalmente pela própria caracterização do problema, como exemplo, podemos descrever o clima de uma região do planeta durante o ano como **muito frio, pouco frio, frio, quente, pouco quente e muito quente**. Assim, é possível gerar uma sequência de valores fuzzy no tempo, tendo como granularidade da série os dias, exemplificando: **[Dia1 - Frio, Dia2 - pouco frio, Dia3 - muito frio, Dia4 - frio...DiaN - Valor Fuzzy]**. Uma outra maneira de se obter uma série temporal fuzzy, segundo [Song and Chissom \(1993a\)](#), é por meio da fuzzificação de uma série temporal, em que os valores numéricos do universo de discurso da série são transformados numa variável linguística, e a série, dessa forma, será descrita na forma de valores fuzzy no tempo. Perceba que diferente de uma série temporal, o próximo valor a ser predito é um valor fuzzy. Logo, a base de dados para a aplicação dos métodos de FTS podem compor-se tanto de dados numéricos, posteriormente fuzzificados, ou de dados categóricos que ocorrem no tempo. A partir de valores fuzzy previstos nas FTS é possível utilizar métodos inferenciais e de defuzzificação para tornar ou retornar os valores nebulosos em numéricos, e com isso prever valores futuros de variados tipos de séries (temporais, regressão, espectrais).

Por último, veja abaixo um exemplo da aplicação de uma FTS nas Fig. 12 e Fig. 13, baseado em [Song and Chissom \(1993b\)](#) e presente no tutorial “*A short tutorial on Fuzzy Time Series*”<sup>1</sup>. A série numérica na Fig. 12 retrata o número de alunos que se matricularam na Universidade do Alabama durante os anos de 1971 até 1992. Essa série é convertida em uma FTS, a qual possui a propriedade de prever valores futuros em formato de regras fuzzy, como explícito na Fig. 13. Perceba que os próximos valores da FTS podem ser previstos com base nos valores dos conjuntos fuzzy passados, por exemplo: **(Regra1: A1 → A2), (Regra2: A2 → A2, A3), (Regra3: A3 → A4), etc.** Perceba que  $A_1, A_2, A_3, \dots, A_n$  refere-se aos conjuntos fuzzy em que a série original foi discretizada e que a partir dele tem-se regras para a previsão futura de conjuntos. Vale lembrar que boa parte dos métodos consegue prever apenas valores dentro do universo de discurso da série fuzzy, para isso a série deve possuir característica estacionária. No entanto, pode-se citar métodos de FTS para séries não estacionárias como é o caso do trabalho desenvolvido em [De Lima e Silva et al. \(2020\)](#).

## 2.4 Explicabilidade e Interpretabilidade

Os algoritmos de inteligência computacional vêm se responsabilizando por tomadas de decisões e resolução de problemas cada vez mais críticos em diversos ramos da sociedade.

<sup>1</sup> <https://towardsdatascience.com/a-short-tutorial-on-fuzzy-time-series-dcc6d4eb1b15>, Acessado em: 02/11/2021.

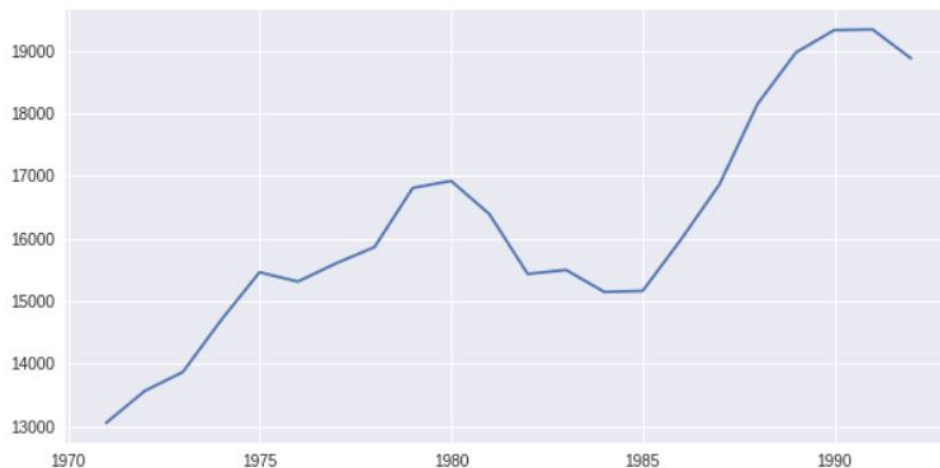


Figura 12 – Série numérica dos alunos matriculados na Universidade do Alabama.

Fonte: (Song and Chissom, 1993b) e “A short tutorial on Fuzzy Time Series”<sup>2</sup>

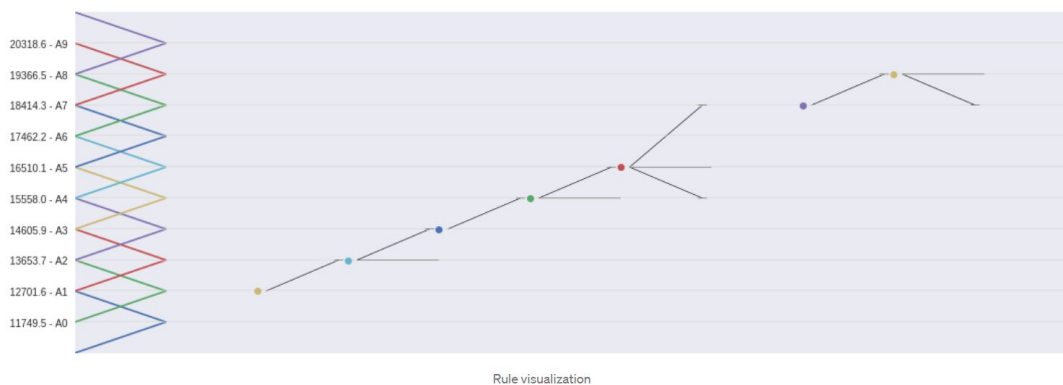


Figura 13 – FTS dos alunos matriculados na Universidade do Alabama.

Fonte: (Song and Chissom, 1993b) e “A short tutorial on Fuzzy Time Series”<sup>3</sup>

Seja na tomada de decisão de compra ou venda de ativos em mercados financeiros Sezer et al. (2020), seja na concepção de carros com direção autônoma baseados em algoritmos de visão computacional (Janai et al., 2020), ou até mesmo na tarefa de recomendação de crédito a determinados clientes em instituições financeiras (Turkson et al., 2016). Nas variadas aplicações de aprendizado de máquina encontramos certos problemas de complexa resolução como: a identificação de determinado viés indesejado durante o treino e teste de algoritmos, a mudança brusca de conceito (*concept drift*) (Pechenizkiy and Zliobaite, 2010) de forma a invalidar algoritmos já concebidos, a identificação de problemas em sistemas compostos de vários subsistemas complexos, sendo um deles composto por um algoritmo de aprendizado de máquina em que há dúvidas se o mal funcionamento do sistema é devida a má qualidade dos algoritmos treinados ou se há erros em outros subsistemas e como o sistema de AI interage quando esses subsistemas apresentam algum erro, além de outras questões (Camarinha-Matos and Martinelli, 1998; Hofkirchner and Schafranek, 2011).

Diante desse cenário, há uma extrema importância em se prover o entendimento de “como” e o “porquê” dos resultados advindos de um algoritmo de aprendizado de máquina. Esse entendimento provê maior segurança na aplicação dos métodos de aprendizado de máquina, além de fornecer ao usuário final e ao engenheiro, o qual concebeu a aplicação, o motivo de uma tomada de decisão. Nesse aspecto tem-se o conceito da explicabilidade e da interpretabilidade:

1. Interpretabilidade: Para [Biran and Cotton \(2017\)](#), interpretabilidade trata-se da capacidade de um modelo ser entendido por humanos (“como” o modelo funciona), seja por meio de uma introspecção profunda do funcionamento desse, ou pela natural produção de interpretabilidade por ele próprio. Ademais, a interpretabilidade pode ser dividida em interpretabilidade intrínseca e extrínseca. A interpretabilidade intrínseca diz respeito a modelos naturalmente interpretáveis e de simples entendimento (modelos caixa branca) como é o caso de árvores de decisão, regressão linear, sistemas baseados em regras, entre outros. Já a interpretabilidade extrínseca diz respeito a modelos de aprendizado de máquina que possuem difícil entendimento (modelos caixa preta). Esses necessitam de um segundo modelo específico que os expliquem. Como exemplo, pode-se citar os algoritmos de redes neurais.
2. Explicabilidade: Para [Biran and Cotton \(2017\)](#), a explicabilidade fundamenta o motivo por de trás de determinado resultado. Diante dos estudos em [Gkatzia et al. \(2016\)](#); [Kim et al. \(2016\)](#); [Ye and Johnson \(1995\)](#); [Biran and Cotton \(2017\)](#), mostrou-se que os usuários de um sistema tendem a aceitar e confiar nas predições de um modelo quando o sistema é explicável. Além disso, a explicabilidade tornou-se uma ferramenta importante ao trazer entendimento quanto a acurácia de um modelo, bem como estabelecer uma relação entre a entrada e a saída desse.

Historicamente, os primeiros algoritmos a terem ampla aceitação, por terem explicabilidade e interpretabilidade, foram os sistemas especialistas, uma vez que são sistemas baseados numa base de regras e fatos ([Giarratano and Riley, 1989](#)). Como exemplo temos em [Clancey and Shortliffe \(1984\)](#), um sistema especialista para apoio em diagnóstico médico, o qual tinha suas conclusões aceitas pelo usuário à medida que fornecia um passo-a-passo do raciocínio realizado em formato de regra. Nesse contexto, os sistemas especialistas baseados em regras foram os primeiros a apresentar um modelo que considerasse a explicabilidade como um dos pontos centrais, para além da preocupação com a acurácia ([Biran and Cotton, 2017](#)). Após os sistemas especialistas baseados em regras, o advento das redes bayesianas ([Charniak, 1991](#)) possibilitou uma nova gama de sistemas especialistas baseado em decisões probabilísticas e em [Lacave and Díez \(2002\)](#) há uma profunda revisão de como obter explicabilidade em relação as redes bayesianas. Perceba

que esses métodos geram uma estrutura auto-explicável e possuem interpretabilidade intrínseca.

Com o tempo, o foco das pesquisas na área de aprendizado de máquina se debruçaram diante das altas taxas de acurácia dos diversos tipos de redes neurais (Geron, 2017). No entanto, a alta acurácia reflete também a alta complexidade desses modelos, o que compromete a interpretabilidade e explicabilidade dos mesmos. Ou seja, tomando como base os vários exemplos da literatura, modelos mais simples tendem a ser explicáveis e com menor acurácia e modelos mais complexos com altas acurácias tendem a ser de difícil compreensão (como exemplo veja a Fig. 14). Na Fig. 14 é válido apontar que conforme o algoritmo cresce em complexidade, há o aumento do seu desempenho preditivo (acurácia), porém, há uma diminuição na sua capacidade de prover explicação intrínseca quanto as suas decisões. É interessante notar, também, na Fig. 14, que as explicações *post-hoc* (técnicas utilizadas para prover explicações realizadas posteriormente ao treinamento e teste dos modelos) decaem de forma acentuada à medida que os algoritmos aplicados são mais complexos. Nesse contexto, surge a XAI (*Explainable Artificial Intelligence*) - Inteligência Artificial Explicável. A XAI é a área de pesquisa destinada a conceber técnicas que consigam interpretar e justificar algoritmos caixa preta, de forma que esses mantenham as altas taxas de performance e forneçam entendimento sobre sua estrutura e resultados (Trindade, 2020).

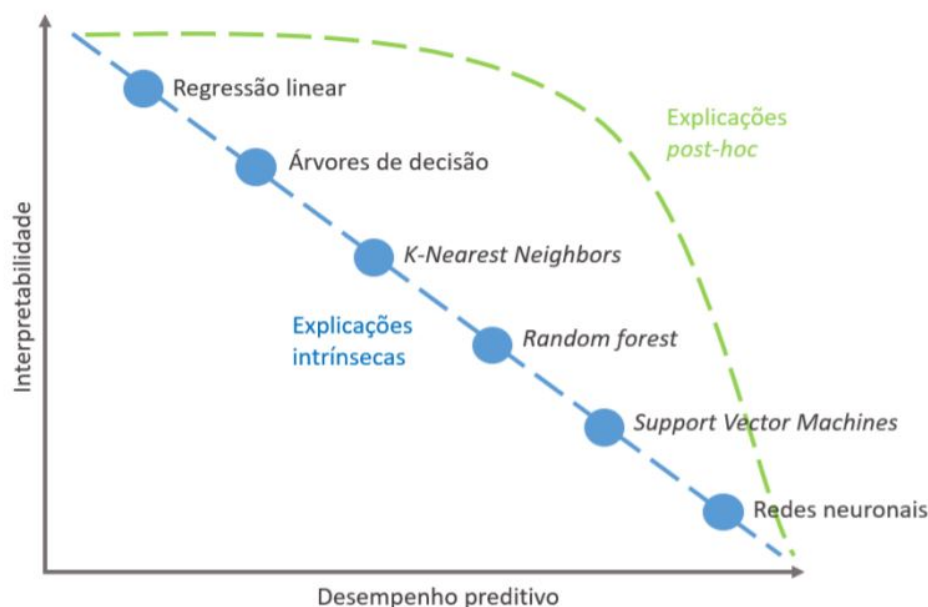


Figura 14 – Interpretabilidade vs Predição.

Fonte: (Trindade, 2020)

Em Du et al. (2019), as técnicas de XAI podem fornecer três importantes aspectos: a validação do modelo, depuração do modelo e a descoberta de conhecimento.

A validação do modelo de aprendizado máquina perpassa por explicar se o modelo

utiliza dados relevantes e com evidências verdadeiras, ou pode mostrar que um modelo pode estar contaminado com algum viés devido a inadequação dos dados. Para se evitar tais problemas pode-se recorrer a transformação dos dados, bem como explorar técnicas de interpretabilidade, até mesmo para impedir que ao sofrer com esse tipo de problema o algoritmo não viole requisitos éticos e legais em sua tomada de decisão (Du et al., 2019).

Quanto a depuração do modelo, é importante notar que a explicabilidade é utilizada para depurar e analisar os algoritmos quando eles fornecem previsões erradas e incomuns (Du et al., 2019). Como exemplo podemos citar, o trabalho realizado por Nguyen et al. (2015), em que redes neurais profundas (DNNs) são enganadas por imagens irreconhecíveis por seres humanos, tendo os resultados de classificação como se essas imagens fossem de um objeto real. Além disso, os resultados dessas redes apresentaram alto grau de certeza. Essas imagens irreconhecíveis são geradas a partir de algoritmos genéticos, os quais enganam a rede neural, a ponto de criar imagens irreais mas com grande similaridade as imagens utilizadas no treinamento das DNNs. Esse estudo mostra o quão distante ainda é a forma de reconhecimento de imagens por seres humanos e DNNs, e como os modelos de ML podem ser falseados, veja a Fig. 15. Assim, destaca-se a importância de se compreender o funcionamento dos modelos de aprendizado de máquina (interpretabilidade), a fim de se evitar qualquer tomada de decisão indesejada por modelos de AI.

Por fim, a descoberta de conhecimento pode ser realizada quando ao se explicar os resultados de um modelo de ML, identifica-se relações escondidas dentro dos dados que caracterizam algum padrão ou algo não visualizado por especialistas da área (Du et al., 2019). Como exemplo, cita-se o trabalho de Caruana et al. (2015), que por meio de um modelo baseado em regras identificou o padrão de um centro hospitalar onde os pacientes que possuíam asma tinham menor chance de morrer de pneumonia que pacientes normais, uma vez que recebiam um tratamento mais cauteloso e agressivo desde o início da doença.

É interessante salientar que boa parte dos modelos de regressão e de séries temporais não possuem boa explicabilidade e interpretabilidade, uma vez que ao final do treinamento concebem um modelo matemático representando uma fórmula, a qual foi parametrizada por algum outro método, como é o caso de um modelo ARIMA que pode ser ajustado pelo algoritmo *Durbin-Levinson* (Morettin and Toloï, 2004). Além disso, boa parte dos algoritmos com boa acurácia para o problema de previsão de séries, como redes neurais recorrentes, possuem complexa parametrização e pouca interpretabilidade, por exemplo, conforme aumenta-se o número das variáveis de entrada e a medida que se testa variadas arquiteturas com um número de neurônios diferente, o modelo pode ter sua acurácia melhorada, porém compreender o funcionamento do algoritmo e o motivo de cada valor de saída pode se tornar uma tarefa hercúlea.

Por fim, em Bhatt et al. (2020), a explicabilidade e interpretabilidade ajudam



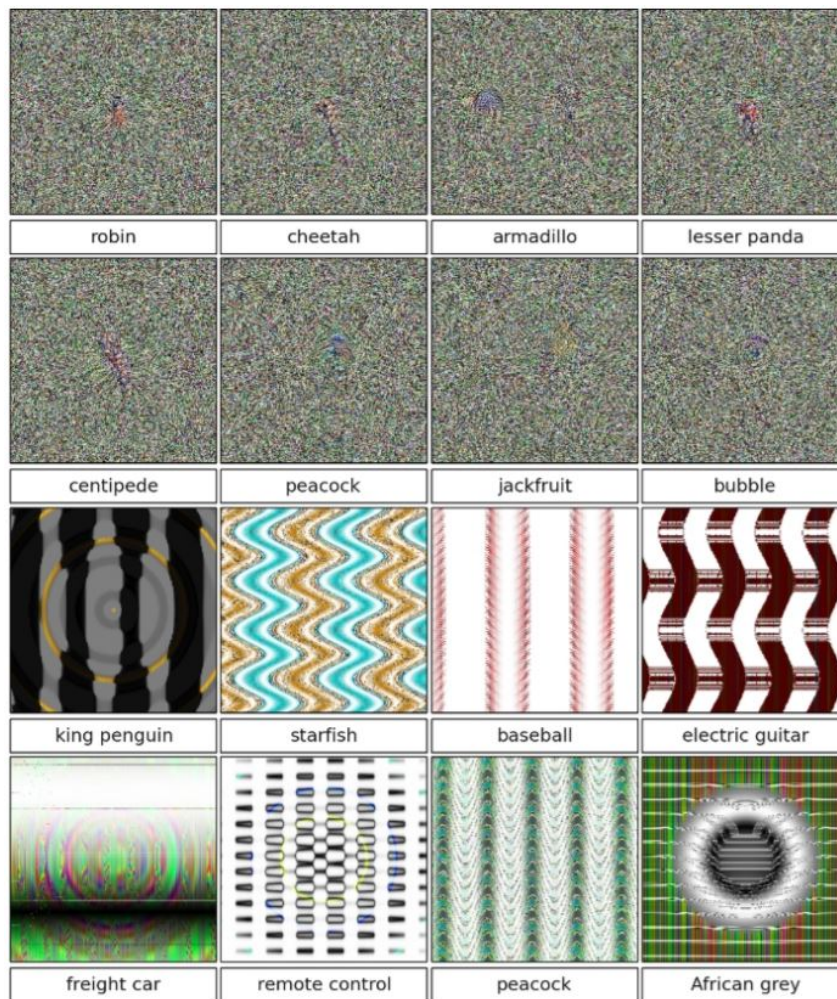


Figura 15 – Imagens criadas por algoritmos genéticos irreconhecíveis por seres humanos, no entanto as redes neurais profundas, treinadas com o banco de dados ImageNet, acreditam com mais 99,6% de certeza ser um objeto familiar

Fonte: (Nguyen et al., 2015)

diversos empreendimentos no mundo a lidar com os modelos de aprendizado máquina no que tange a implantação desses. Quando os algoritmos de ML passam a trabalhar em tempo real é necessário se atentar se esses continuam funcionando adequadamente de acordo com as métricas implantadas. Além disso, no que tange a auditabilidade de um sistema, diversas instituições financeiras têm seus modelos auditados internamente a fim de mitigar riscos e como um requisito legal, como exemplo, tem-se a norma SR 11-7. Além disso, Bhatt et al. (2020) sugere alguns métodos úteis para melhorar a explicabilidade e interpretabilidade dos modelos em produção, como: “Feature Importance”, “Counterfactual Explanations” e “Adversarial Perturbation”.

## 2.5 Utilizando métodos de classificação para regressão

Os problemas de regressão são utilizados para prever valores relativos a variáveis contínuas, em que um problema pode ser descrito na forma de uma função:  $y = f(X_1, X_2, X_3, \dots, X_n)$ . Os trabalhos realizados para prever valores de variáveis contínuas vão desde a simples tarefa de calcular uma regressão linear, até redes neurais utilizadas para tratar de problemas de regressões não lineares, vale destacar que essa evolução se deu conforme o poder computacional aumentou ao longo das décadas. Esses modelos não só são usados para modelar funções, mas também séries temporais, em que boa parte das vezes o desafio é prever valores futuros (como tratado na seção anterior). Mesmo esses modelos apresentando boa acurácia para um dado conjunto de métricas escolhidas, a questão da explicabilidade dos modelos é comprometida. Além disso, quando se trata de muitas dependências de ordem superior entre as variáveis de entrada é necessário o uso de algoritmos como a análise de componentes principais para redução da dimensionalidade. Como exemplo, ao usar algum tipo de rede neural recursiva com centenas de variáveis de entrada, será necessário um pipeline de dados (estrutura que realiza o tratamento dos dados em etapas antes do uso desses em modelos de aprendizado de máquina) para reduzir a dimensionalidade das variáveis de entrada antes do seu uso para treinamento e teste da rede. Ademais, ao se lograr êxito com bons resultados de predição, ao engenheiro de aprendizado de máquina (profissional responsável pela criação e aplicação dos modelos de aprendizado de máquina) restará tentar explicar o modelo (caso seja um dos objetivos primários) por meio de um outro modelo, tarefa essa que pode ser hercúlea (Geron, 2017).

Dessa maneira, nos trabalhos de Weiss and Indurkha (1995); Torgo and Gama (1997); Salman and Kecman (2012b) são mostrados pontos vantajosos ao se transformar um problema de regressão num problema de classificação no que tange a explicabilidade, interpretabilidade, alta dimensionalidade das variáveis de entrada e baixo custo computacional para a execução do modelo.

Em Weiss and Indurkha (1995), é mostrado como se pode explorar algoritmos que geram regras para lidar com problemas de regressão. Uma das abordagens é o uso de árvores de decisão para realizar regressão como é o caso da CART (Breiman et al., 1984). Para Weiss and Indurkha (1995), os modelos de árvore de decisão constituem uma boa forma de gerar bancos de regras capazes de realizar regressões, lidando bem com a questão de alta dimensionalidade e explicação, veja um exemplo de árvore utilizada para regressão na Fig. 16. Para Weiss and Indurkha (1995), deve-se adotar os seguintes passos para realizar uma regressão utilizando algoritmos de classificação a base de regras. Primeiramente deve-se discretizar a série em pseudo classes, as quais podem ser computadas pelo simples particionamento tanto do eixo  $y$  quanto das variáveis  $X_1, X_2, X_3, X_4, X_n$  em regiões. Após, pode-se gerar regras baseadas em algoritmos recursivo como árvores de decisão, e por fim deve-se realizar uma poda das regras, a fim de escolher as regras com

maior acurácia, assim como um refinamento das mesmas pelo uso das métricas de regressão.

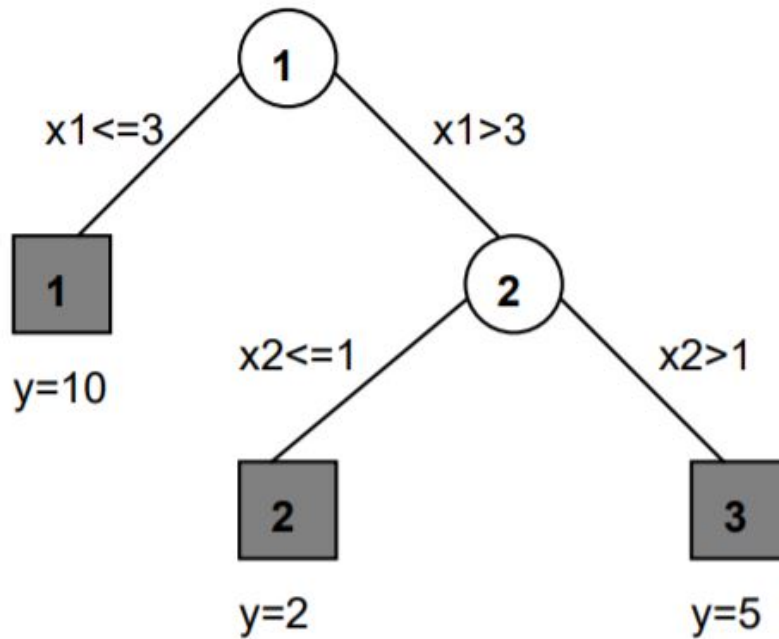


Figura 16 – Exemplo de árvore de decisão

Fonte: (Weiss and Indurkha, 1995)

As regras irão ter como saída algum valor constante relacionado às pseudo classes, como a média ou mediana da região escolhida Weiss and Indurkha (1995). Assim, imagine o seguinte exemplo, dado um conjunto de variáveis de entradas  $X_1, X_2, X_3, X_4 \dots X_n$  discretizadas em  $i$  regiões  $R_{1i}, R_{2i}, R_{3i}, R_{4i}, R_{ni}$ , e uma variável de saída  $y$  discretizadas em  $i$  regiões  $Y_i$ , as regras geradas serão do tipo:

$$X_1 \subseteq R_{1i}, X_2 \subseteq R_{2i} \dots X_n \subseteq R_{ni} \text{ logo } f(x) = \text{mediana}(Y_i)$$

Um dos pontos desvantajosos desse método é que a depender da quantidade de dados usados e do número de partições, a aproximação por valores constantes pode apresentar erros maiores. Uma forma de atenuar essa situação seria usando funções lineares ou não lineares na parte consequente da regra Weiss and Indurkha (1995).

Em Torgo and Gama (1997), é concebido um método denominado RECLA (*Regression Using Classification* - Regressão usando classificação) utilizado como uma ferramenta de pré-processamento que trata de escolher o melhor método de discretização para uma determinada série a fim de possibilitar a aplicação de métodos de classificação para a tarefa de regressão. Esse trabalho explora três métodos de discretização: EP (*Equally probable intervals* - Intervalos igualmente prováveis), EW (*Equal width intervals* - Intervalos de largura iguais), KM (*K-means clustering* - Clusterização realizada pelo K-means). O método RECLA combina esses métodos de discretização com métodos de classificação



como a árvore C4.5 e avalia qual combinação possui maior acurácia. Existem interessantes análises abordadas no trabalho de [Torgo and Gama \(1997\)](#) quanto a discretização e a realizações de regressões utilizando algoritmos de classificação. A primeira dela está contida na Fig. 17, em que os três métodos de discretização (EP, EW, KM) são combinados com o algoritmo C4.5 para resolver o problema de regressão de avaliar o preço de casas. No caso, é salientado no artigo que mesmo ao aumentar o número de discretização na série, o método evolui para um erro constante quanto a distância do grupo classificado em relação ao grupo que o valor real pertence. E, assim, o mais relevante é fazer com que o classificador tenha como resultados conjuntos preditos que estejam perto do conjunto real, ao invés de se preocupar em melhorar o classificador de forma que realize a previsão exata do grupo em que se encontra o valor da série. Isso é notado ao se analisar a Fig. 17, em que ao aumentar o número de classes que discretizam a série o erro total de regressão tende a uma assíntota, enquanto o erro de classificação permanece oscilando em uma faixa de valor constante.

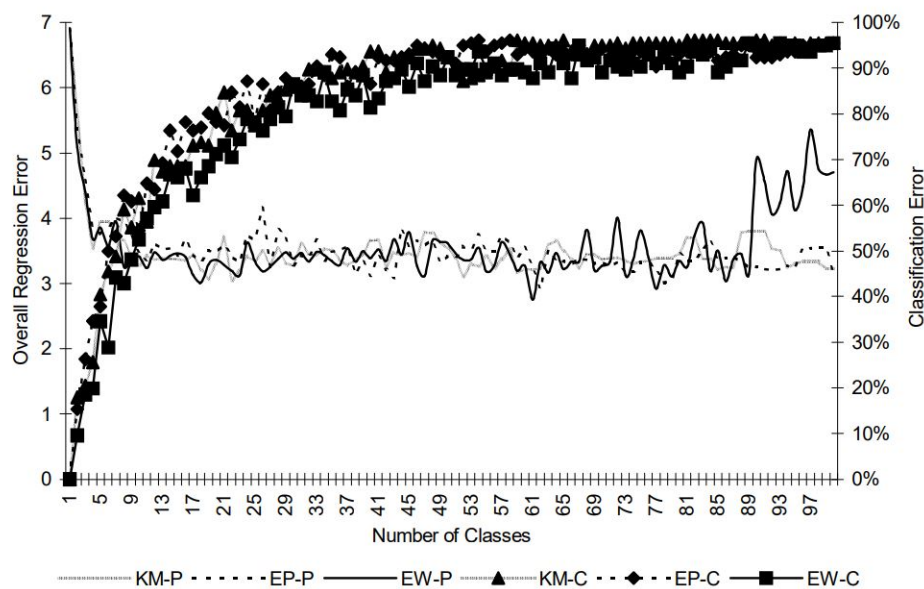


Figura 17 – Erro da Regressão versus Taxa de Erro da Classificação.

Fonte: ([Torgo and Gama, 1997](#))

É interessante citar, também, o trabalho de [Indurkha and Weiss \(2001\)](#), no qual é concebido um método *ensemble* para a geração de regras de classificação utilizadas para resolver problemas de regressão. Esse se mostrou competitivo e com baixo custo computacional em relação aos métodos de regressão tradicionais utilizados por árvores.

Por fim, em [Salman and Kecman \(2012a\)](#), é realizada a discretização de variadas séries relativas a problemas de regressão pelo método  $\varepsilon$ -insensitivity zone - conhecido também como  $\varepsilon$ -tube - podendo ser os intervalos igualmente espaçados ou variantes e, após, aplica-se uma SVM para realizar uma classificação multiclasse. Com isso, a SVM

é capaz de classificar as zonas de valores que foram discretizadas, fazendo com que um modelo de classificação seja capaz de realizar regressão quando uma série é discretizada. Nesse caso, a SVM como modelo de classificação performou um pouco melhor que a SVM de regressão.

## 2.6 Considerações Finais - Capítulo 2

Em resumo, neste capítulo, discorreu-se quanto aos princípios da lógica fuzzy e como essa pode ser aplicada para criar sistemas que podem ser aplicados ao mundo real. Em adição, apresentou-se brevemente alguns conceitos importantes e necessários para a compreensão sobre o que é uma série temporal e as propriedades que podem ser utilizadas para conceber modelos tanto de previsão quanto de análise dessas, além de tangenciar o assunto referente as séries temporais fuzzy (FTS - Fuzzy Time Series), uma abordagem distinta das tradicionais de se representar e analisar séries, a qual possui métodos capazes de lidar com a imprecisão da ocorrência dessas. Ademais, foi abordado o tópico de interpretabilidade e explicabilidade, enaltecendo a necessidade dos motivos pelos quais os modelos de aprendizado de máquina devem ser inteligíveis em suas tomadas de decisão a fim de prover segurança e confiança aqueles que os utilizam. Nesse contexto, apresentou-se também trabalhos que utilizam algoritmos de classificação para prever séries por meio da discretização dessas em classes.

Por fim, o próximo capítulo discorrerá sobre conceitos importantes sobre árvore de decisão e árvores de decisão fuzzy, o que iluminará a compreensão do método que será apresentado nesse trabalho.

## Capítulo 3

# Árvores de Decisão

“De cada semente, um mundo.  
De cada mundo, mil sementes”

---

Yavimaya

O treinamento de uma árvore de decisão gera uma árvore dinâmica ou binária (estrutura de dados na qual a informação é guardada), em que cada um dos seus ramos descendentes (nó pai até os nós folha) representam uma regra do tipo se-então. Essas regras podem ser utilizadas tanto para classificação de conjuntos quanto para regressão de séries. Em adição, as árvores de decisão são métodos supervisionados, ou seja, a indução dessas é realizada durante o treinamento o qual utiliza amostras previamente caracterizadas em classes. Os métodos de árvore decisão tornaram-se populares dentre outros métodos de aprendizado de máquina tendo em vista sua capacidade de gerar regras para dividir um problema em vários subconjuntos para tomada de decisão. Essa divisão ocorre levando em conta cada caminho na árvore de decisão que inicia no nó raiz, perpassa pelos nós internos e termina em algum nó folha, subdividindo o espaço da variável de saída. Dessa maneira, as variáveis de entrada são os atributos das regras responsáveis por realizar essa subdivisão. A visão geométrica dessa subdivisão do espaço de uma variável, no caso de uma árvore de decisão, realiza-se com uma coleção de hiperplanos ortogonais aos eixos das variáveis de entrada e saída (Geron, 2017; Loh, 2011; Breiman et al., 1984). Veja como exemplo a figura 18.

Um dos aspectos relevantes durante a indução de uma árvore é a minimização do erro de generalização considerando o número de nós dessa. Preocupa-se, durante a fase de indução, com o número de nós presentes numa árvore e com a sua profundidade, considerando que ao limitar o número de iterações para a constituição da árvore, evita-se que essa apresente problemas relacionados com *overfitting*, uma vez que árvores com um número de nós e com profundidade em excesso se adequam bem ao conjunto de treinamento, mas possuem pobre poder de generalização quando submetidas aos conjuntos de teste.

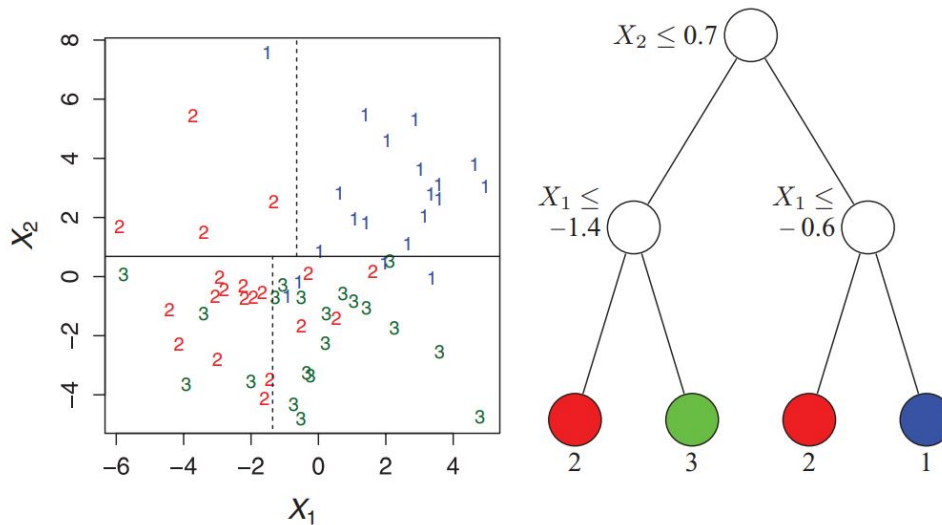


Figura 18 – Exemplo dos particionamento dos dados realizado por uma estrutura de árvore de decisão.

Fonte: (Loh, 2011)

Em adição, as árvores muito extensas podem dificultar a explicabilidade do modelo, dado que se tivermos uma quantidade de regras, por exemplo, na casa de milhões e com cada regra tendo centenas de atributos, um ser humano não teria tempo hábil de entender o funcionamento dessas e extrair algum conhecimento das mesmas (Rokach and Maimon, 2005; Geron, 2017; Kotsiantis, 2011).

Dentre os mecanismos de construção das árvores de decisão tem-se algumas propriedades predefinidas utilizadas na literatura para a estrutura dessas, como é o caso da determinação se a árvore irá ser do tipo *Top-Down* ou *Bottom-up*, se terá critérios de divisão univariada ou multivariada e qual é o critério de parada durante a indução da árvore, como é mostrado em Rokach and Maimon (2005) e Kotsiantis (2011). Adentrando mais detalhadamente nesses conceitos, temos que:

1. Particionamento univariado ou multivariado: Diz respeito a forma que cada nó interno irá se dividir a fim de particionar o espaço de uma dada variável de saída utilizando as variáveis de entrada. Quando se trata de um particionamento univariado tem-se que o método de particionamento utiliza apenas uma métrica para a divisão dos nós internos. Já um método multivariado pode combinar várias métricas para essa divisão. No caso de um particionamento univariado, pode-se usar funções que tenham algum significado estatístico e que melhor caracterizem um conjunto de atributos dado um grau de homogeneidade ou heterogeneidade (grau de impureza), como exemplo: ganho de informação, índice Gini, distância, entre outros. Vale citar que o ganho de informação e o índice Gini são funções tradicionais para os cálculos de critério de impureza na literatura (Hssina et al., 2014; Mienye et al., 2019; Geron,

2017; Rokach and Maimon, 2005).

$$Entropia(D) = - \sum_{i=1}^n p_i (\log_2 p_i) \quad (3.1)$$

Na Eq. (3.1), em que  $p_i$  é a proporção de dados em  $X$  pertencentes à classe  $i$ .

$$\text{GanhoInfo}(D, A) = Entropia(D) - \sum_{j=1}^{|A|} \frac{|D_j|}{|D|} \times Entropia(D_j) \quad (3.2)$$

Na Eq. (3.2). em que  $|D|$  é o tamanho do conjunto  $D$  (representa o número de registros no nó pai),  $D_j$  trata do número de registros associados ao subconjunto  $j$ .

$$\text{Razao Ganho}(D, A) = \frac{\text{GanhoDeInformao}(D, A)}{\text{Entropia}(D, A)} \quad (3.3)$$

$$G_i = 1 - \sum_{k=1}^n (p_{i,k})^2 \quad (3.4)$$

Na Eq. (3.4),  $p_{i,k}$  é a proporção de instâncias da classe  $k$ , nas instâncias de treino do nó  $i$ , e  $G_i$  é o resultado atribuído ao índice Gini.

Como exemplo, tem-se as árvores ID3 (Luger, 2013; Quinlan, 1986), a qual usa ganho de informação, já as árvores C4.5 (Quinlan, 1993) e C5.0 (Pang, 2009; Quinlan, 1996) usam a razão do ganho, Eq.(3.3), e a árvore CART (Breiman et al., 1984; Geron, 2017) usa o índice Gini. Veja nas Eq. (3.1) e (3.2) como se dá o cálculo do ganho de informação Quinlan (2014) e na Eq. (3.4) como o índice Gini é calculado (Rokach and Maimon, 2005; Mienye et al., 2019; Hssina et al., 2014).

2. Critério de Parada: Em Rokach and Maimon (2005), o critério de parada determina quando as iterações que induz uma árvore de decisão terá fim. Não existe uma regra específica para o critério de parada, pois existem conjuntos de condições que podem ser usadas para tal. No caso, pode-se estabelecer um número máximo de nós ou de profundidade que uma árvore pode ter ou estabelecer um valor mínimo para o critério de divisão dos nós, e caso não se obtenha determinado o valor, os nós param de se dividir. Em adição, pode-se usar não apenas uma condição, mas a combinação de várias regras como critério de parada.

Além disso, são utilizados em diversos trabalhos métodos de poda responsáveis por cortar os ramos das árvores que não influenciam em grande medida no erro de generalização, diminuindo assim o risco de *overfitting*. Ademais, os algoritmos de poda tentam resolver o impasse relativo ao problema da escolha do critério de parada. Alguns critérios podem interromper a indução da árvore precocemente e causar o problema de *underfitting* (o

algoritmo possui uma estrutura muito simples e não consegue atingir boas métricas de classificação ou de regressão durante as fases de treinamento e teste). Em oposição, pode-se escolher critérios de parada mais permissivos e que deixem a árvore crescer de forma a se adaptar muito bem ao conjunto de treinamento, mas com péssima capacidade de generalização no conjunto de teste *overfitting*. Com isso, os métodos de poda permitem um critério de parada mais permissivo e ao final do processo de indução esses métodos são aplicados com a finalidade de retirar da árvore alguns ramos que não contribuem para uma boa generalização. Tem-se, também, uma melhora no aspecto da explicabilidade e entendimento do modelo, uma vez que ao reduzir o tamanho de uma árvore gera-se um conjunto com menor número de regras e com menos atributos. Ademais, a poda da árvore também minimiza o danos causados pelos ruídos dos dados no que tange a tomada de decisão (Rokach and Maimon, 2005).

Notadamente, a área de pesquisa para concepção de árvores de decisão evoluiu muito ao longo das últimas décadas e a combinação dos métodos de indução, poda, critério de parada, entre outros, promoveu uma diversidade extraordinária desses algoritmos. Como exemplo, tem-se a CHAID (Milanovic and Stamenkovic, 2016; Rokach and Maimon, 2005), a qual utiliza testes de hipótese para escolher um conjunto de atributos que seja pouco diferente de um conjunto de atributos alvo, com base no  $p$  – valor dos testes utilizados (para atributos contínuos usa-se o teste F, já para atributos discretos, o teste Pearson Qui-Quadrado). Outro exemplo, é a BUTIA (Barros et al., 2011) um algoritmo capaz de induzir árvores utilizando particionamento *bottom-up* e multivariado, concebendo, assim, árvores com capacidade de divisões oblíquas aos eixos da variáveis. Além desses modelos brevemente citados, veja alguns outros modelos tradicionais de árvores de decisão encontrados na literatura:

1. ID3, C4.5 e C5.0: A árvore ID3 é um algoritmo recursivo que utiliza ganho de informação como forma de particionar os nós internos (critério de escolha dos atributos). O critério de parada ocorre na ID3 quando todas as instâncias das variáveis de entrada pertencem a um único valor da variável de saída ou se o ganho de informação não é maior que zero. Vale salientar que a ID3 não possui nenhum método de poda. Já a c4.5 representa uma evolução da ID3, apresentando a razão de ganho como forma de particionamento dos nós e apresenta o método de poda baseada em erro após a fase de indução da árvore. A razão de ganho, também, utiliza as fórmulas de entropia e ganho de informação, mas apresenta o ganho de informação normalizado, como mostrado na Eq. (3.3) para o particionamento dos nós durante o treino da árvore. Perceba que uma das desvantagens da razão do ganho é que não pode ser calculado quando a entropia é zero e para valores pequenos de entropia temos valores grande de razão do ganho. Mesmo com essas desvantagens, um critério de partição normalizado é melhor que o cálculo do ganho de informação, uma

vez que o critério de impureza baseado no valor absoluto do ganho de informação favorece os atributos de entrada com maior número de valores no domínio do que os atributos com menor número. Em adição, os experimentos empíricos quanto ao uso da razão do ganho se mostraram melhores em termos de acurácia para classificação do que o critério do ganho de informação. Já a árvore de decisão c5.0 é sucessora da árvore c4.5, apresentando consideráveis melhorias na questão de desempenho quanto à classificação, menor consumo de memória durante sua indução, menores e mais performáticas árvores, suporte a conjuntos de dados desbalanceados e faltantes, suporte ao uso de *boosting* e realiza uma melhor seleção dos atributos. Essas árvores são criadas a partir do método *top-down*, em que se calcula primeiro o ganho de informação dos atributos (critério de impureza) e depois divide-se de forma recursiva o conjunto de treinamento do nó raiz até o nó folha (Rokach and Maimon, 2005; Quinlan, 1993; Luger, 2013; Quinlan, 1996; Witten et al., 2011).

2. CART: A árvore CART (*Classification and Regression Tree*) é capaz de realizar tanto classificação quanto regressão e utiliza o índice Gini como critério de impureza para a divisão dos nós. É uma árvore induzida, também, de forma *top-down*, porém apresenta apenas estrutura binária, diferentemente das árvores ID3, C4.5 e C5.0 (Breiman et al., 1984; Rokach and Maimon, 2005; Geron, 2017).

As vantagens de se utilizar as árvores de decisão são variadas e não são exaustivas, podemos citar como relevantes as seguintes (Rokach and Maimon, 2005; Geron, 2017; Kotsiantis, 2011):

- Não há necessidade de normalização dos dados de entrada e saída;
- Os modelos de árvore de decisão podem constituir regras, assim é um modelo explicável e interpretável;
- As árvores realizam naturalmente a seleção de atributos mais relevantes (*Feature Selection*);
- É possível trabalhar tanto com valores numéricos quanto categóricos;
- Capacidade de lidar com um conjunto de dados muito grande e implementação paralela (Meng et al., 2016; Amado et al., 2001);
- Podem utilizar conjunto de dados com valores faltantes durante o treinamento;
- Capacidade de lidar com mudanças bruscas de conceito - *concept drift* (Lu et al., 2019);
- Tem-se baixo custo computacional para se percorrer a estrutura da árvore, no caso  $O(\log_2(m))$ , em que  $m$  é o número de nós.

- Possui capacidade de estimar probabilidade de classes;
- Método não paramétrico (não assume que os dados tenham alguma distribuição).

Já algumas das desvantagens são as seguintes (Rokach and Maimon, 2005; Geron, 2017; Kotsiantis, 2011):

- As árvores de decisão podem crescer de forma que gerem um grafo ou regras extensas, o que pode tornar complexa a sua explicabilidade;
- Os modelos tradicionais de árvore realizam particionamentos com hiperplanos ortogonais aos eixos das variáveis, deixando as árvores sensíveis a rotação dos conjuntos de treinamento, o que pode acarretar uma ruim generalização;
- Árvores de decisão são sensíveis a ruídos ou pequenas variações nos dados de treinamento.

Já o método *Random Forest* é um algoritmo *ensemble* de aprendizagem de máquina utilizado tanto para problemas de classificação quanto regressão. Esse utiliza o método de *bagging*, em que são treinadas várias árvores em subconjuntos aleatórios do conjunto original de treinamento. Esses subconjuntos não só são escolhidos de forma aleatória como são re-amostrados utilizando a técnica de *bootstrapping*. O algoritmo *Random Forest* garante grande diversidade das árvores treinadas, reduzindo assim a variância dos resultados da predição sem aumentar o viés devido a agregação dos resultados por meio da votação das árvores. Perante essas vantagens, os métodos *ensemble*, geralmente, apresentam melhores resultados que os algoritmos simples de árvore, no entanto, por serem mais complexos que essas perdem em explicabilidade. (Geron, 2017; Breiman, 2001).

Por fim, dentre as várias aplicações das árvores de decisão podemos destacar modelos utilizados na área da saúde, em modelos financeiros e em processamento de imagens. Exemplificando, em Garcia (2003), é demonstrado a aplicação de árvores de decisão para a mineração de dados na área da saúde, em que essas são usadas para a determinação da internação hospitalar com base nos dados do paciente. Já em Sadorsky (2021), o algoritmo *Random Forest* foi usado para prever a direção de cinco *ETFs* de energia limpa amplamente comercializados no mercado americano, a partir de indicadores técnicos como dados de entrada para o modelo. Além disso, em Jeffrey Kuo et al. (2013), é apresentado um trabalho que analisa a região da laringe por meio de vídeos feitos por um estroboscópio, identificando patologias clínicas nas pregas vocais dos pacientes como paralisia vocal e nódulos vocais. Com o uso de uma árvore de decisão e o pré-processamento das imagens dos dados fisiológicos de pacientes, obteve-se uma acurácia média de 98,7% na classificação dentre as imagens com cordas normais, com paralisia ou com nódulo.



## 3.1 Árvores de Decisão Fuzzy - Fuzzy Decision Trees (FDT)

O advento das árvores de decisão foi um importante passo na evolução dos algoritmos supervisionados na área de aprendizado de máquina. Os diferentes métodos para induzir a estrutura de uma árvore mostraram muita versatilidade para resolver tanto problemas de regressão como de classificação, como mostrado na seção passada. Nesse sentido, a junção da teoria dos conjuntos fuzzy com as árvores de decisão apontaram para um novo caminho em que as árvores obtiveram uma nova propriedade: lidar com dados e problemas imprecisos. Em adição, as árvores fuzzy apresentam a capacidade de gerar regras fuzzy e com isso fornecer uma base de conhecimento para um sistema fuzzy baseado na estrutura de uma árvore que pode ser traduzida em regras. Basicamente, dentre os diversos trabalhos que concebem uma Árvore de Decisão Fuzzy (ADF ou FDT - *Fuzzy Decision Tree*), há a presença de dois modelos dos quais partem a ideia principal de formação de uma FDT: árvores de decisão fuzzy com pré fuzzificação ou sem pré fuzzificação. No que tange esses conceitos, as FDTs com pré-fuzzificação realizam a fuzzificação dos atributos e usam esses como entrada dos métodos de árvore. Já o método sem pré-fuzzificação realiza a fuzzificação dos atributos durante a indução da árvore (Yuan and Shaw, 1995; Kotsiantis, 2011; Lopes, 2016).

Nos trabalhos desenvolvidos por Umanol et al. (1994); Janikow (1998); Jang (1994), tem-se a fusão de árvores de decisão e a lógica fuzzy. Nesse as formas de indução das duas árvores tradicionais a época, a CART e a ID3, foram usadas com atributos pré-fuzificados a fim de se conceber as árvores fuzzy. No caso, uma FDT baseada no método de indução da CART e da ID3 se diferenciam na forma como as regras são inferidas (utiliza-se métodos próprios de inferência da lógica nebulosa), além do que os critérios de impureza não são calculados sobre os atributos numéricos, mas sobre os atributos fuzificados. Com essa nova abordagem, as árvores passam a ganhar novas capacidades, como a de ter menos influência de ruídos presentes no conjunto de treinamento, bem como a capacidade de lidar com imprecisão (Janikow, 1998). Dentre as várias pesquisas relativas ao uso das FDTs vale discorrer brevemente sobre algumas para a compreensão quanto a dimensão da capacidade dos algoritmos de FDT, da sua diversidade e da amplitude de suas aplicações, no caso, temos:

- **Fuzzy Decision Tree ID3:** Em Umanol et al. (1994), foi concebido um método com base na pré-fuzzificação de variáveis, no qual se usa tanto valores fuzzy quanto valores numéricos no conjunto de treinamento da mesma. Para isso foi utilizado um método de indução semelhante a da árvore ID3, em que se concebe um algoritmo interpretável dado os atributos de entrada.

- **LAIVFDT-OPN:** O método das árvores de decisão difusas com valor de intervalo com perímetro de vizinhança ideal, é desenvolvido em [Lertworaprachaya et al. \(2014\)](#) e foi concebido para criar modelos de árvores de decisão fuzzy, com previsão intervalar, a partir dos valores de pertinência difusos associados a intervalos. Esse modelo expressa a dificuldade de se obter valores de pertinência precisos e como os valores da vizinhança desse são significativos para modelar a incerteza. Ao final, o método LAIVFDT-OPN fornecerá uma árvore de decisão fuzzy na qual os nós folha (nós de decisão) apresentaram um intervalo, baseado nos métodos explicitados em [Lertworaprachaya et al. \(2014\)](#). Nos testes realizados com o dataset DDK factory data, o LAIVFDT-OPN performou melhor que outros dois algoritmos, o LAIVFDT ([Lertworaprachaya et al., 2010](#)) e o Fuzzy ID3 ([Umanol et al., 1994](#)), no que diz respeito as métricas de acurácia, sensibilidade e especificidade.
- **FuzzyDT:** A árvore FuzzyDT, desenvolvida em [Cintra et al. \(2013\)](#), é baseada no algoritmo C4.5. Essa realiza a pré-fuzzificação dos atributos numéricos para valores fuzzy e realiza a indução da árvore C4.5. Ao final, a estrutura da árvore, do nó raiz até os nós folha, terá seus caminhos compostos por atributos difusos. Com isso, a árvore poderá ser traduzida em regras fuzzy e pode-se aplicar inferências próprias da matemática fuzzy a essas regras. Nos experimentos realizados em 16 diferentes *datasets*, obteve-se da FuzzyDT, em 10 desses conjuntos, um menor erro de classificação em comparação ao algoritmo C4.5, além do fato das árvores geradas pelo método FuzzyDT possuírem menor tamanho, conseqüentemente menos regras, que a árvore original C4.5.
- **Fuzzy Random Forest:** A *Fuzzy Random Forest* consiste num método semelhante a *Random Forest* desenvolvida por [Bonissone et al. \(2010\)](#), em que se gera um conjunto de árvores fuzzy aleatórias e se utiliza de diferentes caminhos para se extrair o resultado da floresta. Um primeiro caminho é a tomada de decisão pelo resultado de cada árvore, e a partir dessa se retira uma tomada de decisão ponderando-se os resultados individuais, levando em consideração o quão bom é o resultado de cada árvore dado, por exemplo, o ganho de informação. Um segundo caminho seria uma decisão comum tomada pelas árvores que compõe toda a floresta.

Por fim, as FDTs possuem vasta gama de aplicações. Como exemplo, em [Lai et al. \(2009\)](#), é aplicado um método de árvore fuzzy num conjunto com algoritmos genéticos e algoritmos de agregação de dados a fim de realizar a previsão da direção das séries temporais financeiras das ações listadas na bolsa de *Taiwan Stock Exchange Corporation (TSEC)*. Esse modelo utilizou dados de entrada tanto fundamentalistas como técnicos, obtendo, segundo a métrica *Hit Ratio* (métrica usada para verificar o acerto do modelo em prever um dos três grupos: Compra, Venda, Manter), uma performance média de

82%. Outro exemplo se dá em [Idris and Ismail \(2021\)](#), em que se faz o uso de uma Fuzzy-ID3 (FID3) ([Umanol et al., 1994](#)) em conjunto com uma técnica de banco de dados FUZZYDBD ([Cintra et al., 2009](#)) para prover um sistema capaz de detectar de forma precisa casos de câncer de mama. Essa árvore de decisão fuzzy foi aplicada à três diferentes tipos de conjunto de dados: WBCD (Original), WDBC (Diagnóstico) e Coimbra, demonstrando performar de forma superior aos algoritmos tradicionais, no caso para o *dataset* WBCD (Original) obteve para as métricas *Precision*, *Recall* e *F1*, respectivamente, 94.192%, 93.356% e 93.747%. Já para os conjunto de dados de WDBC (Diagnóstico) e Coimbra apresentou-se em relação a *Precision*, *Recall* e *F1*, respectivamente, os resultados de (94.294%, 94.000%, 94.137%) e (69.784%, 69.056%, 69.278%).

## 3.2 Considerações Finais - Capítulo 3

Em suma, foram abarcados conceitos primários sobre as árvores de decisão neste capítulo, como a forma de estrutura da árvore, forma de indução, impureza, critério de parada, entre outros. Abordou-se, brevemente, alguns modelos tradicionais de árvore bem como a *Random Forest*, método *ensemble* de árvores. Além disso, listou-se algumas das vantagens e desvantagens desses, assim como as suas aplicações. Em sequência, foi abordada alguns dos princípios quanto as árvores de decisão fuzzy, trabalhos relacionados e aplicações.

Por último, os conceitos apresentados até este ponto do trabalho apresentam base para o entendimento do método do capítulo relativo a metodologia que se seguirá. Neste apresentaremos os dois métodos propostos de árvores de decisão fuzzy: FDT-FTS Simples e FDT-FTS Composto, assim como será destrinchada a lógica de ambos.



# Capítulo 4

## Métodos Propostos

“Criatividade é uma mente selvagem e um olho disciplinado.”

---

Dorothy Parker

### 4.1 Introdução

Primeiramente, demonstra-se quais os passos iniciais para a abordagem do método de FDT-FTS de acordo com figura 19. Essa representa uma visão global do método.

Explica-se de forma mais detalhada nas sessões a seguir cada um dos passos destacados na Fig. 19.

#### 4.1.1 Leitura e Análise de Dados

Inicialmente, realiza-se a leitura dos dados de alguma fonte como um banco de dados ou um arquivo de texto. É importante nessa etapa a realização da verificação dos tipos de dados, se há dados faltantes, se o formato dos dados e a estrutura dos dados são compatíveis com os requisitos do algoritmo utilizado. Como exemplo, na linguagem Python, os dados podem estar numa lista (estrutura de dados nativa da linguagem *Python*<sup>1</sup>), *Data Frames* (biblioteca *Pandas*<sup>2</sup>), *Array* (biblioteca *NumPy*<sup>3</sup>), entre outros. Salienta-se que antes da fuzziificação dos dados e do treino/teste das árvores é importante que os dados estejam no formato de lista. Além disso, após a leitura dos dados, é imprescindível a realização de uma análise exploratória em que são realizados testes de hipóteses (ex.: teste de estacionariedade), análise gráficas do comportamento dos dados no tempo (ex.: estudos

---

<sup>1</sup> <https://www.python.org/>, Acessado em: Outubro de 2021.

<sup>2</sup> <https://pandas.pydata.org/>, Acessado em: Outubro de 2021.

<sup>3</sup> <https://numpy.org/>, Acessado em: Outubro de 2021.

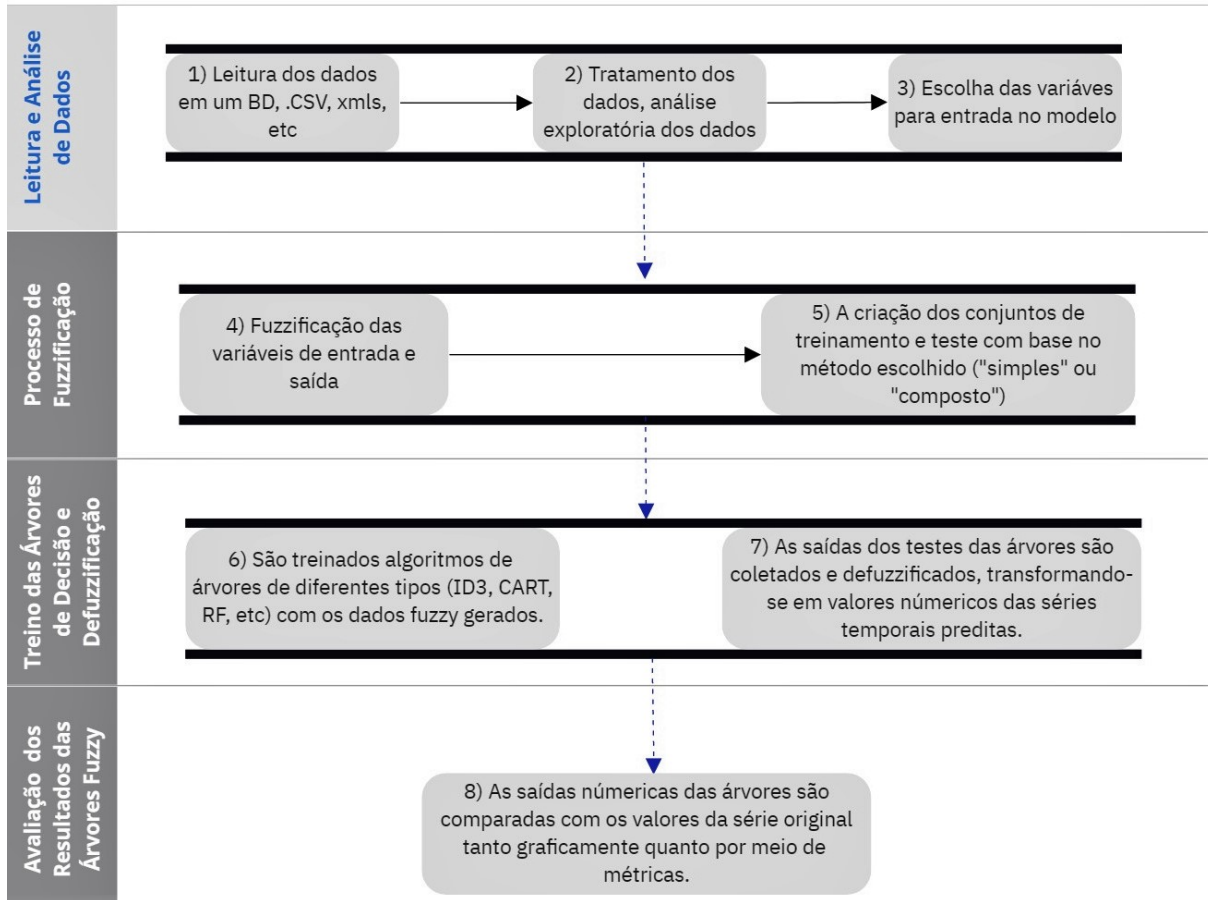


Figura 19 – Arquitetura da aplicação de árvores de decisão fuzzy (*fuzzy decision trees*) para previsão de séries temporais.

Fonte: Autor

de autocorrelação e correlação com variáveis externas), cálculos relativos a estatística descritiva das variáveis do problema (ex.: média, desvio padrão, moda, valores máximos e mínimos), estudos das distribuições das variáveis por meio de histogramas, etc. Essa análise proverá uma visão de quais dados são ou não relevantes para a resolução do problema de previsão e como variáveis externas ao problema podem se correlacionar com a série principal.

#### 4.1.2 Processo de fuzzificação

O processo de fuzzificação é responsável por atribuir um ou mais conjuntos fuzzy aos valores das variáveis de entrada e saída, assim como mapear os valores de pertinência de cada instância relacionada a essas variáveis. Nessa dissertação utilizamos a biblioteca *pyFTS*<sup>4</sup> para realizar a fuzzificação das variáveis de entrada e saída. Essa biblioteca permite a escolha de variados tipos de funções de pertinência (triangular, trapezoidal ou gaussiana), além de realizar a fuzzificação pela escolha do valor máximo de pertinência ou

<sup>4</sup> <http://pyfts.github.io/pyFTS/build/html/index.html>, Acessado em: Outubro de 2021.

pelos valores de pertinência maiores que um dado “ $\alpha$  – corte”. Utilizaremos a notação e as equações presentes em [Lima e Silva \(2019\)](#) para elucidar as duas formas com que a *pyFTS* realiza o processo de fuzzificação.

$$f(t) = A_j \mid \mu_{A_j}(y(t)) = \max \{ \mu_{A_1}(y(t)), \dots, \mu_{A_k}(y(t)) \} \quad (4.1)$$

No caso, a escolha do valor máximo de pertinência pode ser representado pela função  $f(t)$  descrita na Eq. 4.1, em que  $y(t) \in Y$ , sendo  $Y$  a série numérica a ser fuzzificada,  $A_j \in \tilde{A}$  em que  $\tilde{A}$  é uma variável linguística, assim  $A_j$  representa os vários conjuntos fuzzy extraídos de  $\tilde{A}$ .

Essa primeira forma de fuzzificação (valor máximo de pertinência) é vantajosa por ter um cálculo simples e por minimizar as chances de *overfitting* dos métodos que utilizam os valores fuzzy para previsão de séries. No entanto, a fuzzificação pelo valor máximo pode levar a perda de informações, já que resume todos os valores presentes numa região da série em um único número [Lima e Silva \(2019\)](#).

$$f(t) = A_j \mid \mu_{A_j}(y(t)) \geq \alpha \quad \forall A_j \in \tilde{A} \quad (4.2)$$

Já o segundo método de fuzzificação pode ser expresso pela Eq. 4.2, em que os conjuntos fuzzy  $A_j$  são escolhidos pelos valores de pertinência computados por  $\mu_{A_j}(y(t))$ , os quais devem ser maiores ou iguais a  $\alpha$ . Assim, uma vez que a série numérica  $Y$  é fuzzificada, um valor *crisp* dessa pode ser representado por uma coleção de conjuntos fuzzy, o que favorece a representação da região a que um ponto da série pertence. Vale enaltecer que ambos os métodos de fuzzificação descritos acarretam efeitos distintos nos métodos de FTS, assim não existe, a priori, preferência no uso desses.

Veja que o processo de fuzzificação mostrado na Fig. 20, tem a propriedade de discretizar a série temporal em conjuntos fuzzy utilizando funções de pertinência triangular. Assim, na Fig. 20, caso se realize a fuzzificação dos valores iniciais dessa série pelo valor de pertinência máximo, coincidente com o valor médio da função de pertinência triangular, teríamos como resultado uma coleção valores fuzzy a:  $Y_{Fuzzy} = [A_6, A_5, A_4, A_3, A_2, A_1, A_1, A_2, A_3, A_4, A_3 \dots A_j]$ , em que  $Y_{Fuzzy}$  é a série fuzzificada gerada a partir da série numérica  $Y$ , com valores da série original representados por conjuntos fuzzy até a  $j$ -ésima posição, presente na Fig. 20. Assim, por meio da fuzzificação se transforma as séries temporais numéricas em séries temporais fuzzy (Fuzzy Time Series - FTS). Além disso, caso fosse utilizado o segundo método de fuzzificação apresentado ( $\alpha$  – *cut*), poderíamos ter mais de um conjunto de pertinência por valor da série original representados até a  $j$ -ésima e  $i$ -ésima posição (nesse caso, há dois conjuntos fuzzy por valor da série, no entanto, pode-se ocorrer indefinidas quantidades de representações por conjuntos fuzzy para cada valor da sé-

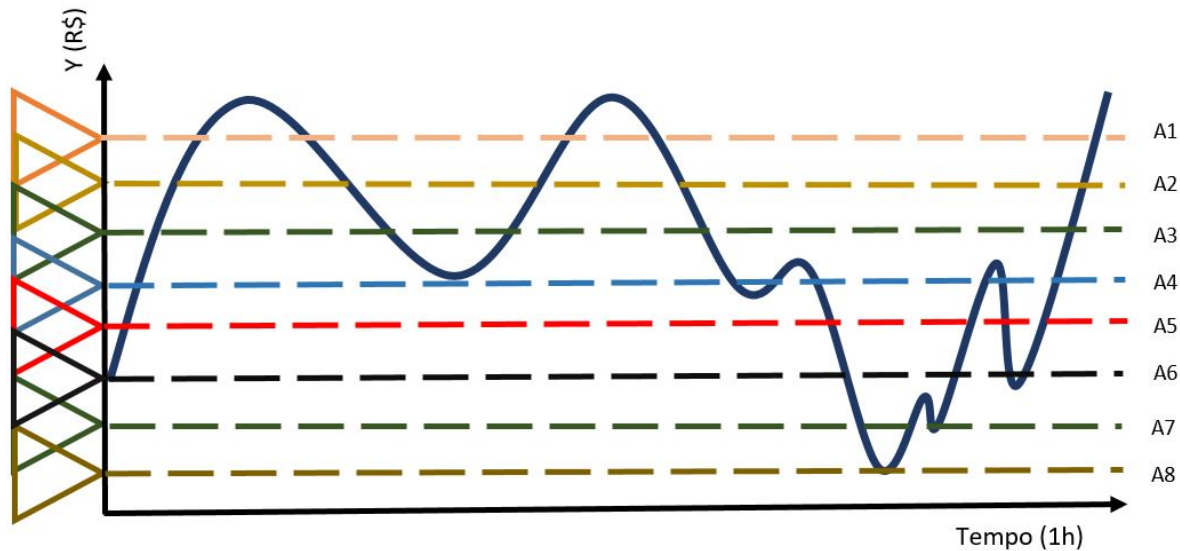


Figura 20 – Exemplo de fuzzição da Série de um Ativo  $Y(R\$)$  no  $Tempo(1h)$ .

Fonte: Autor

rie) :  $Y_{Fuzzy} = [(A_5, A_6), (A_4, A_5), (A_3, A_4), (A_2, A_3), (A_1, A_2), (A_0, A_1), (A_0, A_1), (A_1, A_2), (A_2, A_3), (A_3, A_4), (A_2, A_3) \dots (A_i, A_j)]$

### 4.1.3 Treino da Árvore de Decisão e Defuzzição

O treino das árvores de decisão é realizado com os valores das variáveis de entrada e saída fuzziçados. Nesse caso, pode-se escolher qualquer método de árvore de decisão que realize a classificação dos conjuntos fuzzy. Dessa forma, transforma-se os problemas de regressão num problema de classificação, de forma que quanto melhor é a classificação em relação as proximidades do conjunto fuzzy a que um valor pertence, melhor será a escolha do conjunto fuzzy previsto. Após a defuzzição, o valor numérico que ocorre no tempo será próximo do valor real da série original. Caso a classificação seja ruim, o conjunto fuzzy previsto terá uma classificação distante do conjunto fuzzy real, bem como o valor da série numérica prevista também será distante em relação ao valor numérico real. Pode-se utilizar alguma métrica ou artifício gráfico para visualizar a qualidade da classificação.

Recomenda-se, nesse caso, utilizar para o treinamento e teste das árvores métodos de janelas deslizantes no tempo (ex.: método *Walk-Foward*), uma vez que a ordem dos eventos numa série temporal importa, tendo em vista as características de autocorrelação da série. Dessa forma, não se deve realizar a validação cruzada com dados embaralhados, mas realizar a validação cruzada com janelas deslizantes no tempo. As janelas de treino/teste podem ser fixas e percorrer todo o espaço temporal da série ou podem ser de tamanhos diferentes. Dessa maneira, com a finalidade de se abarcar as diversas possibilidades dos resultados quanto a previsão das séries, ambas as alternativas podem ser utilizadas durante o treino e teste das árvores (Morettin and Toloi, 2004; Falessi et al., 2020; Montgomery



et al., 2015). Ao final do treinamento da árvore de decisão, teremos um grafo no formato de uma árvore, a qual representará nossa base de conhecimento e poderá ser traduzida em regras, considerando-se cada caminho do nó raiz até o nó folha. Com isso, pode-se conceber regras as quais terão atributos fuzzy e com isso todas as vezes que uma regra ocorrer, teremos como saída um valor fuzzy, que poderá então ser defuzzificado. O processo de defuzzificação transformará um valor fuzzy num valor numérico (valor *crisp*). Esse valor é o que de fato será utilizado como previsão da série temporal. Nas sessões seguintes apresentaremos o método FDT-FTS simples e composto, os quais demonstrarão a forma como as árvores serão induzidas com os dados fuzzy e como suas saídas serão defuzzificadas. Além disso, é possível prever valores com diferentes *lags* temporais, previsão multi-passos,  $(t + 1, t + 2, \dots, t + n)$ , bastando para isso realizar um deslocamento nos dados de treino e teste da variável de saída (método direto). Essa estratégia tenta obter o valor previsto diretamente das regras induzidas pela árvore dado o horizonte temporal desejado. Outra forma de realizar a previsão multi-passos seria utilizar um processo recursivo, no entanto o enfoque do trabalho atual é a estratégia de previsão direta uma vez que não se deseja o acúmulo dos erros como ocorre no processo recursivo (Fernandes Costa, 2015). Em suma, para explicitar os passos de treinamento da árvore com uma janela deslizante, tem-se no

algoritmo 1 os passo gerais.

---

**Algoritmo 1:** Algoritmo do treino da árvore e coleta dos valores previstos com base na janela deslizante no tempo

---

**Data:** DadosFuzzy, n, p, l

```

1   $l \leftarrow 1$ ; *l é o tamanho do lag (t+1,t+2...t+n).*
2   $n \leftarrow TamanhoJanelaDeslizante$ ; *n é o tamanho da janela deslizante.*
3   $p \leftarrow QtdPassosJanela$ ; *p é o número de passos que a janela irá desliza.*

4   $DadosTreinoFuzzy \leftarrow []$ ;
5   $DadosTesteFuzzy \leftarrow []$ ;
6   $DadosPrevistosFuzzy \leftarrow []$ ;
7   $DadosPrevistosReais \leftarrow []$ ;

8   $k \leftarrow 0$ ;

9  while  $k < p$  do
10      $InicioTreino \leftarrow (n * k)$ ;
11      $FinalTreino \leftarrow n + (n * k)$ ;
12      $InicioTeste \leftarrow FinalTreino + l$ ;
13      $FinalTeste \leftarrow InicioTeste + n$ ;

14     *Prepara os conjuntos para o treino e teste.*
15      $DadosTreinoFuzzy[k] \leftarrow DadosFuzzy[InicioTreino : FinalTreino]$ ;
16      $DadosTesteFuzzy[k] \leftarrow DadosFuzzy[InicioTeste : FinalTeste]$ ;

17     *Treino e teste da árvore de decisão.*
18      $Arvore[k] \leftarrow TreinoArvore(DadoTreinoFuzzy)$ ;
19      $DadosPrevistosFuzzy[k] \leftarrow TesteArvore(Arvore[k], DadosTesteFuzzy)$ ;

20     *Realiza a defuzzificação dos dados.*
21      $DadosPrevistosReais[k] \leftarrow Defuzzificacao(DadosPrevistoFuzzy)$ ;

22      $k \leftarrow (k + 1)$ ;

```

---

## 4.2 FDT-FTS Método Simples

O modelo FDT-FTS simples está relacionado com a forma de fuzzificação e defuzzificação dos dados. No caso, a criação dos valores fuzzy para treinamento e teste ocorre calculando-se o valor de pertinência máximo para os valores das variáveis de entrada e saída. Como exemplo, suponha um conjunto de dados que possui as variáveis de entrada  $X$  e  $T$  e a variável de saída  $Y$ , e que o universo de discurso dessas três foi dividido em conjuntos fuzzy utilizando uma função de pertinência triangular (processo de fuzzificação).

Além disso, imagine que os valores para uma dada instância de treinamento dentre várias outras, seja  $X_1 = 0.8$  e  $T_1 = 3$  e  $Y_1 = 10$  e que possamos atribuir a cada um desses valores um conjunto fuzzy, calculando-se o valor de pertinência máximo. Com isso, tem-se que ao invés de valores numéricos, o conjunto de treinamento e teste possuirão os valores referentes a conjuntos fuzzy do maior valor de pertinência. Como exemplo, os valores fuzzy da instância apresentada anteriormente são:  $X_{Fuzzy1} = 5$ ,  $T_{Fuzzy1} = 2$ ,  $Y_{Fuzzy1} = 7$ , em que 5, 2, 7, são respectivamente, os conjuntos fuzzy **5**, **2**, **7**, considerando a fuzzificação de cada uma das variáveis do exemplo ( $X$ ,  $T$  e  $Y$ ).

Sendo assim, tanto os algoritmos de árvore de decisão quanto a *Random Forest* não são treinados com os valores numéricos, mas com os valores que representam o conjunto fuzzy de maior valor de pertinência. Além disso, após a indução das árvores, essas terão na sua estrutura, do nó raiz até os nós folha, atributos fuzzy, o que permite que esses caminhos entre a raiz da árvore e suas folhas sejam traduzidos em regras.

Tendo em vista a estrutura de uma árvore induzida com os atributos pré-fuzzificados, percebe-se que, nesse caso, tem-se sempre a situação que para uma instância de entrada, apenas uma instância de treinamento é ativada, ou seja apenas uma regra é ativada, já que os caminhos presentes na estrutura de uma árvore de decisão são autoexcluentes, ou seja, uma única instância de entrada não pode ativar múltiplos nós folha da árvore, apenas um por entrada. Em adição, é válido frisar que os nós folhas da árvore tem como saída conjuntos fuzzy.

Diante do exposto, percebe-se que para cada entrada do conjunto de treinamento fuzzy, tem-se como saída um valor fuzzy. É interessante notar que nesse estágio, ao produzir saídas fuzzy de acordo com as entradas no tempo, a árvore produz uma série temporal fuzzy (FTS - Fuzzy Time Series).

Tem-se, então, que o caminho da árvore pode ser traduzido numa regra no estilo Mamdani Eq.(2.9), o que possibilita a realização do processo de defuzzificação. Com isso, podemos utilizar algum dos métodos apresentados no Capítulo 2. Assim, utiliza-se, nesse caso, o “Critério dos Máximos” como forma de defuzzificação. A partir desse necessita-se apenas computar o valor máximo do conjunto fuzzy que a árvore fornece como saída. Sendo assim, não se aplica o método Mamdani, pois utiliza-se apenas o valor do consequente da regra, tendo a regra apenas um grau de disparo. Ademais, note que se fosse aplicado o método do centroide Eq.(2.11), nesse caso em que o  $k = 1$ , percebe-se que o centroide é o valor do centro do conjunto fuzzy de saída da árvore, equivalente ao valor máximo, considerando as funções presentes na *pyFTS* para a fuzzificação (funções triangulares, trapezoidais e gaussianas simétricas). Para se defuzzificar o valor de saída da árvore, utiliza-se a expressão Eq. (4.3), a qual reflete o método do máximo (princípio de associação

máxima) (Ross, 2009).

$$\mu_C(z^*) \geq \mu_C(z), \forall z \in Z \quad (4.3)$$

Na Eq. (4.3),  $z$  é o valor defuzzificado e  $C$  representa o conjunto fuzzy. Esse método é aplicado a conjuntos fuzzy que tenham apenas um valor de máximo (ex.: funções triangulares e gaussianas). No entanto, caso a função usada seja, por exemplo, uma função trapezoidal, coletamos o valor de máximo que esteja no ponto mediano do conjunto.

Dentre as vantagens desse método de defuzzificação, nota-se que é necessário apenas conhecer a parte consequente da regra, uma vez que se calcula apenas o valor máximo do conjunto fuzzy de saída da árvore. Com isso, não é necessário criar regras para gerar resultados provenientes dos dados de entrada, basta salvar a estrutura da árvore (como é feito pela *scikit-learn*) e a partir dos dados de entrada coletar as saídas provenientes da árvore. Outra vantagem da formulação desse método é que se pode usar qualquer método de árvore implementado em qualquer linguagem, uma vez que o resultado depende apenas dos nós folha, e considerando que o estágio de fuzzificação e defuzzificação ocorrem com funções modulares. Como exemplo, é possível realizar a fuzzificação dos valores das variáveis de entrada e saída, salvá-las num arquivo “.CSV” ou num banco de dados, usá-las para a indução de árvores de decisão que estejam numa linguagem desejada (R, Python, Java, etc) e aplicar a função de defuzzificação nos valores fuzzy da árvore diretamente (caso esteja implementado na mesma linguagem ou sistema), num pipeline ou arquivo de saída (caso esteja em outra linguagem ou sistema).

Resumindo as ideias presentes no método FDT-FTS simples, ao se obter uma árvore de decisão com os atributos fuzzy, realiza-se a previsão dos próximos valores que ocorrerão por meio de uma árvore capaz de gerar uma série fuzzy no estágio intermediário do método. Após a previsão dos valores fuzzy pode-se defuzzificar tais valores e usá-los para prever os valores numéricos de uma série temporal.

### 4.3 FDT-FTS Método Composto

O método FDT-FTS composto realiza uma fuzzificação que considera todos os conjuntos fuzzy ao qual uma instância de uma dada variável tem valor de pertinência maior do que um valor  $\alpha$ . Por exemplo, se uma variável  $X$  possui uma instância  $X_1 = 0.8$ , computa-se quais são seus valores de pertinência levando em consideração os conjuntos fuzzy que dividiram o universo de discurso dessa variável, além de retornar quais são esses conjuntos fuzzy.

Assim, lembrando do exemplo anterior em que  $X_1 = 0.8$  e  $T_1 = 3$  e  $Y_1 = 10$ , nessa nova forma de fuzzificação, teremos não apenas um valor de conjunto fuzzy, mas

$n$  valores dependendo da interseção escolhida entre os conjuntos fuzzy que particionam o universo de discurso. Com isso, suponha que as três instâncias ( $X_1 = 0.8$  e  $T_1 = 3$  e  $Y_1 = 10$ ) irão gerar as seguintes instância de treinamento fuzzyficadas:  $X_{Fuzzy1} = [5, 6]$ ,  $T_{Fuzzy1} = [2, 3]$ ,  $Y_{Fuzzy1} = [7, 8]$ . Com isso, realiza-se todas as combinações possíveis dos conjuntos presentes nas variáveis de entrada e saída.

$$\begin{aligned}
X_{Fuzzy1\_1} &= [5], T_{Fuzzy1\_1} = [2], Y_{Fuzzy1\_1} = [7] \\
X_{Fuzzy1\_2} &= [5], T_{Fuzzy1\_2} = [3], Y_{Fuzzy1\_2} = [7] \\
X_{Fuzzy1\_3} &= [6], T_{Fuzzy1\_3} = [2], Y_{Fuzzy1\_3} = [7] \\
X_{Fuzzy1\_4} &= [6], T_{Fuzzy1\_4} = [3], Y_{Fuzzy1\_4} = [7] \\
X_{Fuzzy1\_5} &= [5], T_{Fuzzy1\_5} = [2], Y_{Fuzzy1\_5} = [8] \\
X_{Fuzzy1\_6} &= [5], T_{Fuzzy1\_6} = [3], Y_{Fuzzy1\_6} = [8] \\
X_{Fuzzy1\_7} &= [6], T_{Fuzzy1\_7} = [2], Y_{Fuzzy1\_7} = [8] \\
X_{Fuzzy1\_8} &= [6], T_{Fuzzy1\_8} = [3], Y_{Fuzzy1\_8} = [8]
\end{aligned} \tag{4.4}$$

No exemplo listado acima, há a expansão das tuplas do conjunto de treinamento num fator de  $2^n$  novas tuplas, como demonstrado em (4.4). Assim, expandindo-se ainda mais o exemplo dado, se o banco de dados ou o arquivo de dados do qual as variáveis  $X$ ,  $T$  e  $Y$  fazem parte contiver 400 tuplas (400 linhas), esse será expandido em  $(2^n) * 400 = 8 * 400 = 3200$ . Assim, é interessante notar para um raciocínio futuro que para cada ponto real da série temporal, nesse exemplo, existem 8 instâncias com atributos fuzzy.

Após essa etapa, os novos dados são separados em dois conjunto, o de treinamento e o de teste, sendo ambos múltiplos de  $2^n$ , no caso do exemplo, múltiplos de 8. Portanto, um conjunto de treinamento/teste aceitável é, respectivamente, (2400,800), o que representa em porcentagem (75%, 25%). Se utilizássemos o conjunto original da série, nas mesmas proporções, teríamos um conjunto de treino/teste de (300,100).

Em adição, feito essa expansão dos dados por meio da fuzzificação, esses são usados para induzir o algoritmo de árvore. Vale salientar que o tamanho da árvore induzida pelo FDT-FTS composto não necessariamente será menor ou maior que as árvores do FDT-FTS simples ou as árvores que sejam induzidas pelo conjunto original da série. Inclusive, os métodos de regularização e poda podem ser aplicados a fim de coibir o crescimento exagerado da árvore e evitar o *overfitting*. (Geron, 2017)

Dando sequência ao exemplo, considere que a árvore será treinada e induzida com o conjunto de 2400 tuplas (como citado anteriormente). Usa-se as 800 tuplas de teste das variáveis  $X_{fuzzy}$  e  $Y_{fuzzy}$  para testar a estrutura da árvore, com isso, teremos 800 valores fuzzy para  $\hat{Y}_{fuzzy}$  (conjunto de valores fuzzy previstos pela árvore de classificação).

Para defuzzificar esses 800 valores, utiliza-se o método “Média dos Máximos”, assim calcularemos a média dos valores máximos de pertinência dos conjuntos fuzzy presentes

na saída da árvore treinada. Não obstante, observa-se que cada 8 instâncias de  $\hat{Y}_{fuzzy}$  (série dos valores fuzzificados) faz referência a uma única instância de  $\hat{Y}_{numerico}$  (série dos valores originais da série), tendo em vista a expansão dos valores anteriormente mostrada. Com isso, cada valor de  $\hat{Y}_{numerico}$  será computado pela média dos máximos de 8 valores de  $\hat{Y}_{fuzzy}$ .

$$x^* = \frac{\sum_{x_i \in M} (x_i)}{|M|}, \text{ considerando que } M = \{x_i \mid \mu(x_i) = h(C)\} \quad (4.5)$$

A “Média dos Máximos” (*MOM - Mean of Maxima* ou *MeOM - Middle of Maxima*) é um dos métodos recorrentemente citado na literatura (Ross, 2009; Saade and Diab, 2004; Chandramohan et al., 2006) e será utilizado na defuzzificação do FDT-FTS composto. Veja na Eq. 4.5 a definição da expressão matemática desse, em que em que  $M$  é o conjunto de pontos com maior valor de pertinência, e  $h(C)$  é a altura do conjunto fuzzy  $C$ . Note que esse realiza, explicitando a grosso modo, a média simples dos valores de  $x_i$ , que promovem os máximos valores de pertinência.

O processo de treinamento das árvores gera um objeto que guarda tanto a estrutura da árvore de decisão ou da *Random Forest*, além das funções que operam essas. Em adição, os valores previstos pela árvore sofrem o processo de defuzzificação pela “Média dos Máximos” produzindo valores numéricos que podem ser comparados com os valores originais da série.

Uma das vantagens adicionais do método FDT-FTS composto, as que foram apresentas no FDT-FTS simples, é a realização de uma média dos valores máximos das regras ativadas. E quanto maior for o número de variáveis de entrada do modelo, tende-se a ter maior número de regras disparadas pelo mesmo. Como é calculada uma média dos máximos, é possível, também, extrair uma medida de variância, com base no conjunto de valores advindos das regras ativadas. Assim, o método FDT-FTS composto possui tanto uma estimativa pontual do valor de uma série temporal (a média), como uma estimativa intervalar (a média e o desvio padrão). A principal desvantagem desse método se traduz no custo computacional para realizar as combinações dos conjuntos fuzzy.

### 4.3.1 Avaliação dos Resultados das Árvores Fuzzy

Pode-se usar métricas de regressão para se avaliar os valores de saída da árvore após a defuzzificação a fim de comparar a série prevista com a real. Em adição, como a classificação de conjuntos fuzzy é parte intermediária do processo de previsão do valor futuro de uma série temporal multivariada, pode-se usar métricas relativas a classificação como uma forma de visualizar a correlação entre a qualidade da classificação e os valores de regressão da série temporal. Por exemplo, pode-se usar uma matriz de confusão a fim de visualizar se os valores estão ou não concentrados na diagonal.

## 4.4 Interpretabilidade e Explicabilidade do Modelo

Primeiramente, vale ressaltar que os modelos de árvore são conhecidos tradicionalmente pela capacidade de criarem regras dada uma árvore induzida. Além disso, a estrutura de uma árvore ou uma base de regras possuem uma interpretação simples e direta dada a forma de inferência, ainda mais quando comparados com modelos complexos existentes, como os de aprendizado profundo.

Embora, as características de interpretabilidade e explicabilidade sejam evidentes nos modelos de árvores, existe uma gama de aplicações implícitas a esses modelos, e que serão exploradas neste trabalho.

Primeiramente, note que ao gerar regras para prever um valor de uma série (como mostrado nos métodos FDT-FTS simples e FDT-FTS composto) existem regras específicas (ou caminhos específicos na árvore) os quais caracterizam determinados pontos ou regiões de uma série. Dessa maneira, é possível separar as regras ou caminhos da árvore em grupos, de acordo com o nível que os valores ocorrem nas séries. Veja na figura 21 um exemplo em que podemos separar grupos de regras a depender das regiões de uma série temporal. Ou seja, é possível usar regras para prever movimentos da série, mas também usar essas para compreender o que está associado a ocorrência dos movimentos. Em adição, pode ser interessante, em certos casos, prever apenas determinados movimentos de uma série, por exemplo, importa saber apenas os períodos de baixa de um rio a fim de prever quando irá faltar água, ou importa prever apenas a alta para que se evite alagamentos.

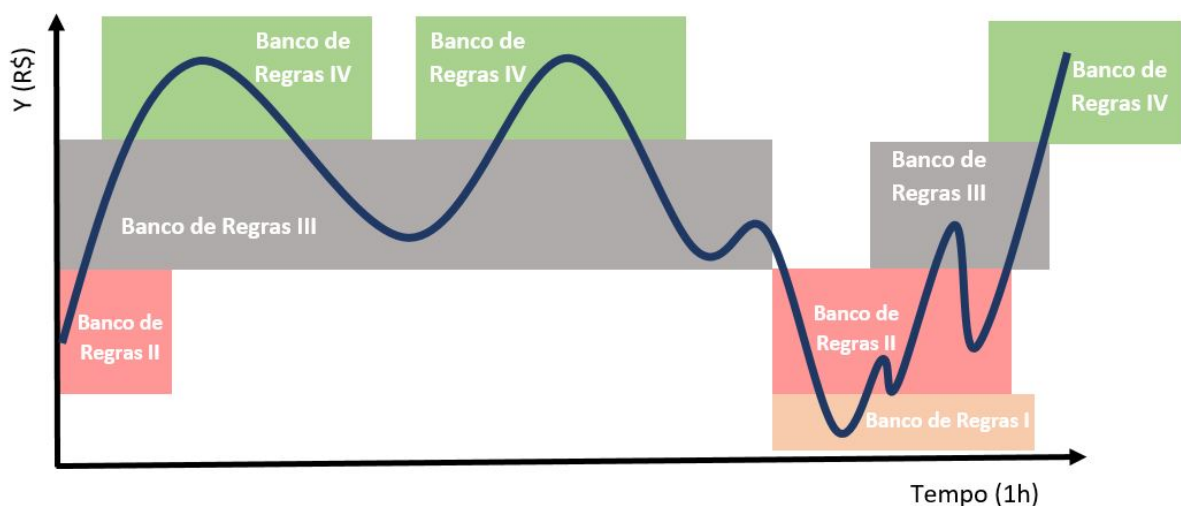


Figura 21 – Discriminação das regras por conjunto de valores da série.

Fonte: Autor

Outra interessante consequência do modelo de árvores, é que o antecedente de uma regra reflete as condições que influem na ocorrência de um valor da série (representado como consequente). Esse processo pode ser bem útil no que tange a descoberta de conhecimento



e da investigação quanto a relação que as variáveis externas exercem no comportamento das séries principais no tempo.

Como exemplo, imagine uma regra que reflita o seguinte cenário quanto a situação volumétrica num rio:

*Dado que em épocas de calor as temperaturas são maiores que um valor maior que  $X$ , dado que a presença de determinadas espécies aumentaram sua população num rio de um valor maior que  $P$ , dado que o volume de precipitação na nascente do rio foi de um valor menor que  $Z$ , então: volume do rio igual a um valor  $Y$ .*

Veja a representação dessa situação por meio de uma regra na Eq. (4.6)

$$\text{Temperatura} > X, \quad \text{Pop\_Rio} > P, \quad \text{Vol\_nascente} < Z, \quad \text{logo:} \quad V\_rio = Y \quad (4.6)$$

É interessante notar como as regras são de simples interpretação e que essas podem ser traduzidas em termos linguísticos como mostrado. Além disso, veja que, nesse exemplo, usa-se uma regra em que temos os operadores ( $>$  e  $<$ ) no antecedente da regra e que essa não é uma regra fuzzy, por não ter o formato em que as variáveis tem um grau de pertencimento ou não a um grupo fuzzy. Não obstante, existem implementações de árvores de decisão que irão prover regras semelhantes as regras fuzzy (variável pertence a um grupo) e outras irão ter regras utilizando os operadores mostrados. Nesse ponto, é válido salientar uma das vantagens desse método, em que se considera apenas o consequente da regra para a defuzzificação. Assim, as regras que apresentam os operadores matemáticos ( $>=$ ,  $<=$ ,  $>$  e  $<$ ) podem ser usadas. Isso pode ser utilizado para se abordar novas interpretações das regras, em que ao invés de se atribuir um conjunto fuzzy a uma variável, atribuí-se um intervalo em que podem ocorrer vários conjuntos fuzzy.

Por fim, como é utilizado critérios de impureza para a indução da árvore, pode-se verificar, por exemplo, quais variáveis possuem maior nível de ganho de informação a fim de verificar quais delas são mais relevantes ao problema.

## 4.5 Considerações Finais - Capítulo 4

Em suma, foi apresentado neste capítulo o modelo FDT-FTS em que por meio das técnicas de fuzzificação e defuzzificação de séries temporais é possível induzir qualquer modelo de árvore de decisão bem como métodos *ensemble*. Com isso, torna-se possível analisar e prever séries temporais multivariadas no tempo, além de buscar uma explicação para os seus movimentos no tempo. O próximo capítulo irá mostrar as aplicações desse método em cinco séries temporais com características distintas.



## Capítulo 5

# Experimentos Computacionais

“Em nossos raciocínios a respeito dos fatos, existem todos os graus imagináveis de certeza. Um homem sábio, portanto, ajusta sua crença à evidência.”

---

David Hume

Neste capítulo apresentaremos cinco conjuntos de dados que possuem diferentes problemas quanto a séries temporais multivariadas. No caso, usaremos os dois métodos (FDT-FTS simples e composto) para prever as séries principais de cada conjunto. Apresentaremos em cada seção o problema referente a cada conjunto de dados, discriminando o significado das variáveis, a análise exploratória dos dados, os resultados, assim como a discussão dos pontos relevantes dos resultados. Vale salientar que não se pretende adentrar de forma rigorosa nas propriedades de cada série, mas demonstrar de forma objetiva a aplicação dos métodos propostos em séries com diferentes características.

Será usado nesse trabalho os algoritmos de árvores de decisão presentes na biblioteca *Scikit-learn*. No caso utilizaremos a função *DecisionTreeClassifier*, a qual é uma função otimizada para gerar árvores de decisão e pode ser parametrizada para apresentar variadas propriedades, como é o caso do parâmetro *criterion* em que se pode escolher a “Entropia” ou o “Índice Gini” como critérios para o particionamento dos nós internos. Essa função foi concebida de forma a ser otimizada em vários aspectos, um deles trata de criar as árvores com estruturas binária (semelhante a CART), uma vez que essas são menos complexas computacionalmente de serem criadas. Ressalta-se que algumas das propriedades não são originais dos algoritmos CART e ID3, mas os pontos mudados em relação a esses na *DecisionTreeClassifier* foram feitos para dar rapidez ao treinamento e melhorar sua acurácia. Assim, apesar do cruzamento das diversas propriedades de diversas árvores estarem presentes na função *DecisionTreeClassifier*, consideraremos como fator

principal o critério de impureza para especificar as árvores, no caso do uso da “Entropia” consideraremos como ID3, e para o uso do “Índice Gini” a árvore CART. Dessa maneira, ao apresentar os resultados utilizaremos essa distinção quanto as árvores treinadas (Pedregosa et al., 2011; Geron, 2017; Buitinck et al., 2013).

Em adição, o trabalho visa explorar o uso diversificado dos métodos FDT-FTS simples e composto, não se aprofundando em questões de otimização dos hiperparâmetros relacionados ao particionamento dos conjuntos fuzzy e aos tipos de funções de pertinência utilizadas. Para isso, poderia-se conceber algoritmos genéticos ou usar heurísticas presentes na *Scikit-Learn* para buscar por esses hiperparâmetros. No caso das árvores, existe a vantagem de que para qualquer conjunto de variáveis de entrada, ela buscará (durante o treinamento) a melhor estrutura de árvore que particione o conjunto de dados de entrada a fim de maximizar a acurácia.

Será, também, computado o resultado da função *DecisionTreeRegressor* (DTR) presente na biblioteca *Scikit-Learn*. Utilizaremos tal método uma vez que esse usa dados numéricos ao invés de valores fuzzy e possui parametrização semelhante as árvores de decisão de classificação utilizadas. Além disso, esse modelo possui, também, explicabilidade e interpretabilidade por criar um modelo baseado em regras, bem como os modelos de árvores de classificação que utilizam os valores fuzzy para realizar a previsão de séries. Assim, para esse algoritmo, também, serão computados os valores de MAPE, SMAPE, RMSE e MAE. Adicionalmente, existirá uma seção para cada experimento computacional em que se apresentará os resultados desse método de comparação em conjunto com os métodos de árvore fuzzy. Vale frisar que os únicos parâmetros que serão controlados nos experimentos das árvores fuzzy serão o nível de profundidade (hiperparâmetro conhecido como *max\_depth*), o número de árvores presentes na *Random Forest* (hiperparâmetro conhecido como *n\_estimators*) e o hiperparâmetro que guarda o estado aleatório do experimento para que esse tenha repetibilidade (conhecido como *random\_state*).

Por fim, os dados extraídos tanto dos métodos FDT-FTS quanto da DTR serão apresentados na seção de resultados. Além disso, no caso do experimento computacional do Rio Xingu iremos comparar também com os resultados presentes no trabalho desenvolvido em Vieira et al. (2021).

## 5.1 QoS (Qualidade do Serviço)

Iniciaremos a demonstração dos métodos analisando uma base de dados relacionada a qualidade de serviço (*QoS - Quality of Service*). *QoS* diz respeito as tecnologias e técnicas responsáveis por garantir o desempenho dos serviços numa rede relacionado as métricas de taxas de transferências, perda de pacotes, flutuações (*Jitter*), atraso na entrega de pacotes, entre outros. Garantindo-se um alto grau de confiabilidade e desempenho a um tipo de

serviço de rede (ex.: Computação em nuvem), garante-se, também, a confiabilidade e desempenho de outros sistemas dependentes desse e a satisfação do usuário final (Ferguson and Huston, 1998).

Os dados coletados são provenientes do trabalho realizado por Zhang et al. (2017), os quais estão disponíveis publicamente no seguinte sítio eletrônico:

“<https://sourceforge.net/projects/qosmonitoring/files/>”, Acessado em: 10/2021.

Tal base de dados conta com quatro arquivos .CSV, relativos a servidores *Web Services* distintos. Esses possuem como variáveis “Tempo de Resposta” (*Response Time*) e a “Taxa de Transferência” (*Throughput*). Assim, usa-se dessas duas variáveis para se prever o valor futuro do “Tempo de Resposta”. Os dados de cada um desses arquivos foram coletados durante três meses enter os horários de 8h00 às 17h00 e o período de coleta de cada ponto é de 15 minutos. Os arquivos possuem um total, cada um, de 2000 pontos.

Veja na Tab. 1 as variáveis que utilizaremos em nosso experimento.

| Descrição das Variáveis de Entrada e Saída utilizadas |  |
|---|--|
| Variáveis   | Informação dos Atributos                                   |
| Tempo de Resposta                                     | Variável principal a ser prevista.                         |
| Tempo de Resposta Defasado (t-1)                      | Tempo de Resposta atrasado em 1 unidade temporal           |
| Tempo de Resposta Defasado (t-2)                      | Tempo de Resposta atrasado em 2 unidades temporais         |
| Tempo de Resposta Defasado (t-3)                      | Tempo de Resposta atrasado em 3 unidades temporais         |
| Taxa de Transferência(t-1)                            | Taxa de Transferência atrasada em uma unidade temporal.    |
| Taxa de Transferência Defasada (t-2)                  | Taxa de Transferência atrasada em duas unidades temporais. |

Tabela 1 – Variáveis utilizadas no problema de previsão do tempo de resposta para o serviço de *QoS*.

No caso, para mostrar o uso dos métodos desse trabalho para previsão multivariada do tempo de resposta, usaremos apenas um dos arquivos (“webservice1.txt”), uma vez que a partir da demonstração desse, pode-se replicar para os outros os mesmos passos.

Nesta análise, todas as série foram fuzzificadas com funções triangulares e foram particionadas em 30 conjuntos fuzzy. Além disso, as configurações das árvores e do método *Random Forest Fuzzy* são as seguintes:

- **ID3** : Max\_Depth=15 e Random\_State=0
- **CART**: Max\_Depth=15 e Random\_State=0

- **Random Forest:** `n_estimators=50` e `Random_State=0`
- **DTR:** Parâmetros padrões da “Scikit-Learn”.

### 5.1.1 Caracterização dos Dados

A caracterização dos dados se dá por meio da visualização inicial das propriedades de autocorrelação, o comportamento da série no tempo e por meio dos resultados do teste de hipótese quanto a estacionariedade da série. Vejamos a seguir algumas dessas análises.

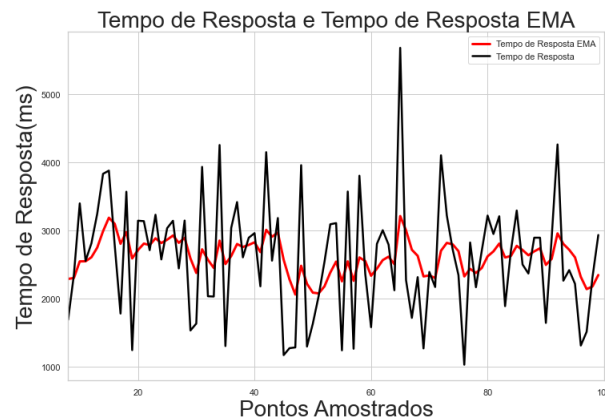


Figura 22 – Série original do tempo de resposta e tempo de resposta utilizando um filtro de média móvel exponencial com  $p=8$ .

Fonte: Autor

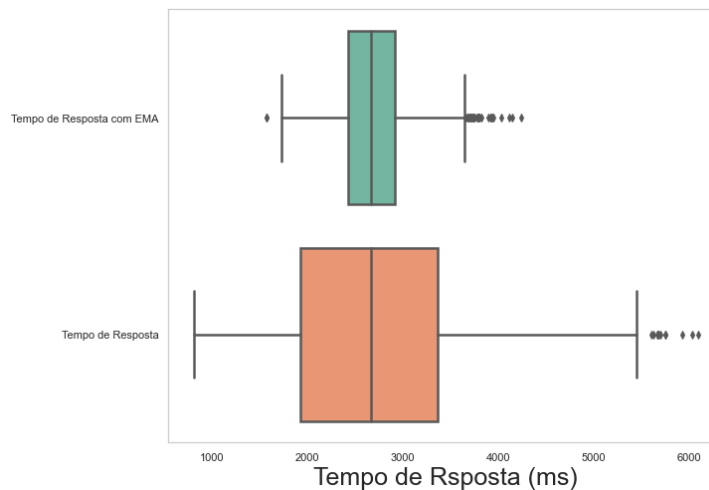


Figura 23 – *Boxplot* da série original de tempo de resposta e da série filtrada.

Fonte: Autor

Veja na Fig. 22, em que se demonstra tanto a série original do tempo de resposta, como essa série filtrada por uma média móvel exponencial com período de 8. A estratégia do uso de filtros para melhorar a previsão da série foi adotada por (Zhang et al., 2017), uma

vez que a série do tempo de resposta possui um ruído de amplitude considerável. Nota-se, também, na Fig. 22 e na Fig. 23 que há uma suavização na série do tempo de resposta e um decaimento na amplitude ao se usar o filtro de média móvel exponencial, principalmente nos níveis mais extremos da série. Essa suavização permite que os algoritmos de séries temporais não sejam influenciados pelas variações bruscas e pelas anomalias da série.

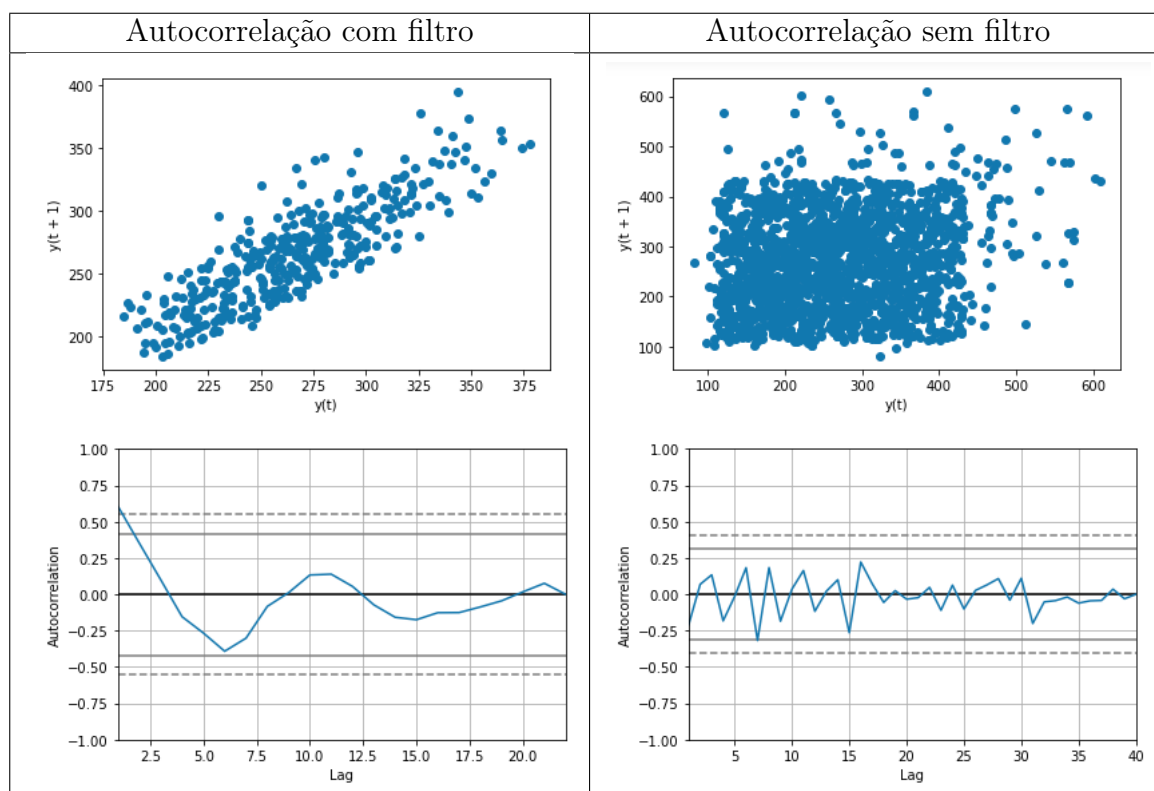


Tabela 2 – Análise da autocorrelação da série de *QoS* com e sem o uso de filtro.

Fonte: Autor

Veja na Tab. 2 a diferença entre a série filtrada e a série sem filtro no que tange a análise de autocorrelação. Nota-se que a série sem o uso do filtro não apresenta nenhum indício de autocorrelação, já a série filtrada sim. Com isso, o uso do filtro é necessário a fim de permitir que os algoritmos de previsão sejam fortemente influenciados pela correlação entre os pontos da série temporal, ao invés de serem influenciados pelo ruído. Em adição, por meio do teste de hipótese “Dickey-Fuller Aumentado” constatou-se que a série de *QoS* é uma série estacionária.

### 5.1.2 Resultados

Mostra-se alguns resultados relativos as previsões do método de FDT-FTS simples e composto com a finalidade de discuti-los posteriormente, além dos resultados da *DecisionTreeRegressor* (DTR). Os gráficos da Tab. 3 apresentam as previsões da série temporal proveniente dos métodos ID3 Fuzzy, CART Fuzzy e *Random Forest Fuzzy* para o método

composto. Expõe-se, também, na Fig. 24 quais foram os intervalos de treino e teste usados para a indução das árvores. Já na Tab. 3, demonstra-se as previsões intervalares realizadas pelas saídas das árvores do método FDT-FTS composto.

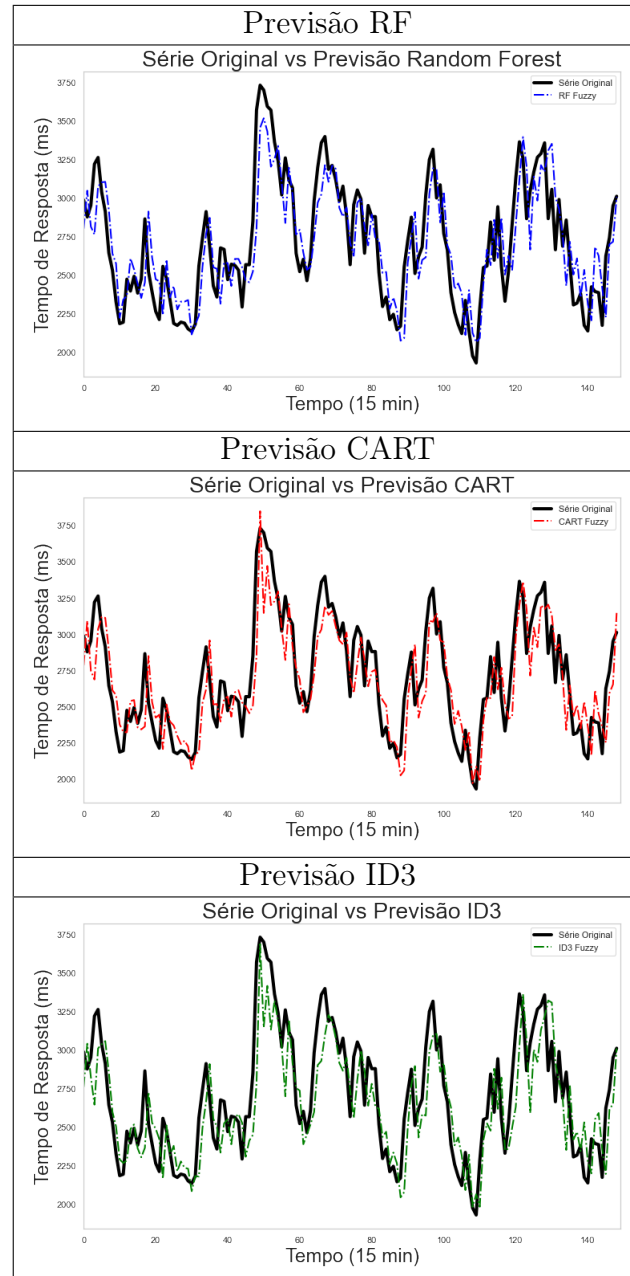


Tabela 3 – Gráficos da série original temporal e as previsões dos métodos de árvore utilizando o método FDT-FTS composto para a série de QoS.

Fonte: Autor

Computa-se a relevância que cada atributo tem para os métodos de árvore a partir dos critérios de impureza que cada método utiliza, como mostrado na Tab. 4. Esses resultados são relevantes, uma vez que explicitam quais as variáveis de entrada são relevantes para a previsão da série temporal.

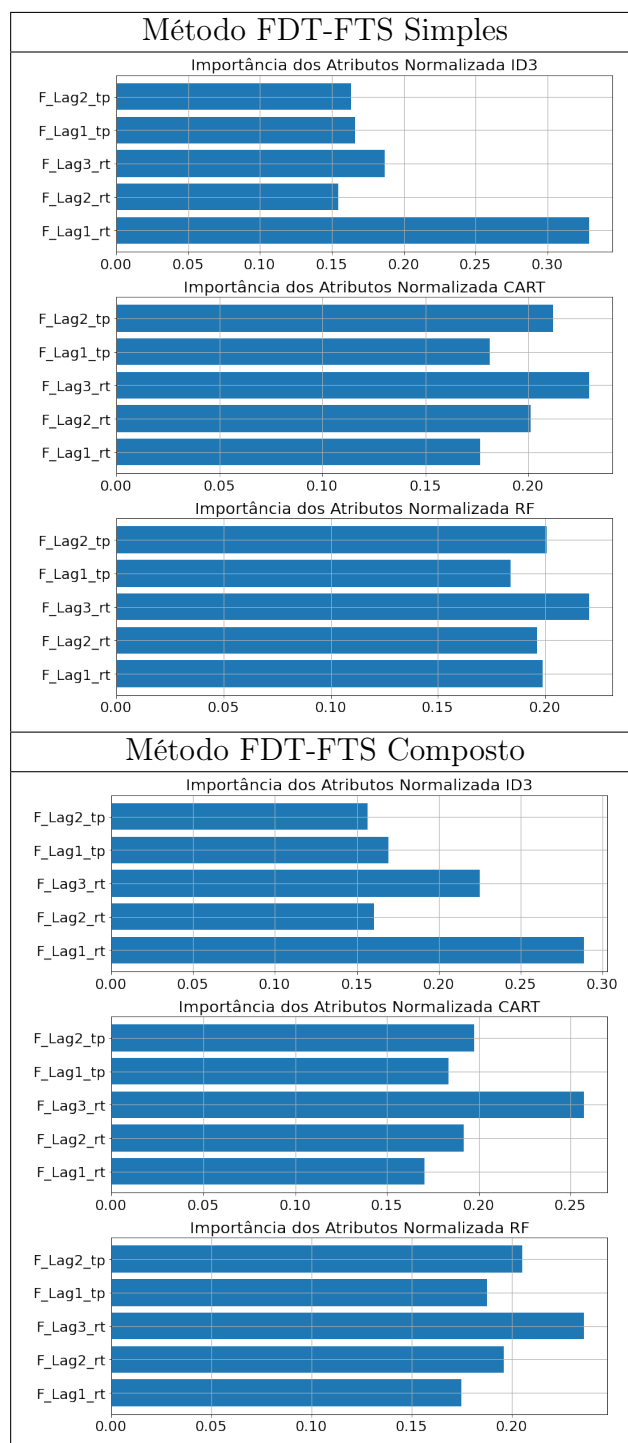


Tabela 4 – Comparação da importância entre as variáveis de entrada do experimento da série de *QoS*.

Fonte: Autor

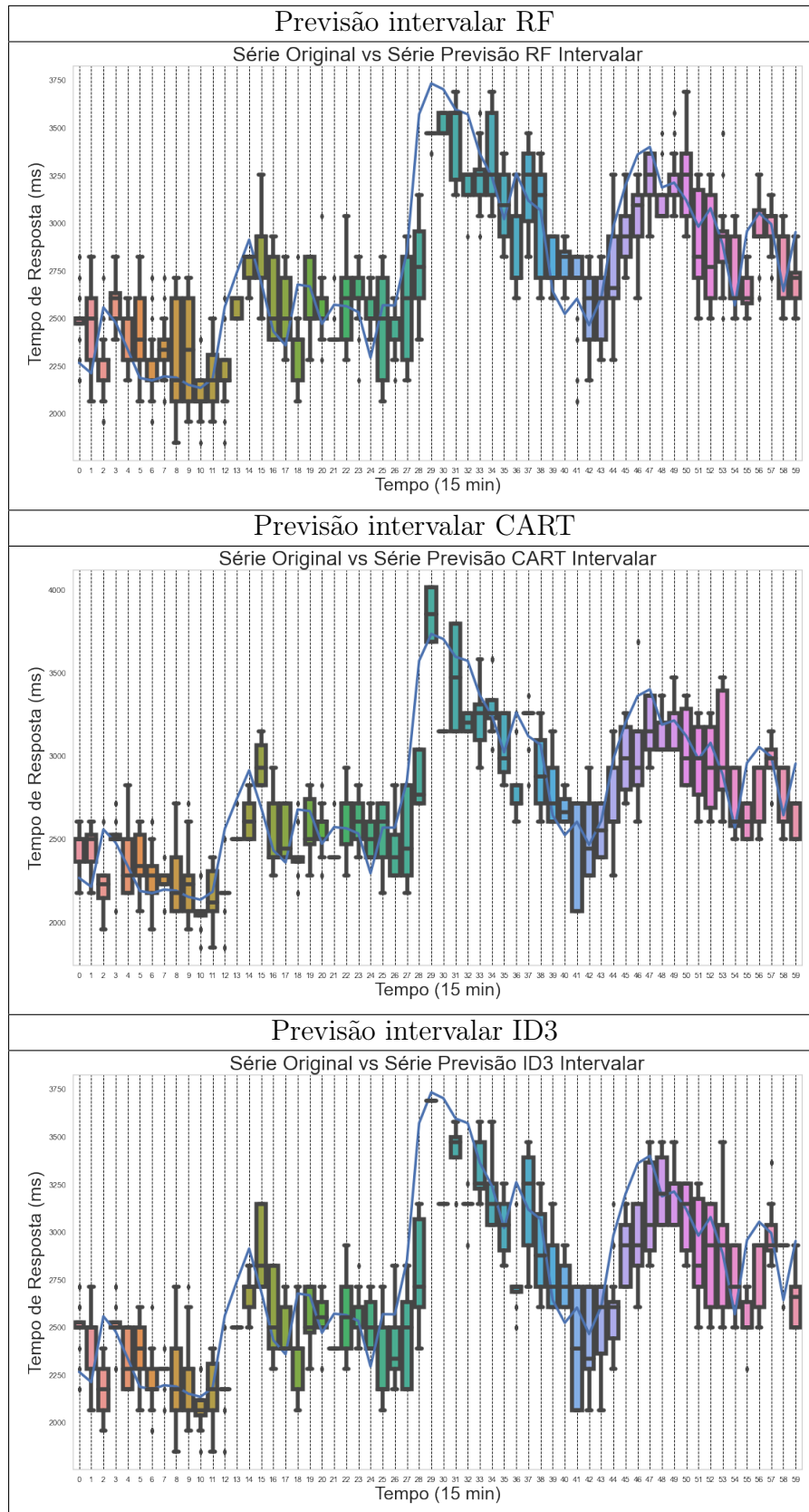


Tabela 5 – Gráficos da série original e as previsões intervalares do método de FDT-FTS composto para a série de QoS.

Fonte: Autor



Apresenta-se na Tab. 5 os gráficos de previsão intervalar para os três métodos de árvore do modelo FDT-FTS composto, tendo em vista que a previsão do FDT-FTS composto é um grupo de valores fuzzy, é possível computar valores dos intervalos da saída da árvore relativos para um único ponto da série temporal. Em adição, os valores das métricas utilizadas para expressar a qualidade dos métodos FDT-FTS simples e composto estão expressos nas Tab. 6 e Tab. 7, além do método DTR na Tab. 8.

## Intervalos de Treinamento:

[[0, 1010], [4, 1014], [12, 1022], [24, 1034], [40, 1050], [60, 1070], [84, 1094], [112, 1122], [144, 1154], [180, 1190], [220, 1230], [264, 1274], [312, 1322], [364, 1374], [420, 1430], [480, 1490], [544, 1554], [612, 1622], [684, 1694], [760, 1770]]

## Intervalos de Teste:

[[1011, 1160], [1015, 1164], [1023, 1172], [1035, 1184], [1051, 1200], [1071, 1220], [1095, 1244], [1123, 1272], [1155, 1304], [1191, 1340], [1231, 1380], [1275, 1424], [1323, 1472], [1375, 1524], [1431, 1580], [1491, 1640], [1555, 1704], [1623, 1772], [1695, 1844], [1771, 1920]]

Figura 24 – Intervalos de treino e teste utilizados no experimento.

Fonte: Autor

| Métodos | Métricas para o FDT-FTS Simples (Média ± Desvio Padrão) |              |               |               |
|---------|---|--------------|---------------|---------------|
|         | MAPE  | SMAPE        | RMSE          | MAE           |
| ID3     | 8.91% ±0.54%  | 8.84% ±0.53% | 296.81 ±19.57 | 238.34 ±16.80 |
| CART    | 8.89% ±0.47%  | 8.91% ±0.46% | 296.99 ±17.40 | 239.23 ±14.39 |
| RF      | 8.67% ±0.43%  | 8.69% ±0.43% | 285.73 ±17.21 | 232.74 ±15.13 |

Tabela 6 – Métricas utilizadas na série de *QoS* com o filtro de média móvel exponencial aplicando-se o método FDT-FTS simples.

| Métodos | Métricas para o FDT-FTS Composto (Média ± Desvio Padrão) |              |               |               |
|---------|--|--------------|---------------|---------------|
|         | MAPE   | SMAPE        | RMSE          | MAE           |
| ID3     | 7.31% ±0.37%   | 7.42% ±0.36% | 243.43 ±15.62 | 198.69 ±14.94 |
| CART    | 7.28% ±0.39%   | 7.37% ±0.40% | 240.02 ±16.69 | 197.45 ±16.65 |
| RF      | 7.34% ±0.44%   | 7.32% ±0.42% | 228.14 ±17.69 | 196.33 ±15.83 |

Tabela 7 – Métricas utilizadas na série de *QoS* com o filtro de média móvel exponencial aplicando-se o método FDT-FTS composto.

| Métodos | Métricas para a DTR (Média ± Desvio Padrão) |              |               |               |
|---------|---|--------------|---------------|---------------|
|         | MAPE  | SMAPE        | RMSE          | MAE           |
| DTR     | 8.951% ±0.65%                               | 8.89% ±0.65% | 294.38 ±21.74 | 238.50 ±20.80 |

Tabela 8 – Métricas utilizadas na série de *QoS* com o filtro de média móvel exponencial aplicando-se o método DTR.

Na Tab. 9, o resultado das métricas usadas para avaliar a qualidade da previsão das séries temporais, bem como o resultado do algoritmo DTR em formato de *boxplot*.

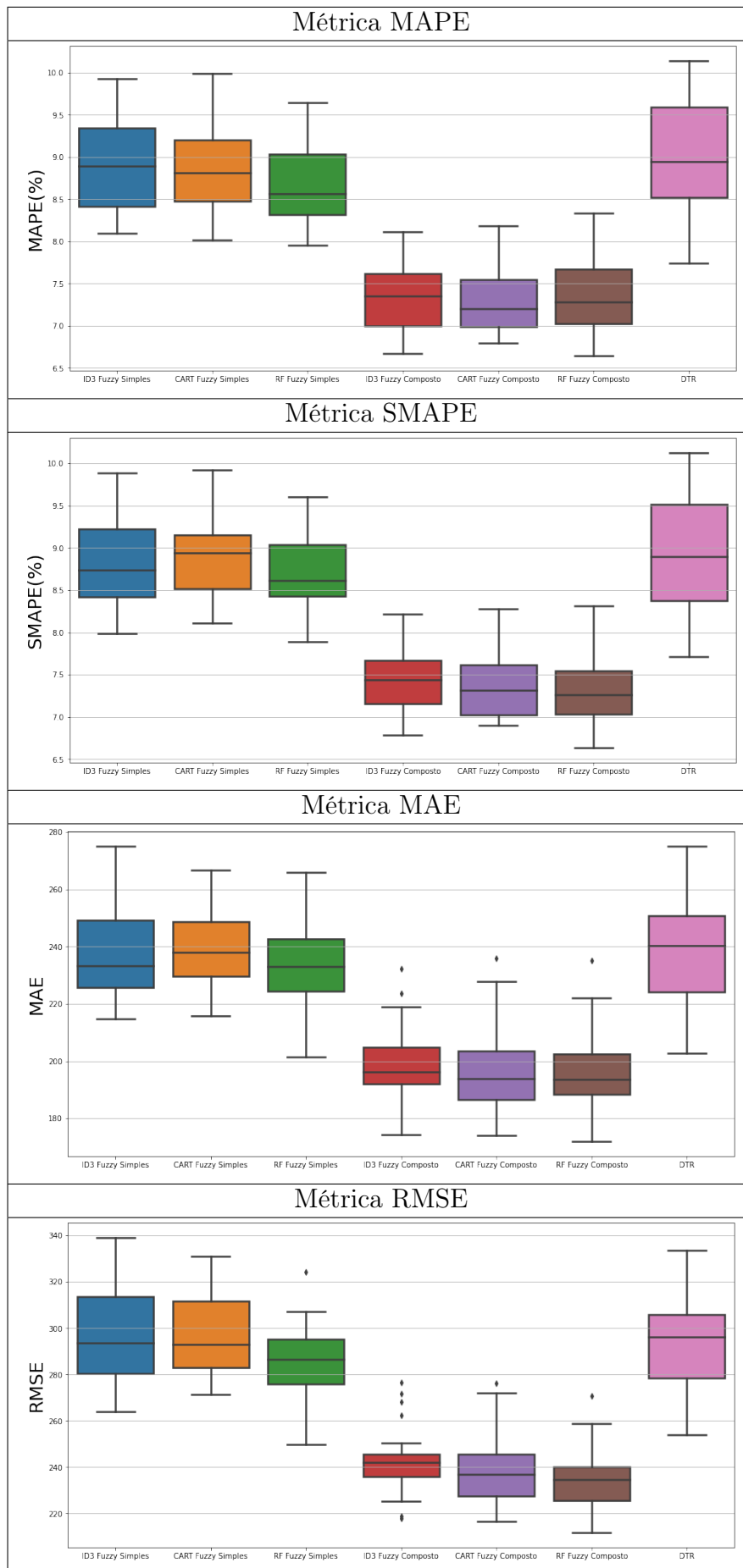


Tabela 9 – Métricas expressas em *boxplots* para o método FDT-FTS simples e composto, além do método de comparação DTR para a série de *QoS*.

### 5.1.3 Discussão dos Resultados

Primeiramente, vale salientar que a série do tempo de resposta como mostrada na Fig. 22 é uma série com movimentos ruidosos atenuados pelo filtro. No caso, tanto as árvores do método simples quanto do composto apresentaram resultados próximos quando se compara as quatro métricas (MAPE, SMAPE, RMSE e MAE). O método FDT-FTS composto, neste caso, apresentou resultados médios melhores em relação aos métodos da FDT-FTS simples e a DTR, como presente nas Tab. 6, Tab. 7 e Tab. 8. Veja que na Tab. 5 se evidencia as várias saídas ativadas do método FDT-FTS composto para prever um único ponto da série, obtendo-se uma série intervalar que representa essas saídas em formato de *boxplot*. Nota-se, também, na Tab. 5 que boa parte dos pontos da série tem interseção ou está próximo dos intervalos. Como essa é uma série com movimentos abruptos, sem sazonalidade e com níveis de amplitude mal definidos, uma das vantagens do método composto é que suas saídas são ponderadas pela média, sendo a média influenciada pela região em que há maior número de regras ativadas.

Outro fato relevante é que as árvores de decisão como a CART Fuzzy, ID3 Fuzzy e a RF Fuzzy compostas apresentaram resultados semelhantes para as várias métricas, mostrando que uma árvore simples teve resultados próximos a uma floresta de árvores. É interessante avaliar, também, que o desvio padrão foi menor para os algoritmos de árvore gerados pelo FDT-FTS composto.

Por fim, nota-se na Tab. 4 que as séries atrasadas tanto do tempo de resposta quanto da taxa de transferência foram uma boa escolha de entrada para as árvores, já que não houve nenhum atributo com valor de importância nulo ou muito destoante dos outros. A árvore ID3 Fuzzy, gerada pelo método FDT-FTS composto, apresentou para a série do tempo de resposta atrasada em  $t - 1$  maior valor de importância de forma mais acentuada que as outras, resultado esse condizente com a análise de autocorrelação da série.

## 5.2 Previsão do Nível do Rio Xingu

No mundo, muitas comunidades têm sofrido com problemas relativos a alagamentos tendo em vista os danos sociais e econômicos causados por esses. Estima-se que o Brasil tenha perdido com secas, queimadas e enchentes cerca de 3,66 bilhões de dólares no ano de 2020 como apontou o relatório da AON <sup>1</sup>. Assim, ao lidarmos com o problema de previsão dos níveis de rios, tem-se um tema de ampla relevância no que tange a capacidade de tomar decisões a fim de mitigar ou evitar situações catastróficas.

A presente seção trará a base de dados usada no trabalho desenvolvido em [Vieira et al. \(2021\)](#). Esse conjunto de dados fornece informações quanto ao nível do rio Xingu,

<sup>1</sup> <https://static.poder360.com.br/2021/05/relatorio-de-analise-de-clima-e-catastrofe-aon-2020.pdf>, Acessado em: Outubro de 2021.

bem como um conjunto de outras séries que possuem relação com as variações do nível do rio. Assim, a finalidade desta parte do trabalho visa conceber um modelo de previsão para o nível do rio Xingu baseado em variáveis externas, por exemplo, atributos climáticos de chuva, humidade, pressão, etc. Dessa maneira, o problema do qual trataremos mostrará como os métodos FDT-FTS simples e composto podem ser utilizados para conceber um modelo de previsão da série de máxima do rio Xingu.

É válido ressaltar que, neste caso, aplicaremos o FDT-FTS simples e o usaremos para selecionar as melhores variáveis de entrada para o modelo FDT-FTS composto. Assim, o FDT-FTS simples servirá como um algoritmo de seleção de atributos para o método FDT-FTS composto. Após, o resultado dos dois serão computados e comparados com os dados do artigo e com o algoritmo DTR. Além disso, foram utilizados como entradas extras as séries atrasadas do nível do rio (série principal) em  $t - 1$  e  $t - 2$ .

O conjunto de dados apresenta 18 variáveis distintas, as quais podem ser listadas e explicadas brevemente pela Tab. 10.

| Descrição das Variáveis de Entrada e Saída utilizadas |  |
|---|--|
| Variáveis   | Informação dos Atributos                               |
| Nino 1_2  | Temperatura da superfície (°C) em Niño 1+2.            |
| Nino 3  | Temperatura da superfície (°C) em Niño 3.              |
| Nino 4  | Temperatura da superfície (°C) em Niño 4.              |
| Nino 3_4  | Temperatura da superfície (°C) em Niño 3.4.            |
| Atlântico Norte                                       | Temperatura da superfície (°C) no Atlântico Norte.     |
| Atlântico Sul   | Temperatura da superfície (°C) no Atlântico Norte.     |
| Atlântico Tropical                                    | Temperatura da superfície (°C). no Atlântico Tropical. |
| Pressão Darwin  | Pressão atmosférica de Darwin (hPa).                   |
| Pressão Tahiti  | Pressão atmosférica de Tahiti (hPa).                   |
| PRP Xingu al01  | Precipitação na sub-bacia 1 do rio Xingu (mm).         |
| PRP Xingu al02  | Precipitação na sub-bacia 2 do rio Xingu (mm).         |
| PRP Xingu al03  | Precipitação na sub-bacia 3 do rio Xingu (mm).         |
| PRP Xingu al04  | Precipitação na sub-bacia 4 do rio Xingu (mm).         |
| PRP Xingu al05  | Precipitação na sub-bacia 5 do rio Xingu (mm).         |
| PRP Xingu al06  | Precipitação na sub-bacia 6 do rio Xingu (mm).         |
| PRP Xingu al07  | Precipitação na sub-bacia 7 do rio Xingu (mm).         |
| PRP Xingu al08  | Precipitação na sub-bacia 8 do rio Xingu (mm).         |
| PRP Xingu al09  | Precipitação na sub-bacia 9 do rio Xingu (mm).         |
| Máxima  | Série do nível do rio Xingu (cm).                      |
| Máxima (t-1)  | Série atrasada em $t - 1$ do nível do rio Xingu (cm).  |
| Máxima (t-2)  | Série atrasada em $t - 2$ do nível do rio Xingu (cm).  |
| Mês   | Meses do ano utilizado na série.                       |

Tabela 10 – Variáveis utilizadas para a previsão da série temporal do Rio Xingu.

Nesta análise, todas as variáveis foram fuzzificadas com funções triangulares e foram particionadas em 20 conjuntos fuzzy, com exceção da série do “Mês” que foi

particionada em 12 conjuntos. Além disso, as configurações das árvores e do método *Random Forest Fuzzy* são as seguintes:

- **ID3** : Max\_Depth=15 e Random\_State=0
- **CART**: Max\_Depth=15 e Random\_State=0
- **Random Forest**:n\_estimators=20 e Random\_State=0
- **DTR**: Parâmetros padrões da “Scikit-Learn”.

### 5.2.1 Caracterização dos Dados

Inicialmente, visualiza-se a distribuição dos dados, assim como a série temporal de máxima nas Fig. 25 e Fig. 26. Adicionalmente, foi computado o gráfico de autocorrelação da série de máxima.

Nota-se pela Fig. 26 que a série de máxima está concentrada em torno de 300 mm e 700 mm, fato que ao ser conflitado com os dados da série temporal mostra que a maior parte do tempo os valores do nível do rio estão ao redor desses dois valores. Outro fato relevante é a autocorrelação presente na série temporal explicita na Fig. 26. Por fim, constatou-se pelo teste de hipótese “Dickey-Fuller Aumentado” que a série de máxima é estacionária.

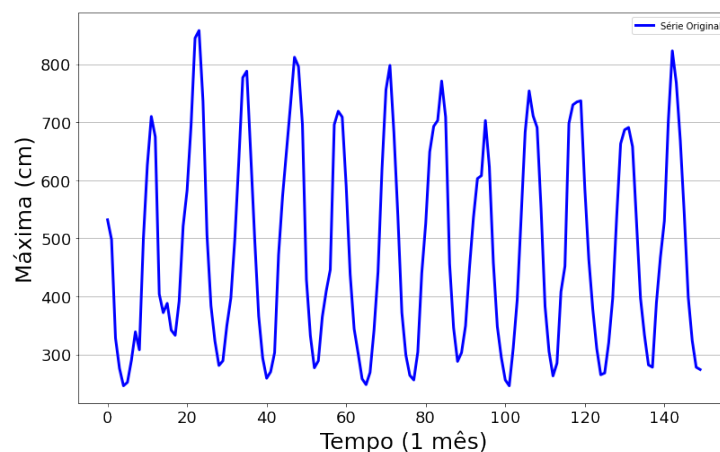


Figura 25 – Série original da máxima do rio Xingu.

Fonte: Autor

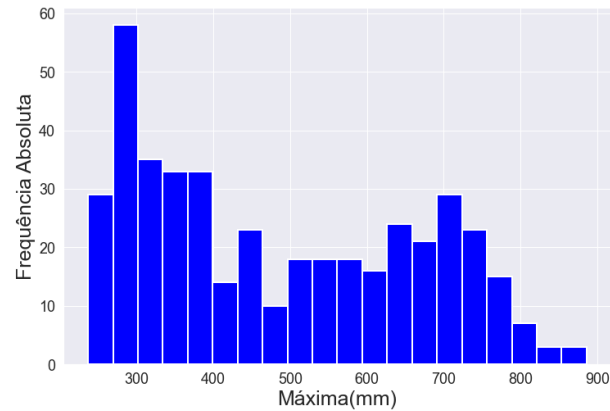


Figura 26 – Histograma da série original da máxima do rio Xingu.

Fonte: Autor

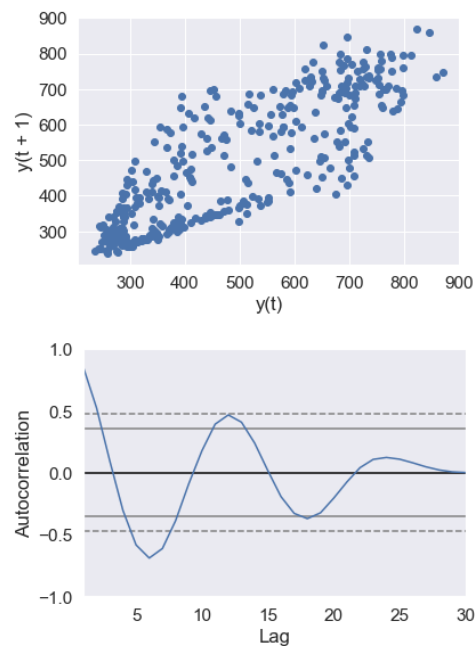


Figura 27 – Autocorrelação da série original do rio Xingu.

Fonte: Autor

## 5.2.2 Resultados

Primeiramente, exibe-se os intervalos de treino e teste das janelas temporais na Fig. 28. Logo após, expõe-se a série original em conjunto com a série prevista na Tab. 11, assim como a série intervalar prevista pelo método FDT-FTS composto na Tab. 13. Por fim, é mostrado os resultados das métricas comparadas de MAPE, SMAPE, RMSE, MAE e o coeficiente de determinação ( $R^2$ ) nas Tab. 14, Tab. 15, Tab. 16, Tab. 17. Aborda-se, especialmente neste experimento, o coeficiente de determinação a fim de comparar com os resultados presentes em [Vieira et al. \(2021\)](#).

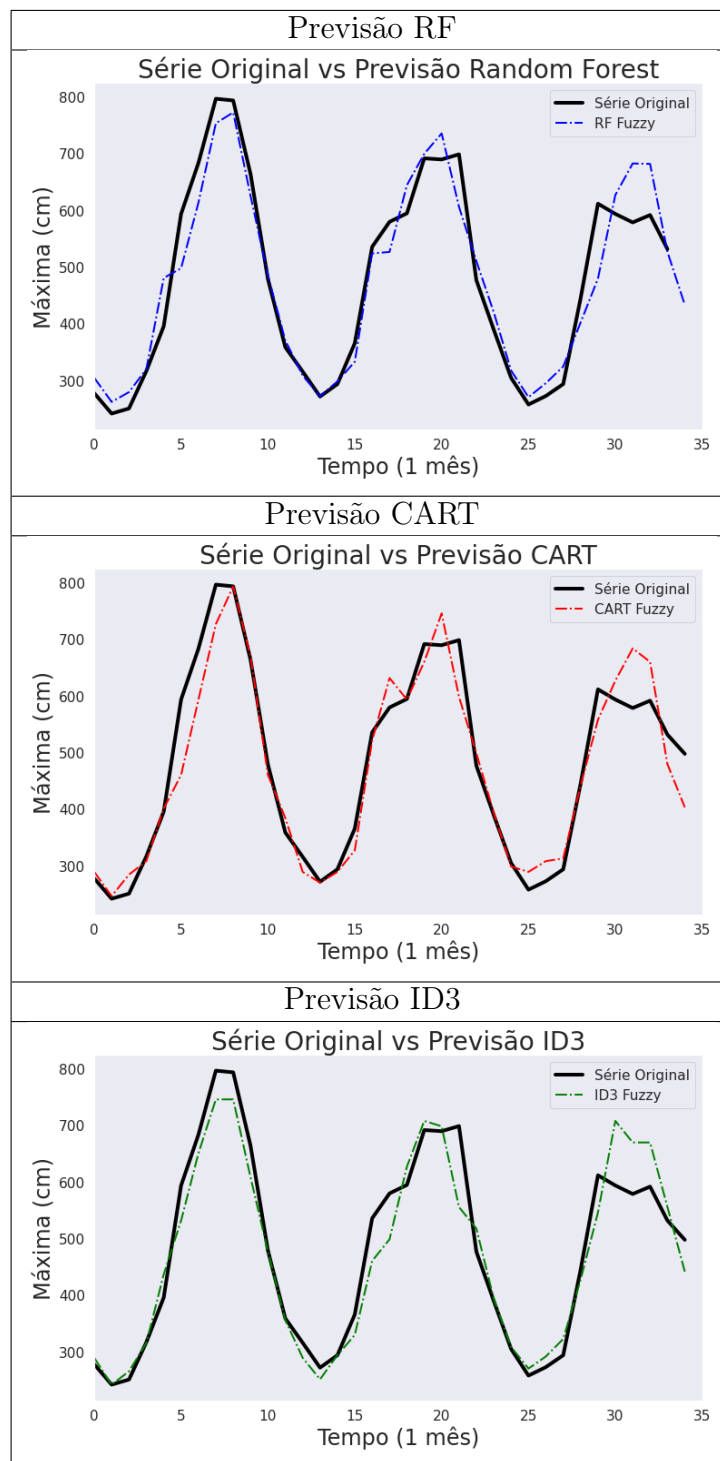


Tabela 11 – Gráficos da série de máxima e as previsões dos métodos de árvore utilizando a FDT-FTS composta para a série de máxima do Rio Xingu.

Fonte: Autor

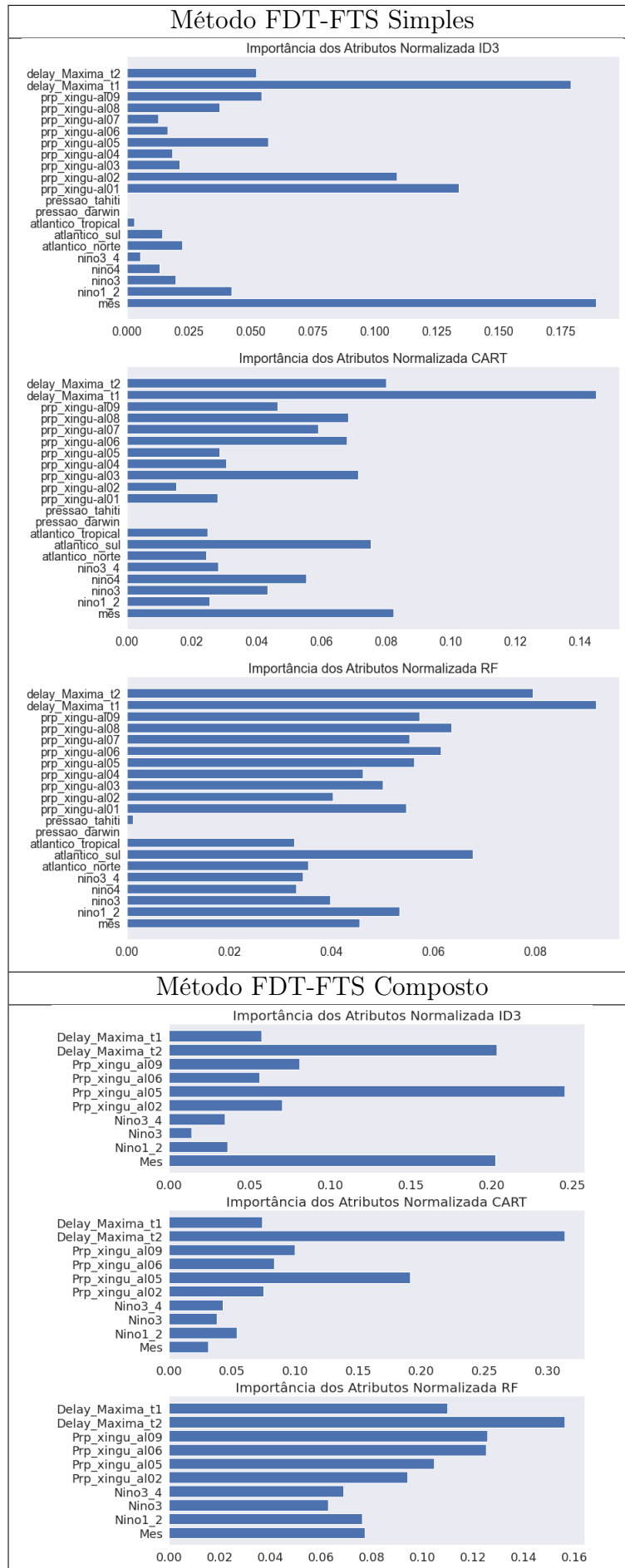


Tabela 12 – Comparação da importância entre as variáveis de entrada do experimento da série de máxima do Rio Xingu.



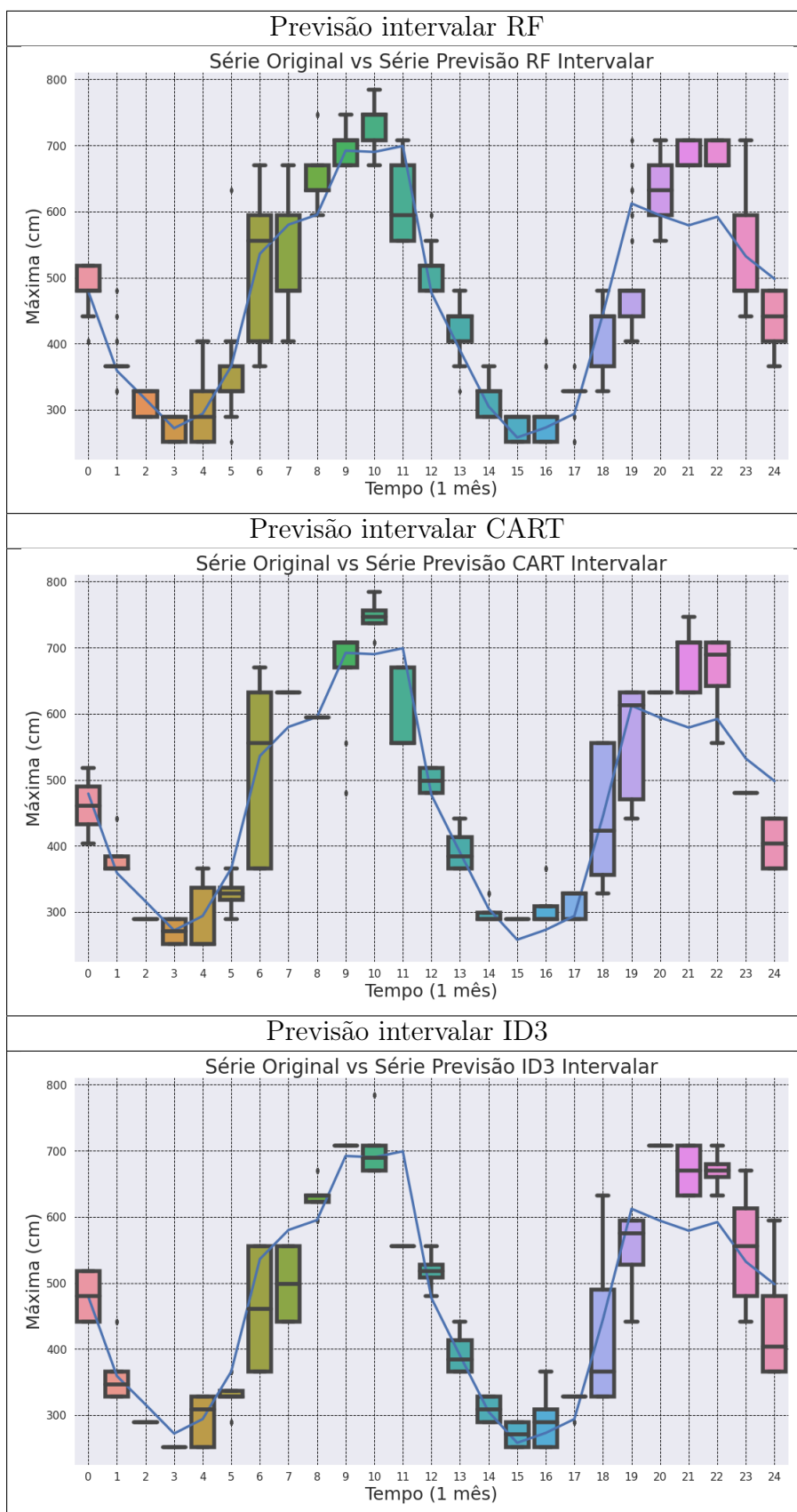


Tabela 13 – Gráficos da série de máxima do Rio Xingu e as previsões intervalares da FDT-FTS composta.

Fonte: Autor

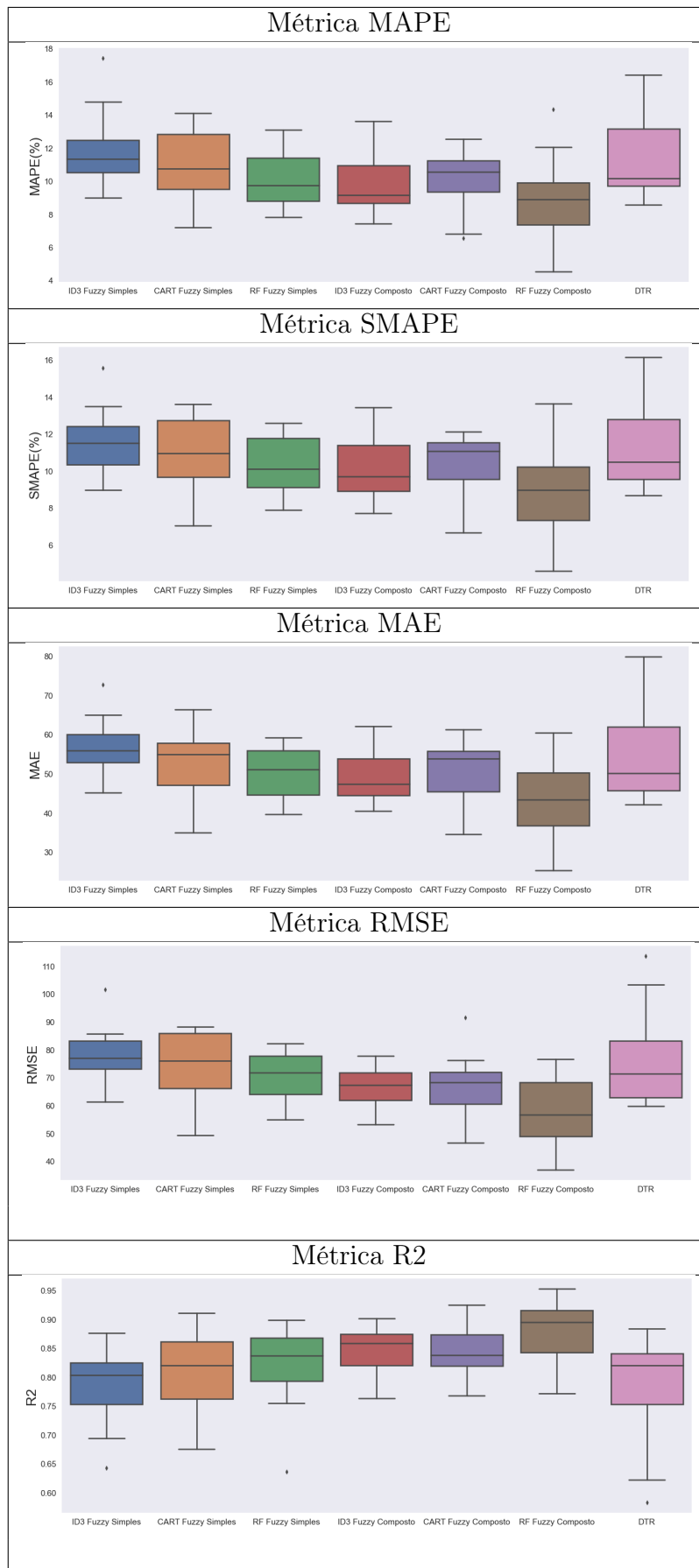


Tabela 14 – Métricas expressas em *boxplots* para o método FDT-FTS simples e composto, além do método de comparação DTR para a série de máxima do rio Xingu.

Fonte: Autor

Intervalos de Treino :  
 [[0, 244], [2, 246], [6, 250], [12, 256], [20, 264], [30, 274], [42, 286], [56, 300], [72, 316], [90, 334], [110, 354], [132, 376]]  
 Intervalos de Teste :  
 [[245, 280], [247, 282], [251, 286], [257, 292], [265, 300], [275, 310], [287, 322], [301, 336], [317, 352], [335, 370], [355, 390], [377, 412]]

Figura 28 – Intervalos de treino e teste para os experimentos da série de máxima do Rio Xingu.

Fonte: Autor

Em adição, mostra-se na Tab. 14, como exemplo, os valores dos *boxplots* das métricas utilizadas para expressar a qualidade dos métodos FDT-FTS simples e composto.

| Métodos | Métricas para o FDT-FTS Simples (Média ± Desvio Padrão) |               |              |             |            |
|---------|---|---------------|--------------|-------------|------------|
|         | MAPE  | SMAPE         | RMSE         | MAE         | R2         |
| ID3     | 11.89% ±2.21%   | 11.61% ±1.73% | 77.91 ±10.28 | 56.49 ±7.31 | 0.78 ±0.06 |
| CART    | 11.04% ±2.08%   | 10.97% ±1.91% | 74.37 ±12.04 | 52.83 ±8.57 | 0.80 ±0.07 |
| RF      | 10.16% ±1.81%   | 10.26% ±1.60% | 71.58 ±9.31  | 49.99 ±6.33 | 0.81 ±0.07 |

Tabela 15 – Métricas utilizadas na série de máxima do Rio Xingu aplicando-se o método FDT-FTS simples.

| Métodos | Métricas para o FDT-FTS Composto (Média ± Desvio Padrão) |                |              |             |            |
|---------|--|----------------|--------------|-------------|------------|
|         | MAPE   | SMAPE          | RMSE         | MAE         | R2         |
| ID3     | 9.84 % ±1.96%  | 10.16 % ±1.83% | 66.44 ±7.04  | 49.42 ±6.82 | 0.84 ±0.04 |
| CART    | 10.10 % ±1.91%   | 10.30 % ±1.85% | 66.31 ±11.90 | 50.21 ±8.59 | 0.84 ±0.04 |
| RF      | 8.88 % ±5.59%  | 8.87 % ±2.39%  | 57.65 ±12.24 | 42.93 ±9.79 | 0.88 ±0.05 |

Tabela 16 – Métricas utilizadas na série de máxima do Rio Xingu aplicando-se o método FDT-FTS composto.

| Métodos | Métricas para DTR (Média ± Desvio Padrão) |               |             |              |            |
|---------|---|---------------|-------------|--------------|------------|
|         | MAPE                                      | SMAPE         | RMSE        | MAE          | R2         |
| DTR     | 11.589% ±2.22%                            | 11.73% ±2.33% | 78.29±16.77 | 57.21 ±12.78 | 0.78 ±0.09 |

Tabela 17 – Métricas utilizadas na série de máxima do Rio Xingu aplicando-se o método DTR.

### 5.2.3 Discussão dos Resultados

Os resultados apontados na seção anterior apontam que os algoritmos de árvore gerados pelo método FDT-FTS composto apresentaram resultado médios melhores que o algoritmo DTR. No caso, o método *Random Forest Fuzzy* apresentou os melhores resultados para todas as métricas como mostrado nas Tab. 15, Tab. 16 e Tab. 17. Além disso, as séries

com previsões intervalares possuem intervalos que estão em confluência com a série original. Perceba que apesar da série ser bem definida considerando seus níveis, sazonalidade e estacionariedade, a variação entre os pontos seguidos da série são bruscas o que leva os modelos de árvore a computar intervalos de grande amplitude, como mostrado na Tab. 13.

No trabalho realizado em [Vieira et al. \(2021\)](#), os valores máximos e mínimos de RMSE, MAE e  $R^2$  alcançados na previsão da série de máxima que combina um modelo de regressão com algoritmo genético estão representados na Tab. 18. No caso, as diferenças entre os dois modelos é que a *Random Forest Fuzzy* gerada pelo método de FDT-FTS composto utiliza em seu treinamento e teste uma janela temporal de tamanho fixo que desliza no tempo, além de ser um modelo baseado em regras e não ter os seus hiperparâmetros otimizados. Já o modelo proposto em [Vieira et al. \(2021\)](#) utiliza um algoritmo genético para a escolha dos melhores atributos considerando inclusive os meses em que esses atributos são relevantes. O algoritmo genético escolhe as variáveis de entrada para um modelo de regressão linear a fim de prever a série de máxima. Vale ressaltar que o modelo de árvore fuzzy realiza a seleção de atributos utilizando, por exemplo, ganho de informação, uma métrica generalizada independente da janela temporal utilizada. A partir dos dados presentes na Tab. 18, conclui-se que o modelo de *Random Forest Fuzzy*, em seus melhores resultados gerados pelo método FDT-FTS composto apresentou resultados próximos do modelo proposto por [Vieira et al. \(2021\)](#), como é o caso dos valores do coeficiente de determinação. No entanto, perceba que o modelo de árvore fuzzy comparado possui maior variância nos resultados do que o modelo proposto em [Vieira et al. \(2021\)](#).

| Métodos   | RMSE Max | RMSE Min     | MAE Max | MAE Min      | R2 Max       | R2 Min |
|-----------|----------|--------------|---------|--------------|--------------|--------|
| GA com RL | 20.13    | <b>18.57</b> | 16.61   | <b>13.43</b> | <b>0.988</b> | 0.986  |
| RF Fuzzy  | 76.54    | <b>36.97</b> | 60.40   | <b>25.34</b> | <b>0.95</b>  | 0.77   |

Tabela 18 – Valores de RMSE, MAE e  $R^2$  comparados entre os algoritmos RF Fuzzy Composto e GA com Regressão Linear ([Vieira et al., 2021](#)).

Por fim, nos resultados das árvores geradas pelos métodos FDT-FTS simples e composto mostra que mesmo retirando alguns atributos não houve penalidade na acurácia para as árvores geradas pelo FDT-FTS composto. Essa seleção de atributos foi necessária levando em consideração que o método composto é computacionalmente mais custoso que o método simples durante a fase de treinamento.

## 5.3 Consumo de Energia

Nesta seção apresentaremos os resultados do método FDT-FTS aplicado a um conjunto de dados que expressam medidas de consumo de energia elétrica numa casa

localizada em Sceaux na França, durante os anos de 2006 e 2010, aproximadamente 47 meses.

Esses dados podem ser encontrados no seguinte sitio eletrônico:

“<https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption>”, Acessado em: 10/2021.

Neste experimento, a série de potência ativa global foi fuzzificada com funções triangulares em 150 conjuntos fuzzy, a variável “Hora” em 24 conjuntos utilizando funções triangulares e todas as outras variáveis foram particionadas em 30 conjuntos fuzzy com funções triangulares.

As variáveis presentes nessa base de dados estão discriminadas na Tab. 19. É importante salientar que a série principal considerada neste experimento é a de energia ativa global, dado que é essa a responsável pela realização do trabalho nos equipamentos da casa. Além disso, não será usado nesse experimento a energia global reativa presente na base de dados.

| Descrição das Variáveis de Entrada e Saída utilizadas |  |
|---|--|
| Variáveis   | Informação dos Atributos   |
| Data  | Data expressa no formato dd/mm/aaaa.   |
| Tempo   | Tempo expresso no formato hh:mm:ss.  |
| Potência Ativa Global                                 | Representa a energia ativa global média por minuto.  |
| Potencia Reativa Global                               | Representa a energia reativa global média por minuto.  |
| Tensão  | Tensão média por minuto.   |
| Intensidade Global                                    | Representa a intensidade de corrente global média por minuto.                                      |
| Sub-medida 1  | É uma medição realizada em Wh, e corresponde aos equipamentos presentes na cozinha.                |
| Sub-medida 2  | É uma medição realizada em Wh, e corresponde aos equipamentos presentes numa lavanderia.           |
| Sub-medida 3  | É uma medição realizada em Wh, e corresponde a um aquecedor elétrico de água e um ar condicionado. |

Tabela 19 – Significado das variáveis presentes no conjunto de dados da demanda de energia.

### 5.3.1 Caracterização dos Dados

Será utilizado um filtro de média móvel exponencial com período igual a 4 na série da potência de energia ativa a fim de dirimir o nível de ruídos, veja na Fig. 29 uma parte da série original e da série com filtro. Já na Fig. 30 é notada a presença da autocorrelação na série filtrada. Ademais, aplica-se o teste de hipótese “Dickey-Fuller Aumentado” que evidenciou a não estacionariedade da série ( $p - valor = 0.371$ ).

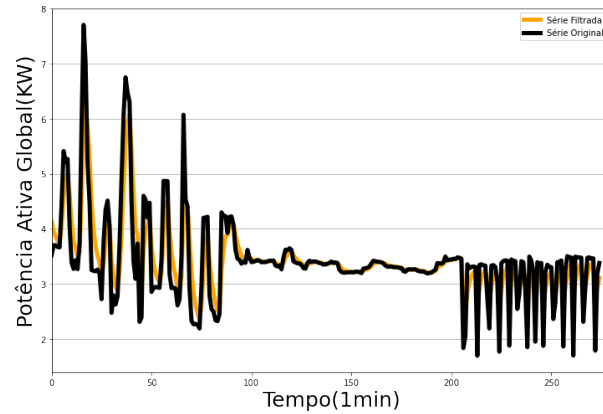


Figura 29 – Série da potência ativa e da série filtrada.

Fonte: Autor

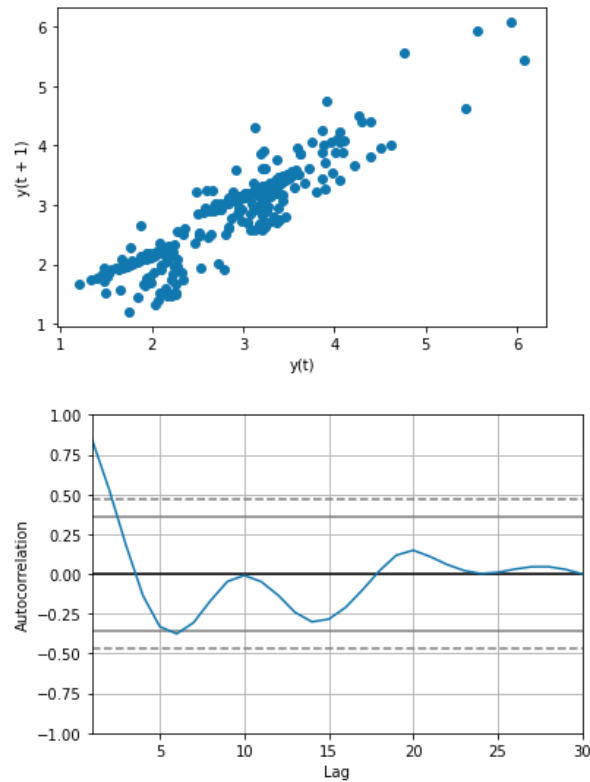


Figura 30 – Autocorrelação da série da demanda de potência ativa em uma casa.

Fonte: Autor

### 5.3.2 Resultados

Explicitaremos nesta seção os resultados quanto a previsão dos três algoritmos de árvore e a curva de energia ativa global. Veja a seguir na Tabela 20, os gráficos da previsão e a série original para cada árvore do método FDT-FTS composto. Veja, também, na Fig. 31 os intervalos de treino e teste.

Na Tab. 22 tem-se como exemplo a previsão intervalar realizada pelas árvores, 50 pontos da série original comparados com a série de intervalos das saídas das árvores do método FDT-FTS composto. Além disso, calcula-se na Tab. 21 a importância de cada atributo utilizado para prever a série global de energia ativa tanto para método simples quanto para o método composto.

Na Tab. 23 tem-se como exemplo os gráficos dos *boxplots* das métricas (MAPE, SMAPE, RMSE e MAE) utilizadas para avaliar a série prevista. Em seguida, apresenta-se as tabelas Tab. 24, Tab. 25 e Tab. 26 os valores numéricos dessas métricas

```
Intervalos de Treino:
[[1000, 4000], [1200, 4200], [1600, 4600], [2200, 5200], [3000, 6000], [4000, 7000], [5200, 8200], [6600, 9600], [8200, 11200], [10000, 13000], [12000, 15000]]
Intervalos de Teste:
[[4010, 4410], [4210, 4610], [4610, 5010], [5210, 5610], [6010, 6410], [7010, 7410], [8210, 8610], [9610, 10010], [11210, 11610], [13010, 13410], [15010, 15410]]
```

Figura 31 – Intervalos de treino e teste utilizados no experimento da demanda de energia.

Fonte: Autor

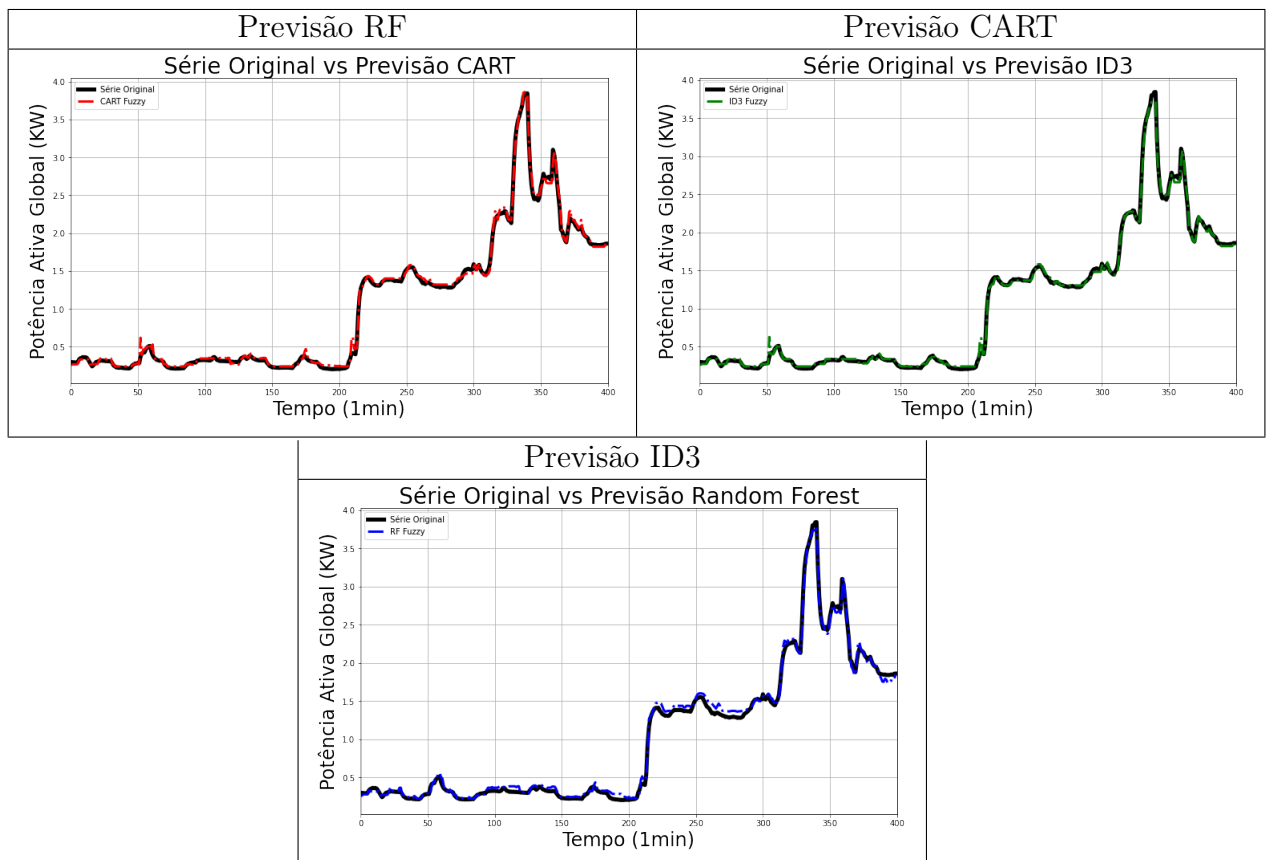


Tabela 20 – Gráficos da série de potência ativa global e as previsões dos métodos de árvore utilizando o FDT-FTS composto.

Fonte: Autor

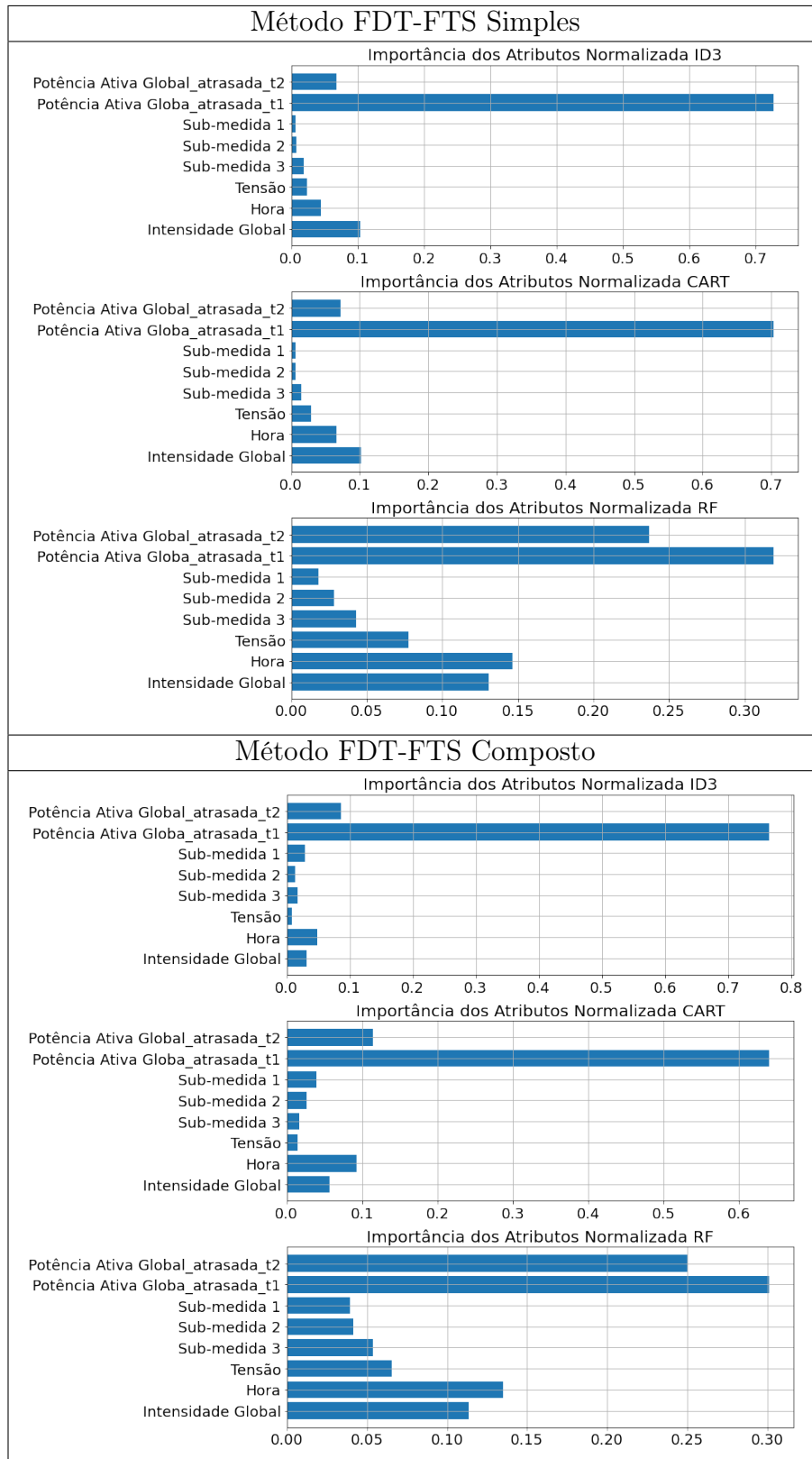


Tabela 21 – Comparação da importância entre as variáveis de entrada que os métodos atribuem para o experimento da série de Energia.

Fonte: Autor



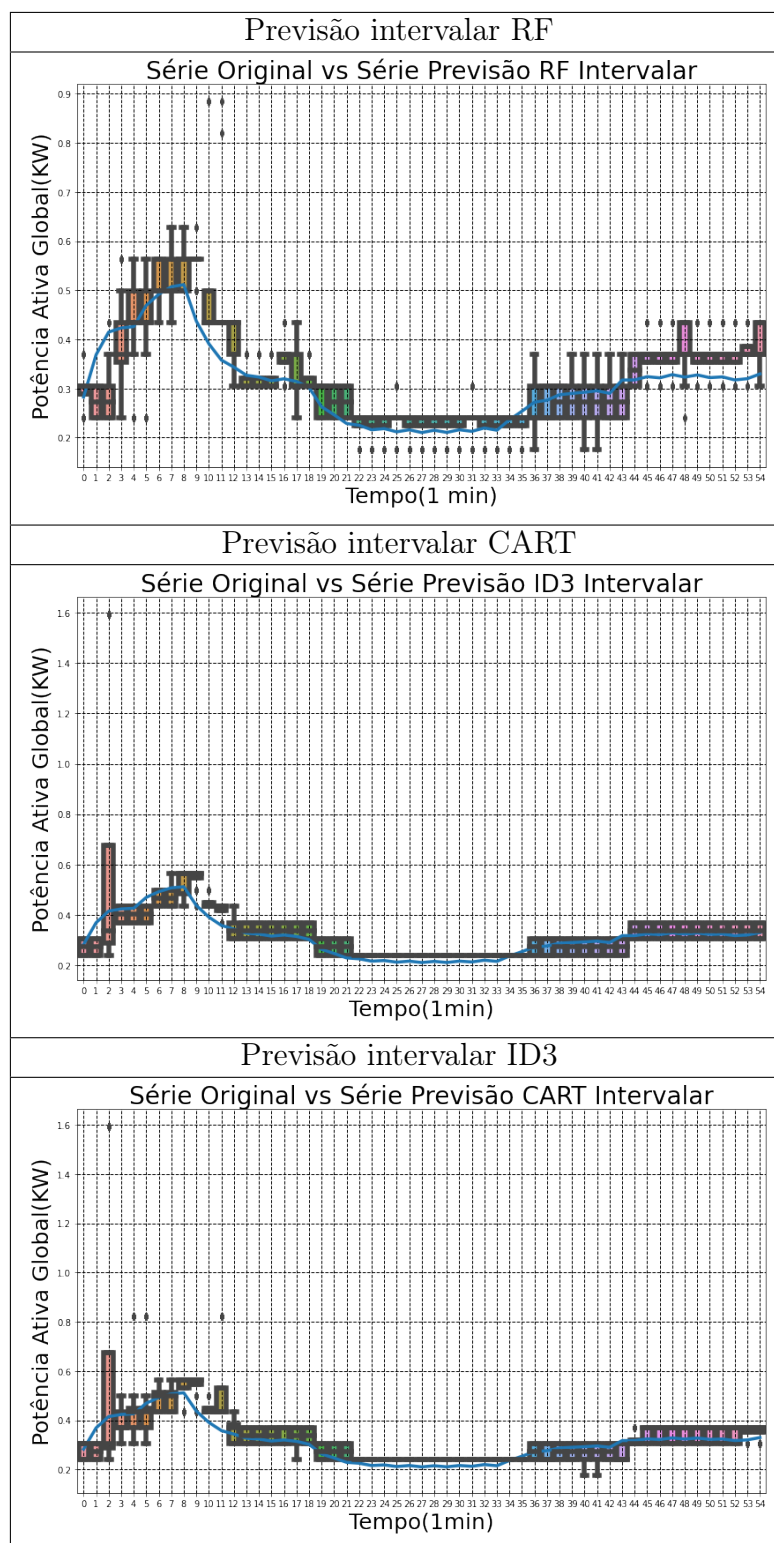


Tabela 22 – Gráficos da série de potência ativa global e as previsões intervalares do método FDT-FTS composto.

Fonte: Autor

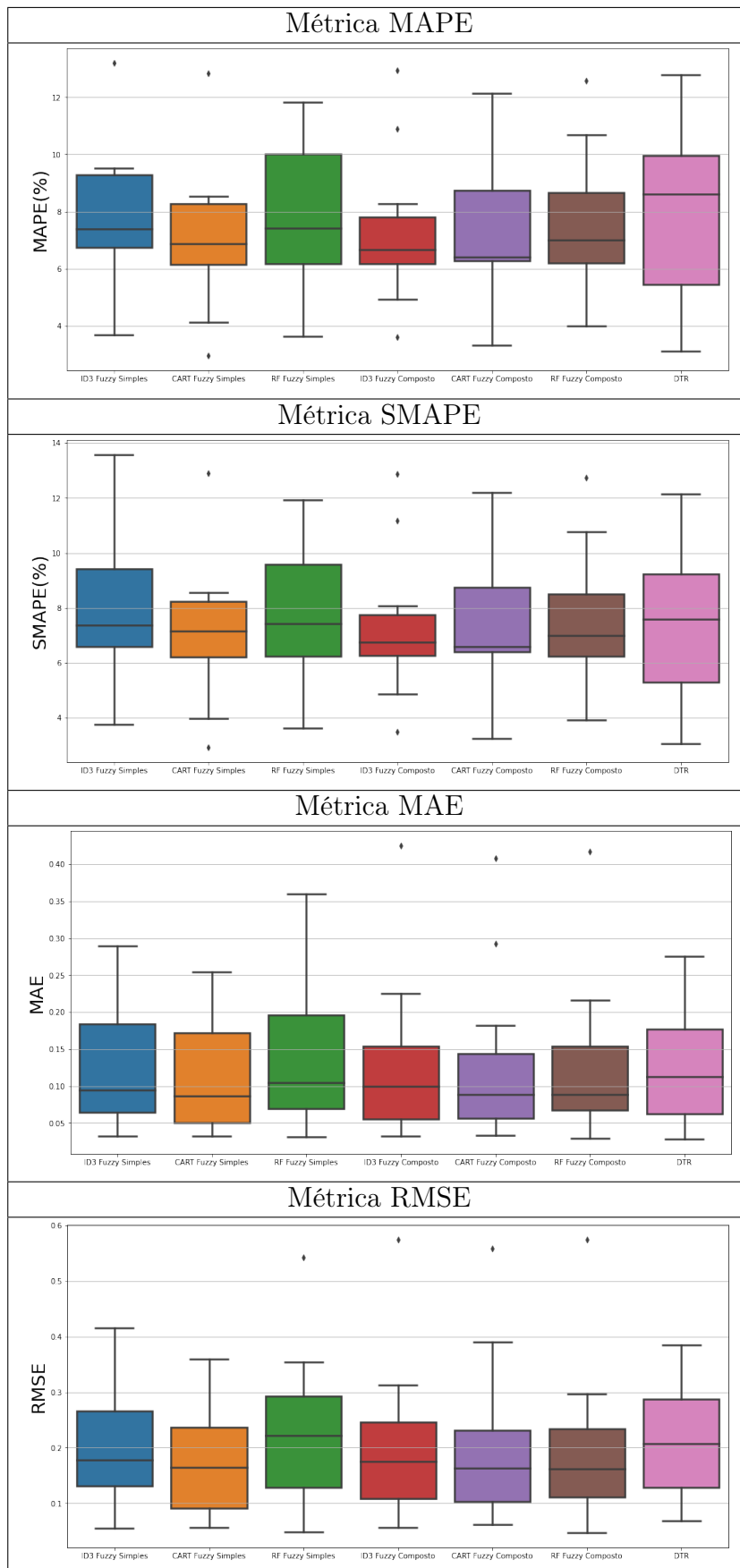


Tabela 23 – Métricas expressas em *boxplots* para o método FDT-FTS simples e composto, além dos métodos de comparação DTR para a série de potência ativa global.

Fonte: Autor

| Métodos | Métricas para o FDT-FTS Simples (Média $\pm$ Desvio Padrão) |                   |                 |                 |
|---------|---|-------------------|-----------------|-----------------|
|         | MAPE  | SMAPE             | RMSE            | MAE             |
| ID3     | 7.711% $\pm$ 2.52%  | 7.76% $\pm$ 2.63% | 0.20 $\pm$ 0.11 | 0.12 $\pm$ 0.08 |
| CART    | 7.114% $\pm$ 2.45%  | 7.16% $\pm$ 2.48% | 0.17 $\pm$ 0.10 | 0.11 $\pm$ 0.07 |
| RF      | 7.909% $\pm$ 2.53%  | 7.90% $\pm$ 2.54% | 0.23 $\pm$ 0.13 | 0.13 $\pm$ 0.09 |

Tabela 24 – Métricas utilizadas na série previsão da potência ativa global com o filtro de média móvel exponencial aplicando-se o método FDT-FTS simples.

| Métodos | Métricas para o FDT-FTS Composto (Média $\pm$ Desvio Padrão) |                   |                 |                 |
|---------|--|-------------------|-----------------|-----------------|
|         | MAPE   | SMAPE             | RMSE            | MAE             |
| ID3     | 7.323% $\pm$ 2.49%   | 7.36% $\pm$ 2.51% | 0.20 $\pm$ 0.14 | 0.13 $\pm$ 0.11 |
| CART    | 7.345% $\pm$ 2.62%   | 7.31% $\pm$ 2.55% | 0.20 $\pm$ 0.14 | 0.12 $\pm$ 0.11 |
| RF      | 7.532% $\pm$ 2.44%   | 7.50% $\pm$ 2.49% | 0.19 $\pm$ 0.14 | 0.12 $\pm$ 0.10 |

Tabela 25 – Métricas utilizadas na série de potência ativa global com o filtro de média móvel exponencial aplicando-se o método FDT-FTS composto.

| Métodos | Métricas para a DTR (Média $\pm$ Desvio Padrão) |                   |                 |                 |
|---------|---|-------------------|-----------------|-----------------|
|         | MAPE  | SMAPE             | RMSE            | MAE             |
| DTR     | 7.96% $\pm$ 3.13%                               | 7.32% $\pm$ 2.66% | 0.21 $\pm$ 0.10 | 0.12 $\pm$ 0.08 |

Tabela 26 – Métricas utilizadas na série de potência ativa global com o filtro de média móvel exponencial aplicando-se o método DTR.

### 5.3.3 Discussão dos Resultados

Primeiramente, vale salientar que apesar da série não possuir estacionariedade, os métodos propostos conseguiram resultados satisfatórios e nesse experimento as árvores simples tiveram resultados ligeiramente superiores as florestas de árvores como exposto nas Tab. 24, Tab. 25 e Tab. 26. Além disso, os resultados da FTD-FTS simples obtiveram resultados semelhantes as árvores geradas na pelo método FDT-FTS composto, a CART Fuzzy simples apresentou o melhor resultado como explicito nas tabelas anteriores.

É interessante ressaltar que na Tab. 22, em que se calcula a série intervalar, os intervalos apresentaram amplitudes consistentes ao longo da série sugerindo uma variabilidade homogênea de regras ativadas ao longo do tempo, além da série intervalar ter uma boa confluência em relação aos movimentos da série original. Em adição, nota-se na Tab. 21 que dos atributo de entrada aquele que tem a maior relevância para os modelos de árvore é a série de potência ativa global atrasada em  $t - 1$ .

## 5.4 Fluxo de Passageiros num Serviço de Metrô

Os dados que serão usados nessa seção dizem respeito ao volume de tráfego de passageiros realizados de hora em hora, na linha da *Interstate 94 Westbound* para a estação *MN DoT ATR 301* que se localiza entre as cidades de *Minneapolis* e *Saint Paul*. Ademais, a ideia principal do problema em questão é: prever e analisar a série de volume do tráfego passageiros por meio não só de valores passados, mas também a partir das variáveis externas que influenciam essa, como o clima, a presença de feriados, a sazonalidade no tempo, entre outras.

Pode-se ter acesso a essa base de dados no sítio eletrônico: “<https://archive.ics.uci.edu/ml/datasets/Metro%20Interstate%20Traffic%20Volume>”, Acessado em: 10/2021.

As variáveis presentes na base de dados do problema de volume do tráfego no metrô estão descritas na Tabela 27.

| Descrição das Variáveis de Entrada e Saída |   |
|--|---|
| Variáveis                                  | Informação dos Atributos  |
| Feriado                                    | Feridos regionais e nacionais nos EUA.  |
| Temperatura                                | Temperatura média em Kelvin.  |
| Chuva (1h)                                 | Quantidade numérica (em <i>mm</i> ) de chuva no período de 1 hora.  |
| Neve (1h)                                  | Quantidade numérica (em <i>mm</i> ) de neve no período de 1 hora.   |
| Nebulosidade                               | Porcentagem numérica de cobertura das nuvens.   |
| Clima                                      | Descrição principal do clima de forma curta e categórica.   |
| Descrição do Clima                         | Descrição longa e categórica do clima.  |
| Data e Hora                                | Contabiliza a data (dia, semana, mês e ano) e as horas da ocorrência dos eventos nas variáveis do problema. |
| Volume do Tráfego                          | Volume do número de passageiros que trafegam no metrô no período de 1h.                                     |

Tabela 27 – Descrição das variáveis de entrada e saída do modelo referente ao problema de fluxo de passageiro no metrô situado entre *Minneapolis* e *Saint Paul*.

Nesta análise, todas as série foram fuzzificadas com funções triangulares e foram particionadas em 10 conjuntos fuzzy, com exceção da série das horas que foi particionada em 24 conjuntos, do feriado que foi particionado em 3 e da variável principal (volume de passageiros) que foi particionado em 100 conjuntos. Além disso, as configurações das árvores e do método *Random Forest* são as seguintes:

- **ID3** : Max\_Depth=10 e Random\_State=0
- **CART**: Max\_Depth=10 e Random\_State=0

- **Random Forest:** `n_estimators=30` e `Random_State=0`
- **DTR:** Parâmetros *default* da “Scikit-Learn”.

### 5.4.1 Caracterização dos Dados

Utilizamos nessa série um filtro de média móvel exponencial com período igual a 4. Veja na Fig. 32 o efeito de suavização que o filtro tem nessa série. Já na Fig. 33 tem-se a presença da autocorrelação na série filtrada do metrô, em adição, aplica-se o teste de hipótese “Dickey-Fuller Aumentado” comprovando que a série em questão é estacionária.

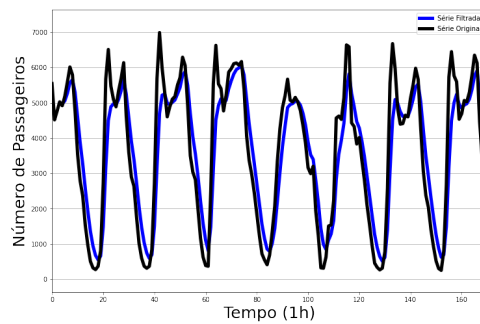


Figura 32 – Série original do fluxo de passageiros no metrô e a série do filtro de média móvel exponencial com  $p=4$ .

Fonte: Autor

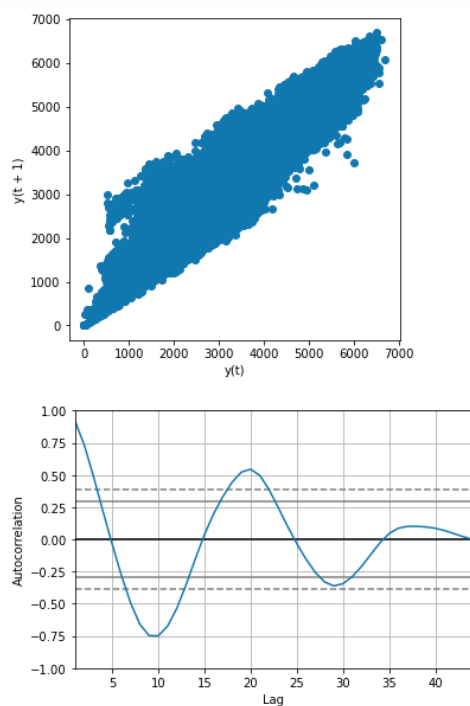


Figura 33 – Autocorrelação da série filtrada do volume de passageiros do metrô.

Fonte: Autor

## 5.4.2 Resultados

Primeiramente, apresenta-se na Tab. 28 a série original e a série prevista pelos modelos de árvore. Além disso, mostra-se, também, a série de previsão intervalar com a série original na Tab. 29. Tem-se na Tab. 30 os gráficos referentes a importância dos atributos calculada pelas árvores do método FDT-FTS. Por fim, apresenta-se nas Tab. 31, Tab. 32, Tab. 33 e Tab. 34 os resultados referentes as métricas usadas para análise de desempenho dos algoritmos para a previsão da série temporal em questão, comparativamente ao algoritmo DTR. Apresenta-se na Fig. 34 os intervalos de treino e teste usados.

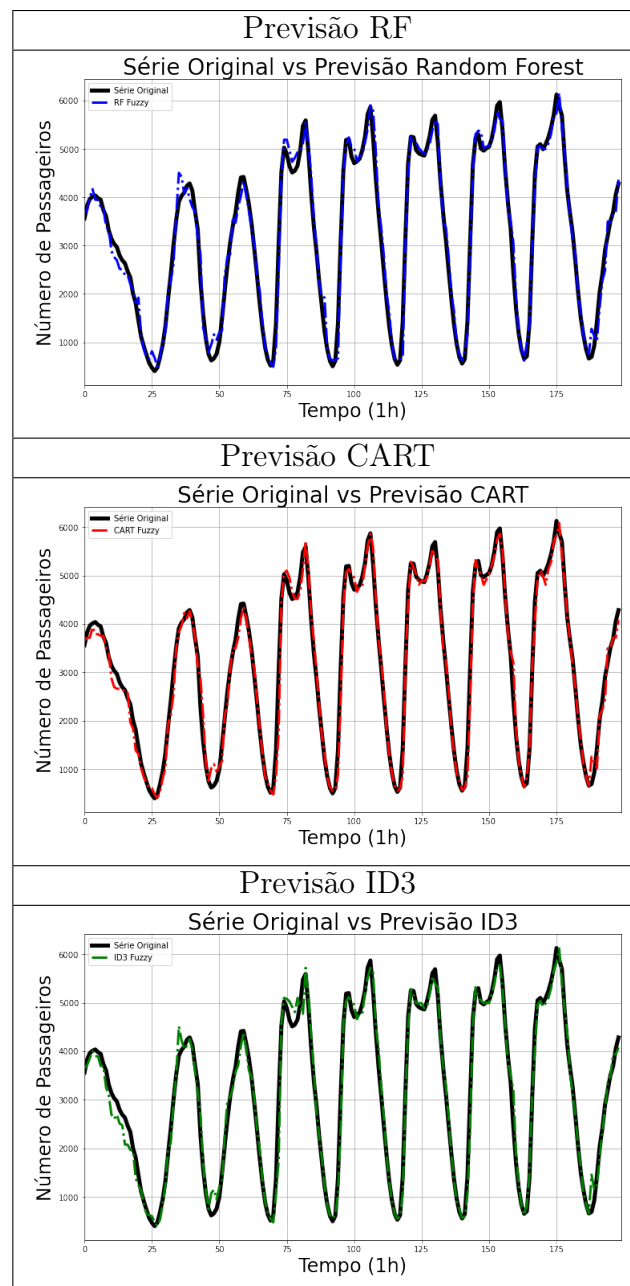


Tabela 28 – Gráficos da série de fluxo de passageiros e as previsões dos métodos de árvore utilizando o FDT-FTS composto.

Fonte: Autor

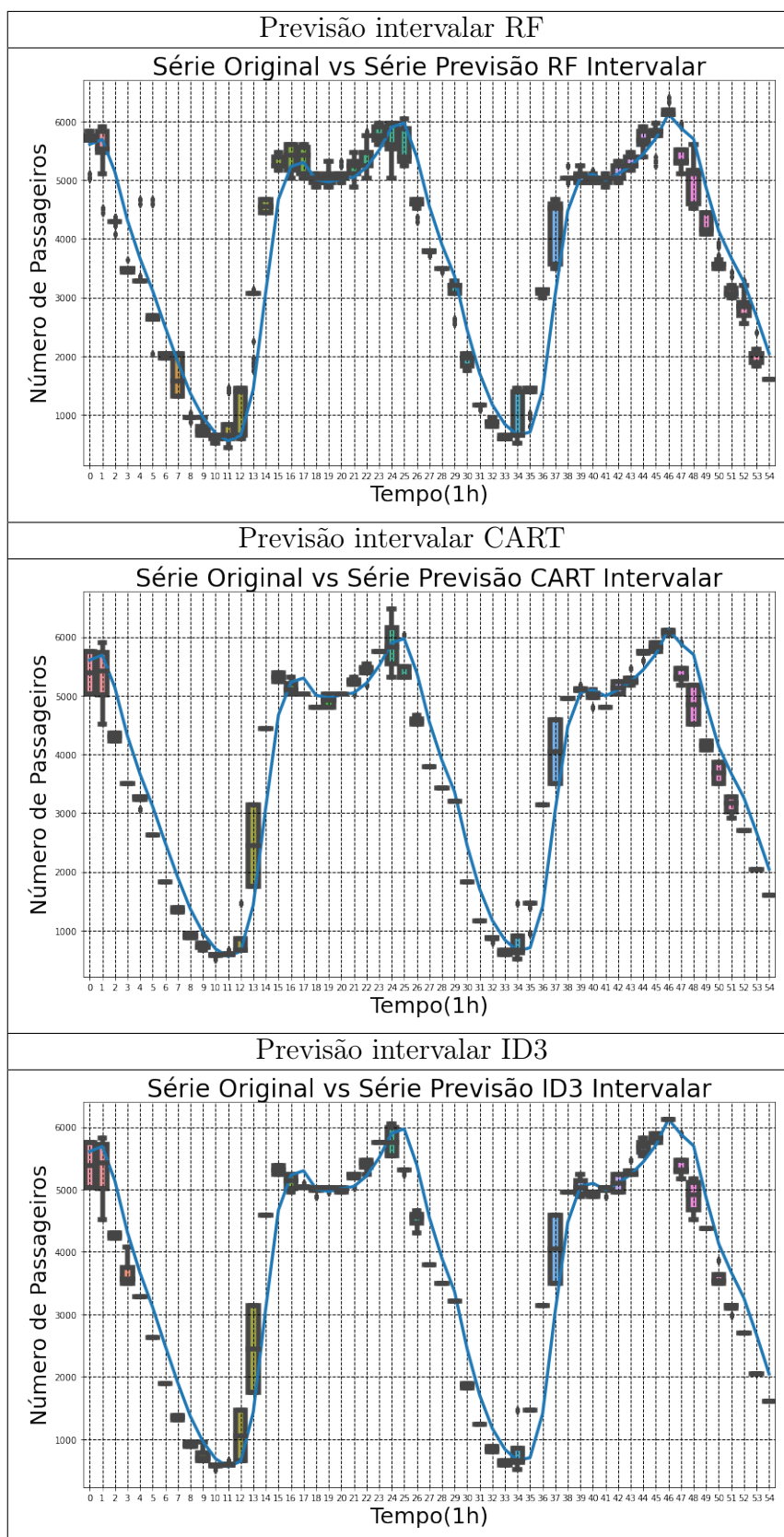


Tabela 29 – Gráficos da série de fluxo de passageiros e as previsões intervalares do FDT-FTS composto.

Fonte: Autor

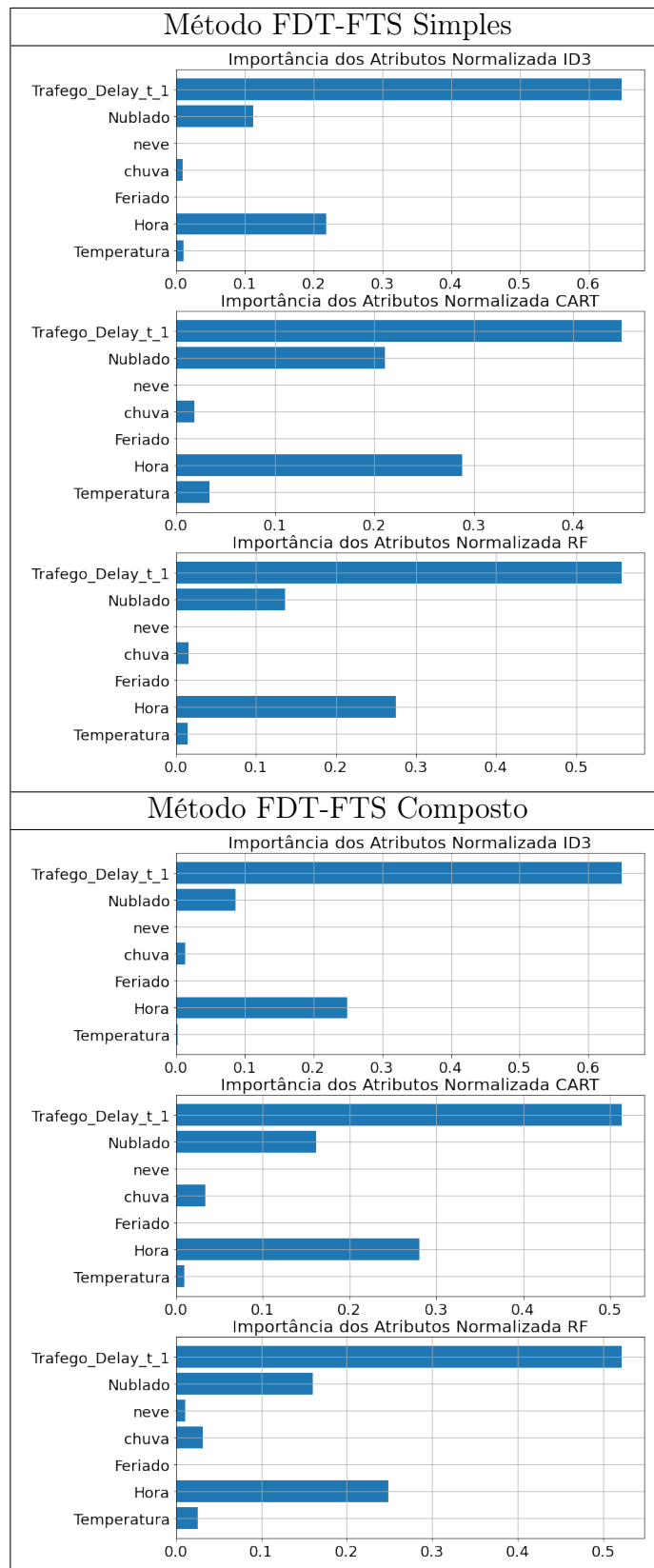


Tabela 30 – Comparação da importância entre as variáveis de entrada que os métodos atribuem para o experimento da série de fluxo de passageiros.

Fonte: Autor



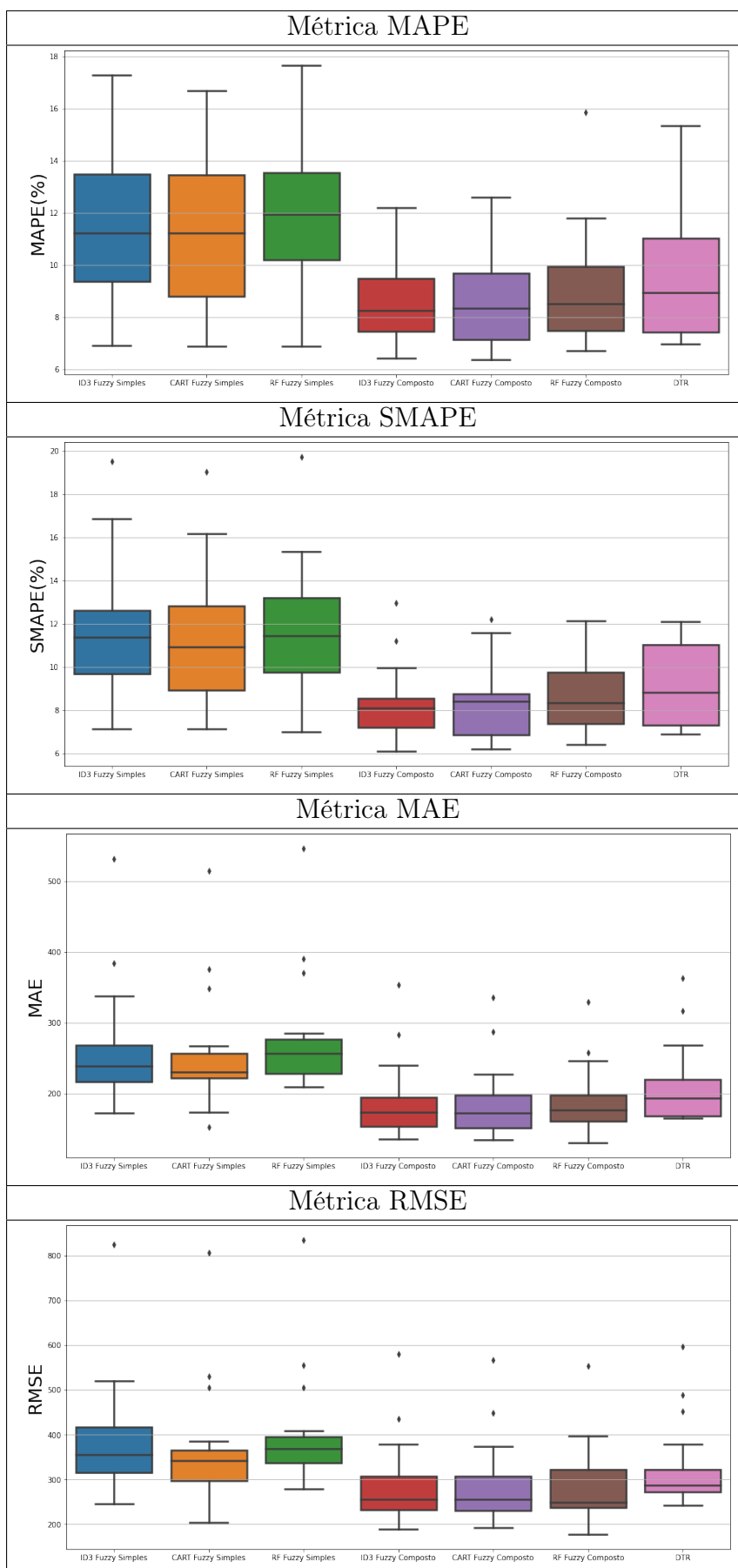


Tabela 31 – Métricas expressas em *boxplots* para o método FDT-FTS simples e composto, além do método DTR para a série de fluxo de passageiros.

Fonte: Autor

Intervalos de Treino:  
 [[0, 1400], [100, 1500], [300, 1700], [600, 2000], [1000, 2400], [1500, 2900], [2100, 3500], [2800, 4200], [3600, 5000], [4500, 5900], [5500, 6900], [6600, 8000], [7800, 9200], [9100, 10500], [10500, 11900], [12000, 13400], [13600, 15000], [15300, 16700]]  
 Intervalos de Teste:  
 [[1401, 1600], [1501, 1700], [1701, 1900], [2001, 2200], [2401, 2600], [2901, 3100], [3501, 3700], [4201, 4400], [5001, 5200], [5901, 6100], [6901, 7100], [8001, 8200], [9201, 9400], [10501, 10700], [11901, 12100], [13401, 13600], [15001, 15200], [16701, 16900]]

Figura 34 – Intervalos de treino e teste utilizados no experimento de fluxo de passageiros em um Metrô.

Fonte: Autor

| Métodos | Métricas para o FDT-FTS Simples (Média ± Desvio Padrão) |               |                |                |
|---------|---|---------------|----------------|----------------|
|         | MAPE  | SMAPE         | RMSE           | MAE            |
| ID3     | 11.47% ±2.95%   | 11.40% ±3.00% | 388.96 ±128.65 | 262.12 ±82.41  |
| CART    | 11.33% ±2.99%   | 11.09% ±3.02% | 369.88 ±130.03 | 254.90 ±81.89  |
| RF      | 12.20% ±2.96%   | 11.70% ±2.84% | 398.41 ±124.74 | 276.84 ±80.929 |

Tabela 32 – Métricas utilizadas na série de fluxo de passageiros com o filtro de média móvel exponencial aplicando-se o método FDT-FTS simples.

| Métodos | Métricas para o FDT-FTS Composto (Média ± Desvio Padrão) |              |               |               |
|---------|--|--------------|---------------|---------------|
|         | MAPE   | SMAPE        | RMSE          | MAE           |
| ID3     | 8.53% ±1.50%   | 8.24% ±1.67% | 286.66 ±94.34 | 187.44 ±53.71 |
| CART    | 8.62% ±1.71%   | 8.28% ±1.63% | 286.74 ±92.23 | 187.05 ±51.03 |
| RF      | 9.11% ±2.22%   | 8.46% ±1.53% | 283.17 ±86.74 | 189.11 ±47.10 |

Tabela 33 – Métricas utilizadas na série de fluxo de passageiros com o filtro de média móvel exponencial aplicando-se o método FDT-FTS composto.

| Métodos | Métricas para a DTR (Média ± Desvio Padrão) |              |               |               |
|---------|---|--------------|---------------|---------------|
|         | MAPE  | SMAPE        | RMSE          | MAE           |
| DTR     | 9.54% ±2.41%                                | 9.06% ±1.89% | 326.39 ±92.65 | 209.40 ±54.05 |

Tabela 34 – Métricas utilizadas na série de fluxo de passageiros com o filtro de média móvel exponencial aplicando-se o método DTR.

### 5.4.3 Discussão dos Resultados

A série do fluxo de passageiros que utilizam o metrô é uma série sazonal, estacionária e com níveis de amplitude que se repetem ao longo do tempo. Nesse caso, as árvores do método FDT-FTS composto obtiveram melhor desempenho, embora com medianas e médias próximas dos outros métodos (veja as Tab. 32, Tab. 33 e Tab. 34). Nota-se que a série intervalar apresenta em muitas das suas previsões *boxplots* com pequenas extensões,

o que indica a pouca variabilidade nas regras ativadas por pontos da série original (poucas regras conseguem caracterizar um ponto da série)(veja a Tab. 29).

Ademais, outro fator relevante é que as variáveis de entrada como neve, chuva, temperatura e feriado tem pouca ou nenhuma importância para o modelo e que as variáveis como hora e a série atrasada em  $t - 1$  possuem muita relevância para os algoritmos de árvore, como demonstrado na Tab. 30.

De forma geral, as árvores apresentaram um bom resultado quanto a generalização para a previsão da série temporal de fluxo de passageiros, como é mostrado na Tab. 28, em que a curva da série prevista apresenta boa confluência com a curva original.

## 5.5 Séries Temporais Financeiras - IBOVESPA

A presente seção irá discorrer sobre o experimento de previsão de séries temporais financeiras. Considerando o intenso volume de ordens disparadas por algoritmos nas bolsas do mundo, essa área de pesquisa tornou-se uma das mais relevantes considerando as decisões críticas tomadas por algoritmos e o volume financeiro que esses manejam na compra e venda de ativos e derivativos. No caso, abordaremos, como exemplo, a série do contrato futuro do IBOVESPA, de forma que os passos aplicados na previsão dessa podem ser extrapolados para outras séries financeiras. Para a coleta dos dados dessa série foi escrito um *script* em “MQL5” para download dos dados e compilado na plataforma MT5<sup>2</sup>, a qual possui link com uma corretora de valores mobiliários da bolsa brasileira B3<sup>3</sup>. Além disso, foram coletadas 10.000 tuplas referentes à série cronológica dos minicontratos do IBOVESPA (WIN), de 26/09/2019 a 05/11/2020. Foram coletados o preço de fechamento, preço da máxima, preço da mínima, preço de abertura e volume real dos contratos negociados durante o período de 15 minutos.

$$\begin{aligned}\delta(t) &= P_f(t+1) - P_f(t) \\ \hat{P}_f(t+1) &= \hat{\delta}(t) + P_f(t)\end{aligned}\tag{5.1}$$

Nesse trabalho realizaremos a previsão da série do preço de fechamento futuro, no entanto usaremos os algoritmos de árvore para prever, primeiramente, a primeira diferença (variação) da série do preço de fechamento. A série da primeira diferença do preço de fechamento possui a vantagem de ser estacionária e por isso a usaremos para prever indiretamente o preço de fechamento futuro. Isso é possível considerando a Eq. 5.1 em que  $\delta(t)$  é a variação do preço de fechamento,  $P_f(t+1)$  é o preço de fechamento futuro e  $P_f(t)$  é o preço de fechamento presente. Assim, ao prever a primeira diferença do preço

<sup>2</sup> <https://www.metatrader5.com/>, acessado em Outubro de 2021.

<sup>3</sup> [https://www.b3.com.br/pt\\_br/](https://www.b3.com.br/pt_br/), acessado em Outubro de 2021.

de fechamento e somando-a com o preço de fechamento presente obteremos a previsão do preço de fechamento futuro, como expresso na Eq. 5.1, em que  $\hat{P}_f(t+1)$  é o valor previsto do preço de fechamento e  $\hat{\delta}(t)$  é o valor previsto da primeira diferença.

Para realizar a análise desta série, fuzzificamos todas as variáveis de entrada com funções triangulares, as quais foram particionadas em 20 conjuntos fuzzy, com exceção da série de primeira diferença que foi particionada em 100 conjuntos. Além disso, as configurações das árvores e do método *Random Forest* são as seguintes:

- **ID3** : Max\_Depth=10 e Random\_State=0
- **CART**: Max\_Depth=10 e Random\_State=0
- **Random Forest**:n\_estimators=50 e Random\_State=0
- **DTR**: Parâmetros *default* da “Scikit-Learn”.

Por fim, veja na Tab. 35 a descrição das variáveis utilizadas para a previsão da série financeira do IBOVESPA.

| Descrição das Variáveis de Entrada e Saída utilizadas |  |
|---|--|
| Variáveis   | Informação dos Atributos   |
| RSI   | Série do indicador técnico RSI.  |
| EMA_Close   | Série da diferença entre o preço de fechamento e uma média móvel exponencial de período igual a. |
| EMAII_EMAI  | Série da diferença entre duas médias móveis exponenciais.  |
| Dif_High_Low  | Série da diferença entre máxima e mínima.  |
| Primeira Diferença (t-1)                              | Série da primeira diferença atrasada.  |
| Primeira Diferença                                    | Série da primeira diferença.   |
| Primeira Fechamento                                   | Série do preço de fechamento.  |

Tabela 35 – Variáveis utilizadas para a previsão do preço de fechamento do IBOVESPA.

### 5.5.1 Caracterização dos Dados

Apresentamos na Fig. 35 a série do preço de fechamento do IBOVESPA e na Fig.36 a série da primeira diferença do preço de fechamento. Já na Fig. 37 temos a análise da autocorrelação da série da primeira diferença. É válido ressaltar que na Fig. 37 não se nota a presença de autocorrelação linear entre os pontos da série, podendo haver alguma correlação não linear ou correlação com séries externas de forma que essas possam influenciar a direção e amplitude da série de primeira diferença. Pelo teste de “Dickey-Fuller Aumentado” a série do preço de fechamento não é estacionária ( $p - valor = 0.61$ ) e a série da primeira diferença é estacionária.

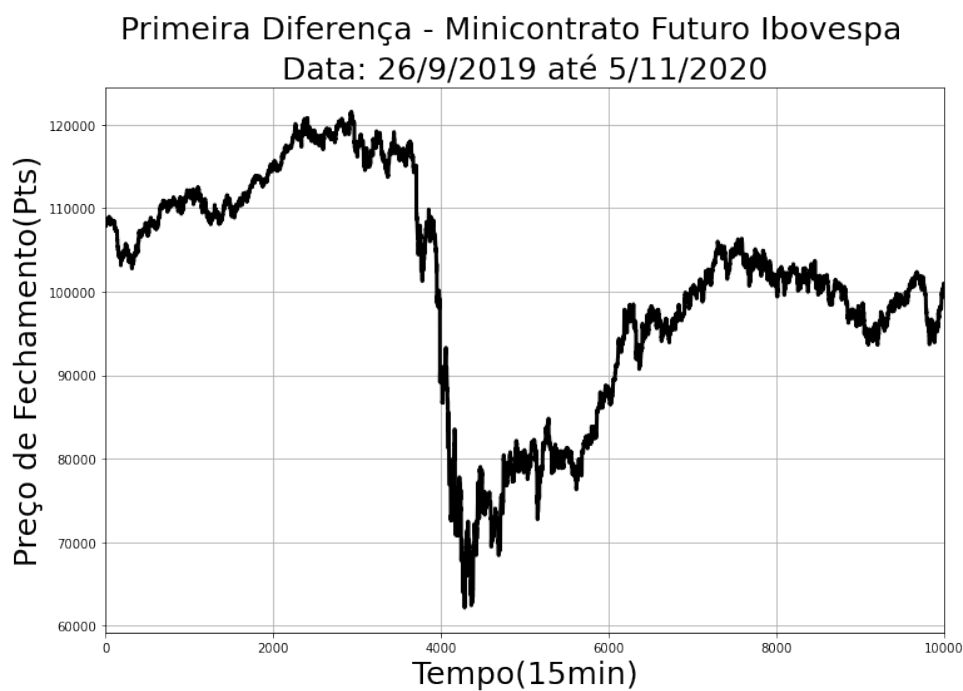


Figura 35 – Série do preço de fechamento do minicontrato futuro do IBOVESPA.

Fonte: Autor

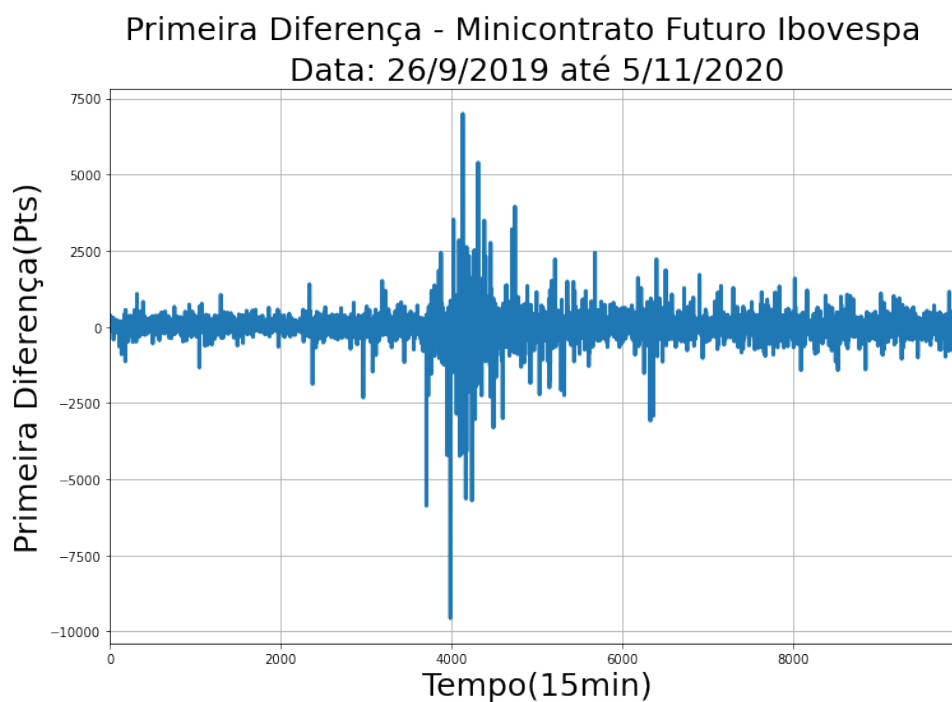


Figura 36 – Série da primeira diferença do preço de fechamento do minicontrato do IBOVESPA.

Fonte: Autor

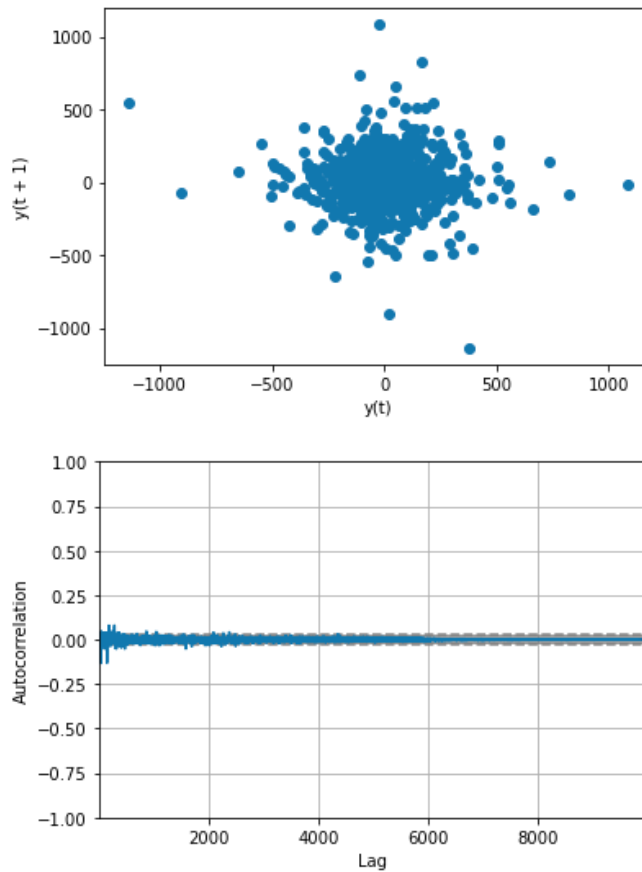


Figura 37 – Autocorrelação da série de primeira diferença.

Fonte: Autor

## 5.5.2 Resultados

Na Tab. 36 temos a série do preço de fechamento original e a série prevista pelos algoritmos gerados pelo método FDT-FTS. Já na Tab. 37 é apresentado a série de previsão intervalar contraposta a série da primeira diferença. Em adição, na Tab. 38 temos a importância que cada atributo teve para cada método, e, por fim, mostra-se nas Tab. 40, Tab. 41, Tab. 42 e Tab. 39 o resultado das métricas de acurácia para os modelos de FDT-FTS simples e composto tanto em tabelas como em *boxplots*. A Fig. 38 apresenta os intervalos de treino e teste.

### Intervalos de Treinamento:

[[4200, 6200], [4210, 6210], [4230, 6230], [4260, 6260], [4300, 6300], [4350, 6350], [4410, 6410], [4480, 6480], [4560, 6560], [4650, 6650], [4750, 6750], [4860, 6860], [4980, 6980], [5110, 7110], [5250, 7250], [5400, 7400], [5560, 7560], [5730, 7730], [5910, 7910], [6100, 8100], [6300, 8300], [6510, 8510]]

### Intervalos de Teste:

[[6210, 6610], [6220, 6620], [6240, 6640], [6270, 6670], [6310, 6710], [6360, 6760], [6420, 6820], [6490, 6890], [6570, 6970], [6660, 7060], [6760, 7160], [6870, 7270], [6990, 7390], [7120, 7520], [7260, 7660], [7410, 7810], [7570, 7970], [7740, 8140], [7920, 8320], [8110, 8510], [8310, 8710], [8520, 8920]]

Figura 38 – Intervalos de treino e teste utilizados no experimento de previsão da série temporal do contrato futuro do IBOVESPA.

Fonte: Autor

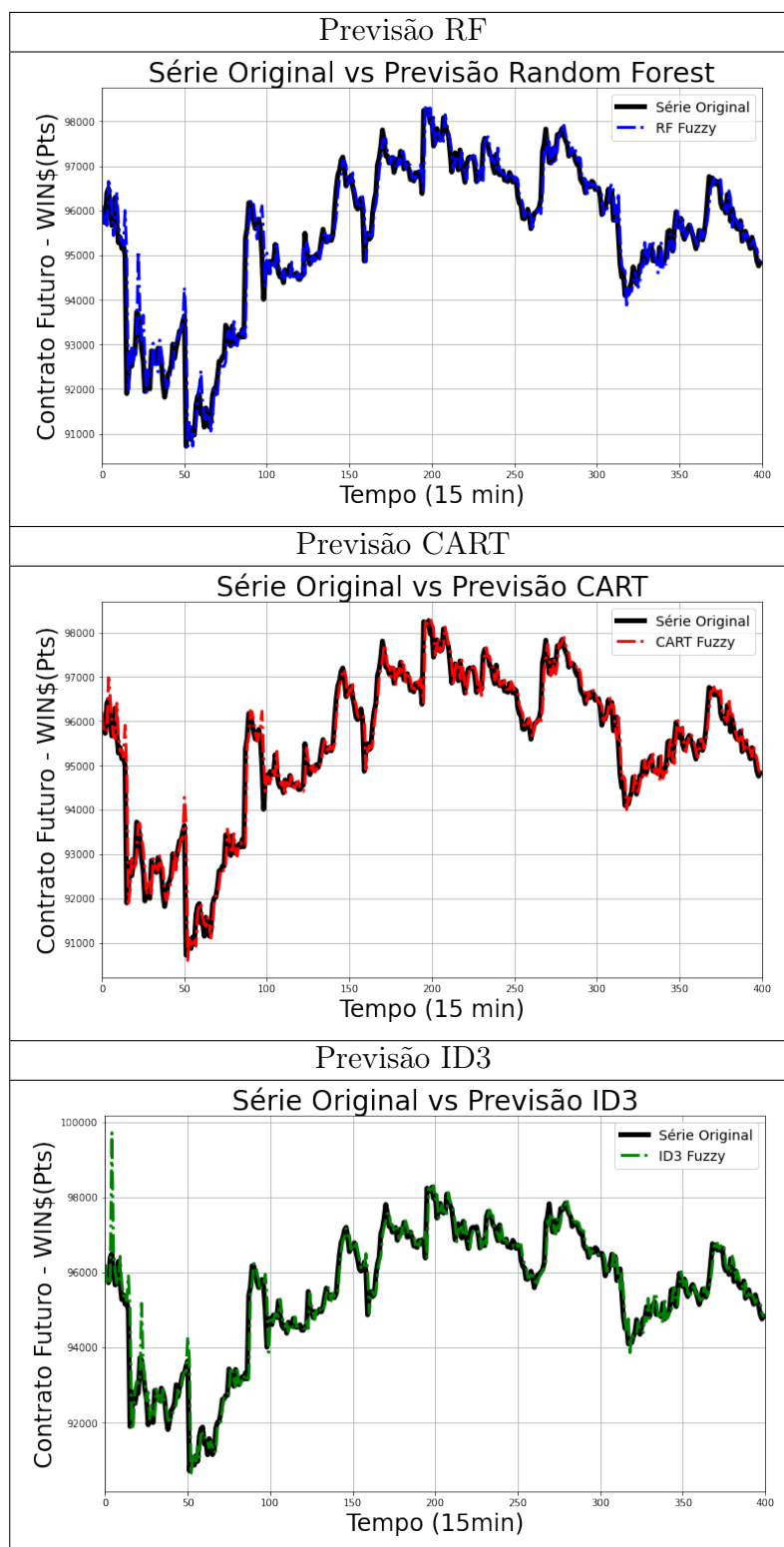


Tabela 36 – Gráficos da série original temporal e as previsões dos métodos de árvore utilizando a FDT-FTS composta.

Fonte: Autor

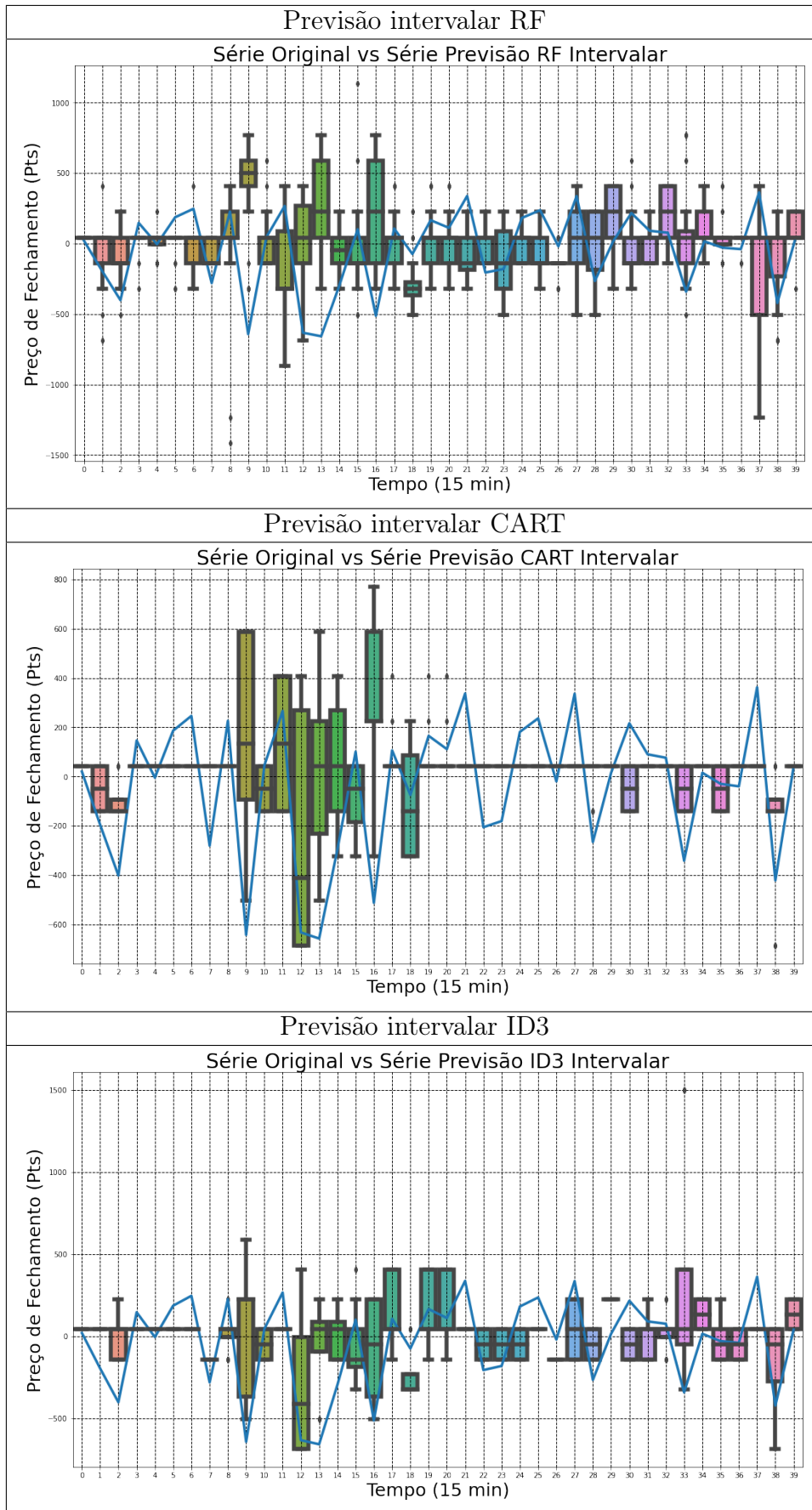


Tabela 37 – Gráficos da série da primeira diferença e as previsões intervalares da FDT-FTS composta.

Fonte: Autor



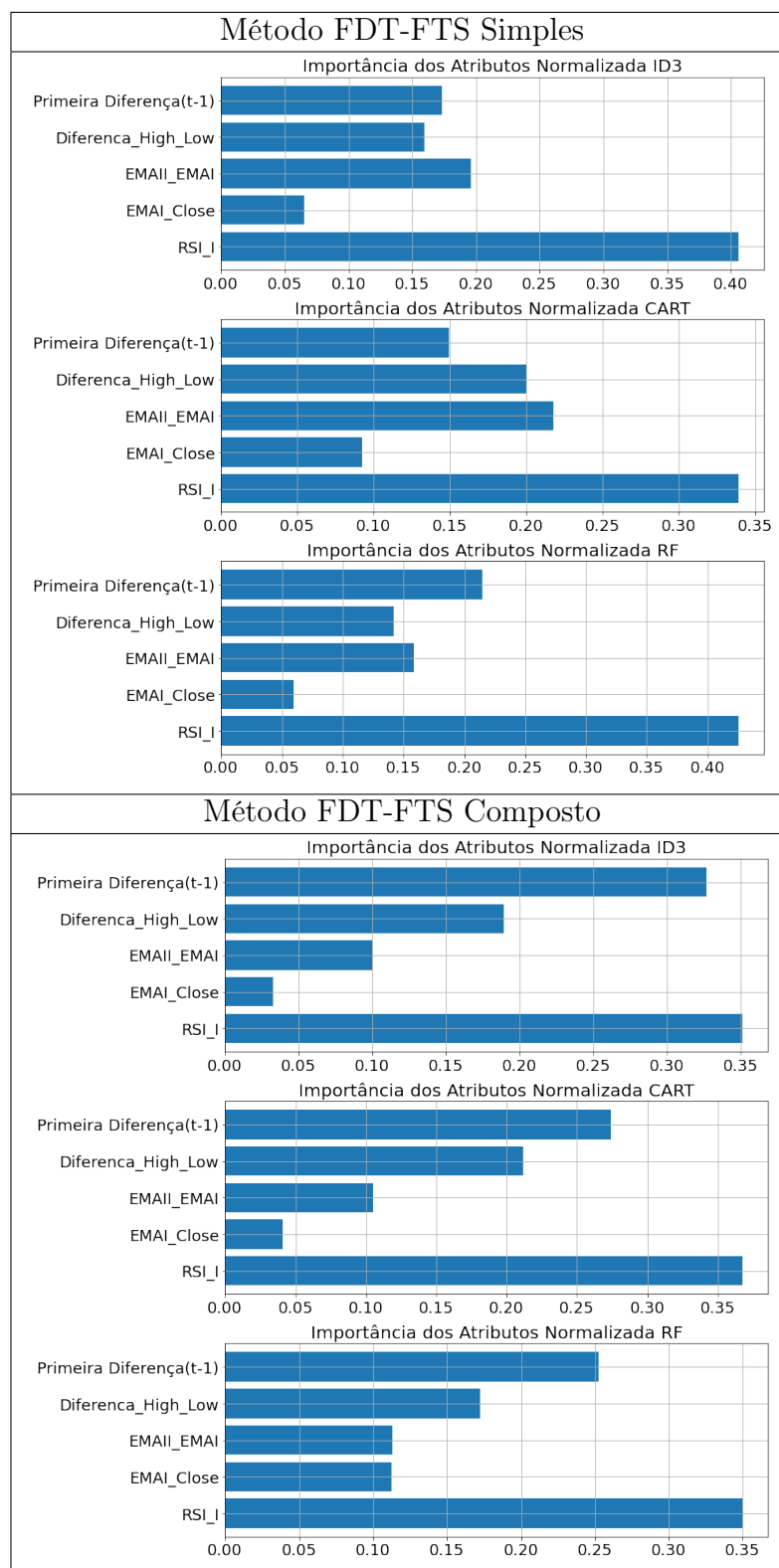


Tabela 38 – Comparação da importância entre as variáveis de entrada que os métodos atribuem para o experimento da série de primeira diferença.

Fonte: Autor

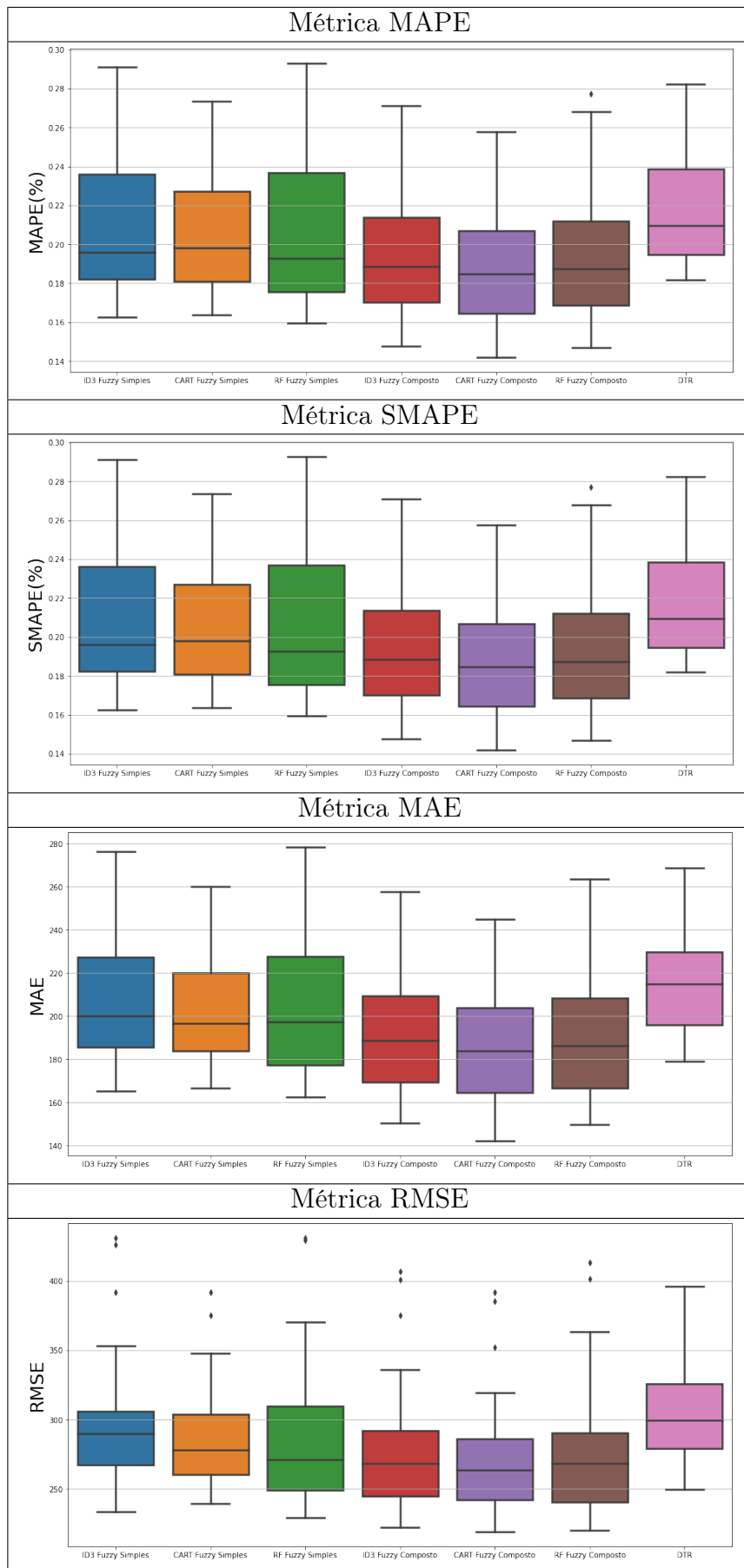


Tabela 39 – Métricas para a série do preço de fechamento expressas em *boxplots* para o método FDT-FTS simples e composto, além do método DTR.

Fonte: Autor

| Métodos | Métricas para o FDT-FTS Simples (Média $\pm$ Desvio Padrão) |                    |                    |                    |
|---------|---|--------------------|--------------------|--------------------|
|         | MAPE  | SMAPE              | RMSE               | MAE                |
| ID3     | 0.21% $\pm$ 0.03%   | 0.21 % $\pm$ 0.03% | 299.32 $\pm$ 53.55 | 209.28 $\pm$ 31.25 |
| CART    | 0.20% $\pm$ 0.03%   | 0.20% $\pm$ 0.03%  | 289.56 $\pm$ 40.23 | 203.88 $\pm$ 24.77 |
| RF      | 0.20% $\pm$ 0.03%   | 0.20% $\pm$ 0.03%  | 291.84 $\pm$ 57.45 | 205.05 $\pm$ 33.64 |

Tabela 40 – Métricas utilizadas para medição da acurácia na série do preço de fechamento para o método FDT-FTS simples.

| Métodos | Métricas para o FDT-FTS Composto (Média $\pm$ Desvio Padrão) |                   |                    |                    |
|---------|--|-------------------|--------------------|--------------------|
|         | MAPE   | SMAPE             | RMSE               | MAE                |
| ID3     | 0.19% $\pm$ 0.03%  | 0.19% $\pm$ 0.03% | 275.92 $\pm$ 54.02 | 189.24 $\pm$ 27.96 |
| CART    | 0.18% $\pm$ 0.03%  | 0.18% $\pm$ 0.03% | 274.44 $\pm$ 47.76 | 187.92 $\pm$ 27.91 |
| RF      | 0.19% $\pm$ 0.03%  | 0.19% $\pm$ 0.03% | 278.47 $\pm$ 52.59 | 193.09 $\pm$ 30.72 |

Tabela 41 – Métricas utilizadas para medição da acurácia na série do preço de fechamento para o método FDT-FTS composto.

| Métodos | Métricas para DTR (Média $\pm$ Desvio Padrão) |                   |                    |                    |
|---------|---|-------------------|--------------------|--------------------|
|         | MAPE  | SMAPE             | RMSE               | MAE                |
| DTR     | 0.21% $\pm$ 0.03%                             | 0.21% $\pm$ 0.03% | 308.07 $\pm$ 39.79 | 217.21 $\pm$ 24.16 |

Tabela 42 – Métricas utilizadas na série de preço de fechamento para o método DTR.

### 5.5.3 Discussão dos Resultados

A série do preço de fechamento prevista pelos métodos FDT-FTS apresentaram boas métricas de acurácia considerando os valores absolutos, além dos valores relativos expressos em porcentagem. Percebe-se pelas Tab. 40, Tab. 41 e Tab. 42 que a árvore CART Fuzzy gerada pelo método FDT-FTS composto obteve o melhor resultado para esse experimento. É válido apontar na Tab. 36, em que se compara a série original com a série prevista, a presença de alguns movimentos previstos de grande amplitude, em que mesmo acertando a direção do movimento a árvore extrapola o valor de um pico ou vale. No caso, a série do ativo financeiro em questão possui muitas anomalias que não foram extraídas da série original de treinamento e teste, tendo em vista que essas são recorrentes na série de contrato futuro do índice IBOVESPA, assim esses vários movimentos anômalos podem influenciar no treino da árvore.

Outro aspecto notável é relacionado ao gráfico de previsão intervalar da série de variação do preço de fechamento, em que as árvores capturam momentos em que há uma alta variância nos valores da série, produzindo saídas com maior amplitude intervalar e

quando os valores da serie possuem pouca variância, as árvores produzem saídas com intervalos curtos e que se concentram em torno de um valor médio.

Por fim, na Tab. 38 o valor da série de primeira diferença atrasada em  $t - 1$  apresentou relevância mesmo diante da autocorrelação inexistente na Fig. 37, além do indicador RSI (usado para estratégias de reversão) ter apresentado maior relevância para as árvores treinadas.

## 5.6 Considerações Finais

Os resultados apresentados objetivaram testar o método FDT-FTS simples e composto em diferentes tipos de séries de maneira a perceber como seria o comportamento do método para cada uma. Dessa forma, em face dos resultados anteriores e visto a flexibilidade das árvores para lidar com as séries temporais, abordamos tangencialmente outros aspectos relevantes no uso dessas. Primeiramente, é interessante mostrar que é possível retirar algumas análises do estágio de classificação realizada pelas árvores quanto aos conjuntos fuzzy. Como exemplo temos a Fig. 39, a qual mostra uma matriz de confusão expressa em formato de gráfico de calor, demonstrando que melhor é a classificação quanto mais populosa for a região da diagonal. Isso significa que o algoritmo de classificação proverá bons resultados não somente acertando as classes de forma exata, mas também prevendo valores em regiões próximas. Assim, os valores previstos de uma série estarão sempre em regiões próximas aos valores reais.

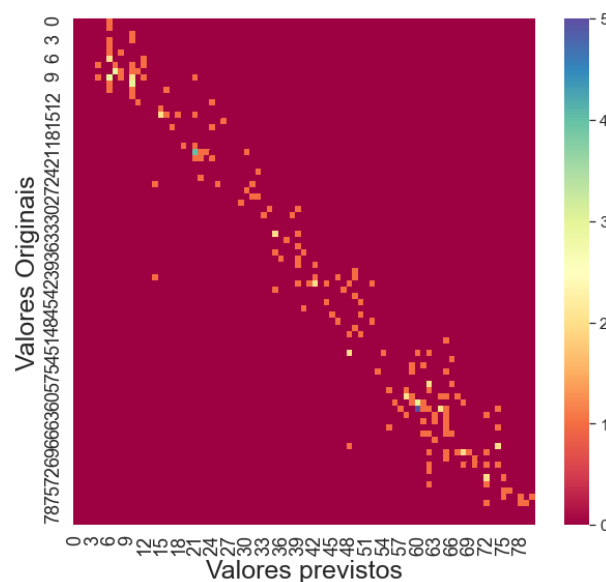


Figura 39 – Matriz de confusão do algoritmo RF Fuzzy para o método FDT-FTS simples referente ao problema do fluxo de passageiros.

Fonte: Autor

Além disso, na Fig. 40 e na Tab. 43 mostra que as regras podem ser extraídas

das árvores de maneira a explicar os pontos previstos e fornecer evidências de forma a construir algum conhecimento sobre o movimento das séries.

Na Fig. 40, tem-se a previsão da série temporal fuzzy da série de fluxo de passageiros para 50 pontos realizada pelo algoritmo ID3 Fuzzy gerado pela FDT-FTS simples. Já na Tab. 43 computamos as regras dos pontos 0 até 9 da Fig. 40, além dos endereços do nó da árvore e o valor que esses preveem. Nota-se alguns aspectos interessantes nessas regras, como o fato de diferentes regras com endereços de nós diferentes preverem o mesmo valor (ex.: regras do ponto 2 e 3) e a ocorrência de regras repetidas (ex.: regras do ponto 2 e 3). Assim, considerando a possibilidade de se compor conjuntos de regras para explicar o movimento de uma série, por consequência podemos extrair interpretações da relação da série principal com as séries externas.

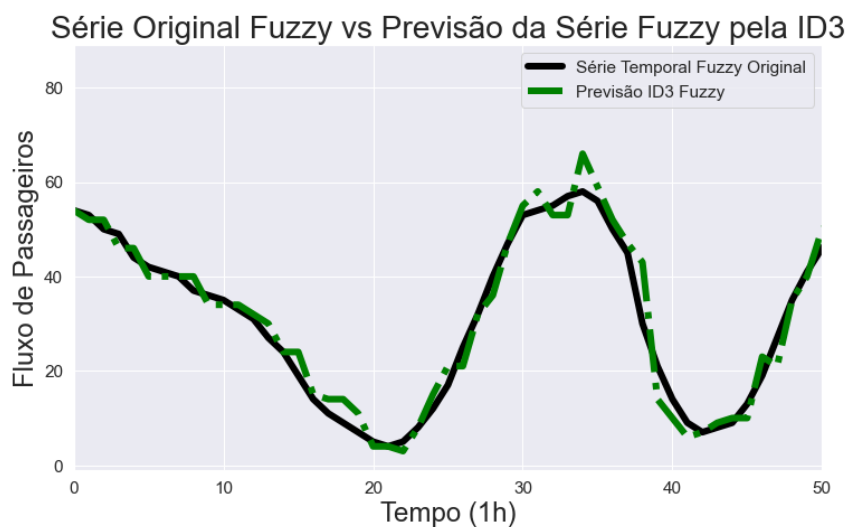


Figura 40 – Previsão da série fuzzy pelo ID3 Fuzzy gerado pelo método FDT-FTS simples para o problema de fluxo de passageiros.

Fonte: Autor

Em adição, resalta-se quais foram os melhores resultados em cada experimento, levando-se em consideração a mediana das métricas utilizadas, veja na Tab. 44. No caso, percebe-se que para boa parte dos problemas obteve-se um melhor resultado para os algoritmos RF Fuzzy Composto e CART Fuzzy Composto, quando comparados com o valor da mediana. Assim, nota-se na Tab. 44 que as árvores fuzzy apresentaram melhores resultados do que a implementação padrão da árvore de decisão de regressão presente na *Scikit Learn* em todos os experimentos, tendo destaque os resultados da Random Forest Fuzzy Composta e a CART Fuzzy Composta. É válido salientar que o melhor desempenho da Random Forest Fuzzy Composta era esperado tendo em vista que se trata de um método *ensemble*. É importante, também, ressaltar que a comparação feita na Tab. 44 evidencia que o uso dos métodos de fuzzificação e defuzzificação propostos neste trabalho modificam a forma como uma série é tratada uma vez discretizada em conjuntos fuzzy.

| Regras do ponto 0 até 4:   |
|--|
| <p>Endereço do nó folha: 11<br/> Regra para o ponto: 0<br/> (Atributo['Trafego_Delay_t_1']&lt;= 47) e (Atributo['Trafego_Delay_t_1']&lt;= 24) e (Atributo['Trafego_Delay_t_1']&lt;= 9) e (Atributo['Trafego_Delay_t_1']&lt;= 6) e (Atributo['Trafego_Delay_t_1']&lt;= 4) e (Atributo['chuva']&lt;= 0) e (Atributo['Hora']&lt;= 2) e (Atributo['Trafego_Delay_t_1']&lt;= 3) e (Atributo['Nublado']&lt;= 5) e (Atributo['Temperatura']&gt; 8)<br/> Valor previsto pela regra: 54</p> <p>-----</p> <p>Endereço do nó folha: 26<br/> Regra para o ponto: 1<br/> (Atributo['Trafego_Delay_t_1']&lt;= 47) e (Atributo['Trafego_Delay_t_1']&lt;= 24) e (Atributo['Trafego_Delay_t_1']&lt;= 9) e (Atributo['Trafego_Delay_t_1']&lt;= 6) e (Atributo['Trafego_Delay_t_1']&lt;= 4) e (Atributo['chuva']&lt;= 0) e (Atributo['Hora']&gt; 2) e (Atributo['Hora']&lt;= 3) e (Atributo['Nublado']&lt;= 7) e (Atributo['Nublado']&gt; 3)<br/> Valor previsto pela regra: 52</p> <p>-----</p> <p>Endereço do nó folha: 161<br/> Regra para o ponto: 2<br/> (Atributo['Trafego_Delay_t_1']&lt;= 47) e (Atributo['Trafego_Delay_t_1']&lt;= 24) e (Atributo['Trafego_Delay_t_1']&gt; 9) e (Atributo['Trafego_Delay_t_1']&lt;= 15) e (Atributo['Hora']&gt; 3) e (Atributo['Hora']&lt;= 4) e (Atributo['Trafego_Delay_t_1']&lt;= 11) e (Atributo['Trafego_Delay_t_1']&gt; 10) e (Atributo['Nublado']&lt;= 7) e (Atributo['Nublado']&lt;= 3)<br/> Valor previsto pela regra: 52</p> <p>-----</p> <p>Endereço do nó folha: 396<br/> Regra para o ponto: 3<br/> (Atributo['Trafego_Delay_t_1']&lt;= 47) e (Atributo['Trafego_Delay_t_1']&gt; 24) e (Atributo['Trafego_Delay_t_1']&gt; 37) e (Atributo['Trafego_Delay_t_1']&lt;= 42) e (Atributo['Hora']&lt;= 6) e (Atributo['Trafego_Delay_t_1']&gt; 38) e (Atributo['Temperatura']&gt; 8) e (Atributo['Trafego_Delay_t_1']&lt;= 40) e (Atributo['Nublado']&lt;= 7) e (Atributo['Nublado']&lt;= 3)<br/> Valor previsto pela regra: 46</p> <p>-----</p> <p>Endereço do nó folha: 909<br/> Regra para o ponto: 4<br/> (Atributo['Trafego_Delay_t_1']&gt; 47) e (Atributo['Trafego_Delay_t_1']&gt; 62) e (Atributo['Trafego_Delay_t_1']&gt; 71) e (Atributo['Hora']&lt;= 9) e (Atributo['Hora']&lt;= 6) e (Atributo['Trafego_Delay_t_1']&lt;= 76) e (Atributo['Trafego_Delay_t_1']&lt;= 75) e (Atributo['Nublado']&lt;= 5) e (Atributo['Trafego_Delay_t_1']&gt; 74) e (Atributo['Temperatura']&gt; 8)<br/> Valor previsto pela regra: 46</p> <p>-----</p> |
| Regras do ponto 5 até 9:   |
| <p>-----</p> <p>Endereço do nó folha: 851<br/> Regra para o ponto: 5<br/> (Atributo['Trafego_Delay_t_1']&gt; 47) e (Atributo['Trafego_Delay_t_1']&gt; 62) e (Atributo['Trafego_Delay_t_1']&lt;= 71) e (Atributo['Trafego_Delay_t_1']&gt; 66) e (Atributo['Hora']&lt;= 8) e (Atributo['Trafego_Delay_t_1']&gt; 68) e (Atributo['Nublado']&lt;= 4) e (Atributo['Trafego_Delay_t_1']&gt; 69)<br/> Valor previsto pela regra: 40</p> <p>-----</p> <p>Endereço do nó folha: 851<br/> Regra para o ponto: 6<br/> (Atributo['Trafego_Delay_t_1']&gt; 47) e (Atributo['Trafego_Delay_t_1']&gt; 62) e (Atributo['Trafego_Delay_t_1']&lt;= 71) e (Atributo['Trafego_Delay_t_1']&gt; 66) e (Atributo['Hora']&lt;= 8) e (Atributo['Trafego_Delay_t_1']&gt; 68) e (Atributo['Nublado']&lt;= 4) e (Atributo['Trafego_Delay_t_1']&gt; 69)<br/> Valor previsto pela regra: 40</p> <p>-----</p> <p>Endereço do nó folha: 731<br/> Regra para o ponto: 7<br/> (Atributo['Trafego_Delay_t_1']&gt; 47) e (Atributo['Trafego_Delay_t_1']&lt;= 62) e (Atributo['Trafego_Delay_t_1']&gt; 57) e (Atributo['Trafego_Delay_t_1']&gt; 59) e (Atributo['Trafego_Delay_t_1']&gt; 60) e (Atributo['Hora']&lt;= 10) e (Atributo['Nublado']&gt; 2) e (Atributo['chuva']&lt;= 1) e (Atributo['Trafego_Delay_t_1']&gt; 61)<br/> Valor previsto pela regra: 40</p> <p>-----</p> <p>Endereço do nó folha: 549<br/> Regra para o ponto: 8<br/> (Atributo['Trafego_Delay_t_1']&gt; 47) e (Atributo['Trafego_Delay_t_1']&lt;= 62) e (Atributo['Trafego_Delay_t_1']&lt;= 57) e (Atributo['Hora']&lt;= 11) e (Atributo['Trafego_Delay_t_1']&gt; 53) e (Atributo['Nublado']&lt;= 7) e (Atributo['Hora']&lt;= 9) e (Atributo['Trafego_Delay_t_1']&lt;= 56) e (Atributo['Trafego_Delay_t_1']&lt;= 55)<br/> Valor previsto pela regra: 40</p> <p>-----</p> <p>Endereço do nó folha: 727<br/> Regra para o ponto: 9<br/> (Atributo['Trafego_Delay_t_1']&gt; 47) e (Atributo['Trafego_Delay_t_1']&lt;= 62) e (Atributo['Trafego_Delay_t_1']&gt; 57) e (Atributo['Trafego_Delay_t_1']&gt; 59) e (Atributo['Trafego_Delay_t_1']&gt; 60) e (Atributo['Hora']&lt;= 10) e (Atributo['Nublado']&lt;= 2) e (Atributo['Trafego_Delay_t_1']&gt; 61) e (Atributo['Hora']&gt; 9)<br/> Valor previsto pela regra: 34</p> <p>-----</p>   |

Tabela 43 – Regras extraídas da árvore de decisão ID3 fuzzy gerada pelo método FDT-FTS simples relativa a série da Fig. 40.

Fonte: Autor

Isso permite um maior ganho na simplicidade como a série é tratada pelos algoritmos de classificação, além de demonstrar que os algoritmos de classificação tradicionais como as árvores podem ter um ganho de desempenho notável pela simples transformação da série numérica em uma série fuzzy.

Ademais, outra característica do método proposto trata da capacidade de se

combinar atributos numéricos e categóricos nas variáveis de entrada para o treino das árvores. Além disso, ao se testar uma maior ou menor discretização das séries presentes nas variáveis de entrada e saída, torna-se possível a redução da sensibilidade da árvore em relação aos ruídos das séries previstas, uma vez que ao discretizar a série em poucos conjuntos fuzzy captura-se as variações de maior amplitude do movimento de uma série e à medida que se aumenta a discretização, torna-se possível capturar os movimentos de menor amplitude.

| Experimentos | Melhores resultados considerando a mediana das métricas. |               |               |               |
|--------------|--|---------------|---------------|---------------|
|              | MAPE   | SMAPE         | RMSE          | MAE           |
| QoS          | CART F. Comp.  | RF F. Comp.   | RF F. Comp.   | RF F. Comp.   |
| Rio Xingu    | RF F. Comp.  | RF F. Comp.   | RF F. Comp.   | RF F. Comp.   |
| Energia      | CART F. Comp.  | CART F. Comp. | RF F. Comp.   | CART F. Simp. |
| Mêtro        | ID3 F. Comp.   | ID3 F. Comp.  | ID3 F. Comp.  | RF F. Comp.   |
| IBOVESPA     | CART F. Comp.  | CART F. Comp. | CART F. Comp. | CART F. Comp. |

Tabela 44 – Comparação dos métodos em face aos resultados das medianas proveniente das métricas utilizadas.

Por fim, no capítulo seguinte apresentaremos a conclusão e teceremos as considerações finais acerca do trabalho realizado.





## Capítulo 6

### Conclusão

“Nunca nos devemos deixar  
persuadir senão por evidência da  
razão.”

---

René Descartes

O presente trabalho propôs um método com duas variações para a geração de árvores de decisão fuzzy a partir de métodos já implementados de árvores de classificação. Esse método, nomeado como FDT-FTS, utilizou da pré-fuzzificação de atributos e posterior defuzzificação para permitir que qualquer árvore de decisão de classificação possa prever séries temporais multivariadas. Tal método de previsão utiliza os conceitos de árvore, de lógica fuzzy e dos trabalhos que utilizam métodos de discretização de séries e algoritmos de classificação para previsão de vários tipos de séries. No caso, a discretização das séries é realizada pela fuzzificação dessas por funções de pertinência, transformando uma série temporal numérica numa série temporal fuzzy. Assim, os algoritmos de árvore podem ser utilizados para classificar as classes de uma série, no caso, conjuntos fuzzy e prever o próximo valor ou região em que um determinado valor ocorrerá.

O FDT-FTS em suas duas variações (simples e composta) foi testada em diferentes séries temporais com diferentes propriedades (séries não estacionárias, estacionárias, alta autocorrelação, baixa autocorrelação, etc.), além de ter sido comparada com o método de árvore de decisão presente na *Scikit-Learn*, o qual utiliza valores numéricos e não fuzzy. Para isso testou a acurácia da FDT-FTS em quatro métricas distintas (MAPE, SMAPE, RMSE e MAE), cada qual com diferentes vantagens e desvantagens no que tange as análises feitas e os tipos de séries previstas. Vale ressaltar que os resultados apresentados mostraram que a FDT-FTS logrou êxito ao prever séries temporais pelo uso de regras com atributos fuzzy, em que boa parte dos valores das métricas tanto relativas quanto absolutas apresentaram erros pequenos e úteis a tomada de decisão, além de permitir a explicabilidade quanto aos movimentos das séries coletadas pela tradução do grafo das

árvores em regras.

Além das métricas usadas, as quais permitem estimar um erro na previsão das séries temporais, tem-se nas árvores do método FDT-FTS composto a possibilidade da previsão intervalar levando em conta o fato de que essas ativam várias saídas para prever um único ponto da série. Isso possibilita novas análises quanto a variância e a acurácia das previsões, uma vez que reflete, por vezes, as oscilações bruscas ou mais suaves da própria série que se deseja prever.

Em adição, deve-se enaltecer a confluência das várias linhas de pesquisa que foram base para esse trabalho e que permitiram a produção de funções que combinam a lógica fuzzy com árvores de decisão, dando novas utilidades a essas e permitindo inúmeros novos caminhos a serem explorados futuramente. Dentre esses novos caminhos futuros, pode-se abordar a otimização dos hiperparâmetros do método FDT-FTS, podendo ser mono ou multiobjetivo (otimizar com base em uma métrica ou várias métricas de regressão). Adicionalmente, seria relevante testar o método FDT-FTS em diversas outras árvores como as árvores oblíquas e árvores evolutivas. Além do mais, a ideia presente no âmago do método FDT-FTS pode ser expandida e modificada para abarcar problemas de classificação.

Por fim, destaca-se a relevância deste trabalho não somente por alcançar os principais objetivos traçados e mostrados no capítulo introdutório, mas de propor a ideia de rebuscar e combinar conceitos da lógica fuzzy e das árvores de decisão, a fim de gerar um método simples, com boa acurácia e que possa prover explicabilidade e interpretabilidade nos problemas de previsão que abordam séries temporais multivariadas.

## Referências

- N. Amado, J. Gama, and F. Silva. Parallel implementation of decision tree learning algorithms. volume 2258, pages 6–13, 12 2001. ISBN 978-3-540-43030-8. doi: 10.1007/3-540-45329-6\_4.
- M. Barros. *Processos Estocásticos*. Papel Virtual, Rio de Janeiro, 1 edition, 2004. ISBN 8574930725.
- R. Barros, R. Cerri, P. A Jaskowiak, A. de Carvalho, and A. F. A bottom-up oblique decision tree induction algorithm. 11 2011. doi: 10.1109/ISDA.2011.6121697.
- U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. F. Moura, and P. Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAT\* '20*, page 648–657, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369367. doi: 10.1145/3351095.3375624. URL <https://doi.org/10.1145/3351095.3375624>.
- O. Biran and C. Cotton. Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI (XAI)*, volume 8, pages 8–13, 2017.
- P. Bonissone, J. M. Cadenas, M. Carmen Garrido, and R. Andrés Díaz-Valladares. A fuzzy random forest. *International Journal of Approximate Reasoning*, 51(7):729–747, 2010. ISSN 0888-613X. doi: <https://doi.org/10.1016/j.ijar.2010.02.003>. URL <https://www.sciencedirect.com/science/article/pii/S0888613X10000435>.
- A. d. P. Braga, T. B. Ludermir, and A. C. P. d. L. F. Carvalho. *Redes neurais artificiais: teoria e aplicações*. LTC, 2000.
- L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL <https://doi.org/10.1023/A:1010933404324>.
- L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.
- L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt,

- and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.
- L. M. Camarinha-Matos and F. J. Martinelli. Application of machine learning in water distribution networks. *Intelligent Data Analysis*, 2(1):311–332, 1998. ISSN 1088-467X. doi: [https://doi.org/10.1016/S1088-467X\(98\)00030-4](https://doi.org/10.1016/S1088-467X(98)00030-4). URL <https://www.sciencedirect.com/science/article/pii/S1088467X98000304>.
- R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 1721–1730, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2788613. URL <https://doi.org/10.1145/2783258.2788613>.
- A. Chandramohan, M. V. C. Rao, and M. S. Arumugam. Two new and useful defuzzification methods based on root mean square value. *Soft computing (Berlin, Germany)*, 10(11): 1047–1059, 2006. ISSN 1432-7643.
- C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- E. Charniak. Bayesian networks without tears. *AI Magazine*, 12(4):50, Dec. 1991. doi: 10.1609/aimag.v12i4.918. URL <https://ojs.aaai.org/index.php/aimagazine/article/view/918>.
- F. A. L. Cichini. Aplicação de um sistema de inferência fuzzy de suporte à decisão para estimação de valores de ações cotadas na bolsa de valores de são paulo. Master's thesis, Universidade Estadual Paulista, Faculdade de Engenharia de Bauru, 02 2009.
- M. Cintra, M.-C. Monard, and H. Camargo. A fuzzy decision tree algorithm based on c4.5 - mathware soft computing. 20:56–62, 01 2013.
- M. E. Cintra, H. A. Camargo, and T. P. Martin. Optimising the fuzzy granulation of attribute domains. In *IFSA/EUSFLAT Conf.*, 2009.
- W. J. Clancey and E. H. Shortliffe. *Readings in Medical Artificial Intelligence: The First Decade*. Addison-Wesley Longman Publishing Co., Inc., USA, 1984. ISBN 0201108542.
- B. F. Crabtree, S. C. Ray, P. M. Schmidt, P. T. O'Connor, and D. D. Schmidt. The individual over time: time series applications in health care research. *Journal of clinical epidemiology*, 43(3):241–260, 1990.

- J. G. De Gooijer and R. J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443–473, 2006. ISSN 0169-2070. doi: <https://doi.org/10.1016/j.ijforecast.2006.01.001>. URL <https://www.sciencedirect.com/science/article/pii/S0169207006000021>. Twenty five years of forecasting.
- P. C. De Lima e Silva, C. A. Severiano, M. A. Alves, R. Silva, M. W. Cohen, and F. G. Guimarães. Forecasting in non-stationary environments with fuzzy time series. *Applied Soft Computing*, 97:106825, 2020. ISSN 1568-4946. doi: <https://doi.org/10.1016/j.asoc.2020.106825>. URL <https://www.sciencedirect.com/science/article/pii/S1568494620307638>.
- M. Du, N. Liu, and X. Hu. Techniques for interpretable machine learning. *Commun. ACM*, 63(1):68–77, Dec. 2019. ISSN 0001-0782. doi: 10.1145/3359786. URL <https://doi.org/10.1145/3359786>.
- J. L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. ISSN 0364-0213. doi: [https://doi.org/10.1016/0364-0213\(90\)90002-E](https://doi.org/10.1016/0364-0213(90)90002-E). URL <https://www.sciencedirect.com/science/article/pii/036402139090002E>.
- T. Evgeniou and M. Pontil. *Support Vector Machines: Theory and Applications*, pages 249–257. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. ISBN 978-3-540-44673-6. doi: 10.1007/3-540-44673-7\_12. URL [https://doi.org/10.1007/3-540-44673-7\\_12](https://doi.org/10.1007/3-540-44673-7_12).
- D. Falessi, J. Huang, L. Narayana, J. F. Thai, and B. Turhan. On the need of preserving order of data when validating within-project defect classifiers. *Empir. Softw. Eng.*, 25(6):4831–4832, 2020.
- P. Ferguson and G. Huston. *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*. Wiley, 1998. ISBN 9780471243588. URL <https://books.google.com.br/books?id=rulSAAAAMAAJ>.
- G. Fernandes Costa. Previsão multi-passos em séries temporais: Estratégias clássicas e de aprendizagem automática. Master’s thesis, Universidade do Porto, Faculdade de Economia, 2015.
- S. C. Garcia. O uso de árvores de decisão na descoberta de conhecimento na área da saúde. Master’s thesis, Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.
- A. Geron. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. O’Reilly Media, Sebastopol, CA, 2017. ISBN 978-1491962299.
- J. C. Giarratano and G. Riley. *Expert systems: principles and programming*. Brooks/Cole Publishing Co., 1989.

- D. Gkatzia, O. Lemon, and V. Rieser. Natural language generation enhances human decision-making with uncertain information. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 264–268, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2043. URL <https://aclanthology.org/P16-2043>.
- F. Gomide, R. R. Gudwin, and R. Tanscheit. Conceitos fundamentais da teoria de conjuntos fuzzy, lógica fuzzy e aplicações. In *Proc. 6 th IFSA Congress-Tutorials*, volume 59, pages 1–38, 1995.
- S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, Nov. 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL <https://doi.org/10.1162/neco.1997.9.8.1735>.
- W. Hofkirchner and M. Schafranek. General system theory. In C. Hooker, editor, *Philosophy of Complex Systems*, volume 10 of *Handbook of the Philosophy of Science*, pages 177–194. North-Holland, Amsterdam, 2011. doi: <https://doi.org/10.1016/B978-0-444-52076-0.50006-7>. URL <https://www.sciencedirect.com/science/article/pii/B9780444520760500067>.
- B. Hssina, A. Merbouha, H. Ezzikouri, and M. Erritali. A comparative study of decision tree id3 and c4.5. *International Journal of Advanced Computer Science and Applications*, 4, 2014.
- N. Idris and M. A. Ismail. Breast cancer disease classification using fuzzy-id3 algorithm with fuzzydbd method: automatic fuzzy database definition. *PeerJ Computer Science*, 7, 2021.
- N. Indurkha and S. M. Weiss. Solving regression problems with rule-based ensemble classifiers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01*, page 287–292, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 158113391X. doi: 10.1145/502512.502553. URL <https://doi.org/10.1145/502512.502553>.
- H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks—with an erratum note'. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148, 01 2001.
- J. Janai, F. Güney, A. Behl, and A. Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *Found. Trends Comput. Graph. Vis.*, 12:1–308, 2020.

- J. Jang, C. Sun, and E. Mizutani. *Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. MATLAB curriculum series. Prentice Hall, 1997. ISBN 9780132610667.
- J.-S. Jang. Structure determination in fuzzy modeling: a fuzzy cart approach. In *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*, pages 480–485 vol.1, 1994. doi: 10.1109/FUZZY.1994.343738.
- C. Janikow. Fuzzy decision trees: issues and methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(1):1–14, 1998. doi: 10.1109/3477.658573.
- C.-F. Jeffrey Kuo, P.-C. Wang, Y.-H. Chu, H.-W. Wang, and C.-Y. Lai. Using image processing technology combined with decision tree algorithm in laryngeal video stroboscope automatic identification of common vocal fold diseases. *Computer Methods and Programs in Biomedicine*, 112(1):228–236, 2013. ISSN 0169-2607. doi: <https://doi.org/10.1016/j.cmpb.2013.06.021>. URL <https://www.sciencedirect.com/science/article/pii/S0169260713002204>.
- B. Kim, R. Khanna, and O. O. Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL <https://proceedings.neurips.cc/paper/2016/file/5680522b8e2bb01943234bce7bf84534-Paper.pdf>.
- S. B. Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39: 261–283, 2011.
- C. Lacave and F. J. Díez. A review of explanation methods for bayesian networks. *The Knowledge Engineering Review*, 17(2):107–127, 2002. doi: 10.1017/S026988890200019X.
- R. K. Lai, C.-Y. Fan, W.-H. Huang, and P.-C. Chang. Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Systems with Applications*, 36(2, Part 2):3761–3773, 2009. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2008.02.025>. URL <https://www.sciencedirect.com/science/article/pii/S0957417408001474>.
- Y. Lertworaprachaya, Y. Yang, and R. John. Interval-valued fuzzy decision trees. In *International Conference on Fuzzy Systems*, pages 1–7, 2010. doi: 10.1109/FUZZY.2010.5583973.
- Y. Lertworaprachaya, Y. Yang, and R. John. Interval-valued fuzzy decision trees with optimal neighbourhood perimeter. *Applied Soft Computing*, 24:851–866, 2014. ISSN 1568-4946. doi: <https://doi.org/10.1016/j.asoc.2014.08.060>. URL <https://www.sciencedirect.com/science/article/pii/S1568494614004256>.

- H. P. d. Lima. Uma abordagem para construção de sistemas fuzzy baseados em regras integrando conhecimento de especialistas e extraído de dados. Master's thesis, Universidade Federal de São Carlos, São Carlos, 2015. Disponível em: <https://repositorio.ufscar.br/handle/ufscar/7224>.
- P. C. d. Lima e Silva. *Scalable Models For Probabilistic Forecasting With Fuzzy Time Series*. PhD thesis, Universidade Federal de Minas Gerais - UFMG, 2019. URL <https://doi.org/10.5281/zenodo.3449195>.
- W.-Y. Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1:14 – 23, 01 2011. doi: 10.1002/widm.8.
- M. V. R. Lopes. Tratamento de imprecisão na geração de árvores de decisão. Master's thesis, Universidade Federal de São Carlos, São Carlos, 03 2016. URL <https://repositorio.ufscar.br/handle/ufscar/8954>.
- J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2019. doi: 10.1109/TKDE.2018.2876857.
- G. Luger. *Inteligência Artificial*. PEARSON BRASIL, 2013. ISBN 9788581435503.
- A. G. Maia. *ECONOMETRIA - CONCEITOS E APLICAÇÕES: CONCEITOS E APLICAÇÕES*. Saint Paul Institute of Finance, 1 edition, 2017. ISBN 9788580041286.
- S. Makridakis and M. Hibon. *Evaluating Accuracy (or Error) Measures*. INSEAD working paper. INSEAD, 1995. URL <https://books.google.com.br/books?id=pdRYGwAACAAJ>.
- E. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. *Proceedings of the Institution of Electrical Engineers*, 121:1585–1588(3), December 1974. ISSN 0020-3270. URL <https://digital-library.theiet.org/content/journals/10.1049/piee.1974.0328>.
- Q. Meng, G. Ke, T. Wang, W. Chen, Q. Ye, Z.-M. Ma, and T.-Y. Liu. A communication-efficient parallel algorithm for decision tree. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL <https://proceedings.neurips.cc/paper/2016/file/10a5ab2db37feedfdeaab192ead4ac0e-Paper.pdf>.
- I. D. Mienye, Y. Sun, and Z. Wang. Prediction performance of improved decision tree-based algorithms: a review. *Procedia Manufacturing*, 35:698–703, 2019. ISSN 2351-9789. doi: <https://doi.org/10.1016/j.promfg.2019.06.011>. URL <https://www.sciencedirect.com/science/article/pii/S235197891930736X>. The 2nd International Conference



- on Sustainable Materials Processing and Manufacturing, SMPM 2019, 8-10 March 2019, Sun City, South Africa.
- M. Milanovic and M. Stamenkovic. Chaid decision tree: Methodological frame and application. *Economic Themes*, 54, 12 2016. doi: 10.1515/ethemes-2016-0029.
- D. Montgomery, C. Jennings, and M. Kulahci. *Introduction to Time Series Analysis and Forecasting*. Wiley Series in Probability and Statistics. Wiley, 2015. ISBN 9781118745151. URL <https://books.google.com.br/books?id=Xeh8CAAAQBAJ>.
- D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley and Sons, 2003.
- P. A. Morettin and C. M. d. C. Toloí. *Análise de séries temporais*. Edgard Blucher, 2004.
- A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2015. doi: 10.1109/CVPR.2015.7298640.
- S. A. Otto, M. Kadin, M. Casini, M. A. Torres, and T. Blenckner. A quantitative framework for selecting and validating food web indicators. *Ecological Indicators*, 84:619–631, 2018. ISSN 1470-160X. doi: <https://doi.org/10.1016/j.ecolind.2017.05.045>. URL <https://www.sciencedirect.com/science/article/pii/S1470160X1730300X>.
- J.-z. Pang, Su-lin ;Gong. C5.0 classification algorithm and application on individual credit evaluation of banks. *Systems Engineering - Theory Practice*, 29(12):94–104, 2009. ISSN 1874-8651. doi: [https://doi.org/10.1016/S1874-8651\(10\)60092-0](https://doi.org/10.1016/S1874-8651(10)60092-0). URL <https://www.sciencedirect.com/science/article/pii/S1874865110600920>.
- A. Pankratz. *Forecasting with univariate Box-Jenkins models : concepts and cases / Alan Pankratz*. Wiley New York, 1983. ISBN 0471090239.
- M. Pechenizkiy and I. Zliobaite. Handling concept drift in medical applications: Importance, challenges and solutions. In *2010 IEEE 23rd International Symposium on Computer-Based Medical Systems (CBMS)*, pages 5–5, 2010. doi: 10.1109/CBMS.2010.6042653.
- F. Pedregosa, G. Varoquaux, A. Gramfort, B. Michel, V. and Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- J. R. Quinlan. Induction of decision trees. *MACH. LEARN*, 1:81–106, 1986.
- J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1558602380.

- J. R. Quinlan. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.
- J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.
- J. A. RESENDE, M. D. V. de; STURION. *Análise genética de dados com dependência espacial e temporal no melhoramento de plantas perenes via modelos geoestatísticos e de séries temporais empregando REML/BLUP ao nível individual*, volume 1. Embrapa Cerrados, 2001.
- L. Rokach and O. Maimon. *Decision Trees*, pages 165–192. Springer US, Boston, MA, 2005. ISBN 978-0-387-25465-4. doi: 10.1007/0-387-25465-X\_9. URL [https://doi.org/10.1007/0-387-25465-X\\_9](https://doi.org/10.1007/0-387-25465-X_9).
- T. Ross. *Fuzzy Logic with Engineering Applications*. Wiley, 2009. ISBN 9780470748510. URL [https://books.google.com.br/books?id=nhz1f9j6\\_SMC](https://books.google.com.br/books?id=nhz1f9j6_SMC).
- S. Ryu, J. Noh, and H. Kim. Deep neural network based demand side short term load forecasting. *Energies*, 10(1), 2017. ISSN 1996-1073. doi: 10.3390/en10010003. URL <https://www.mdpi.com/1996-1073/10/1/3>.
- J. J. Saade and H. B. Diab. Defuzzification methods and new techniques for fuzzy controllers. *Iranian journal of electrical and computer engineering*, 3:161–174, 2004.
- P. Sadowsky. A random forests approach to predicting clean energy stock prices. *Journal of Risk and Financial Management*, 14(2), 2021. ISSN 1911-8074. doi: 10.3390/jrfm14020048. URL <https://www.mdpi.com/1911-8074/14/2/48>.
- R. Salman and V. Kecman. Regression as classification. In *2012 Proceedings of IEEE Southeastcon*, pages 1–6, 2012a. doi: 10.1109/SECon.2012.6196887.
- R. Salman and V. Kecman. Regression as classification. pages 1–6, 03 2012b. ISBN 978-1-4673-1374-2. doi: 10.1109/SECon.2012.6196887.
- J. Sen and S. Mehtab. A time series analysis-based stock price prediction using machine learning and deep learning models. *International Journal of Business Forecasting and Marketing Intelligence*, 6(1):272, 2020. doi: 10.1504/ijbfmi.2020.10038524.
- O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu. Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. *Applied Soft Computing*, 90: 106181, 2020. ISSN 1568-4946. doi: <https://doi.org/10.1016/j.asoc.2020.106181>. URL <https://www.sciencedirect.com/science/article/pii/S1568494620301216>.
- Q. Song and B. Chissom. Fuzzy time series and its model. *Fuzzy Sets and Systems*, 54: 269–277, 03 1993a. doi: 10.1016/0165-0114(93)90372-O.

- Q. Song and B. S. Chissom. Forecasting enrollments with fuzzy time series — part i. *Fuzzy Sets and Systems*, 54(1):1–9, 1993b. ISSN 0165-0114. doi: [https://doi.org/10.1016/0165-0114\(93\)90355-L](https://doi.org/10.1016/0165-0114(93)90355-L). URL <https://www.sciencedirect.com/science/article/pii/016501149390355L>.
- M. Steurer, R. J. Hill, and N. Pfeifer. Metrics for evaluating the performance of machine learning based automated valuation models. *Journal of Property Research*, 38(2):99–129, 2021. doi: 10.1080/09599916.2020.1858937.
- J. Stoyanov. Handbook of financial time series by t. g. andersen; r. a. davis; j.-p. kreiss; t. mikosch. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 173, 01 2010. doi: 10.2307/40925407.
- T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15:116–132, 1985.
- R. Tanscheit. Sistemas fuzzy. *Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro*, pages 338–353, 2004.
- L. Torgo and J. Gama. Regression using classification algorithms. *Intelligent Data Analysis*, 1(1):275–292, 1997. ISSN 1088-467X. doi: [https://doi.org/10.1016/S1088-467X\(97\)00013-9](https://doi.org/10.1016/S1088-467X(97)00013-9). URL <https://www.sciencedirect.com/science/article/pii/S1088467X97000139>.
- J. F. V. Trindade. Interpretabilidade em modelos de sistemas de recomendação. Master’s thesis, Universidade do Porto, 2020.
- R. S. Tsay. *Multivariate time series analysis: with R and financial applications*. John Wiley & Sons, 2013.
- R. E. Turkson, E. Y. Baagyere, and G. E. Wenya. A machine learning approach for predicting bank credit worthiness. In *2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR)*, pages 1–7, 2016. doi: 10.1109/ICAIPR.2016.7585216.
- M. Umanol, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, and J. Kinoshita. Fuzzy decision trees by fuzzy id3 algorithm and its application to diagnosis systems. In *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*, pages 2113–2118 vol.3, 1994. doi: 10.1109/FUZZY.1994.343539.
- A. Vieira, P. De Souza, J. Ueyama, G. Pessin, L. Perdigão Cota, G. Garcia, and R. Correa Pabon. Improving flood forecasting through feature selection by a genetic algorithm -experiments based on real data from an amazon rainforest river. *Earth Science Informatics*, 03 2021. doi: 10.1007/s12145-020-00528-8.

- M. L. Wang. Advanced multivariate time series forecasting models. *Journal of Mathematics and Statistics*, 14(1):253–260, Dec. 2018. doi: 10.3844/jmssp.2018.253.260. URL <https://thescpub.com/abstract/jmssp.2018.253.260>.
- R. A. Watanabe. Um estudo sobre um método de defuzzificação para eventos fuzzy em sistemas baseados em regras. Master’s thesis, Universidade Estadual de Campinas, Instituto de Matemática Estatística e Computação Científica, Campinas, 2016.
- S. M. Weiss and N. Indurkha. Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research*, 3:383–403, 1995.
- I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, Amsterdam, 3 edition, 2011. ISBN 978-0-12-374856-0. URL <http://www.sciencedirect.com/science/book/9780123748560>.
- L. Ye and P. Johnson. The impact of explanation facilities on user acceptance of expert systems advice. *MIS Quarterly: Management Information Systems*, 19(2):157–172, June 1995. ISSN 0276-7783. doi: 10.2307/249686.
- Y. Yuan and M. J. Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, 69(2):125–139, 1995. ISSN 0165-0114. doi: [https://doi.org/10.1016/0165-0114\(94\)00229-Z](https://doi.org/10.1016/0165-0114(94)00229-Z). URL <https://www.sciencedirect.com/science/article/pii/016501149400229Z>.
- L. A. Zadeh. Fuzzy sets. 8(3):338–353, 1965. ISSN 0019-9958. doi: [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X). URL <https://www.sciencedirect.com/science/article/pii/S001999586590241X>.
- P. Zhang, L. Wang, W. Li, H. Leung, and W. Song. A web service qos forecasting approach based on multivariate time series. In *2017 IEEE International Conference on Web Services (ICWS)*, pages 146–153, 2017. doi: 10.1109/ICWS.2017.27.
- X. Zhang, T. Zhang, A. A. Young, and X. Li. Applications and comparisons of four time series models in epidemiological surveillance data. *PLOS ONE*, 9(2):1–16, 02 2014. doi: 10.1371/journal.pone.0088075. URL <https://doi.org/10.1371/journal.pone.0088075>.