

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

Patrícia de Oliveira e Lucas

**Um Novo Método de AutoML para Séries Temporais: Decomposição,  
Causalidade, Otimização Evolutiva Multitarefa e Agentes**

Belo Horizonte

2026

Patrícia de Oliveira e Lucas

**Um Novo Método de AutoML para Séries Temporais: Decomposição,  
Causalidade, Otimização Evolutiva Multitarefa e Agentes**

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Frederico Gadelha Guimarães  
Coorientador: Eduardo Mazoni Andrade  
Marçal Mendes

Belo Horizonte

2026

L933n

Lucas, Patrícia de Oliveira e.

Um novo método de AutoML para séries temporais [recurso eletrônico] : decomposição, causalidade, otimização evolutiva multitarefa e agentes / Patrícia de Oliveira e Lucas. – 2026.

1 recurso online (109 f. : il., color.) : pdf.

Orientador: Frederico Gadelha Guimarães.

Coorientador: Eduardo Mazoni Andrade Marçal Mendes.

Tese (doutorado) – Universidade Federal de Minas Gerais, Escola de Engenharia.

Inclui bibliografia.

1. Engenharia elétrica – Teses. 2. Análise de séries temporais – Teses. 3. Previsão – Teses. I. Guimarães, Frederico Gadelha. II. Mendes, Eduardo Mazoni Andrade Marçal. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS

Escola de Engenharia

COLEGIADO DO CURSO DE GRADUAÇÃO / PÓS-GRADUAÇÃO EM Engenharia Elétrica

## **FOLHA DE APROVAÇÃO**

**"Um Novo Método de AutoML para Séries Temporais: Decomposição, Causalidade, Otimização Evolutiva Multitarefa e Agentes"**

**PATRÍCIA DE OLIVEIRA E LUCAS**

Tese de Doutorado 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 Doutor em Engenharia Elétrica.

Aprovada em 03 de fevereiro de 2026.

Por:

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

**Prof. Ph.D. Eduardo Mazoni Andrade Marçal Mendes**  
DELT (UFMG) - Coorientador

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

**Prof. Dr. Péricles Barbosa Cunha de Miranda**  
Departamento de Computação (UFRPE)

**Prof. Dr. Bruno Riccelli dos Santos Silva**  
(UFC Crateús)

**Prof. Dr. Adriano Alonso Veloso**  
DCC (UFMG)



Documento assinado eletronicamente por **Frederico Gadelha Guimaraes, Professor do Magistério Superior**, em 03/03/2026, às 08:48, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Walmir Matos Caminhas, Membro**, em 03/03/2026, às 09:05, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Eduardo Mazoni Andrade Marcal Mendes, Professor do Magistério Superior**, em 03/03/2026, às 09:44, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Adriano Alonso Veloso, Professor do Magistério Superior**, em 03/03/2026, às 13:51, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Bruno Riccelli dos Santos Silva, Usuário Externo**, em 05/03/2026, às 12:38, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Péricles Barbosa Cunha de Miranda, Usuário Externo**, em 05/03/2026, às 14:11, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site [https://sei.ufmg.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador 4992035 e o código CRC 287FFEE6.

## AGRADECIMENTOS

Agradeço à minha família, em especial à minha filha Duda e ao meu companheiro Marcelo, pelo amor, apoio incondicional e paciência em todos os momentos desta jornada.

Ao carinho constante dos meus pets, Loki, Shiva, Fibi, Mimi, Pedro e Joice.

Aos colegas dos laboratórios MINDS e FutureLab, deixo meu sincero reconhecimento pela parceria, pelas conversas que ampliaram horizontes e pelo ambiente colaborativo que tanto contribuiu para o desenvolvimento deste trabalho.

Às orientações atentas, competentes e sempre encorajadoras dos meus orientadores, Frederico e Eduardo, registro minha gratidão. Suas contribuições foram fundamentais para minha formação e para a construção desta tese.

Por fim, agradeço ao IFNMG e ao Instituto Kunumi pelo apoio, incentivo à pesquisa e por proporcionar condições que favoreceram o avanço deste estudo.

## RESUMO

A previsão de séries temporais desempenha um papel fundamental em diversos domínios práticos, auxiliando no planejamento, na tomada de decisão e na identificação antecipada de padrões e anomalias. A eficácia dessas aplicações depende da capacidade de extrair informações relevantes de observações históricas, tornando essencial o desenvolvimento contínuo de metodologias capazes de aprimorar a precisão e a interpretabilidade dos modelos. Embora existam estratégias de automação para esse processo, a criação de soluções verdadeiramente universais permanece desafiadora, especialmente devido à dificuldade de integrar, em uma única abordagem, diferentes tipos de modelos e técnicas adequadas a múltiplos domínios de aplicação. Nesse contexto, esta tese apresenta o AutoDCE-TS, uma nova abordagem de AutoML baseada em Decomposição, Descoberta Causal e Otimização Evolutiva Multitarefa para previsão de séries temporais multivariadas. O AutoDCE-TS adota uma arquitetura de Múltiplas Entradas e Saída Única composta por quatro camadas responsáveis por automatizar a extração e seleção de características, seleção de modelos, otimização de hiperparâmetros e previsão. A metodologia cria fluxos explicáveis ao estruturar subpipelines específicos para cada variável da série multivariada e para os componentes gerados pela decomposição. Para cada variável, um grafo causal é construído e utilizado como entrada para um modelo baseado em árvores de regressão. A similaridade entre as variáveis é explorada por meio de otimização evolutiva multitarefa, permitindo selecionar simultaneamente os modelos e seus hiperparâmetros com maior eficiência computacional. Foram realizados experimentos em 18 conjuntos de dados, envolvendo a comparação com 11 métodos. Os resultados demonstram que o AutoDCE-TS é competitivo e aplicável a diferentes domínios, configurando-se como uma solução eficiente para previsão de séries temporais. Além disso, esta tese introduz o AutoDCE-TS-Agentic, um sistema multiagente baseado em modelos de linguagem (LLMs) projetado para ampliar a automação e facilitar o uso do AutoDCE-TS. O sistema coordena agentes especializados para realizar tarefas como imputação de valores ausentes, análise das séries temporais de entrada, configuração automática dos hiperparâmetros críticos e geração de relatórios explicativos. O AutoDCE-TS-Agentic oferece uma interface mais acessível ao usuário, melhora a interpretabilidade dos fluxos de previsão e torna o processo mais transparente e reprodutível, complementando e estendendo as capacidades do AutoDCE-TS.

Palavras-chave: Séries temporais; Previsão; AutoML, Multivariadas, Agentes.

## ABSTRACT

Time series forecasting plays a fundamental role in several practical domains, supporting planning, decision-making, and the early identification of patterns and anomalies. The effectiveness of these applications depends on the ability to extract relevant information from historical observations, making the continued development of methodologies that enhance model accuracy and interpretability essential. Although automation strategies exist for this process, designing truly universal solutions remains challenging, particularly due to the difficulty of integrating, within a single approach, different types of models and techniques suitable for multiple application domains. In this context, this thesis presents AutoDCE-TS, a novel AutoML approach for multivariate time series forecasting based on Decomposition, Causal Discovery, and Multitask Evolutionary Optimization. AutoDCE-TS adopts a Multiple-Input Single-Output architecture composed of four layers responsible for automating feature extraction and selection, model selection, hyperparameter optimization, and forecasting. The methodology generates explainable pipelines by structuring specific sub-pipelines for each variable in the multivariate series and for the components generated through decomposition. For each variable, a causal graph is constructed and used as input to a regression tree-based model. Variable similarity is leveraged through multitask evolutionary optimization, enabling the simultaneous selection of models and hyperparameters with improved computational efficiency. Experiments were conducted on 18 datasets, involving comparisons with 11 methods. The results demonstrate that AutoDCE-TS is competitive and applicable across different domains, establishing it as an efficient solution for time series forecasting. Furthermore, this thesis introduces AutoDCE-TS-Agentic, a multi-agent system based on large language models (LLMs) designed to extend automation and facilitate the use of AutoDCE-TS. The system coordinates specialized agents to perform tasks such as missing-value imputation, analysis of the input time series, automatic configuration of critical hyperparameters, and generation of explanatory reports. AutoDCE-TS-Agentic provides a more accessible user interface, enhances the interpretability of forecasting pipelines, and makes the process more transparent and reproducible, thereby complementing and extending the capabilities of AutoDCE-TS.

Keywords: Time series; Forecasting; AutoML; Multivariate; Agents.



## LISTA DE FIGURAS

Figura 1 – Representação da integração entre Decomposição Empírica de Modos, Seleção de Características com Descoberta Causal e Otimização Multitarefa Evolutiva no AutoML proposto. . . . .	24
Figura 2 – Ilustração do problema de recomendação do pipeline $P^*$ por um sistema AutoML usando um conjunto $A$ de algoritmos e seus hiperparâmetros $\lambda$ aplicados a um conjunto de dados $D$ . . . . .	25
Figura 3 – Tipos de estruturas de pipelines. $P^1$ : linear, $P^2$ : baseado em conjunto e $P^3$ : composto. . . . .	26
Figura 4 – Ilustração de como o espaço de busca com dimensão dois é explorado pelos métodos de otimização: pesquisa em grade, pesquisa aleatória e por métodos guiados. Os círculos são as combinações das variáveis 1 e 2 avaliadas. A cor é a ordem das avaliações, do mais escuro para o mais claro. A melhor solução é representada pelo círculo vermelho. Adaptado de (PASSOS; MISHRA, 2022). . . . .	27
Figura 5 – Principais operadores dos métodos de otimização baseados em população. . . . .	28
Figura 6 – Ilustração da otimização bayesiana adaptada de (GARNETT, 2023). . . . .	29
Figura 7 – Ilustração do método PCMCI. . . . .	40
Figura 2 – Estrutura geral do AutoDCE-TS, composta pelas camadas Decomposição, Seleção Causal de Características, Otimização de Modelos Multitarefa e Previsão. $D$ e $x_1, x_2, \dots, x_m$ representam a variável endógena e as variáveis exógenas, respectivamente. O método EMD ( <i>Empirical Mode Decomposition</i> ) decompõe $D$ em <i>Intrinsic Mode Functions</i> (IMFs) $d_1, d_2, \dots, d_n$ . O método PCMCI gera os grafos causais $g_1, g_2, \dots, g_{n+m}$ , onde $n$ é o número de IMFs e $m$ é o número de variáveis exógenas. O Algoritmo Evolutivo Multifatorial para Otimização de Hiperparâmetros (AEMOH) seleciona os melhores modelos e hiperparâmetros $M_1, M_2, \dots, M_{n+m}$ . O símbolo $\sum$ representa a reconstrução da variável endógena $D$ no tempo $t + 1$ . . . . .	44
Figura 3 – Fluxograma do AEMOH. Os hiperparâmetros $h_1, h_2, h_3$ e $h_4$ representam, respectivamente, o número de árvores de decisão, o número máximo de características consideradas na busca pela melhor divisão, o número mínimo de amostras necessárias em um nó folha e o estimador. $i$ denota o número de indivíduos na população e $j$ o número de tarefas. $\psi$ é o custo factorial, $r$ o ranqueamento factorial, $\varphi$ a aptidão escalar, $C$ a complexidade do estimador, e $\tau$ o fator de habilidade. O indivíduo $p_j^*$ representa o melhor indivíduo para a tarefa $j$ . . . . .	47

Figura 4 – Metodologia usada na realização dos experimentos 1, 2 e 3. . . . .	52
Figura 5 – Número de variáveis selecionadas pelos métodos PCMCI, Correlacional, AG e Lasso para todos os conjuntos de dados estudados. . . . .	55
Figura 6 – NRMSE para horizontes de previsão de 1, 3 e 5 usando os métodos PCMCI, Correlacional, GA e Lasso para as bases de dados analisadas. . . . .	56
Figura 7 – Média do NRMSE para os horizontes de previsão 1 a 5 para as estruturas avaliadas. . . . .	59
Figura 8 – Número de variáveis utilizadas nas estruturas A, B e C. . . . .	60
Figura 9 – Mapa de calor de NRMSE para horizontes de 1, 15 e 30 dos modelos AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT, AUTOTS, RF, XGBOOST, LIGHTGBM, LSTM, TCN, N-BEATS, TFT e iTFM nos conjuntos de dados ECONOMICO_1, ECONOMICO_2, ECONOMICO_3, ECONOMICO_4 e ECONOMICO_5. . . . .	62
Figura 10 – Mapa de calor de NRMSE para horizontes de 1, 15 e 30 dos modelos AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT, AUTOTS, RF, XGBOOST, LIGHTGBM, LSTM, TCN, N-BEATS, TFT e iTFM nos conjuntos de dados ENERGIA_1, ENERGIA_2, ENERGIA_3, ENERGIA_4 e ENERGIA_5. . . . .	63
Figura 11 – Mapa de calor de NRMSE para horizontes de 1, 15 e 30 dos modelos AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT, AUTOTS, RF, XGBOOST, LIGHTGBM, LSTM, TCN, N-BEATS, TFT e iTFM nos conjuntos de dados IOT_1, IOT_2, IOT_3, IOT_4 e IOT_5. . . . .	64
Figura 12 – Mapa de calor de NRMSE para horizontes de 1, 15 e 30 dos modelos AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT, AUTOTS, RF, XGBOOST, LIGHTGBM, LSTM, TCN, N-BEATS, TFT e iTFM nos conjuntos de dados CLIMATICO_1, CLIMATICO_2 e CLIMATICO_3. . . . .	65
Figura 13 – Diagrama de diferenças críticas dos métodos comparados neste estudo para cada domínio dos conjuntos de dados, considerando os horizontes de previsão de 1, 15 e 30 conjuntamente. CD é a distância crítica, representada pelas barras pretas e indicam que todos os métodos dentro desse limite possuem média de NRMSE estatisticamente iguais. A régua acima do diagrama mostra a classificação média dos métodos, onde uma classificação média mais alta corresponde a uma colocação mais alta. Os modelos avaliados incluem: AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT, AUTOTS, Random Forest (RF), XGBOOST, LIGHTGBM, LSTM, TCN, N-BEATS, TFT e iTFM. . . . .	66

Figura 14 – Grafo causal gerado pelo AutoDCE-TS com base no modelo Random Forest aplicado ao conjunto de dados CLIMATICO_2. Os nós representam as variáveis com suas respectivas defasagens, enquanto as setas indicam relações causais direcionadas para a variável endógena ETO (em azul). A coloração dos nós reflete a importância relativa de cada variável defasada na previsão do ETO, de acordo com a escala mostrada à direita. As variáveis envolvidas são: evapotranspiração de referência (ETO), temperatura máxima do ar (Tmax), umidade relativa (RH), radiação solar (RS) e velocidade do vento (U2). . . . .	68
Figura 15 – Diagrama ilustrativo do AutoDCE-TS-Agentic. . . . .	71
Figura 16 – Variáveis alvo das bases de dados ECONOMICO_1, ECONOMICO_2, ECONOMICO_3, ECONOMICO_4 e ECONOMICO_5. . . . .	99
Figura 17 – Variáveis alvo das bases de dados ENERGIA_1, ENERGIA_2, ENERGIA_3, ENERGIA_4 e ENERGIA_5. . . . .	102
Figura 18 – Variáveis alvo das bases de dados IOT_1, IOT_2, IOT_3, IOT_4, IOT_5 e IOT_6. . . . .	104
Figura 19 – Variáveis alvo das bases de dados CLIMATICO_1, CLIMATICO_2, CLIMATICO_3. . . . .	108

## LISTA DE TABELAS

Tabela 1 – Descrição das bases de dados . . . . .	52
Tabela 2 – Tempo de execução em segundos dos gráficos gerados pelos métodos PCMCI, Correlacional, GA e Lasso para todos os conjuntos de dados do estudo. . . . .	55
Tabela 3 – NRMSE de previsão para os horizontes de previsão 1, 3 e 5 dos conjuntos de dados estudados. . . . .	56
Tabela 4 – Número de links espúrios gerados pelos métodos PCMCI, Correlacional, GA e Lasso para o conjunto de dados CLIMATICO_2 em três localidades diferentes. . . . .	57
Tabela 5 – Espaço de busca de hiperparâmetros para os métodos TCN, LSTM, NBEATS, TFT e iTFM. . . . .	61
Tabela 6 – Resultados do teste não paramétrico de Friedman e diferenças críticas (CD) do teste pós-hoc de Nemenyi. O teste de Friedman avalia a hipótese nula de que não há diferenças significativas entre os valores medianos das populações. . . . .	63
Tabela 7 – Tempo médio de execução em minutos dos métodos AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT e AUTOTS por domínio de conjunto de dados. . . . .	63
Tabela 8 – NRMSE da previsão para os horizontes de 1, 15 e 30 dos modelos AutoDCE-TS, AutoCE-TS e AutoDCE-TS-Agentic. . . . .	83
Tabela 9 – Descrição das variáveis contidas na base de dados ECONOMICO_1. . . . .	98
Tabela 10 – Descrição das variáveis contidas na base de dados ECONOMICO_2. . . . .	99
Tabela 11 – Descrição das variáveis contidas na base de dados ECONOMICO_3. . . . .	100
Tabela 12 – Descrição das variáveis contidas na base de dados ECONOMICO_4. . . . .	100
Tabela 13 – Descrição das variáveis contidas na base de dados ECONOMICO_5. . . . .	101
Tabela 14 – Descrição das variáveis contidas na base de dados ENERGIA_1. . . . .	101
Tabela 15 – Descrição das variáveis contidas na base de dados ENERGIA_2, ENERGIA_3, ENERGIA_4 e ENERGIA_5. . . . .	103
Tabela 16 – Descrição da série temporal das variáveis do conjunto de dados IOT_1. . . . .	103
Tabela 17 – Descrição da série temporal das variáveis do conjunto de dados IOT_2. . . . .	105
Tabela 18 – Descrição da série temporal das variáveis do conjunto de dados IOT_3. . . . .	106
Tabela 19 – Descrição da série temporal das variáveis do conjunto de dados IOT_4. . . . .	106
Tabela 20 – Descrição da série temporal das variáveis do conjunto de dados IOT_5. . . . .	107
Tabela 21 – Descrição da série temporal das variáveis do conjunto de dados CLIMATICO_1. . . . .	107

Tabela 22 – Descrição da série temporal das variáveis do conjunto de dados CLI-MATICO_2. . . . .	108
Tabela 23 – Descrição da série temporal das variáveis do conjunto de dados CLI-MATICO_3. . . . .	109

## LISTA DE ABREVIATURAS E SIGLAS

ADF	Augmented Dickey-Fuller
AEMOH	Algoritmo Evolutivo Multifatorial para Otimização de Hiperparâmetros
AG	Algoritmo Genético
AIC	Critério de Informação de Akaike
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
AutoDCE-TS	Automated Machine Learning Based on Decomposition, Causality, and Evolutionary Multitask Optimization for Time Series Forecasting
AutoML	Automated Machine Learning
BOHB	Bayesian optimization and Hyperband
CASH	Combined Algorithm Selection and Hyperparameter optimization problem
CD	Distância Crítica
DAG	Grafo Acíclico Dirigido
DL	Deep Learning
EMD	Empirical Mode Decomposition
EMO	Evolutionary Multitask Optimization
ETo	Evapotranspiração de Referência
FTS	Fuzzy Times Series
HFCM	High Order Fuzzy Cognitive Maps
IA	Inteligência Artificial
IMF	Intrinsic Mode Functions
IoT	Internet das Coisas
iTFM	Inverted Transformers

KPSS	Kwiatkowski-Phillips-Schmidt-Shin
LGBM	Light Gradient Boosting Machine
LLMs	Large Language Models
LSTM	Long Short-Term Memory
MB	Markov Blanket
MCI	Teste de independência condicional momentânea
MISO	Múltiplas Entradas e Saída Única
ML	Machine Learning
N-BEATS	Neural Basis Expansion Analysis for Interpretable Time Series Forecasting
NAS	Neural Architecture Search
NRMSE	Erro Quadrático Médio Normalizado
RF	Random Forest
RNN	Recurrent Neural Networks
TCN	Temporal Convolutional Networks
TFT	Temporal Fusion Transformer
XGBoost	Extreme Gradient Boosting

## LISTA DE SÍMBOLOS

$D$	variável endógena
$x_q$	q-ésima variável exógena
$q$	índice das variáveis exógenas
$m$	número de variáveis exógenas
$d_j$	j-ésima função de modo intrínseco (IMF)
$j$	índice das IMFs
$n$	número de IMFs
$g_i$	i-ésimo grafo causal
$M_i$	modelo selecionado para a variável $i$
$t$	instante temporal
$t + 1$	horizonte de previsão
$\tau_{max}$	defasagem máxima considerada
$f_1$	função objetivo de acurácia
$f_2$	função objetivo de complexidade
$N$	número de avaliações
$y(t)$	valor real da série no tempo $t$
$\hat{y}(t)$	valor previsto no tempo $t$
$y_{max}$	valor máximo da série
$y_{min}$	valor mínimo da série
$\mathcal{C}$	medida de complexidade do modelo
$A$	número de árvores do modelo
$P_i$	profundidade da i-ésima árvore
$h_1, h_2, h_3, h_4$	hiperparâmetros do indivíduo



$f_{1i}^j$	valor da função objetivo do indivíduo $i$ na tarefa $j$
$\psi_j^i$	custo factorial
$r_j^i$	ranqueamento factorial
$\varphi_i$	aptidão escalar do indivíduo $i$
$\tau_i$	fator de habilidade do indivíduo $i$
$C_i^j$	complexidade do indivíduo $i$ na tarefa $j$
$p^*$	melhor indivíduo global
$p_j^*$	melhor indivíduo da tarefa $j$
$M_s$	matriz de similaridade entre séries temporais
$P, Q$	distribuições de probabilidade
$M$	distribuição média
$D_{KL}$	divergência de Kullback-Leibler
$JSD(P \parallel Q)$	distância de Jensen-Shannon
$L$	limiar de similaridade
$i_1, i_2$	indivíduos pais
$i_N$	indivíduo gerado

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>19</b>
1.1	Objetivos	21
1.2	Contribuições	21
1.3	Organização do Trabalho	22
<b>2</b>	<b>REVISÃO DA LITERATURA</b>	<b>24</b>
2.1	Aprendizado de máquina automatizado	24
2.1.1	Estratégias de Otimização de Pipeline de Aprendizado de Máquina	26
2.1.1.1	Estratégias para reduzir o custo computacional de sistemas AutoML	30
2.1.2	Automação das Etapas do Pipeline de Aprendizado de Máquina	31
2.1.2.1	Preparação dos dados	32
2.1.2.2	Engenharia de características	33
2.1.2.3	Geração do modelo	34
2.1.3	AutoML para Previsão de Séries Temporais	35
2.2	Decomposição Empírica de Modos	36
2.3	Seleção de características	37
2.3.1	Descoberta causal	38
2.3.2	PCMCI	39
2.4	Otimização Evolutiva Multitarefa	41
2.5	Conclusão	42
<b>3</b>	<b>AUTOML PARA PREVISÃO DE SÉRIES TEMPORAIS MULTIVARIADAS</b>	<b>44</b>
3.0.1	Algoritmo Evolutivo Multifatorial para Otimização de Hiperparâmetros	46
3.0.2	Estrutura do AEMOH	46
3.1	Conclusão	50
<b>4</b>	<b>EXPERIMENTOS COMPUTACIONAIS</b>	<b>51</b>
4.1	Comparação de métodos de seleção de características não causais e o PCMCI	53
4.1.1	Resultados	54
4.1.2	Discussão	57
4.2	Estudo de ablação	57
4.2.1	Resultados e discussão	58
4.3	Comparação do Método AutoDCE-TS com outros Métodos de Previsão de Séries Temporais Multivariadas	58
4.3.0.1	Resultados e Discussão	61
4.4	Conclusão	67

<b>5</b>	<b>AutoDCE-TS AGÊNTICO PARA PREVISÃO DE SÉRIES TEMPO-RAIS</b>	<b>70</b>
5.1	Execução do pipeline	70
5.2	Correção	74
5.3	Resumo	74
5.4	Explicação	75
5.5	Estudo de caso sobre a base de dados CLIMATIC_2	77
5.5.1	Comparação AutoDCE-TS versus AutoDCE-TS-Agentic	82
5.6	Conclusão	82
<b>6</b>	<b>CONCLUSÕES</b>	<b>84</b>
6.0.1	Trabalhos Futuros	86
6.1	Publicações	87
6.1.1	Publicações em revista	87
6.1.2	Publicações em congresso	87
	<b>REFERÊNCIAS</b>	<b>89</b>
	<b>Appendix A Conjuntos de dados</b>	<b>98</b>
A.1	ECONOMICO_1	98
A.2	ECONOMICO_2	98
A.3	ECONOMICO_3	98
A.4	ECONOMICO_4	100
A.5	ECONOMICO_5	100
A.6	ENERGIA_1	101
A.7	ENERGIA_2, ENERGIA_3, ENERGIA_4, ENERGIA_5	101
A.8	IOT_1	102
A.9	IOT_2	102
A.10	IOT_3	104
A.11	IOT_4	105
A.12	IOT_5	105
A.13	CLIMATICO_1	107
A.14	CLIMATICO_2	107
A.15	CLIMATICO_3	108

## 1 INTRODUÇÃO

Os sistemas inteligentes tornaram-se parte intrínseca do cotidiano, influenciando decisões, oferecendo recomendações e transformando interações humanas em dados valiosos que alimentam um ciclo contínuo de aprendizado. Esse cenário evidencia uma transição de uma era predominantemente industrial para uma era orientada pela informação, na qual a análise de dados se consolidou como elemento estratégico para a tomada de decisões em organizações dos mais diversos setores (DEGUCHI et al., 2020).

Nesse contexto de transformação, conceitos como Indústria 5.0, cunhado em 2021 pela União Europeia, e Sociedade 5.0, promovido pelo Japão, reforçam a ideia de que vivemos em uma sociedade superinteligente. Nela, a automação não busca substituir o ser humano, mas ampliar suas capacidades, apoiando-se em tecnologias como Inteligência Artificial (IA), Internet das Coisas (IoT) e análise avançada de dados (COMMISSION et al., 2021; DEGUCHI et al., 2020; PASSALACQUA et al., 2025).

A IA, na forma de Machine Learning (ML), tem como objetivo aprender de forma automática os padrões e relacionamentos significativos a partir de exemplos e observações, sem a necessidade de ser explicitamente programada para tal (BISHOP, 2011). Nos últimos anos, temos testemunhado avanços significativos em técnicas baseadas em ML, impulsionando o surgimento de novas arquiteturas aplicadas na resolução de uma ampla gama de tarefas em diversos campos.

No entanto, para que essas técnicas sejam amplamente adotadas e capazes de transformar profundamente a realidade das empresas e da indústria, Luo, Han e Liu (2023) ressaltam a necessidade de uma revolução no desenvolvimento de sistemas de ML. Os autores argumentam que o processo atual de desenvolvimento desses sistemas ainda depende fortemente de arquiteturas projetadas manualmente, com modelagens específicas para cada tarefa, e requer um esforço considerável na manipulação manual dos dados.

Embora a comunidade de ML já tenha disponibilizado uma ampla variedade de ferramentas prontas para uso por meio de pacotes de código aberto, o desafio de desenvolver sistemas de ML acessíveis, escalonáveis e flexíveis persiste (OLSON et al., 2016). O objetivo primordial de um modelo de ML é aprender padrões generalizáveis, ou seja, ser capaz de fazer previsões precisas quando apresentado a novos conjuntos de dados (VIEIRA; PINAYA; MECHELLI, 2020). Alcançar esse objetivo não é uma tarefa trivial, uma vez que projetar um sistema de ML eficaz demanda habilidades consideráveis, além de ser um processo demorado e propenso a erros.

Cada etapa desse processo implica em uma infinidade de escolhas potenciais a serem feitas: como os dados devem ser pré-processados? Qual modelo é mais adequado para

ajustar os dados em questão? Quais são os hiperparâmetros ideais para o aprendizado? Essas decisões podem influenciar significativamente o desempenho e a eficácia do modelo. Portanto, a pesquisa contínua e o desenvolvimento de técnicas e ferramentas mais eficientes são essenciais para enfrentar esses desafios e impulsionar o avanço do ML.

O *Automated Machine Learning* (AutoML) tem como objetivo automatizar essas escolhas, fornecendo métodos que constroem modelos de ML adequados com o mínimo de intervenção humana possível, ou até mesmo sem nenhuma intervenção. Isso beneficia tanto os não especialistas, que podem aproveitar a facilidade de uso, quanto os especialistas, que podem expandir seu conhecimento para uma variedade de problemas sem a necessidade de intervenções manuais frequentes (TUGGENER et al., 2019).

Recentemente, agentes inteligentes baseados em Large Language Models (LLMs) têm acrescentado uma nova camada de automação aos sistemas AutoML, aprimorando a interação entre o sistema e o usuário e auxiliando em tarefas como o ajuste de hiperparâmetros, a explicação dos pipelines empregados e até mesmo o planejamento e a execução completos do processo de ML (TRIRAT; JEONG; HWANG, 2024; GUO et al., 2024; LIU; GAO; LI, 2024; XU et al., 2024).

No contexto da previsão de séries temporais — uma tarefa essencial para atividades de planejamento e controle em diversas áreas — métodos AutoML assumem papel particularmente relevante, uma vez que, além dos desafios já apontados que motivaram o surgimento dos métodos AutoML, a previsão de séries temporais requer ainda habilidades técnicas específicas para lidar com características peculiares desses dados, como tendência, sazonalidade, outliers, desvios e mudanças abruptas (PALDINO et al., 2021).

Diversas ferramentas de AutoML, como AutoGluonTS (ALEXANDROV et al., 2019), PyCaret (ALI, 2020), Fedot (NIKITIN et al., 2022a), AutoTS (WANG et al., 2022b) e TimeSeriesScientist (ZHAO et al., 2025) oferecem suporte a dados de séries temporais. Outras, como TPOT (OLSON et al., 2016), Auto-Sklearn (FEURER et al., 2020), GAMA (GIJSBERS; VANSCHOREN, 2020) e Auto-WEKA (KOTTHOFF et al., 2017), são projetadas para automatizar fluxos de regressão, mas não possuem suporte nativo para lidar com dados de séries temporais. Consequentemente, os usuários precisam pré-processar os dados para preservar sua ordem temporal. Essa limitação é particularmente crítica para dados de previsão multivariados e de múltiplas etapas, que exigem fluxos de trabalho mais especializados.

A necessidade de avanços nessa área foi destacada em pesquisas recentes. De acordo com Pilyugina et al. (2024), o AutoML para previsão de séries temporais representa um desafio urgente, visto que selecionar um pipeline ideal dentre uma gama de candidatos é um problema altamente complexo. Além disso, Fischer e Saadallah (2024) enfatizam que a seleção do melhor modelo deve levar em conta não apenas o desempenho preditivo, mas também a eficiência computacional e a explicabilidade.

Este contexto motiva a investigação do problema de AutoML na previsão de séries temporais multivariadas, a partir da seguinte questão de pesquisa: até que ponto é possível automatizar um pipeline de previsão de séries temporais multivariadas, mantendo níveis competitivos de precisão, baixo custo computacional e transparência?

Para responder a essa questão, foi projetada uma metodologia voltada à automação das etapas que compõem o pipeline de previsão de séries temporais multivariadas. Na construção dessa metodologia buscou-se identificar quais componentes podem ser automatizados sem incorrer em custos computacionais excessivos e quais técnicas podem ser integradas a um sistema AutoML de modo a garantir a geração de pipelines eficientes e que consigam manter a transparência para o usuário.

## 1.1 Objetivos

O objetivo geral deste trabalho é desenvolver uma metodologia para a automação de pipelines de previsão de séries temporais multivariadas. Para alcançar esse propósito, foram definidos os seguintes objetivos específicos:

1. Identificar estratégias bem-sucedidas e lacunas existentes no desenvolvimento de sistemas AutoML aplicados à previsão de séries temporais, considerando as etapas específicas do processo e a natureza dos dados temporais.
2. Propor uma metodologia para a automação das etapas de um pipeline de previsão de séries temporais multivariadas.
3. Implementar um método AutoML que operacionalize essa metodologia.
4. Avaliar experimentalmente o método proposto em diferentes conjuntos de dados, comparando seu desempenho com abordagens de referência.
5. Desenvolver um sistema multiagente de apoio ao método AutoML, capaz de automatizar etapas como a imputação de valores faltantes e a configuração orientada pelos dados de entrada, bem como de fornecer uma interface mais interativa e mecanismos de explicação para os resultados produzidos pelo pipeline.

## 1.2 Contribuições

Orientada pelos objetivos delineados anteriormente, esta tese apresenta as seguintes contribuições principais:

- Desenvolvimento de um sistema AutoML para previsão de séries temporais, fundamentado nos princípios de Decomposição, Causalidade e Otimização Evolutiva

Multitarefa. O AutoDCE-TS automatiza as etapas de um pipeline do tipo MISO (múltiplas entradas, uma saída), abrangendo desde a extração e seleção de características até a construção de modelos e a geração de previsões. O sistema foi concebido como uma estrutura AutoML de propósito geral, capaz de lidar com séries temporais de diferentes domínios, dimensionalidades e frequências amostrais.

- Desenvolvimento do AEMOH (Algoritmo Evolutivo Multifatorial para Otimização de Hiperparâmetros) para selecionar modelos e hiperparâmetros. O AEMOH é um algoritmo multitarefa que explora a similaridade entre séries temporais para compartilhar conhecimento entre tarefas relacionadas, promovendo ganhos de desempenho com redução do custo computacional em cenários compostos por séries de um mesmo domínio.
- Realização de experimentos em 18 conjuntos de dados com diferentes dimensionalidades e domínios de aplicação. O AutoDCE-TS foi comparado a 11 métodos, incluindo AutoMLs e modelos deep learning de última geração para previsão de séries temporais.
- Desenvolvimento de uma versão multiagente do AutoDCE-TS, denominada AutoDCE-TS-Agentic, cujo objetivo é apoiar a automação e explicação dos pipelines gerados.

### 1.3 Organização do Trabalho

Esta tese está organizado da seguinte forma:

- O Capítulo ?? apresenta uma revisão bibliográfica sobre o tema AutoML, bem como o estado da arte tanto em AutoML de forma geral quanto em abordagens específicas para previsão de séries temporais. Também descreve os conceitos que serviram de base para o desenvolvimento do método AutoML proposto, a saber: Decomposição Empírica de Modos, Seleção de Características com Descoberta Causal e Otimização Evolutiva Multitarefa.
- O Capítulo 3 detalha o método proposto, denominado Aprendizado de Máquina Automatizado Baseado em Decomposição, Causalidade e Otimização Evolutiva Multitarefa para Previsão de Séries Temporais (AutoDCE-TS).
- O Capítulo 4 apresenta os experimentos realizados, descrevendo a configuração de cada um deles e os resultados obtidos.
- O Capítulo 5 introduz um sistema multiagente desenvolvido para apoiar a automação e explicação dos pipelines gerados pelo AutoDCE-TS, acompanhado de um estudo de caso que demonstra seu funcionamento.

- Por fim, o Capítulo 6 apresenta as conclusões e os trabalhos futuros.
- O Apêndice A descreve as bases de dados utilizadas nos experimentos.



## 2 REVISÃO DA LITERATURA

Este capítulo tem como objetivo apresentar os conceitos de Aprendizado de máquina automatizado, Decomposição Empírica de Modos, Seleção de Características com Descoberta Causal e Otimização Multitarefa Evolutiva. Esses conceitos servirão de base para a construção do método AutoML proposto nas etapas de extração e seleção de características e seleção de modelos (Figura 1), que será apresentado no próximo capítulo.

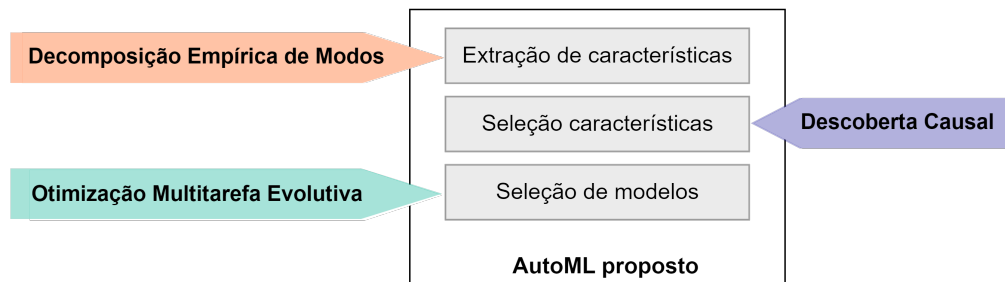


Figura 1 – Representação da integração entre Decomposição Empírica de Modos, Seleção de Características com Descoberta Causal e Otimização Multitarefa Evolutiva no AutoML proposto.

### 2.1 Aprendizado de máquina automatizado

O aprendizado de máquina automatizado (em inglês *Automated Machine Learning* - AutoML) é o termo usado para definir um sistema que recomenda automaticamente pipelines de aprendizado de máquina otimizados para tarefas específicas (ZÖLLER; HUBER, 2021). Um pipeline de aprendizado de máquina é um conjunto de operações que transformam um vetor de característica  $X$  em um valor alvo  $y$ . Essas operações podem ser agrupadas em etapas de acordo com a finalidade de cada uma. Aqui, dividiremos o pipeline em 3 etapas: preparação dos dados, engenharia de características e geração do modelo (seleção do modelo e otimização de hiperparâmetros) (HE; ZHAO; CHU, 2021).

Um sistema AutoML completo cria combinações de várias dessas operações para encontrar um pipeline de alto desempenho e que atenda aos objetivos da aplicação. Sendo o AutoML um problema combinacional, é possível formulá-lo como um problema de otimização. A estrutura usada para modelar os pipelines será um Grafo Acíclico Dirigido (DAG), onde cada nó representa uma operação aplicada aos dados e as arestas o fluxo dos dados. Sendo  $G$  o conjunto de estruturas de pipeline válidas,  $A$  o conjunto de algoritmos disponíveis e  $\lambda^i$  o conjunto de hiperparâmetros configurados para o algoritmo  $A^i$  do domínio  $\Lambda_{A^i}$ , um pipeline pode ser definido como  $P_{(g, \vec{A}, \vec{\lambda})}$ , onde  $g \in G$ ,  $\vec{A}$  é o vetor de

algoritmos selecionados para os nós de  $g$  e  $\vec{\lambda}$  é o vetor de hiperparâmetros de todos os algoritmos selecionados (CHEN; SONG; HU, 2019).

Dessa forma, o problema de escolher um pipeline  $P^*$  para um conjunto de dados específico  $D$  consiste em encontrar uma estrutura  $g$  com um vetor de algoritmos e seus respectivos hiperparâmetros que minimize a função de perda  $\mathcal{L}$  (Equação 2.1) (ZÖLLER; HUBER, 2021).

$$P_{(g, \vec{A}, \vec{\lambda})}^* = \operatorname{argmin}_{g \in G, \vec{A} \in A, \vec{\lambda} \in \Lambda} \mathcal{L}(P_{(g, \vec{A}, \vec{\lambda})}, D) \quad (2.1)$$

A Figura 2 ilustra três pipelines  $P^1$ ,  $P^2$  e  $P^3$  formados pelos grafos cujos vértices são  $\{\{A^1, \lambda^1\}, \{A^4, \lambda^4\}, \{A^7, \lambda^7\}\}$ ,  $\{\{A^3, \lambda^3\}, \{A^5, \lambda^5\}, \{A^8, \lambda^8\}\}$ ,  $\{\{A^2, \lambda^2\}, \{A^5, \lambda^5\}, \{A^9, \lambda^9\}\}$ , respectivamente.

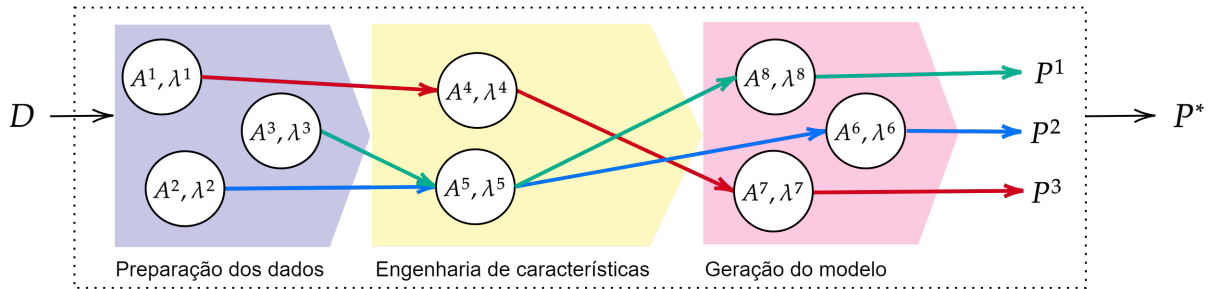


Figura 2 – Ilustração do problema de recomendação do pipeline  $P^*$  por um sistema AutoML usando um conjunto  $A$  de algoritmos e seus hiperparâmetros  $\lambda$  aplicados a um conjunto de dados  $D$ .

A flexibilidade na escolha do número de operações em cada etapa define a complexidade da estrutura criada para os pipelines. Nikitin et al. (2022a) classificam as estruturas de pipeline em lineares, baseados em conjunto e compostos. Nos pipelines com estrutura linear os grafos gerados são caminhos. As estruturas baseadas em conjunto são grafos de extensão mínima (todos os caminhos entre todos os pares de nós têm o mesmo comprimento), permitindo a combinação de modelos de aprendizagem de máquina (*ensemble learning*). Essas estruturas, definidas em Zöller e Huber (2021) como fixas, não possuem flexibilidade para conseguirem se adaptar a dados complexos. Os pipelines compostos, por sua vez, são muito flexíveis, pois só exigem a existência de um nó final de destino para todos os fluxos de dados. Isso permite resolver tarefas complexas com a formação de blocos de processamento de dados de subtarefas com propósitos distintos (classificação, agrupamento, regressão, etc.) (NIKITIN et al., 2022a).

A Figura 3 mostra um exemplo de cada tipo de pipeline descrito. O pipeline  $P^1$  é linear e usa os dados  $D^1$  como entrada. O pipeline  $P^2$  é baseado em conjunto e tem como

entrada os dados  $D^2$ , já o pipeline  $P^3$  é composto e usa as subtarefas processadas por  $P^1$  e  $P^2$  para realizar uma tarefa final.

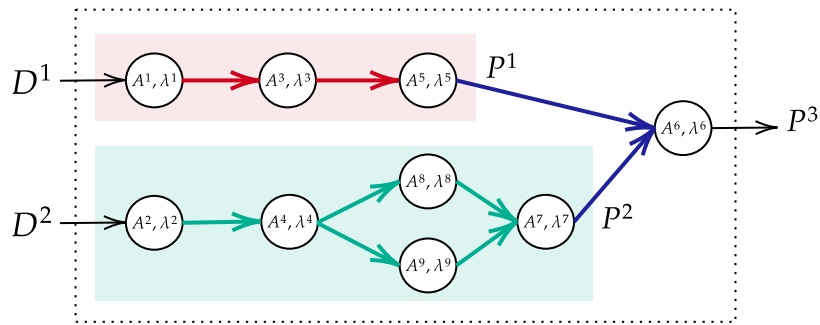


Figura 3 – Tipos de estruturas de pipelines.  $P^1$ : linear,  $P^2$ : baseado em conjunto e  $P^3$ : composto.

### 2.1.1 Estratégias de Otimização de Pipeline de Aprendizado de Máquina

O problema de otimização AutoML é desafiador, pois trabalha com espaços de busca complexos e as funções objetivo demandam muito tempo de processamento e um alto custo computacional. Essas características podem tornar o problema AutoML impossível de ser resolvido manualmente (MOTZ; KRAUSS; SCHMITT, 2022). Além disso, sua função objetivo é uma caixa preta, o que significa que, quando  $f$  é avaliado, observa-se apenas  $f(x)$ , sem ter acesso as derivadas (FRAZIER, 2018). Nessa seção serão abordadas as estratégias de otimização estado da arte que tratam essas características dos sistemas AutoML e ainda recursos que podem ser usados em conjunto com essas estratégias para reduzir o custo computacional da otimização.

Os métodos de otimização mais simples são aqueles que não utilizam nenhuma estratégia que leve em consideração iterações anteriores para guiar a busca e tornar o método mais eficiente. A pesquisa em grade e aleatória são exemplos desses métodos.

**Pesquisa em grade (em inglês *grid search*):** avalia cada combinação do produto cartesiano dos valores definidos no espaço de busca de cada variável tratada no problema. Apesar da simplicidade e facilidade de implementação, a pesquisa em grade sofre da maldição da dimensionalidade, uma vez que à medida que o número de variáveis ou o espaço de busca aumenta, o número de avaliações cresce exponencialmente (BELETE; HUCHAIAH, 2021).

**Pesquisa aleatória (em inglês *random search*):** ao contrário da pesquisa em grade, a pesquisa aleatória gera as combinações das variáveis aleatoriamente para serem avaliadas. Isso garante um espaço de busca que não se limita a uma grade de valores pré-definida. Isso elimina o problema da maldição da dimensionalidade, já que o número

de avaliações pode ser definido de acordo com os recursos de tempo e computacionais disponível (FEURER, 2019).

A Figura 4 ilustra como o espaço de busca com dimensão dois é explorado pela pesquisa em grade, pesquisa aleatória e por métodos guiados. Os círculos representam as combinações das variáveis 1 e 2 que serão avaliadas e a cor é a ordem dessas avaliações, iniciando do mais escuro para o mais claro. A melhor solução é representada pelo círculo laranja. Observa-se que a pesquisa aleatória flexibiliza a varredura do espaço de busca ao trabalhar com intervalos de valores e não com valores fixos como na pesquisa em grade. Apesar disso, a aleatoriedade não permite que o método guie a busca para um ótimo local.

Para resolver esse problema, estratégias mais sofisticadas foram desenvolvidas para aprender como aproximar a solução ótima através do uso de informações de configurações avaliadas anteriormente, aqui chamado de métodos guiados. Ainda na Figura 4, percebe-se na estratégia de busca dos métodos guiados que conforme as configurações são avaliadas, a busca é direcionada para se aproximar da solução ótima local.

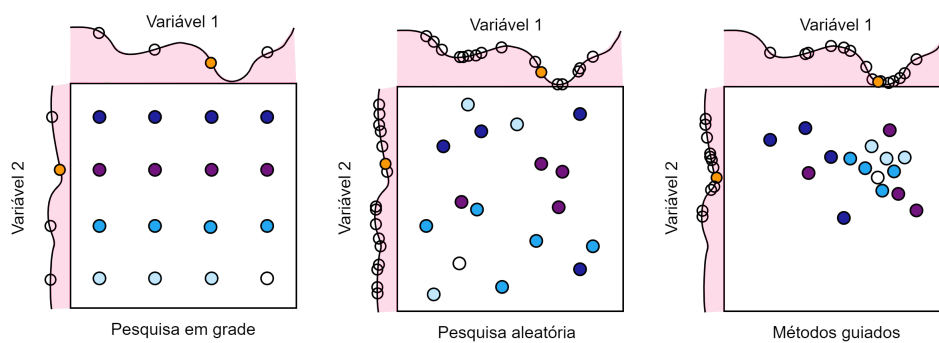


Figura 4 – Ilustração de como o espaço de busca com dimensão dois é explorado pelos métodos de otimização: pesquisa em grade, pesquisa aleatória e por métodos guiados. Os círculos são as combinações das variáveis 1 e 2 avaliadas. A cor é a ordem das avaliações, do mais escuro para o mais claro. A melhor solução é representada pelo círculo vermelho. Adaptado de (PASSOS; MISHRA, 2022).

Duas famílias de métodos de otimização guiados serão abordadas: baseados em população e baseados em modelos substitutos.

**Métodos baseados em população:** operam sobre um conjunto de vetores de soluções  $X = \{x^1, x^2, \dots, x^N\}$  denominado população, onde  $N$  é o tamanho da população.  $\{x^i\} = \{\{x^{i,1}\}, \{x^{i,2}\}, \dots, \{x^{i,n}\}\}$  são os indivíduos da população, sendo  $n$  o número de variáveis (genes) de cada indivíduo  $i$ . No geral, esses métodos apresentam 3 operadores principais em seus algoritmos: (1) operador que seleciona indivíduos promissores para participarem do cruzamento (seleção), (2) operador que combina soluções para encontrar indivíduos mais bem adaptados (cruzamento), (3) operador que cria perturbações em alguns indivíduos para aumentar a exploração do espaço de busca e escapar de ótimos locais

(mutação) (WU; MALLIPEDDI; SUGANTHAN, 2019; GIAGKIOZIS; PURSHOUSE; FLEMING, 2013).

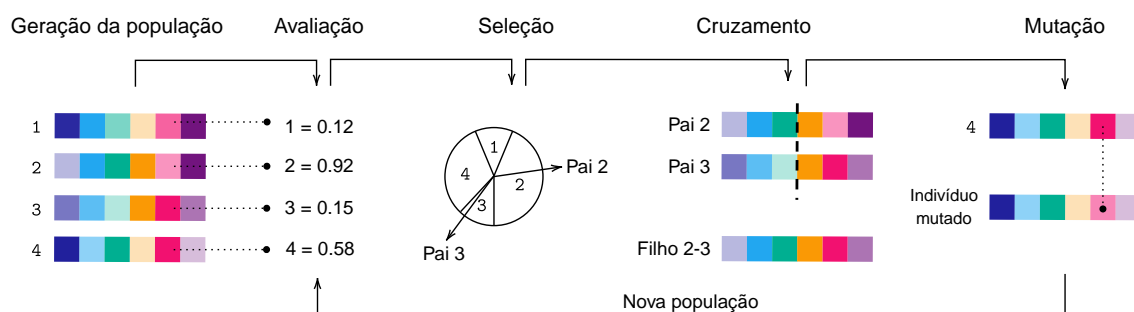


Figura 5 – Principais operadores dos métodos de otimização baseados em população.

A Figura 5 ilustra como esses operadores trabalham para gerar novos indivíduos. O algoritmo inicia com a geração da população inicial, normalmente aleatória. Para exemplificar, foram criados 4 indivíduos com 6 variáveis cujos valores são representados pela variação de cores. Em seguida, todos os indivíduos são avaliados usando a função objetivo (*fitness*). A seleção define dois indivíduos (Pais) para participarem do cruzamento com probabilidade proporcional às avaliações. No exemplo, os indivíduos 2 e 3 foram escolhidos para combinarem parte de seus genes em um novo indivíduo (Filho 2-3). Para a mutação, o indivíduo 4 foi escolhido aleatoriamente e o novo indivíduo (Indivíduo mutado) é uma cópia do 4, com uma alteração no gene 5. A partir desse fluxo (denominado geração), uma nova população é gerada e o fluxo recomeça até que um critério de parada seja atingido.

Os métodos baseados em população chamam a atenção por imporem poucas restrições à estrutura do problema, por trabalharem com diferentes tipos de dados e serem aplicáveis em problemas onde a busca exaustiva é impraticável, como nos problemas de otimização AutoML. São exemplos de métodos baseados em população: algoritmos genéticos (AG), estratégias evolutivas, programação genética (PG) e enxame de partículas (FEURER, 2019; GIAGKIOZIS; PURSHOUSE; FLEMING, 2013).

**Métodos baseados em modelos substitutos (em inglês *surrogate model*):** operam por meio da criação de um modelo auxiliar (substituto) por meio de aproximações de informações amostradas da função objetivo original. A ideia é que o modelo substituto seja uma representação confiável do modelo original, mas computacionalmente barato (KOZIEL; CIAURRI; LEIFSSON, 2011). O processo básico de um método baseado em modelos substitutos pode ser dividido nas seguintes etapas (FORRESTER; KEANE, 2009):

1. Um conjunto de pontos  $X$  são avaliados usando a função objetivo original.
2. Agregando as informações da avaliação na forma  $\mathcal{D} = \{X, y\}$  um modelo substituto é treinado.

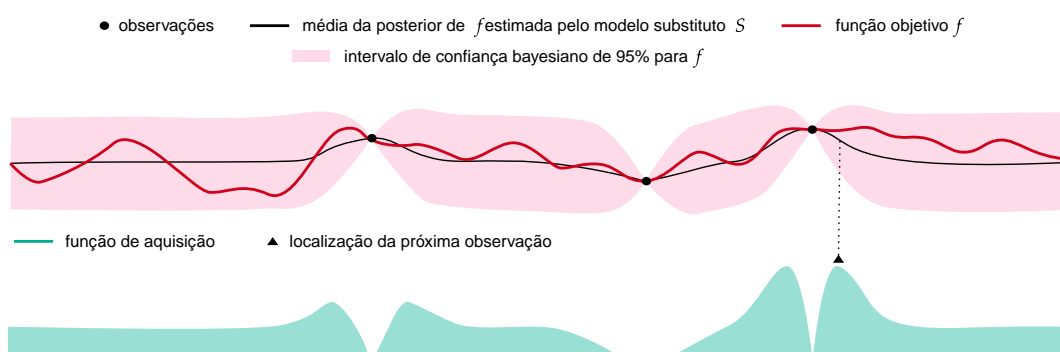


Figura 6 – Ilustração da otimização bayesiana adaptada de (GARNETT, 2023).

3. Realiza-se uma busca no modelo substituto para identificar um novo ponto a ser avaliado.
4. As novas informações são agregadas ao conjunto  $\mathcal{D}$  e, caso o critério de parada não tenha sido atingido, o processo volta para a etapa 2.

Nesses métodos o modelo original é avaliado apenas uma vez por iteração e como o modelo substituto é computacionalmente mais barato, em muitos casos, o custo de otimização é determinado pela avaliação do modelo original (KOZIEL; CIAURRI; LEIFSSON, 2011).

A otimização bayesiana é um exemplo de método baseado em modelo substituto que utiliza modelos estatísticos bayesianos para representar a função objetivo original e uma função de aquisição para decidir qual será a próxima solução a ser avaliada (FRAZIER, 2018). Para exemplificar, a Figura 6 ilustra como esse algoritmo funciona. Dada uma função objetivo desconhecida  $f$  que queremos maximizar e um conjunto de três observações avaliadas em  $f$ , um modelo substituto  $S$  (geralmente um processo gaussiano) estima a distribuição posterior de  $f$  com base nas observações. Uma função de aquisição (geralmente a melhoria esperada) atribui uma pontuação à novos pontos de observação proporcional à sua capacidade de beneficiar o processo de otimização com base na distribuição posterior de  $f$  (GARNETT, 2023). Perceba que a função de aquisição assume valores 0 nos pontos já avaliados, pois eles não fornecem informações úteis para a solução. À medida que se afasta desses pontos, a incerteza aumenta e a função de aquisição assume valores maiores, pois são locais candidatos a exploração. Perceba também que os valores são ainda maiores em locais onde a média posterior é maior, pois tais pontos tendem a estarem próximos de ótimos globais aproximados (FRAZIER, 2018).

### 2.1.1.1 Estratégias para reduzir o custo computacional de sistemas AutoML

Com um custo computacional extremamente alto, é imprescindível que um sistema AutoML possua estratégias que reduzam tanto o tempo de processamento, quanto o custo de hardware (ZÖLLER; HUBER, 2021). Técnicas eficazes encontradas na literatura para lidar com esses desafios são a otimização multi-fidelidade, Successive Halving, previsão da curva de aprendizado, Meta-Aprendizado e escalabilidade.

**Otimização multi-fidelidade (em inglês *multi-fidelity optimization*):** são heurísticas que usam aproximações de baixa fidelidade da função de perda real para acelerar o ajuste. São exemplos de heurísticas: usar subconjuntos de dados, treinar os modelos em apenas algumas iterações, usar subconjunto de recursos e excluir pipelines ruins (FEURER, 2019).

O *Successive Halving* é um método iterativo de otimização multi-fidelidade onde o algoritmo aloca um orçamento (tamanho do conjunto de dados, número de recursos ou número de iterações do algoritmo) para selecionar pipelines mais promissoras e descartar os demais. A cada iteração, o orçamento dobra para que os modelos sobreviventes sejam avaliados com mais confiança. Esse processo se repete até restar apenas um modelo (JAMIESON; TALWALKAR, 2015). O *Hyperband* (LI et al., 2016) melhora o *Successive Halving* usando a Pesquisa Aleatória para testar diferentes orçamentos e selecionar as melhores configurações. E ainda o Bayesian optimization and Hyperband (BOHB) (FALKNER; KLEIN; HUTTER, 2018) substitui a Pesquisa aleatória do *Hyperband* pela Otimização Bayesiana, apresentando resultados que superaram os métodos citados.

**Previsão da curva de aprendizado:** durante o processo de treinamento de um algoritmo, é possível definir o orçamento alocado com base na curva de aprendizado. Conforme mencionado em Klein et al. (2022), configurações de hiperparâmetros semelhantes geralmente produzem curvas de aprendizado semelhantes. Ao modelar essa relação, pode-se prever as curvas de aprendizado para novas configurações, sem a necessidade de observar seu desempenho inicial. Essa estratégia é usada por Swersky, Snoek e Adams (2014) que aproveita informações de modelos parcialmente treinados para prever o resultado final com base em um modelo de processo Gaussiano aproximado. Essa previsão é então utilizada durante a otimização bayesiana para determinar a ação mais promissora. Usando essa mesma ideia, mas com redes bayesianas para prever as curvas de aprendizado, Klein et al. (2022) estende o *Hyperband* e escolhe aqueles pipelines previstos como mais promissoras.

**Aprendizado em conjunto (em inglês *ensemble learning*):** é a combinação das saídas de vários modelos diferentes para fazer uma previsão final que seja melhor do que qualquer um dos modelos individuais e muitas vezes reduzindo a variância (GANAIÉ et al., 2022). Uma combinação pode ser desde a média das previsões até outro modelo ou camadas de modelos empilhadas. No aprendizado em conjunto empilhado, as previsões



dos modelos combinados (modelos base) são usadas para treinar outro modelo (modelo superior) que é responsável por encontrar a combinação que retorna a melhor previsão. Exemplos de frameworks que implementam aprendizado em conjunto empilhado são: H2O (LEDELL; POIRIER, 2020) e AutoGluon (ERICKSON et al., 2020). No AutoML, o aprendizado em conjunto substitui o processo de busca de modelos pela combinação dos pipelines gerados.

**Meta-Aprendizado (em inglês *meta-learning*):** aprendizado orientado por metadados obtidos de experiências com tarefas anteriores. Os metadados podem abranger todo tipo de informação como estrutura do pipeline, hiperparâmetros dos modelos, precisão, tempo de execução, etc. Uma vez definida a semelhança entre tarefas, esses metadados podem ser usados em várias etapas do AutoML para aumentar sua eficiência (VANSCHOREN, 2019; ZÖLLER; HUBER, 2021). O Auto-sklearn 2.0 (FEURER et al., 2020) usa o meta-aprendizado para decidir qual a melhor política de otimização (validação cruzada ou hold-out e *successive halving* ou orçamento completo) para um determinado conjunto de dados. O Auto-PyTorch (ZIMMER; LINDAUER; HUTTER, 2021) inicia a primeira iteração do BOHB com um conjunto de configurações aprendidas de outras bases de dados.

**Escalabilidade:** refere-se à capacidade de um sistema AutoML lidar com grandes conjuntos de dados, com modelos que demandam muito tempo de treinamento e com grandes espaços de busca. O framework H2O (LEDELL; POIRIER, 2020) torna seu sistema escalável através da paralelização do treinamento de modelos individuais em núcleos de CPU em uma única máquina ou em um cluster de máquinas em rede. Para os modelos XGBoost, também oferecem suporte à aceleração de GPU para acelerar ainda mais o treinamento. Por sua vez, o AutoStacker (CHEN et al., 2018) paraleliza o processos de otimização realizado pelo AG.

### 2.1.2 Automação das Etapas do Pipeline de Aprendizado de Máquina

A automação das etapas do pipeline de aprendizado de máquina não é uma tendência nova. Especialmente na geração de modelos, trabalhos como Kohavi e John (1995) já alertavam sobre o efeito da configuração de hiperparâmetros na precisão de um modelo e a dificuldade de realização desse problema em um tempo razoável. Também, do teorema “*No Free Lunch*”, já é sabido que nenhum algoritmo é capaz de funcionar perfeitamente em todos os cenários possíveis (WOLPERT; MACREADY, 1997). Desde 1990 estudos já tentavam automatizar a escolha dos hiperparâmetros (DUBRAWSKI, 1997; DUAN; KEERTHI; POO, 2003). Com o surgimento de modelos cada vez mais complexos e do *big data* o interesse pela automação de outras etapas se consolidou e soluções com automação cada vez mais completas já estão disponíveis (OLSON et al., 2016; NIKITIN et al., 2022a; SHAH et al., 2021). Nas próximas seções serão apresentadas as especificidades de cada etapa do pipeline e métodos de automação disponíveis na literatura.



### 2.1.2.1 Preparação dos dados

A etapa de preparação de dados envolve a realização de operações nos dados brutos, a fim de torná-los aptos a serem usados como entrada pelos algoritmos de aprendizado de máquina. Operações comuns dessa etapa são detecção e correção de erros, imputação de valores ausentes e codificação de atributos categóricos.

As operações realizadas na etapa de preparação de dados lidam diretamente com dados brutos e, por isso, dependem em grande parte do conhecimento de domínio do problema. Assim, a escolha das operações a serem aplicadas em um conjunto de dados específico ainda exige intervenção humana significativa. Embora existam na literatura muitas soluções que permitam implementar diferentes algoritmos para essa tarefa, nenhuma delas está madura o suficiente para lidar com precisão da recomendação de um pipeline de preparação de dados (BILAL et al., 2022). O resto dessa seção apresenta algumas dessas soluções.

O AlphaClean (KRISHNAN; WU, 2019) se concentra na limpeza dos dados e modela o problema como uma árvore de busca gulosa. Cada ramo da árvore é uma sequência de reparos que otimiza uma função objetivo. As operações de reparo e a função objetivo são informadas pelo usuário. O AlphaClean retorna os resultados progressivamente para evitar expandir caminhos pouco promissores.

O BoostClean (KRISHNAN et al., 2017) automatiza o processo de limpeza dos dados usando o Boosting (FREUND; SCHAPIRE, 1997). Cada operação de reparo é um aprendiz fraco e a sequência de reparos ( $S$ ) é aplicada ao conjunto de treinamento, que então, é usado para treinar o classificador. O conjunto de teste é usado para avaliar os aprendizes fracos e gerar os seus respectivos pesos. Ao final da execução, é encontrada uma sequência  $S^*$  que retorna o classificador com maior precisão de teste.

O HoloClean (REKATSINAS et al., 2017) assume que os erros nos dados ocorrem devido a atribuições imprecisas nos registros dos recursos e procura reparar os erros atualizando esses valores. O HoloClean detecta primeiro as células com valores potencialmente imprecisos. A saída dessa etapa (células ruidosas e limpas) é usada para criar um modelo gráfico probabilístico que descreve a distribuição de variáveis aleatórias  $A_c$  para cada célula  $c$  na base de dados. Para reparar os valores de  $c$ , o HoloClean executa aprendizado estatístico e inferência sobre a distribuição conjunta das variáveis  $A_c$  para calcular a probabilidade marginal  $P(A_c = d; \Omega, \Sigma)$  para todos os valores  $d \in \text{domínio}(c)$ , e atribuir  $\hat{v}_c$  ao valor que maximiza a probabilidade da variável  $T_c$ .  $\Omega$  são os valores iniciais observados para todas as células e  $\Sigma$  é o conjunto de operação de restrições de reparos disponíveis.

### 2.1.2.2 Engenharia de características

Aplicar modelos de aprendizado de máquina após a preparação dos dados pode não ser suficiente para criar modelos interpretáveis e com bom desempenho. A engenharia de características consiste em manipular os dados de entrada de forma a extrair informações relevantes (ZHANG; FOGELMAN-SOULIÉ; LARGERON, 2018). Para isso, os dados brutos são transformados, combinados e selecionados de maneira a produzir um conjunto de características relevantes para o problema. Essa manipulação pode incluir técnicas como redução da dimensionalidade, expansão de dados e seleção de atributos (HE; ZHAO; CHU, 2021).

Além da identificação de características relevantes, a engenharia de características também contribui para a mitigação de variáveis colineares, reduzindo os efeitos da multicolinearidade. A multicolinearidade ocorre quando existe uma relação aproximadamente linear entre duas ou mais variáveis independentes, o que pode comprometer a estabilidade dos modelos e levar a interpretações equivocadas sobre a influência individual das variáveis no resultado (CHAN et al., 2022). Ressalta-se que, no presente trabalho, não foi empregada nenhuma técnica específica adicional para a remoção de colinearidade, sendo sua mitigação decorrente exclusivamente do processo de engenharia de características adotado.

Não existe um padrão definitivo que contenha as melhores técnicas de engenharia de características. Por esse motivo, é crucial experimentar diversas abordagens para uma base de dados específica e observar seus efeitos no desempenho do modelo (VERDONCK et al., 2021). Assim como a etapa anterior, a engenharia de recursos é um processo demorado e requer conhecimento de domínio. Portanto, a automatização desse processo pode economizar tempo e esforços significativos.

O AutoLearn (KAUL; MAHESHWARY; PUDI, 2017), por exemplo, visa gerar novos atributos em duas etapas. Primeiramente são gerados novos recursos usando o erro retornado pela regressão *Ridge* de um par de variáveis (dependente e independente). Os novos recursos são então submetidos a um método *wrapper* com *Randomized Lasso* para selecionar os mais relevantes.

Em Khurana, Samulowitz e Turaga (2017), a automação da engenharia de características é baseada em aprendizado por reforço. O problema é representado como um DAG, chamado grafo de transformações, onde cada nó corresponde a um conjunto de dados com mesmo rótulo e número de linhas. As arestas representam as transformações aplicadas aos nós. Nós especiais (nós soma) podem ter mais de um nó pai e a transformação resultante é a soma dos conjuntos de dados. O DAG é usado para treinar o agente para aprender uma estratégia eficaz de explorar as combinações de transformações disponíveis sob um determinado orçamento.

O AutoCross (LUO et al., 2019) trabalha com o cruzamento de recursos para

capturar interações entre recursos categóricos. O problema é estruturado como uma árvore, onde o nó raiz corresponde ao conjunto de recursos original e cada filho é construído adicionando a si mesmo um cruzamento de pares de seus próprios elementos. O *Beam Search* é usado para explorar a árvore e expandir apenas os nós mais promissores até que o nó solução seja encontrado.

O CAAFE (HOLLMANN; MÜLLER; HUTTER, 2023) utiliza um agente LLM para gerar códigos que modificam o conjunto de dados de entrada, permitindo que esses dados transformados sejam posteriormente processados por algoritmos clássicos. Além disso, o CAAFE produz comentários que explicam a utilidade de cada característica gerada. Para produzir o código, o LLM recebe um prompt contendo detalhes sobre o conjunto de dados e sobre a tarefa de engenharia de características. O código resultante é então executado, e o desempenho do modelo é avaliado para selecionar a melhor transformação.

### 2.1.2.3 Geração do modelo

A etapa de geração de modelos envolve dois problemas, a seleção do modelo e a otimização de hiperparâmetros (OHP). Quando esses problemas são abordados de forma combinada são chamados de CASH (*Combined Algorithm Selection and Hyperparameter optimization problem*) e a seleção do modelo é tratada como um novo hiperparâmetro. O AutoStacker (CHEN et al., 2018) utiliza essa abordagem com algoritmos genéticos. Os pipelines são formados pelo empilhamento de camadas de modelos, que possuem hiperparâmetros como o modelo para cada nó, os hiperparâmetros dos modelos escolhidos em cada nó, o número de camadas e o número de nós em cada camada.

O DarwinML (QI et al., 2019) aborda a geração de modelos de forma separada. Nesse sistema, a arquitetura de uma combinação de modelos é representada por um DAG, em que as arestas indicam o fluxo de dados e os vértices correspondem aos modelos. Para encontrar a melhor arquitetura, é utilizado um AG. Após a seleção da melhor arquitetura, a otimização de hiperparâmetros é aplicada aos 5 melhores indivíduos da população final com otimização bayesiana.

Devido a conquistas notáveis dos algoritmos *Deep Learning* (DL) em diversas tarefas, a otimização de arquiteturas neurais é vista como um campo de pesquisa emergente dentro do AutoML (HE; ZHAO; CHU, 2021) e foi nomeado primeiramente por Zoph e Brain (2017) como *Neural Architecture Search* (NAS). Um exemplo de framework é o BANANAS (WHITE et al., 2019) que organiza os pipelines em um DAG e usa a otimização bayesiana para encontrar a melhor arquitetura. Em Yang et al. (2020), o NSGA-III (*Non-dominated Genetic Sorting Algorithm*) é aplicado em conjunto com uma estratégia de compartilhamento de parâmetros para reduzir a complexidade computacional.

Métodos como DS-Agent (GUO et al., 2024), HuggingGPT (SHEN et al., 2023)

e AgentHPO (LIU; GAO; LI, 2024), fazem seleção e ajuste de hiperparâmetros usando agentes LLM.

### 2.1.3 AutoML para Previsão de Séries Temporais

Embora existam várias ferramentas disponíveis para automatizar pipelines de regressão, muitas delas não fornecem a mesma facilidade na preparação dos dados quando se trata de séries temporais (OLSON et al., 2016; FEURER et al., 2020; GIJSBERS; VANSCHOREN, 2020; KOTTHOFF et al., 2017). Isso significa que o usuário deve organizar os dados de forma que o sistema consiga preservar a ordem temporal da série.

Além disso, a previsão de séries temporais é um problema que requer métodos específicos para lidar com as características peculiares desses dados, como tendências, sazonalidades e mudanças abruptas (PALDINO et al., 2021). Métodos como transformações são amplamente utilizadas para tornar séries estacionárias e extrair suas propriedades. Também existem vários modelos específicos para previsão de séries que devem ser contemplados nesses sistemas como Autoregressive (AR), Autoregressive Integrated Moving Average (ARIMA), Fuzzy Times Series (FTS), Recurrent Neural Networks (RNN), Temporal Convolutional Networks (TCN) e Long Short-Term Memory (LSTM).

Abaixo são descritos frameworks AutoML de código aberto que propõem automatizar todo ou parte do pipeline de previsão de séries temporais.

O FEDOT (NIKITIN et al., 2022a) é um framework que cria pipelines compostos e os organiza como um DAG. Os nós do DAG representam operações que podem ser modelos de previsão, métodos de engenharia de recursos ou de preparação dos dados. A implementação dessas operações pode ser feita em bibliotecas básicas e populares de aprendizado de máquina como Scikit-Learn, Keras, Statsmodel, etc. A escolha do melhor pipeline é feita com programação genética que busca otimizar dois objetivos: desempenho e complexidade do DAG. Na otimização dos hiperparâmetros, o FEDOT usa Otimização Bayesiana com o erro sendo calculado para todo o pipeline.

O AutoTS (WANG et al., 2022a) trabalha com módulos de processamento para resolver um problema NAS para previsão de séries temporais multivariadas. São 5 módulos responsáveis pelo processamento das entradas, extração, aprimoramento e combinação de características e processamento de saída. Em cada módulo é possível usar vários modelos existentes e também otimizar seus hiperparâmetros. Os modelos utilizados são *deep learning*, o que gera um enorme espaço de busca para seleção dos modelos e dos hiperparâmetros em cada módulo. Para resolver esse problema, um mecanismo de busca baseado em poda em dois estágios e análise de grafos de conhecimento foram aplicados.

A busca inicia com o método de poda vertical para otimizar sequencialmente cada módulo e extrair características de desempenho dos componentes do espaço de busca. No

meio da busca, passa-se a usar o método de poda horizontal para analisar efetivamente a importância de cada opção de módulo de acordo com as características aprendidas e filtrar as opções de módulo menos importantes. Muitas opções de módulo no espaço de busca têm muitas sobreposições na configuração do hiperparâmetro e na arquitetura, por isso, os autores ainda projetaram um grafo de conhecimento para ajudar o algoritmo de busca a entender melhor as características de cada opção de módulo e, assim, tomar decisões mais eficazes e melhorar a qualidade da pesquisa (WANG et al., 2022a).

O AutoGluon (SHCHUR et al., 2023) fornece uma ferramenta que automatiza a preparação dos dados e geração de modelos para gerar previsões pontuais e probabilísticas de séries temporais. Os modelos são combinados em uma pilha multicamadas. A primeira camada (base) é formada por vários modelos que usam hiperparâmetros padrão. As saídas dessa camada são concatenadas para alimentar a próxima camada, juntamente com as entradas originais. As próximas camadas são formadas por modelos empilhadores que reutilizam os modelos da camada base para evitar o problema CASH. A última camada agrega as previsões dos modelos de empilhamento usando média ponderada. O AutoGluon retorna um ranking contendo os resultados de todos os modelos e também do conjunto.

Por fim, o TimeSeriesScientist (ZHAO et al., 2025) é um sistema multiagente que automatiza todo o pipeline de previsão de séries temporais univariadas com o auxílio de 4 agentes que são responsáveis pelas tarefas de diagnóstico sobre a série temporal de entrada, seleção de modelos e hiperparâmetros, previsão e geração de relatório final em linguagem natural.

## 2.2 Decomposição Empírica de Modos

A Decomposição Empírica de Modos (EMD, do inglês *Empirical Mode Decomposition*) é um método de decomposição de sinais desenvolvido com base na suposição de que qualquer série temporal não estacionária e não linear consiste em diferentes funções de modos intrínsecos (IMF, do inglês *Intrinsic Mode Functions*) simples de oscilação. A EMD busca identificar essas funções por meio das características de escala de tempo do próprio sinal, sem a necessidade de uma função de base, como ocorre em outros métodos conhecidos, tais como a decomposição de Fourier e Wavelet (HUANG et al., 1998; BARBOSH; SINGH; SADHU, 2020).

Uma IMF deve satisfazer às seguintes condições: (a) o número de extremos e o número de cruzamentos por zero devem ser iguais ou diferir no máximo por um em todo o conjunto de dados e (b) em qualquer ponto, o valor médio do envelope gerado pelos máximos e mínimos locais deve ser zero (HUANG et al., 1998). Esta última condição exige que uma IMF seja estacionária, facilitando seu processo de previsão (ZEILER et al., 2010).

O processo para extrair  $n$  IMFs de um sinal  $S(t)$  é chamado peneiramento e segue

as seguintes etapas:

1. Usando uma spline cúbica, conecte todos os máximos e mínimos locais para gerar os envelopes superior e inferior.
2. Calcule a média  $m(t)$  dos envelopes.
3. Calcule a diferença:  $I(t) = S(t) - m(t)$ .
4. Verifique se  $I(t)$  atende às condições (a) e (b) para ser uma IMF.
  - 4.1. Se sim, então  $IMF_n = I$  e o sinal  $S(t)$  é substituído pelo resíduo  $r_n(t) = S(t) - I$ .
  - 4.1. Senão,  $S(t) = I(t)$ .
5. Repita os passos 1 a 4 até que  $S(t)$  se torne uma função monotônica da qual nenhuma outra IMF pode ser extraída.

Ao final do peneiramento, o sinal  $S(t)$  pode ser expresso como na Equação 2.2, onde  $IMF_i(t)$  representa o  $i$ -ésimo componente e  $r_n$  é o resíduo final que é uma constante ou uma tendência.

$$S(t) = \sum_{i=1}^n IMF_i(t) + r_n(t) \quad (2.2)$$

Na literatura, o EMD é amplamente utilizado na criação de modelos híbridos combinados com modelos de aprendizado de máquina para previsão de séries temporais. Em trabalhos recentes, os autores verificaram que a modelagem usando as IMFs geradas pelo EMD pode melhorar a capacidade de previsão em comparação com a modelagem utilizando apenas a série original (MOUNIR; OUADI; JRHILIFA, 2023; LI; ZHANG; LI, 2023; LEE, 2020). Em (LIU; LIU, 2020), os autores compararam a combinação do modelo *High Order Fuzzy Cognitive Maps* (HFCM) com EMD e com Wavelet. Os resultados mostraram que o EMD-HFCM foi superior em seis dos oito conjuntos de dados avaliados.

### 2.3 Seleção de características

A seleção de características é um processo essencial na construção de modelos de previsão, no qual um subconjunto representativo e relevante de características é selecionado. Esse subconjunto deve ser parcimonioso, ao mesmo tempo em que oferece o melhor desempenho para o modelo (HIRA; GILLIES, 2015; KOPRINSKA; RANA; AGELIDIS, 2015). Esse processo tornou-se indispensável em aplicações que lidam com dados de alta dimensão, como séries temporais multivariadas, pois permite reduzir a dimensionalidade dos dados e eliminar características irrelevantes e redundantes. Um conjunto de características

menor resulta em treinamento mais rápido e menor complexidade do modelo de previsão (SHEIKHPOUR et al., 2017).

Encontrar o melhor subconjunto de características não é uma tarefa simples devido aos extensos espaços de busca envolvidos. Nos modelos de previsão de séries temporais, valores defasados são utilizados como entrada, o que significa que a dimensionalidade do problema aumenta à medida que mais dados históricos são considerados, especialmente em séries temporais multivariadas.

Os métodos de seleção de características podem ser categorizados em (HE; ZHAO; CHU, 2021):

- **Filtro:** classifica cada característica com base em alguma medida estatística e, em seguida, seleciona-as com base em um limiar. Esses métodos são independentes dos modelos de previsão e, portanto, computacionalmente menos exigentes.
- **Wrapper:** as características são classificadas de acordo com a precisão fornecida pelo modelo de previsão. Esses métodos tendem a ter um desempenho melhor, uma vez que levam em consideração a hipótese do modelo treinado no espaço de características. No entanto, eles são computacionalmente mais caros (HIRA; GILLIES, 2015).
- **Embutido:** a seleção de características é realizada como parte do procedimento de aprendizado, o que os torna específicos para um único modelo.

Com o crescente volume de dados de alta dimensão, os métodos de seleção de filtro ganharam destaque devido à sua rápida velocidade de processamento, independência de modelos de previsão e capacidade de lidar com o sobre-ajuste (*overfitting*) (YU; LIU; LI, 2018). Além disso, a seleção de recursos causais tornou-se um tópico de pesquisa emergente em aprendizado de máquina (RUNGE et al., 2019). Esses métodos podem fornecer uma compreensão mais profunda do processo subjacente de geração de dados, o que pode melhorar o poder explicativo e a robustez dos modelos de previsão (YU et al., 2020).

### 2.3.1 Descoberta causal

Um grafo causal  $G = (V, E)$  é uma representação gráfica que descreve as relações causais de um sistema. Em outras palavras, uma aresta direcionada  $X \rightarrow Y$  estabelece uma relação de causa e efeito entre a causa  $X$  e seu efeito  $Y$ . Dado um conjunto  $\mathcal{G}$  que consiste em todos os grafos definidos pelas variáveis  $V$  de um conjunto de dados  $D$ , o problema da descoberta causal é encontrar o grafo  $G^*$  que seja uma possível explicação para  $D$  (ZANGA; OZKIRIMLI; STELLA, 2022).

Em grafos direcionados acíclicos (DAG), a condição *d-separação* deve implicar independência condicional na distribuição de probabilidade. Embora a *d-separação* forneça



condições necessárias, ela não é suficiente para estabelecer relações de independência condicional, pois a distribuição de probabilidade pode conter outras relações de independência condicional que não são necessariamente implicadas pela *d-separação* aplicada ao grafo. Quando assumimos a inexistência dessas relações adicionais, dizemos que a Suposição de Fidelidade Causal é aceita (GLYMOUR; ZHANG; SPIRITES, 2019).

Quando um grafo satisfaz a suposição de fidelidade, o cobertor de Markov (*Markov Blanket*) (MB) de uma variável  $X$  no grafo é único e composto por seus pais (causas diretas), filhos (efeitos diretos) e cônjuges (outros pais dos filhos). De acordo com a Condição Causal de Markov (equivalente à condição *d-separação*), todas as outras variáveis que não pertencem ao MB são probabilisticamente independentes quando a variável em questão é condicionada em seu MB. Portanto, o MB de  $X$  é o subconjunto ideal mínimo de recursos de  $X$  a ser usado na seleção de características (EBERHARDT, 2017; YU; LIU; LI, 2018).

Considerando o problema da seleção de *lags* em séries temporais, o MB de uma variável é reduzido apenas aos seus pais, pois valores no tempo  $t$  não podem causar valores no tempo  $t - 1$ . Ou seja, o condicionamento apenas nos pais de uma variável é suficiente para estabelecer independência condicional e identificar links espúrios (RUNGE et al., 2019).

Em teoria, os subconjuntos de características obtidos por métodos de seleção causal estão mais próximos do MB de uma variável em comparação com métodos não causais. Como resultado, eles podem melhorar a capacidade explicativa dos modelos de previsão e torná-los mais robustos. Isso ocorre porque os relacionamentos causais implicam no mecanismo subjacente dos dados, o que faz com que esses relacionamentos sejam persistentes em diferentes ambientes (YU et al., 2020).

### 2.3.2 PCMCI

O PCMCI, desenvolvido por (RUNGE et al., 2019), é um método de descoberta causal que gera um grafo causal diretamente de séries temporais multivariadas. O PCMCI consiste de duas etapas: primeiro, usa o algoritmo PC para identificar os pais  $\hat{\mathcal{P}}(X_t^j)$  para todas as variáveis da série temporal  $X_t^j \in X_t^1, \dots, X_t^N$ , e segundo, aplica o teste de independência condicional momentânea (MCI) para testar ligações indiretas  $X_{t-\tau}^i \rightarrow X_t^j$ .

O PC é um algoritmo que usa testes iterativos de independência para descoberta de conjuntos de Markov. Para cada variável  $X_t^j$  é feita a inicialização dos pais preliminares  $\hat{\mathcal{P}}(X_t^j) = (X_{t-1}, X_{t-2}, \dots, X_{t-\tau_{max}})$ . Primeiro, testes de independência incondicionais são aplicados para remover  $X_{t-\tau}^i$  de  $\hat{\mathcal{P}}(X_t^j)$ , caso a hipótese nula  $X_{t-\tau}^i \perp\!\!\!\perp X_t^j$  não for rejeitada a um nível de significância  $\alpha_{PC}$ . Os pais preliminares são classificados por seu valor absoluto estatístico de teste.



A Figura 7 ilustra o  $PC_1$  para duas variáveis  $X_1$  e  $X_3$  onde a intensidade da cor representa o valor absoluto estatístico de teste das variáveis dependentes (cores escuras indicam valores maiores). Os nós cinza representam as variáveis independentes. Em seguida, testes de independência condicional  $X_{t-\tau}^i \perp\!\!\!\perp X_t^j | \mathcal{L}$  são executados, onde  $\mathcal{L}$  são os pais mais fortes em  $\hat{\mathcal{P}}(X_t^j) \setminus X_{t-\tau}^i$  e os pais independentes são removidos de  $\hat{\mathcal{P}}(X_t^j)$ . Dessa forma, o PC converge para apenas algumas condições relevantes que incluem os pais causais com alta probabilidade (rosa escuro/azul escuro) e potencialmente alguns falsos positivos (setas pontilhadas).

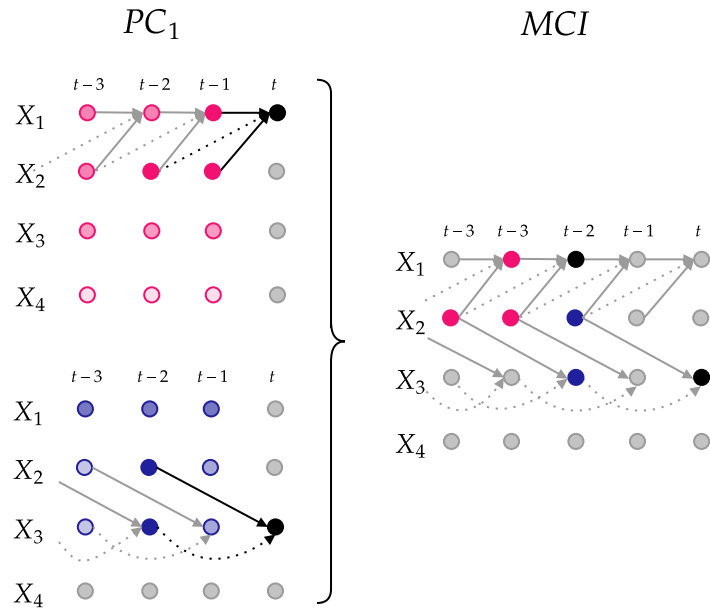


Figura 7 – Ilustração do método PCMCI.

Na segunda etapa, o MCI usa os pais estimados pelo PC para identificar causas indiretas. No exemplo da Figura 7, as condições  $\hat{\mathcal{P}}(X_t^3)$  são suficientes para estabelecer a independência condicional para testar  $X_{t-2}^1 \rightarrow X_t^3$ . Além disso, os pais defasados de  $\hat{\mathcal{P}}(X_1^{t-2})$  são inseridos como condições adicionais e são responsáveis por manter a taxa de falsos positivos a um nível esperado.

O método PCMCI fundamenta-se na Condição Causal de Markov e na Suposição de Fidelidade. Além disso, não faz a suposição de existência de efeitos causais contemporâneos, presume estacionariedade e ausência de confundidores, ou seja, de causas comuns diretas não observadas (RUNGE et al., 2019).

O PCMCI possui complexidade polinomial no número de variáveis  $N$  e  $\tau_{max}$ . No pior caso, onde o grafo está completamente conectado, a complexidade computacional do estágio de seleção de condição  $PC_1$  para  $N$  variáveis equivale a  $N^3\tau_{max}^2$ . A etapa MCI envolve ainda testes  $N^2\tau_{max}$  (para  $\tau > 0$ ). Portanto, o pior caso de complexidade computacional total no número de variáveis é polinomial e dado por  $N^3\tau_{max}^2 + N^2\tau_{max}$ .

## 2.4 Otimização Evolutiva Multitarefa

A Otimização Multitarefa Evolutiva (EMO, do inglês *Evolutionary Multitask Optimization*) representa uma vertente dos Algoritmos Evolutivos focada na abordagem simultânea de múltiplos problemas de otimização. A EMO está fundamentada na premissa de que, se houver conhecimento comum entre diferentes tarefas, esse conhecimento deve ser compartilhado durante o processo de resolução. Essa transferência de conhecimento confere uma vantagem competitiva em relação à abordagem isolada de cada problema, evidenciando melhorias em termos de convergência, obtenção de soluções mais eficazes e otimização no consumo de recursos computacionais (OSABA et al., 2022; ZHAO et al., 2023).

Duas formas estado da arte em OME são (OSABA et al., 2022; TAN; FENG; JIANG, 2021):

1. Homogênea: concentra-se em problemas que compartilham o mesmo espaço de busca, possibilitando a resolução através de um único processo de pesquisa em população única. A transferência de conhecimento entre as tarefas ocorre mediante a aplicação de operadores de cruzamento.
2. Heterogênea: direciona-se para problemas que apresentam espaços de busca distintos. Nesse cenário, cada tarefa é abordada por um processo de pesquisa individual, executados de forma paralela e trocando informações periodicamente conforme uma política predefinida de transferência de conhecimento.

O principal algoritmo na categoria de OME Homogênea, discutido na literatura, é o Algoritmo Evolutivo Multifatorial (AEMF), desenvolvido por Gupta, Ong e Feng (2016). O AEMF foi inspirado nos modelos de herança multifatorial da biologia evolutiva que propõem que as iterações de fatores culturais e genéticos influenciam o desenvolvimento de características complexas entre descendentes. Os fatores culturais são comumente originados do aprendizado social ou da transmissão de costumes de pais para filhos. O equivalente da herança multifatorial no AEMF é que cada problema de otimização contribui com um ambiente cultural distinto no qual a prole pode se desenvolver (GUPTA; ONG; FENG, 2016).

O AEMF segue a mesma premissa dos algoritmos evolutivos clássicos, com a incorporação de quatro propriedades que possibilitam comparar os indivíduos  $p_i$  da população  $\mathcal{P}$  em um ambiente multitarefa, com  $i \in \{1, 2, \dots, |\mathcal{P}|\}$  (GUPTA; ONG; FENG, 2016).

**Custo Fatorial:** para uma tarefa  $T_j$ , o custo fatorial  $\psi_j^i$  do indivíduo  $p_i$  é igual ao valor da função objetivo correspondente  $f_j^i$ .

**Classificação fatorial:** a classificação fatorial  $r_j^i$  do indivíduo  $p_i$  na tarefa  $T_j$  é o índice de  $p_i$  na lista de indivíduos classificados em ordem crescente em relação a  $\psi_j$ .

**Aptidão escalar:** é a capacidade que um indivíduo possui para resolver tarefas versáteis. A aptidão escalar  $\varphi_i$  é calculada com base na melhor classificação fatorial de um indivíduo em todas as tarefas:  $\varphi_i = \frac{1}{\min_j\{r_j^i\}}$ .

**Fator de habilidade:** o fator de habilidade  $\tau_i$  é a tarefa na qual o indivíduo  $p_i$  é mais eficaz e pode ser calculado como  $\tau_i = \operatorname{argmin}\{r_j^i\}$ .

**Otimidade Multifatorial:** um indivíduo  $p^*$  é considerado ótimo no sentido multifatorial, se  $\psi_j^* \leq \psi_j^i$  para todo  $p_i \in \mathcal{P}$ .

## 2.5 Conclusão

Neste capítulo, o AutoML é abordado como um problema de otimização único que engloba as três etapas principais de um pipeline de aprendizado de máquina: preparação de dados, engenharia de características e geração de modelos. A literatura revisada revela uma variedade de métodos de otimização empregados em AutoML, juntamente com estratégias para reduzir tanto o tempo de processamento quanto o custo computacional desses métodos.

O conceito de AutoML não se restringe à automação de todo o pipeline de aprendizado de máquina. Portanto, uma revisão por etapa foi apresentada para abordar métodos específicos. Vale ressaltar que, como o AutoML incorpora todos os processos relacionados a um pipeline, o qual é complexo devido à diversidade de tipos de dados de entrada e objetivos (como classificação, regressão, previsão de séries temporais, clusterização, entre outros), diversas técnicas são utilizadas para lidar com esses problemas. Em especial, ficou evidenciado que o uso de algoritmos baseados em população e a otimização bayesiana são abordagens muito exploradas pelos AutoML pesquisados.

Mesmo quando a pesquisa é restrita aos AutoML para previsão de séries temporais encontra-se o mesmo padrão de diversidade em relação aos dados (univariados e multivariados), aos modelos (estatísticos, aprendizado de máquina, deep learning) utilizados, bem como nas técnicas de pré-processamento empregadas. Apesar disso, a abordagem de previsão probabilística e multivariada ainda é pouco explorada pelos métodos. Também se constatou a ausência de métodos causais em qualquer etapa dos sistemas de AutoML analisados.

Outros conceitos discutidos nesta seção fornecem a base necessária para o desenvolvimento do método AutoML proposto nesta tese. A EMD possibilita a extração de padrões intrínsecos de séries temporais não lineares e não estacionárias, fornecendo características significativas para modelagem. A seleção causal de características permite

---

identificar os subconjuntos mais relevantes e parcimoniosos, reduzindo a complexidade do modelo e melhorando seu desempenho em dados de alta dimensão. Por fim, a Otimização Multitarefa Evolutiva oferece uma estratégia para selecionar e ajustar modelos de forma eficiente, aproveitando a transferência de conhecimento entre tarefas relacionadas.

### 3 AUTOML PARA PREVISÃO DE SÉRIES TEMPORAIS MULTIVARIADAS

Neste capítulo apresenta-se o método Aprendizado de Máquina Automatizado Baseado em Decomposição, Causalidade e Otimização Evolutiva Multitarefa para Previsão de Séries Temporais (em inglês: *Automated Machine Learning Based on Decomposition, Causality, and Evolutionary Multitask Optimization for Time Series Forecasting* - AutoDCE-TS). O AutoDCE-TS é um sistema de Múltiplas Entradas e Saída Única (MISO), desenvolvido dentro de uma estrutura de pipeline fixa de 4 camadas que automatiza as etapas de extração e seleção de características, seleção e geração de modelos e previsão. A Figura 2 apresenta a estrutura geral do AutoDCE-TS.

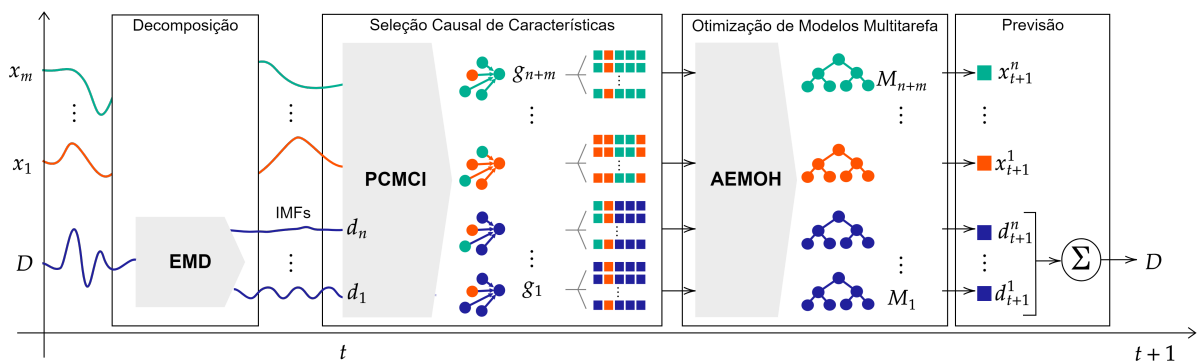


Figura 2 – Estrutura geral do AutoDCE-TS, composta pelas camadas Decomposição, Seleção Causal de Características, Otimização de Modelos Multitarefa e Previsão.  $D$  e  $x_1, x_2, \dots, x_m$  representam a variável endógena e as variáveis exógenas, respectivamente. O método EMD (*Empirical Mode Decomposition*) decompõe  $D$  em *Intrinsic Mode Functions* (IMFs)  $d_1, d_2, \dots, d_n$ . O método PCMCI gera os grafos causais  $g_1, g_2, \dots, g_{n+m}$ , onde  $n$  é o número de IMFs e  $m$  é o número de variáveis exógenas. O Algoritmo Evolutivo Multifatorial para Otimização de Hiperparâmetros (AEMOH) seleciona os melhores modelos e hiperparâmetros  $M_1, M_2, \dots, M_{n+m}$ . O símbolo  $\Sigma$  representa a reconstrução da variável endógena  $D$  no tempo  $t+1$ .

A entrada do método consiste em uma série temporal multivariada composta por uma variável endógena  $D$  e variáveis exógenas  $x_q$ , com  $q = \{1, 2, \dots, m\}$ . A camada Decomposição é responsável por decompor a variável  $D$  em IMFs  $d_j$ , com  $j = \{1, 2, \dots, n\}$ , utilizando o método EMD. Nenhuma das IMFs geradas, incluindo o resíduo final, é descartada.

A camada Seleção Causal de Características utiliza as  $n$  IMFs juntamente com as  $m$  variáveis exógenas para gerar  $n+m$  grafos causais  $g_i$ , com  $i = \{1, 2, \dots, n+m\}$ , usando o método PCMCI. Dessa forma, um grafo causal é gerado para cada variável (IMFs e variáveis exógenas). Os grafos para as IMFs são construídos considerando as defasagens de todas as variáveis exógenas e a própria IMF correspondente. Em contraste, os grafos

para as variáveis exógenas são construídos considerando apenas as defasagens da variável exógena correspondente e das outras variáveis exógenas, ou seja, sem usar informações das IMFs.

As relações causais extraídas de cada grafo são então usadas para selecionar as características que irão compor o conjunto de dados para o treinamento dos modelos de previsão de cada variável. Note que as variáveis do problema (IMFs e exógenas) são tratadas individualmente, e que o número de IMFs tem impacto no aumento da dimensionalidade do conjunto de variáveis. Por essa razão, a decomposição é aplicada apenas à variável endógena.

A adoção do método PCMCI para seleção de variáveis é justificada por sua capacidade de reduzir a dimensionalidade do problema com base nas relações causais entre as variáveis (ver Seção 5.1 do Capítulo 5). Além disso, o uso do EMD é motivado pelos ganhos observados na acurácia da previsão (ver Seção 5.2 do Capítulo 5).

O PCMCI também requer que a série temporal seja estacionária, portanto, os testes Augmented Dickey-Fuller (ADF) (CHEUNG; LAI, 1995) e Kwiatkowski-Phillips-Schmidt-Shin (KPSS) (KWIATKOWSKI et al., 1992) são realizados com um nível de confiança de 95%. Quando um comportamento não estacionário é identificado, aplica-se a primeira diferença. O PCMCI também assume suficiência causal, significando que todas as variáveis de confusão foram medidas e incluídas. No AutoDCE-TS, essa suposição se reflete na escolha de hiperparâmetros como a defasagem máxima ( $\tau_{max}$ ) e o conjunto de variáveis de entrada  $D$  e  $x_q$ , que são assumidos como suficientes para descrever o sistema subjacente e prever a variável-alvo  $D$ .

A camada Otimização de Modelos Multitarefa realiza a seleção do melhor modelo e de seus hiperparâmetros  $M_i$  como um problema CASH para cada variável a ser prevista. Nesse contexto, os  $n + m$  problemas CASH são resolvidos simultaneamente utilizando o Algoritmo Evolutivo Multifatorial para Otimização de Hiperparâmetros (AEMOH), conforme descrito na Seção 3.1. É importante destacar que o AEMOH é um algoritmo de otimização multitarefa, no qual cada problema CASH é tratado como uma tarefa separada e todas as tarefas são otimizadas simultaneamente, permitindo transferência de conhecimento entre elas. A aplicação do AEMOH é bem adequada ao AutoDCE-TS, uma vez que séries temporais extraídas de um determinado domínio de aplicação frequentemente exibem similaridades, o que possibilita o compartilhamento de informações durante a seleção de modelos e hiperparâmetros. Além disso, a otimização simultânea contribui para a redução do custo computacional.

Após a conclusão do processo AEMOH, os modelos selecionados  $M_j$  são treinados e encaminhados para a camada Previsão. Essa camada final utiliza os modelos treinados  $M_i$  para gerar previsões para cada uma das IMFs  $d_j$  e das variáveis exógenas  $x_q$ . Os valores previstos das IMFs são então somados para reconstruir a previsão da variável endógena

$D$  no tempo  $t + 1$ . Para previsões multi-passos à frente, o método emprega um processo recursivo de previsão, no qual todas as variáveis são atualizadas iterativamente com base nos valores previstos dos passos anteriores.

### 3.0.1 Algoritmo Evolutivo Multifatorial para Otimização de Hiperparâmetros

O Algoritmo Evolutivo Multifatorial para Otimização de Hiperparâmetros (AE-MOH) foi inspirado no algoritmo evolutivo desenvolvido por (SILVA et al., 2020). Neste algoritmo, os autores formulam o problema de otimização de hiperparâmetros com a intenção de encontrar um equilíbrio entre dois objetivos conflitantes: acurácia e complexidade do modelo. Para tratar o problema como uma otimização de objetivo único, foi implementada uma estratégia de Torneio Duplo para equilibrar a seleção entre os dois objetivos.

O problema de otimização individual para cada modelo foi formulado da seguinte forma:

**Otimizar:**

$$\text{minimizar } f_1 = N^{-1} \sum_{i=0}^N \text{NRMSE}(i) \quad (3.1)$$

$$\text{minimizar } f_2 = \mathcal{C} \quad (3.2)$$

**Onde:**

$$\text{NRMSE} = \frac{\sqrt{\sum_{t=0}^n (y(t) - \hat{y}(t))^2}}{y_{max} - y_{min}} \quad (3.3)$$

$$\mathcal{C} = A * \sum_{i=1}^A (P_i) \quad (3.4)$$

A função objetivo  $f_1$  na Eq. (3.1) determina a acurácia do modelo, quantificada pela média do Erro Quadrático Médio Normalizado (NRMSE) (Eq. 3.3) em  $N$  avaliações, onde  $\hat{y}(t)$  e  $y(t)$  representam os valores previstos e reais no tempo  $t$ , respectivamente. A função objetivo  $f_2$  na Eq. (3.2) aborda a complexidade do modelo  $\mathcal{C}$ , que é definida pelo número de árvores de regressão ( $A$ ) e suas profundidades ( $P$ ), conforme mostrado na Eq. (3.4).

### 3.0.2 Estrutura do AEMOH

Seguindo o fluxograma da Figura 3, todas as etapas do desenvolvimento do AEMOH serão descritas.

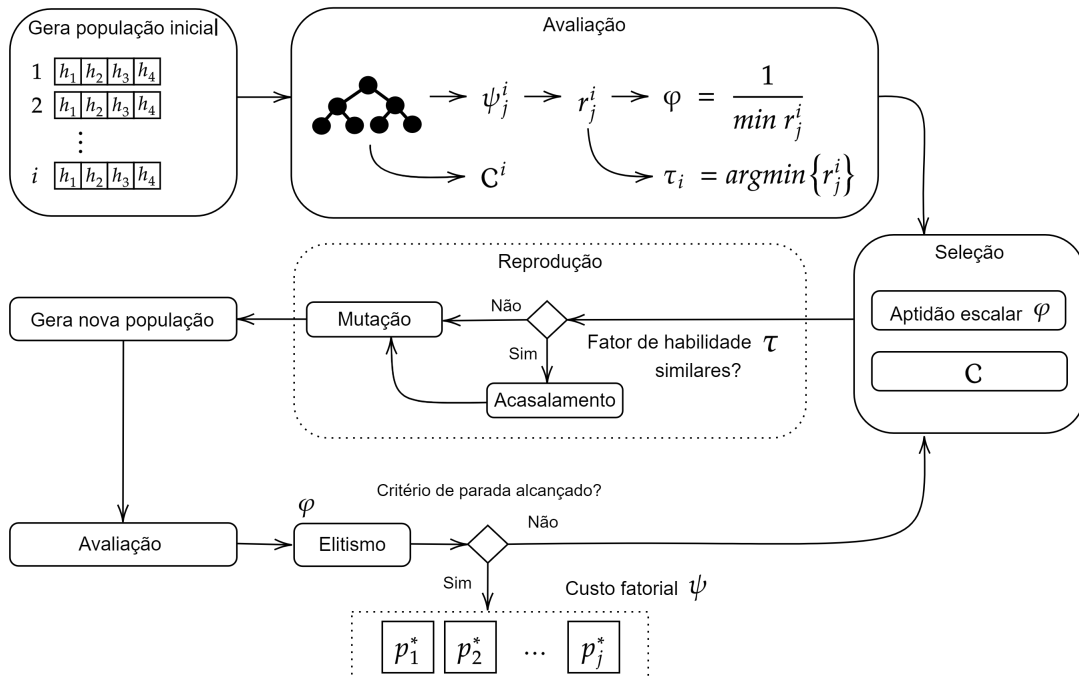


Figura 3 – Fluxograma do AEMOH. Os hiperparâmetros  $h_1$ ,  $h_2$ ,  $h_3$  e  $h_4$  representam, respectivamente, o número de árvores de decisão, o número máximo de características consideradas na busca pela melhor divisão, o número mínimo de amostras necessárias em um nó folha e o estimador.  $i$  denota o número de indivíduos na população e  $j$  o número de tarefas.  $\psi$  é o custo fatorial,  $r$  o ranqueamento fatorial,  $\varphi$  a aptidão escalar,  $C$  a complexidade do estimador, e  $\tau$  o fator de habilidade. O indivíduo  $p_j^*$  representa o melhor indivíduo para a tarefa  $j$ .

## 1. Geração da População Inicial

Os indivíduos que constituem a população do AEMOH são representados por um vetor (genótipo) que incorpora os hiperparâmetros  $h_1$ ,  $h_2$ ,  $h_3$  e  $h_4$ . Neste trabalho, foram empregados *ensembles* baseados em árvores de regressão, a saber: Random Forest (RF) (BREIMAN, 2001), Light Gradient Boosting Machine (LGBM) (KE et al., 2016) e Extreme Gradient Boosting (XGBoost) (CHEN; GUESTRIN, 2016a). Os hiperparâmetros otimizados foram: o número de árvores de regressão ( $h_1$ ), o número máximo de características consideradas na busca pela melhor divisão ( $h_2$ ), o número mínimo de amostras necessárias em um nó folha ( $h_3$ ) e o estimador ( $h_4$ ).

A população inicial é gerada aleatoriamente, respeitando os limites do espaço de busca para cada hiperparâmetro. Esses limites são definidos como:  $h_1 \in 10, 20, \dots, 500$ ,  $h_2 \in 0.2, 0.3, \dots, 0.7$ ,  $h_3 \in 1, 2, 3, \dots, 30$ , e  $h_4 \in \text{RF, LGBM, XGBoost}$ .

O intervalo adotado para  $h_1$  baseia-se nos resultados apresentados por Probst e Boulesteix (2017), que demonstram que o aumento do número de árvores de 10 para aproximadamente 500 produz ganhos substanciais de desempenho na maior parte dos cenários. A partir desse ponto, a melhoria tende a se estabilizar, de modo que



valores superiores raramente justificam o aumento do custo computacional.

O hiperparâmetro  $h_2$ , que controla o número máximo de características disponíveis em cada divisão, influencia diretamente a diversidade das árvores. Embora seu valor teórico varie entre 0 e 1, adotou-se o intervalo de 0.2 a 0.7 para evitar extremos: valores muito baixos tendem a gerar árvores excessivamente aleatórias, enquanto valores muito altos produzem árvores quase determinísticas, reduzindo a diversidade do ensemble.

Por fim,  $h_3$  regula a complexidade das árvores ao definir o número mínimo de amostras em um nó folha. O intervalo de 1 a 30 foi selecionado por equilibrar dois comportamentos: valores próximos de 1 permitem árvores profundas e altamente adaptadas aos dados, enquanto valores maiores restringem o crescimento e podem tornar as árvores rasas demais, prejudicando a capacidade de modelagem.

## 2. Avaliação

Na etapa de avaliação, um indivíduo é transformado em um *ensemble* treinado e então avaliado em cada tarefa usando uma janela deslizante. O valor de  $f_{1_i}^j$  para o indivíduo  $i$  na tarefa  $j$  é usado como o custo factorial,  $\psi_j^i = f_{1_i}^j$ . A partir de  $\psi$ , outras propriedades são derivadas, como ranqueamento factorial ( $r$ ), aptidão escalar ( $\varphi$ ) e fator de habilidade ( $\tau$ ), conforme descrito anteriormente.

## 3. Seleção

O operador de seleção é responsável por escolher os indivíduos mais aptos da população para a reprodução. No AEMOH, a estratégia adotada é o método do torneio duplo. Na primeira etapa, dois pares de indivíduos são selecionados aleatoriamente da população, e cada par compete com base em sua aptidão escalar. Na segunda rodada, os vencedores da etapa anterior competem entre si, desta vez com base na soma da complexidade  $C_i^j$ .

## 4. Reprodução

No AEMOH, os indivíduos podem se reproduzir por acasalamento intracultural ou via mutação. O acasalamento intracultural é o operador que combina características de pais semelhantes durante a reprodução. Embora o AEMF clássico permita acasalamento intercultural (entre pais distintos), um estudo recente de (TAN; LUO; ZHONG, 2023) demonstra que a transferência de conhecimento entre problemas distintos, conhecida como transferência negativa, impacta significativamente o desempenho da otimização. Devido ao impacto da transferência negativa, o autor enfatiza a importância de medir adequadamente a similaridade dos problemas.

Para realizar essa tarefa, o AEMOH usa uma matriz de similaridade  $M_s$  que compara todas as séries temporais por meio da distância de Jensen-Shannon. Essa

medida é simétrica e avalia a similaridade entre duas distribuições de probabilidade (MENÉNDEZ et al., 1997), como mostrado na Eq. 3.5, onde  $M = (P + Q)/2$  e  $D_{KL}$  representa a divergência de Kullback-Leibler. A distância de Jensen-Shannon é limitada entre 0 e 1, onde  $JSD(P \parallel Q) = 0$  indica distribuições semelhantes e  $JSD(P \parallel Q) = 1$  indica que as distribuições são muito diferentes. O limiar  $L$ , que determina se duas séries temporais são consideradas suficientemente semelhantes no contexto dos problemas, é calculado pela Eq. 3.6.

$$JSD(P \parallel Q) = \sqrt{\frac{D(P \parallel M) + D(Q \parallel M)}{2}} \quad (3.5)$$

$$L = \frac{\max(M_s)}{2} \quad (3.6)$$

Para que o acasalamento ocorra, dois indivíduos selecionados (pais),  $i_1$  e  $i_2$ , devem ter fatores de habilidade semelhantes, ou seja,  $JSD(\tau_{i_1} \parallel \tau_{i_2}) < L$ . Durante o acasalamento, os pais são ordenados como  $i_1$  e  $i_2$  de acordo com seus objetivos  $f_1$ . Para todos os genes, o processo de combinação contribuirá significativamente para o indivíduo de melhor classificação (com taxas de 0.7 e 0.3). Para genes codificados como valores reais, será aplicada uma combinação linear como  $i_N = 0.7i_1 + 0.3i_2$ . Para genes categóricos, o valor de  $i_N$  será  $i_1$  com probabilidade de 0.7 ou  $i_2$  com probabilidade de 0.3.

Na reprodução por mutação, um dos pais é selecionado aleatoriamente e sofre perturbações aleatórias, restritas aos espaços de busca de cada gene. A mutação pode ocorrer sempre que o acasalamento não acontecer ou após o acasalamento. Incluir este último caso visa garantir que, em situações de problemas muito semelhantes, a diversidade genética da população seja preservada.

## 5. Elitismo

O AEMOH adota uma estratégia elitista que garante a sobrevivência dos melhores indivíduos ao longo das gerações. Esses indivíduos de destaque são identificados pela sua Aptidão Escalar ( $\varphi_i$ ).

## 6. Critério de Parada

O AEMOH será interrompido quando um número predefinido de gerações for atingido. Na etapa final, um indivíduo  $p^*$  é selecionado para cada tarefa  $j$ , utilizando como critério de seleção o menor custo factorial observado para a respectiva tarefa.

### 3.1 Conclusão

Neste capítulo, foi apresentado o método AutoDCE-TS, que automatiza as etapas de extração e seleção de características, além da escolha e geração do modelo. A extração de características é realizada por meio do método de decomposição EMD, enquanto a seleção utiliza o método de descoberta causal PCMCI, responsável por gerar grafos causais para cada variável da série temporal multivariada. Já a escolha dos modelos e de seus hiperparâmetros é formulada como um problema de otimização evolutiva multitarefa.

No capítulo seguinte, são descritos três experimentos distintos conduzidos para avaliar o desempenho do AutoDCE-TS.

## 4 EXPERIMENTOS COMPUTACIONAIS

Neste capítulo serão apresentados três experimentos, a saber:

1. Comparação de métodos de seleção de características não causais e o PCMCI. O objetivo desse experimento foi comparar o método de descoberta causal PCMCI com outros métodos não causais para seleção de características em séries temporais multivariadas em diferentes bases de dados. Para isso, foram considerados métodos das categorias filtro, wrapper e embutidos. A comparação levou em conta o tempo de execução, o número de variáveis selecionadas, a capacidade preditiva do modelo para horizontes de 1, 3 e 5 passos à frente utilizando essas variáveis e a quantidade de relações espúrias identificadas.
2. Estudo de ablação. Este experimento teve como objetivo investigar como a seleção causal de características e a decomposição impactam no desempenho do método AutoDCE-TS em termos de precisão de previsão e número de variáveis selecionadas.
3. Comparação do Método AutoDCE-TS com outros Métodos de Previsão de Séries Temporais Multivariadas. O objetivo principal desta análise foi verificar se a abordagem de previsão automática proposta, AutoDCE-TS, é competitiva em relação a outras abordagens AutoML específicas para previsão de séries temporais, bem como modelos de aprendizado profundo. A comparação com modelos de aprendizado profundo é justificada pela capacidade desses modelos de aprenderem representações dos dados, o que também pode ser visto como uma forma de automatização da seleção e extração de características. Neste experimento, compara-se a precisão de previsão para previsões pontuais de 1 a 30 passo à frente. Além disso, o tempo de execução foi analisado para os métodos AutoML.

A Figura 4 ilustra a metodologia usada na realização de cada experimento. Foram utilizadas diferentes bases de dados, nomeadas de acordo com seus respectivos domínios de aplicação: econômico, energia, internet das coisas (IOT) e clima. Uma descrição resumida pode ser vista na Tabela 1. Foi realizado um pré-processamento nas bases de dados originais para imputar valores faltantes e corrigir erros. Detalhes sobre os dados são apresentados no Apêndice A.

A métrica para calcular a precisão de previsão dos experimentos foi o Erro Quadrático Médio Normalizado (NRMSE), calculado de acordo com a Equação (4.1), onde  $y_{max}$  e  $y_{min}$  são os valores máximo e mínimo do conjunto de dados de teste.

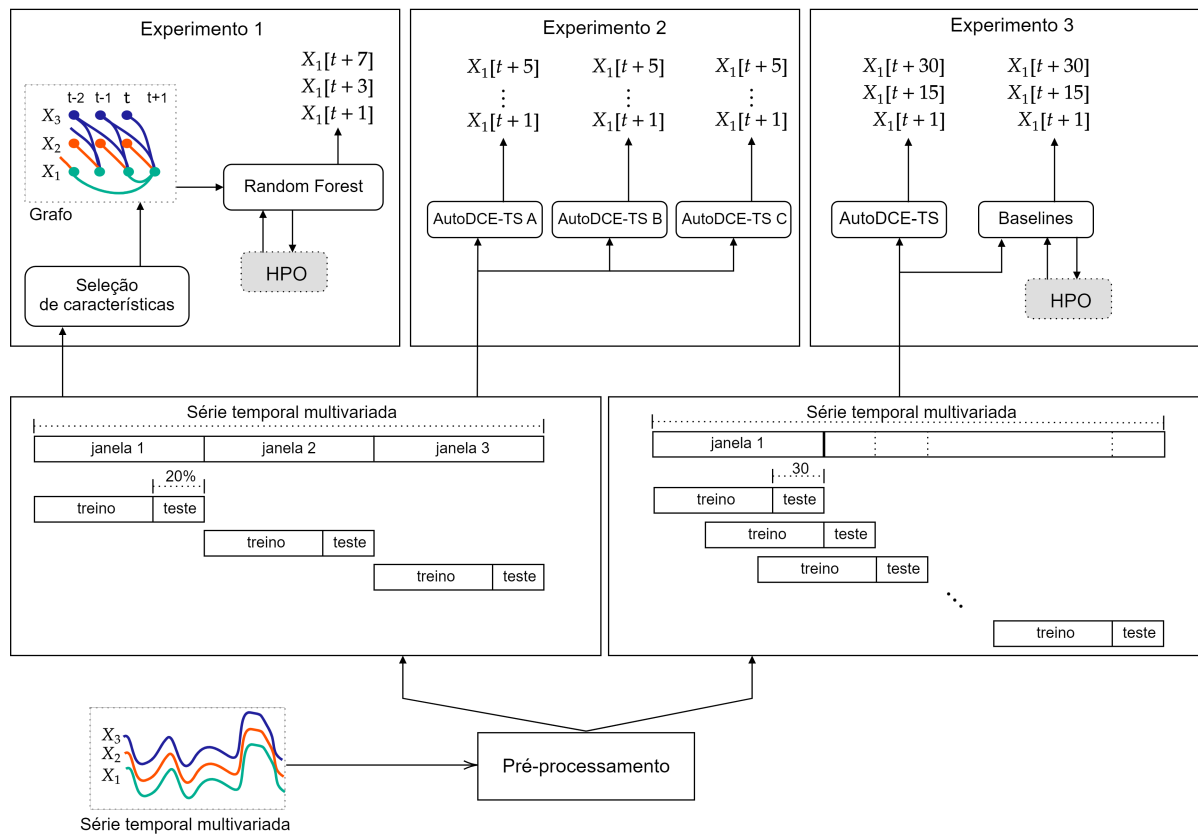


Figura 4 – Metodologia usada na realização dos experimentos 1, 2 e 3.

Tabela 1 – Descrição das bases de dados

Base de dados	Granularidade	Número de amostras	Número de variáveis	Alvo
ECONOMICO_1	diária	2.970	6	Índice médio de Bitcoin para USD
ECONOMICO_2	diária	8.469	6	Índice médio Dow Jones
ECONOMICO_3	diária	5.496	6	Índice médio do mercado FOREX EUR-GBP
ECONOMICO_4	diária	5.496	6	Índice médio do mercado FOREX EUR-USD
ECONOMICO_5	diária	5.496	6	Índice médio do mercado FOREX GBP-USD
ENERGIA_1	horário	43.802	9	Produção de energia eólica
ENERGIA_2	horário	35.003	24	Consumo de energia elétrica
ENERGIA_3	horário	35.003	24	Consumo de energia solar
ENERGIA_4	horário	35.003	24	Consumo de energia eólica
ENERGIA_5	horário	35.003	24	Preço da energia elétrica
IOT_1	minuto	100.000	14	Consumo de energia por eletrodomésticos
IOT_2	minuto	100.00	27	Consumo de energia por eletrodomésticos
IOT_3	minuto	19.735	28	Consumo de energia por eletrodomésticos
IOT_4	horária	9.359	12	Qualidade do ar
IOT_5	horária	35.052	10	Qualidade do ar
CLIMATICO_1	horária	57.240	12	Radiação global
CLIMATICO_2	diária	7.307	7	Evapotranspiração
CLIMATICO_3	mensal	492	19	Vazão do Rio Xingu

$$NRMSE = \frac{\sqrt{\sum_{t=0}^n (y(t) - \hat{y}(t))^2}}{y_{max} - y_{min}} \quad (4.1)$$

Para promover a transparência e a reprodutibilidade dos resultados, os códigos, conjuntos de dados e bancos de dados de resultados estão disponíveis em <https://github.com/PatriciaLucas/AutoML/tree/main>.

A implementação computacional foi realizada na linguagem de programação Python, utilizando bibliotecas da comunidade científica para análise de dados e aprendizado de máquina, como pandas, NumPy, scikit-learn, tigramite, statsmodels, emd, lightgbm, xgboost, pycaret, fedot, AUTOTSF, neuralforecast e darts. Os experimentos foram executados em duas configurações de hardware distintas: a primeira máquina contava com processador Intel i7, 16 GB de RAM e sem GPU dedicada, enquanto a segunda utilizou uma GPU T4 com 12 GB de RAM em ambiente de Google Colab.

#### 4.1 Comparação de métodos de seleção de características não causais e o PCMCI

Os algoritmos foram implementados utilizando uma interface MISO, onde múltiplas entradas indicam o número de variáveis na série temporal, ou simplesmente  $(X_1, X_2, X_3)$ . A saída é representada como  $X_1$ , a ser prevista. A série temporal multivariada é utilizada para a seleção de características. Com as defasagens selecionadas, um grafo é gerado e utilizado para estruturar os dados de entrada para o modelo de previsão *Random Forest*. O método *Grid Search* foi empregado para otimizar os hiperparâmetros dos modelos. Levando em consideração o custo computacional e o número de defasagens significativamente autocorrelacionadas na série temporal em estudo, foi empregada uma janela de defasagem de tamanho 10.

Os hiperparâmetros do PCMCI são:  $\alpha_{PC}$ ,  $\tau_{max}$  e o teste de independência condicional.  $\alpha_{PC}$  atua como um parâmetro de regularização. Utilizando níveis muito elevados, como  $\alpha_{PC} = 1$ , nenhum pai é removido. Isso significa que todas as variáveis são selecionadas como pais.  $\tau_{max}$  representa o atraso máximo usado na geração do grafo.  $\alpha_{PC} = 0.1$  foi utilizado para todas as variáveis. O teste de independência condicional selecionado foi o linear.

O método baseado em correlação (nomeado de Correlacional), foi implementado utilizando autocorrelação e correlação cruzada, selecionando defasagens com correlação significativamente diferente de 0 a um nível de confiança de 99%.

Um modelo substituto foi aplicado para mensurar a significância da correlação cruzada. Para uma compreensão mais detalhada desse método, consulte o exemplo da Figura ?? com  $X_1$  e  $X_2$ : primeiramente, calcule a correlação cruzada entre cada ponto de  $X_1$  e  $X_2$  ( $C_{1,2}$ ). Para avaliar se  $C_{1,2}$  é diferente de 0,  $X_1$  foi utilizado para gerar 1000 séries permutadas diferentes, preservando suas propriedades estatísticas individuais e removendo qualquer correlação cruzada entre elas. Em seguida, calcule a correlação cruzada de  $X_1$  com cada uma das séries permutadas e gere uma série temporal de distribuições de correlação ( $D$ ). Note que a média dessas distribuições deve ser 0. Finalmente, um teste estatístico foi realizado para determinar se  $C_{1,2}$  é significativamente diferente de 0 a um nível de confiança de 99%, utilizando  $D$ .

Como abordagem *wrapper*, o algoritmo genético (AG) foi implementado com os seguintes parâmetros: validação cruzada com K-fold e  $k = 5$ , tamanho da população = 80, probabilidade de crossover = 0.5, taxa de mutação = 0.2 e número de gerações = 100. Os indivíduos foram avaliados usando regressão linear.

O Lasso (TIBSHIRANI, 1996) foi usado como método embutido e foi implementado usando regressão linear com um  $\alpha$  constante multiplicando o termo  $L_1$  igual a 0.1. O grafo foi gerado selecionando defasagens cujos coeficientes eram maiores que 0.

Tanto o AG quanto o Lasso são avaliados utilizando um modelo diferente daquele utilizado para gerar o grafo. O objetivo principal é investigar se as características selecionadas por esses métodos dependentes de modelo podem ser aplicadas com sucesso a diferentes modelos.

O desempenho foi medido em termos de tempo de execução, número de variáveis selecionadas e o NRMSE médio calculado a partir dos modelos. Os dados foram divididos em duas partes iguais: a primeira para seleção de características e otimização de hiperparâmetros, e a segunda para treinar e testar o modelo em 3 segmentos diferentes da série temporal, com 20% dos dados reservados para teste. Os testes de Augmented Dickey-Fuller (ADF) (CHEUNG; LAI, 1995) e Kwiatkowski-Phillips-Schmidt-Shin (KPSS) (KWIATKOWSKI et al., 1992) foram empregados com um nível de confiança de 95% para determinar a estacionaridade. A diferenciação foi aplicada nos casos de comportamento não estacionário.

Além da avaliação de desempenho, foi realizada uma análise dos grafos gerados pelos métodos estudados. Para esse fim, três conjuntos de dados climáticos foram utilizados para criar grafos que devem corresponder às influências de defasagens de variáveis na evapotranspiração de referência ( $ET_o$ ).  $ET_o$  é uma variável bem conhecida e estabelecida na literatura, conforme (ALLEN; PEREIRA; RAES, 2006). De acordo com os autores, as variáveis que influenciam  $ET_o$  são temperaturas, radiação solar, umidade relativa e velocidade do vento. A série temporal de precipitação também foi incluída nos conjuntos de dados para determinar se os métodos de seleção criam ligações espúrias entre precipitação e  $ET_o$ . Como a janela de defasagem disponível é 10, serão avaliadas dez ligações.

#### 4.1.1 Resultados

Os resultados na Figura 5 e na Tabela 2 fornecem informações sobre o número de variáveis selecionadas pelos modelos, bem como o tempo de execução necessário para gerá-los.

O método PCMCI selecionou um conjunto menor de variáveis, exceto para o conjunto de dados IOT\_2, onde o Lasso apresentou um conjunto significativamente menor em comparação com todos os métodos. Observa-se que o método Correlacional gerou os

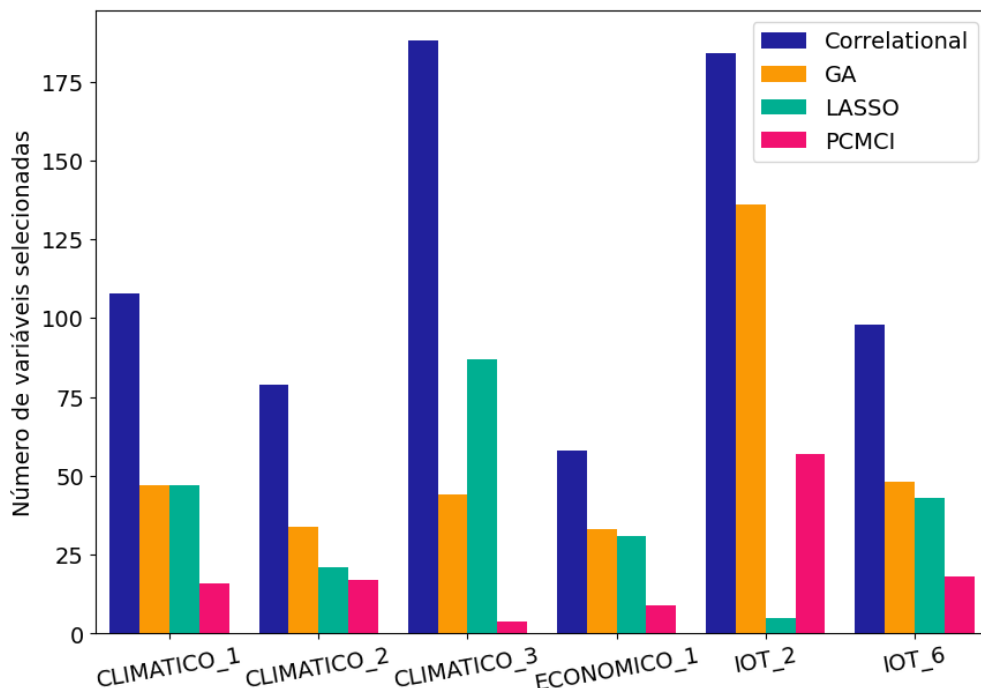


Figura 5 – Número de variáveis selecionadas pelos métodos PCMCI, Correlacional, AG e Lasso para todos os conjuntos de dados estudados.

Tabela 2 – Tempo de execução em segundos dos gráficos gerados pelos métodos PCMCI, Correlacional, GA e Lasso para todos os conjuntos de dados do estudo.

Datasets	PCMCI	Correlacional	AG	Lasso
ECONOMICO_2	9.89	1.53	53.95	12.46
IOT_2	408.85	6.84	1131.66	238.82
CLIMATICO_2	81.60	2.02	244.89	83.17
IOT_6	158.81	2.62	322.81	116.43
CLIMATICO_1	181.27	3.06	529.89	138.30
CLIMATICO_3	26.08	4.59	25.89	0.43

maiores conjuntos de variáveis para todas as bases de dados. O GA, embora tenha gerado um conjunto menor que o Lasso em XINGU, selecionou muito mais variáveis do que o Lasso e o PCMCI nas outras bases de dados.

Em termos de tempo de execução, o método Correlacional mostra tempos significativamente menores em comparação com os outros métodos. PCMCI e Lasso têm tempos de processamento semelhantes, com o Lasso superando significativamente nos conjuntos de dados IOT\_2 e CLIMATICO\_3. O GA apresenta os piores tempos de execução. Além disso, métodos dependentes de modelo como GA e LASSO demonstram custos computacionais significativamente reduzidos quando o conjunto de dados contém poucas amostras, como observado em XINGU.

Na Tabela 3, estão os resultados do NRMSE médio e desvio padrão para previsões de 1, 3 e 5 passos à frente. Esses resultados também são mostrados nos boxplots da Figura 6. Em geral, todos os métodos apresentaram diferença entre as médias dentro do desvio-



padrão, sugerindo que os métodos apresentam desempenhos comparáveis em diferentes horizontes de previsão, exceto para os conjuntos de dados IOT\_2 e CLIMATICO\_3 na previsão de 1 passo à frente. Em IOT\_2, o método Lasso exibiu um NRMSE médio mais alto em comparação com os outros métodos. Em CLIMATICO\_3, o PCMCI mostrou um desempenho melhor do que os outros.

Tabela 3 – NRMSE de previsão para os horizontes de previsão 1, 3 e 5 dos conjuntos de dados estudados.

Datasets	Horizonte	PCMCI	Correlational	GA	Lasso
ECONOMICO_2	1	0.11 ±0.01	0.11 ±0.01	0.13 ±0.02	0.12 ±0.01
	3	0.14 ±0.03	0.14 ±0.02	0.14 ±0.02	0.14 ±0.02
	7	0.14 ±0.02	0.14 ±0.02	0.14 ±0.02	0.13 ±0.02
IOT_2	1	0.05 ±0.01	0.05 ±0.01	0.06 ±0.01	0.13 ±0.05
	3	0.10 ±0.03	0.10 ±0.03	0.10 ±0.03	0.13 ±0.05
	7	0.14 ±0.08	0.15 ±0.11	0.13 ±0.06	0.14 ±0.07
IOT_6	1	0.05 ±0.01	0.05 ±0.01	0.05 ±0.01	0.05 ±0.01
	3	0.11 ±0.01	0.11 ±0.00	0.10 ±0.01	0.11 ±0.01
	7	0.16 ±0.01	0.16 ±0.01	0.15 ±0.01	0.16 ±0.01
CLIMATICO_1	1	0.02 ±0.01	0.02 ±0.01	0.02 ±0.01	0.02 ±0.01
	3	0.04 ±0.01	0.05 ±0.02	0.04 ±0.01	0.05 ±0.02
	7	0.08 ±0.01	0.09 ±0.00	0.09 ±0.01	0.09 ±0.01
CLIMATICO_2	1	0.18 ±0.02	0.18 ±0.02	0.18 ±0.02	0.20 ±0.02
	3	0.25 ±0.04	0.23 ±0.03	0.24 ±0.03	0.24 ±0.03
	7	0.24 ±0.03	0.24 ±0.02	0.25 ±0.02	0.26 ±0.02
CLIMATICO_3	1	0.14 ±0.03	0.22 ±0.07	0.27 ±0.15	0.24 ±0.11
	3	0.28 ±0.19	0.23 ±0.06	0.21 ±0.07	0.23 ±0.07
	7	0.27 ±0.17	0.32 ±0.25	0.39 ±0.38	0.24 ±0.13

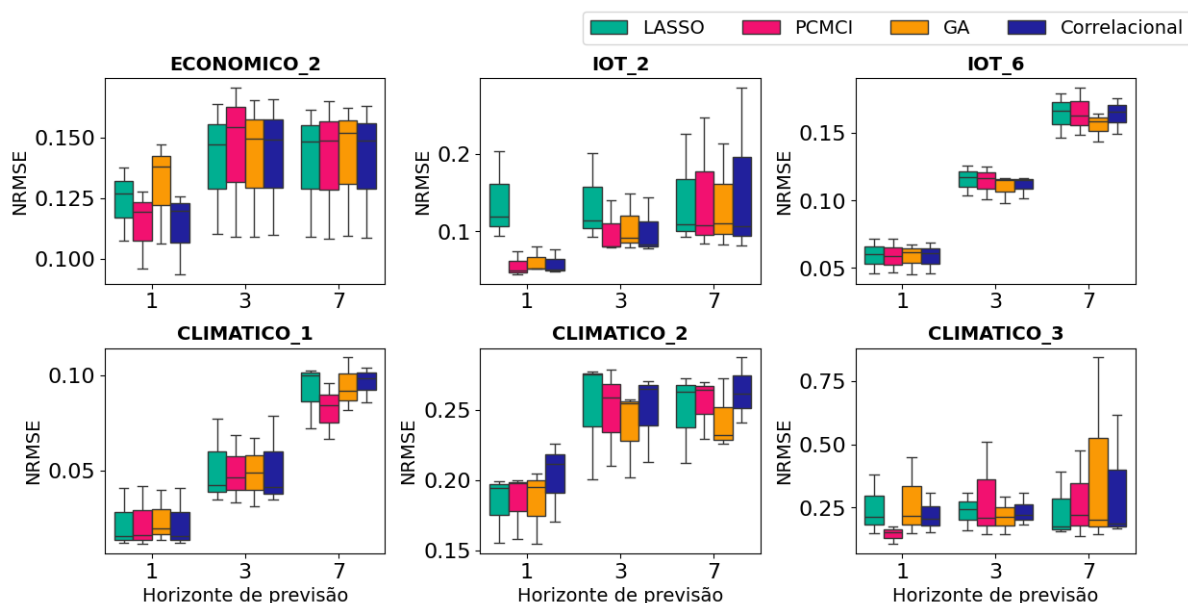


Figura 6 – NRMSE para horizontes de previsão de 1, 3 e 5 usando os métodos PCMCI, Correlational, GA e Lasso para as bases de dados analisadas.

Finalmente, a Tabela 4 apresenta o número de relações espúrias gerados pelos métodos. Os métodos Correlational e Lasso selecionaram quase todas as relações disponíveis

entre  $ETo$  e precipitação. O método GA selecionou apenas três relações em Jequitinhonha, mas nos outros locais, selecionou mais de 50%. Por outro lado, o método PCMCI selecionou apenas duas relações em Doce e nenhuma em Mucuri e Jequitinhonha.

Tabela 4 – Número de links espúrios gerados pelos métodos PCMCI, Correlacional, GA e Lasso para o conjunto de dados CLIMATICO\_2 em três localidades diferentes.

Datasets	PCMCI	Correlational	GA	Lasso
Mucuri	0	10	08	08
Doce	02	09	06	09
Jequitinhonha	0	09	03	09

#### 4.1.2 Discussão

Nos resultados do número de variáveis selecionadas pelos métodos, observou-se que o PCMCI é capaz de gerar conjuntos de características significativamente menores em comparação com os outros métodos testados. Também é evidente que as características selecionadas pelo PCMCI são relevantes, pois exibem um NRMSE estatisticamente similar aos outros modelos. Esse resultado demonstra a capacidade do PCMCI de gerar modelos de previsão mais parcimoniosos, conseqüentemente mais interpretáveis e menos suscetíveis ao overfitting.

Vale ressaltar que o Lasso gerou um conjunto de características muito reduzido no conjunto de dados HOME. No entanto, o NRMSE alcançado foi estatisticamente mais alto do que nos outros modelos. O oposto foi observado no conjunto de dados XINGU, onde, apesar de ter um número muito limitado de variáveis, o PCMCI conseguiu superar os outros modelos ou manteve uma média de NRMSE significativamente igual aos demais métodos.

Por fim, observou-se que um conjunto maior de características não resultou em modelos mais precisos, o que pode indicar a criação de relações espúrias entre os lags. Essas relações espúrias foram confirmadas ao incluir a variável de precipitação nos conjuntos de dados de evapotranspiração de referência. O PCMCI foi o único método que não selecionou links espúrios em dois dos três conjuntos de dados avaliados.

## 4.2 Estudo de ablação

Neste experimento avalia-se o impacto das camadas Decomposição (1) e Seleção Causal de Características (2) na estrutura do AutoDCE-TS. Para isso, conduziu-se o experimento com a comparação da precisão de previsão e do número de variáveis selecionadas.

Foram avaliadas três estruturas: (A) método proposto sem as camadas 1 e 2, (B) método proposto com a camada 2 e (C) método proposto com as camadas 1 e 2.

As configurações empregadas nos experimentos foram as seguintes:  $\alpha_{PC} = 0.1$ , teste de independência condicional linear selecionado, tamanho da população do AEMOH igual a 20, número de gerações do AEMOH igual a 10 e probabilidade de mutação após acasalamento de 0.3.

O experimento foi realizado nas bases de dados CLIMATICO\_2, ECONOMICO\_2, ENERGIA\_1 e IOT\_5. Cada estrutura foi avaliada em três janelas sem sobreposição, sendo 80% dos dados usados para treino e os demais para teste. A média do NRMSE das três janelas foi usada para medir o desempenho das estruturas em 5 horizontes de previsão. O número de variáveis foi calculado somando-se todos os lags de todas as variáveis usadas para treinar os modelos.

#### 4.2.1 Resultados e discussão

A Figura 7 apresenta as médias do NRMSE separadas por horizontes de previsão, enquanto a Figura 8 mostra o número de variáveis selecionado em cada estrutura.

Esses resultados indicam que a inclusão da camada 2 (observando A e B) melhora ou não altera o desempenho do método proposto em todas as bases de dados e horizontes de previsão. É importante notar que B atinge esse resultado utilizando um número significativamente reduzido de variáveis em comparação com A.

Em relação à inclusão da camada 1 (observando B e C), nota-se uma melhora no desempenho geral, tanto na redução do NRMSE quanto na diminuição da incerteza do método, especialmente em horizontes de previsão mais longos. Essa melhora é acompanhada por um aumento no número de variáveis, o que é explicado pela inserção das componentes da variável endógena. Ainda assim, o número de variáveis permanece significativamente inferior ao de A.

### 4.3 Comparação do Método AutoDCE-TS com outros Métodos de Previsão de Séries Temporais Multivariadas

Nesta avaliação, o AutoDCE-TS foi comparado a métodos AutoML e modelos de aprendizado profundo especializados em previsão de séries temporais. Os frameworks AutoML selecionados foram escolhidos com base em três critérios: especialização em tarefas de previsão de séries temporais, compatibilidade com a interface MISO empregada pelo AutoDCE-TS e disponibilidade código em python. Além disso, o método foi comparado aos modelos *ensemble* usados na camada de Seleção de Modelos. Todos os conjuntos de dados descritos na Tabela 1 foram utilizados.

O modelo AutoDCE-TS foi configurado da seguinte forma:  $\alpha_{PC} = 0.1$ , teste de independência condicional linear, tamanho da população do AEMOH definido em

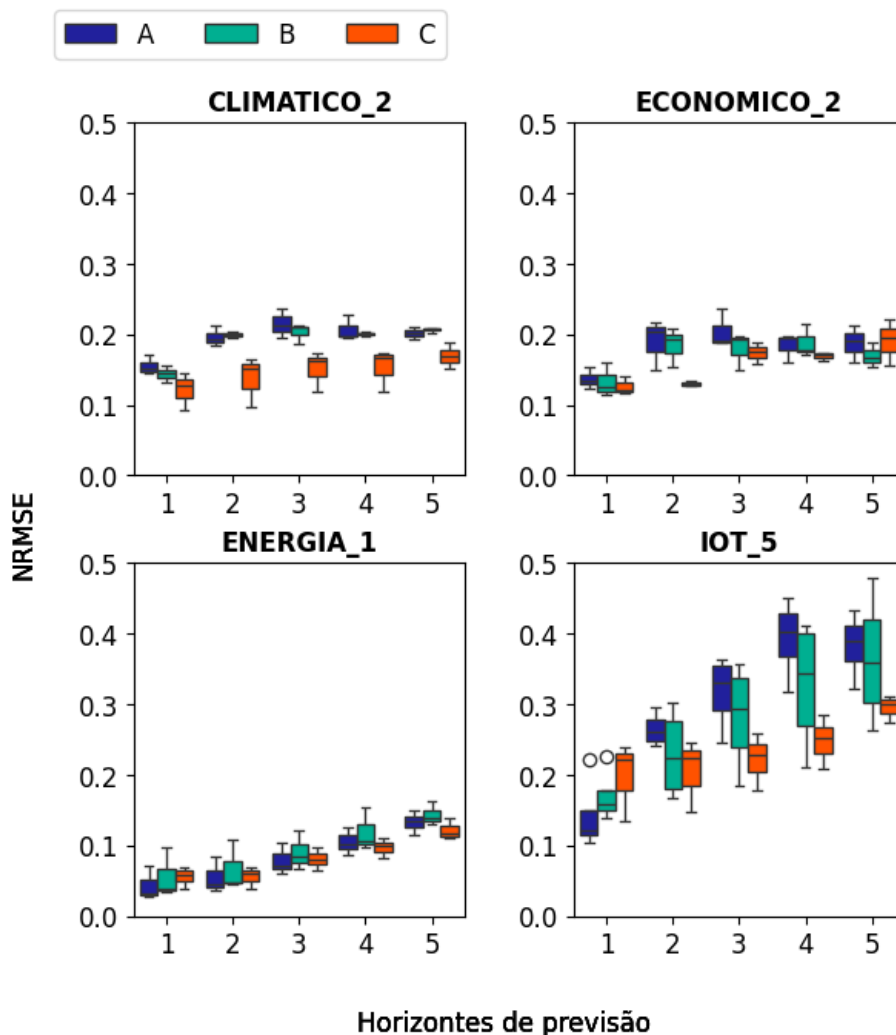


Figura 7 – Média do NRMSE para os horizontes de previsão 1 a 5 para as estruturas avaliadas.

20, número de gerações do AEMOH definido em 10 e probabilidade de mutação após o cruzamento definida em 0.3. O parâmetro  $\alpha_{PC}$  foi selecionado empiricamente a partir do espaço de busca [0.05, 0.1, 0.2, 0.3, 0.4, 0.5] usando o Critério de Informação de Akaike (AIC). De modo geral, essa configuração mostrou-se suficientemente robusta para o conjunto de experimentos realizados, caracterizando-se como uma escolha inicial apropriada. Como a camada de Decomposição adiciona variáveis e, conseqüentemente, aumenta o custo computacional do método, optou-se por expandir a avaliação do AutoDCE-TS sem utilizar essa camada. Para essa configuração, o método é denominado AutoCE-TS.

Os métodos AutoML selecionados foram PYCARET, FEDOT e AUTOTS. Cada um desses métodos possui pacotes implementados em Python disponíveis em (ALI, 2020; NIKITIN et al., 2022b; WANG et al., 2022b), respectivamente. Os hiperparâmetros padrão desses métodos foram mantidos, exceto para o número de iterações do algoritmo de otimização. No FEDOT e no AUTOTS, que utilizam algoritmos evolucionários, o número de indivíduos foi definido em 20 e o número de gerações em 10. Para o PYCARET, que

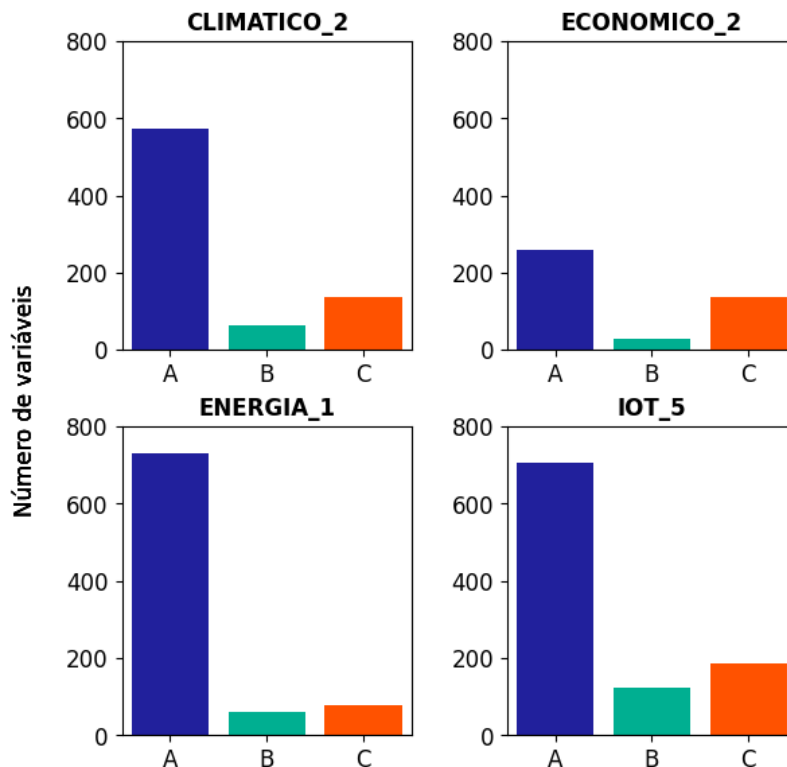


Figura 8 – Número de variáveis utilizadas nas estruturas A, B e C.

emprega Random Search, o número de avaliações foi definido em 50. Em todos os métodos, foram utilizados modelos multivariados.

Os modelos de aprendizado profundo selecionados foram Long Short-Term Memory (LSTM) (HOCHREITER; SCHMIDHUBER, 1997), Temporal Convolutional Network (TCN) (BAI; KOLTER; KOLTUN, 2018), Neural Basis Expansion Analysis for Interpretable Time Series Forecasting (N-BEATS) (ORESHKIN et al., 2020), Temporal Fusion Transformer (TFT) (LIM et al., 2021) e Inverted Transformers (iTfM) (LIU et al., 2024). Os hiperparâmetros foram escolhidos por Random Search dentro dos espaços de busca descritos na Tabela 5. Todos os demais hiperparâmetros foram mantidos nos valores padrão.

Os modelos *ensemble* RF, LGBM e XGBoost foram treinados utilizando os hiperparâmetros padrão de suas respectivas bibliotecas (PEDREGOSA et al., 2011; GUYON et al., 2017; CHEN; GUESTRIN, 2016b).

O experimento foi conduzido para avaliar os métodos em 10 janelas de tamanho  $0.3 * |D|$  com sobreposição de 10% ao longo da série temporal multivariada de tamanho  $|D|$ . As janelas foram divididas em conjuntos de treino/validação e teste. Os dados de teste consistiram em 30 amostras, que foram comparadas às saídas previstas de 1 a 30 passos à frente produzidas pelos métodos. O tamanho da janela de defasagem usada como entrada para todos os modelos foi definido em 20, exceto para PYCARET, FEDOT e

AUTOTS, onde essa escolha é automatizada. O NRMSE médio das 10 janelas foi usado para medir o desempenho.

A análise estatística foi conduzida para cada domínio dos conjuntos de dados utilizando o pacote Python AutoRank (HERBOLD, 2020), empregando o NRMSE dos métodos ao longo dos três horizontes de previsão. A significância estatística das diferenças no NRMSE médio entre todos os métodos foi avaliada utilizando o teste não-paramétrico de Friedman, já que a normalidade não pôde ser assumida para nenhuma das amostras. Um nível de significância de 0.05 foi utilizado para os testes. O teste de Friedman avalia a hipótese nula de que não há diferenças significativas entre os valores medianos das populações. Em todos os casos, a hipótese nula foi rejeitada, indicando diferenças estatisticamente significativas entre os valores medianos das populações. O teste pós-hoc de Nemenyi foi utilizado para identificar diferenças significativas.

Para avaliar o custo computacional, os tempos de execução (em minutos) dos métodos AutoML AutoDCE-TS, AutoCE-TS, PyCaret, FEDOT e AutoTS foram medidos.

AutoDCE-TS, PYCARET, FEDOT, AUTOTS, RF, LGBM e XGBoost foram executados em uma máquina com CPU e 16 GB de RAM, com limite de tempo de execução de 5 horas por janela. Os modelos de aprendizado profundo foram executados em uma máquina equipada com GPU T4 e 51 GB de RAM.

Tabela 5 – Espaço de busca de hiperparâmetros para os métodos TCN, LSTM, NBEATS, TFT e iTFM.

Hiperparâmetros	LSTM	TCN	N-BEATS	TFT / iTFM
Tamanho do kernel	-	2, 5, 11	-	-
Número de camadas conv.	-	1, 3, 5	-	-
Número de filtros da camada conv.	-	16, 32, 64	-	-
Dilatação	-	2, 3, 4	-	-
N. de pilhas	-	-	16, 24, 30	-
N. de camadas totalmente conv.	-	-	2, 3, 4	-
N. de neurônios na camada totalmente conv.	-	-	64, 128, 256	-
N. de unidades	32, 64, 96, 128	-	-	-
N. de camadas LSTM	1, 2, ..., 5	-	-	-
N. de unidades de embeddings e encoders	-	-	-	16, 32, 64, 92, 144
N. de cabeças de atenção	-	-	-	2, 3, 4
Percentual de dropout	0.2, 0.5, 0.8	0.2, 0.5, 0.8	0.2, 0.5, 0.8	0.2, 0.5, 0.8

#### 4.3.0.1 Resultados e Discussão

Os resultados são apresentados em mapas de calor gerados a partir do NRMSE médio das 10 janelas experimentais, considerando horizontes de previsão de 1, 15 e 30. Os mapas de calor nas Figuras 9, 10, 11 e 12 correspondem aos conjuntos de dados dos domínios ECONOMICO, ENERGIA, IOT e CLIMATICO, respectivamente. Espaços em branco nos mapas de calor do FEDOT indicam erros durante a execução do método,

enquanto aqueles nos mapas do AUTOTS representam tempos de execução superiores ao limite de 5 horas definido para os métodos AutoML nos experimentos.

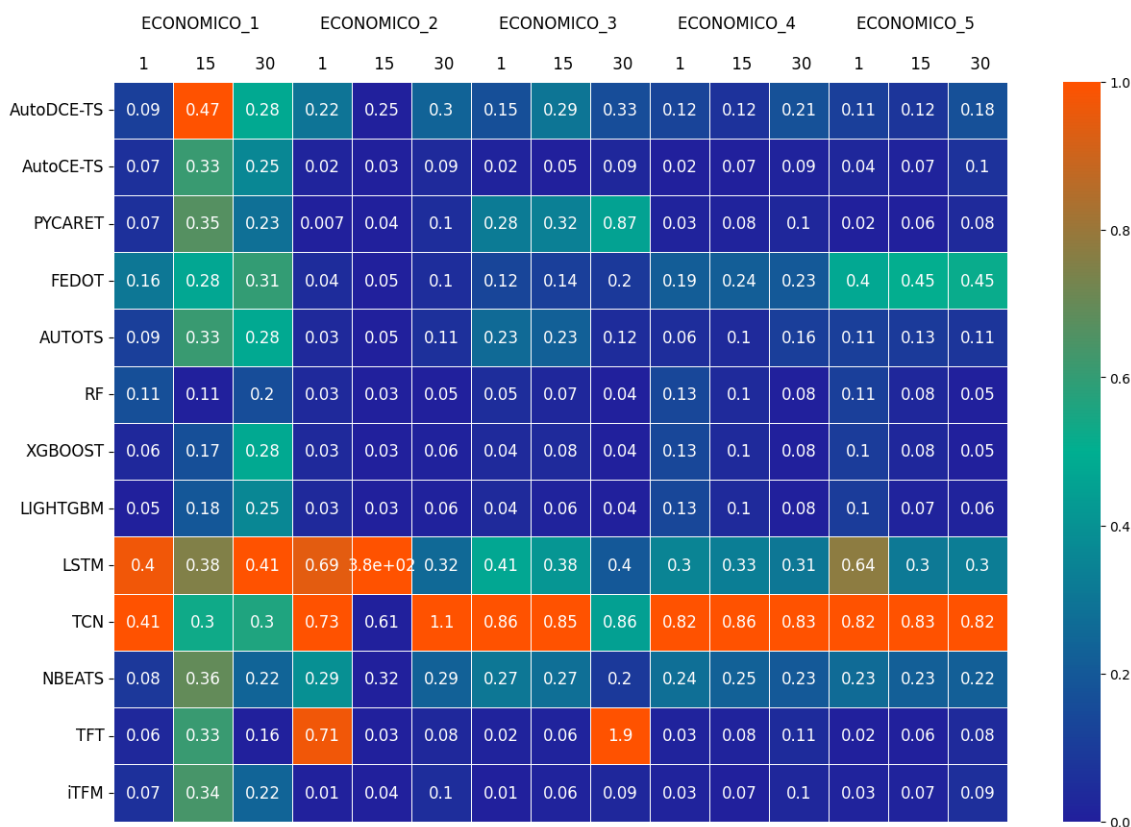


Figura 9 – Mapa de calor de NRMSE para horizontes de 1, 15 e 30 dos modelos AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT, AUTOTS, RF, XGBOOST, LIGHTGBM, LSTM, TCN, N-BEATS, TFT e iTFM nos conjuntos de dados ECONOMICO\_1, ECONOMICO\_2, ECONOMICO\_3, ECONOMICO\_4 e ECONOMICO\_5.

Os resultados do teste pós-hoc de Nemenyi são apresentados no diagrama de diferenças críticas (Figura 13). Nesse diagrama, as diferenças entre populações são consideradas significativas se a diferença de classificação média exceder a distância crítica (CD) descrita na Tabela 6 e indicada pela barra preta. A régua também mostra a classificação média dos métodos, onde uma classificação média mais alta corresponde a uma colocação mais alta.

Os métodos FEDOT e AUTOTS foram excluídos da análise estatística em (b), (c) e (d) devido à ausência de resultados.

Os tempos de execução em minutos para os métodos AutoML AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT e AUTOTS são apresentados na Tabela 7. Os espaços em branco indicam que o método não foi executado em mais de um conjunto de dados do domínio devido a erro de execução ou por exceder o limite de tempo de 5 horas.

Os resultados obtidos por AutoDCE-TS e AutoCE-TS revelam uma diferença

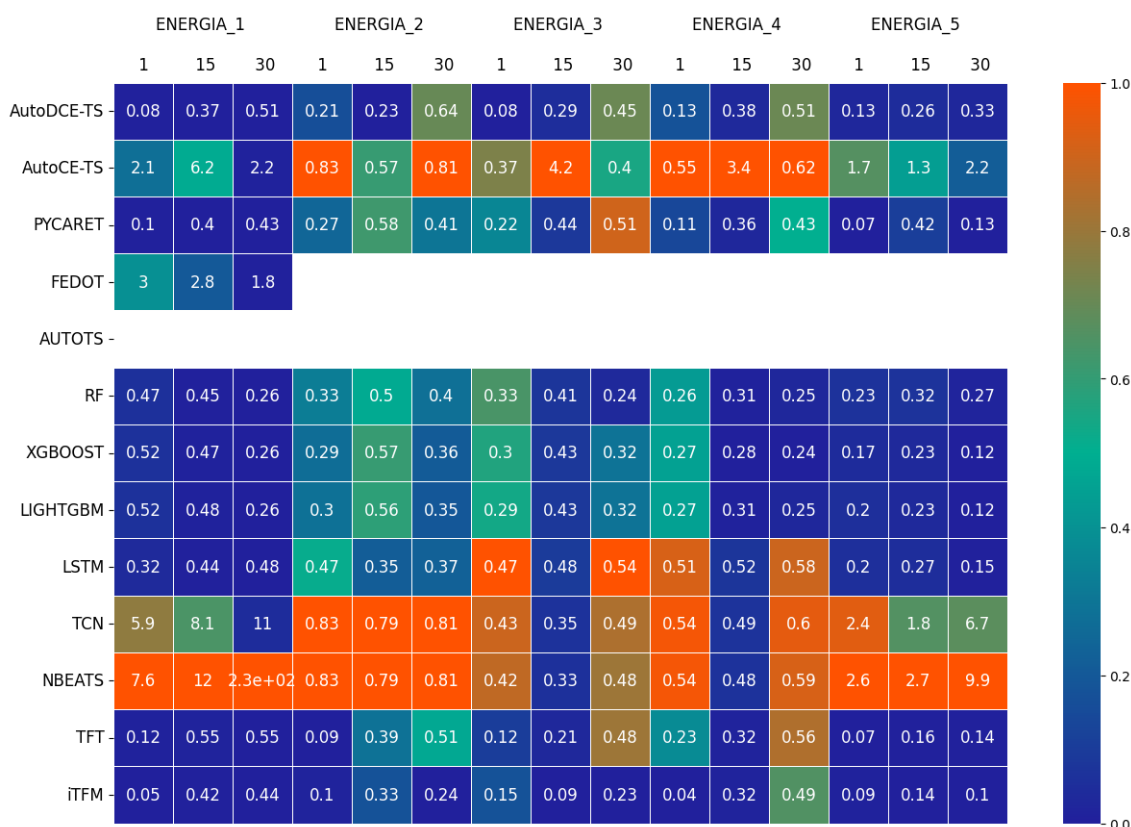


Figura 10 – Mapa de calor de NRMSE para horizontes de 1, 15 e 30 dos modelos AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT, AUTOTS, RF, XGBOOST, LIGHTGBM, LSTM, TCN, N-BEATS, TFT e iTFM nos conjuntos de dados ENERGIA\_1, ENERGIA\_2, ENERGIA\_3, ENERGIA\_4 e ENERGIA\_5.

Tabela 6 – Resultados do teste não paramétrico de Friedman e diferenças críticas (CD) do teste pós-hoc de Nemenyi. O teste de Friedman avalia a hipótese nula de que não há diferenças significativas entre os valores medianos das populações.

	CD	p-valor	Resultado
ECONOMICO	4.303	0.00000	H0 é rejeitada
ENERGIA	3.498	0.00000	H0 é rejeitada
IOT	3.498	0.00053	H0 é rejeitada
CLIMATICO	4.515	0.00039	H0 é rejeitada

Tabela 7 – Tempo médio de execução em minutos dos métodos AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT e AUTOTS por domínio de conjunto de dados.

	ECONOMICO	ENERGIA	IOT	CLIMATICO
AutoDCE-TS	8.54	46.27	25.27	11.24
AutoCE-TS	5.17	35.37	13.06	8.34
PYCARET	2.46	3.44	3.15	2.72
FEDOT	24.12	-	-	-
AUTOTS	124.72	-	-	59.60



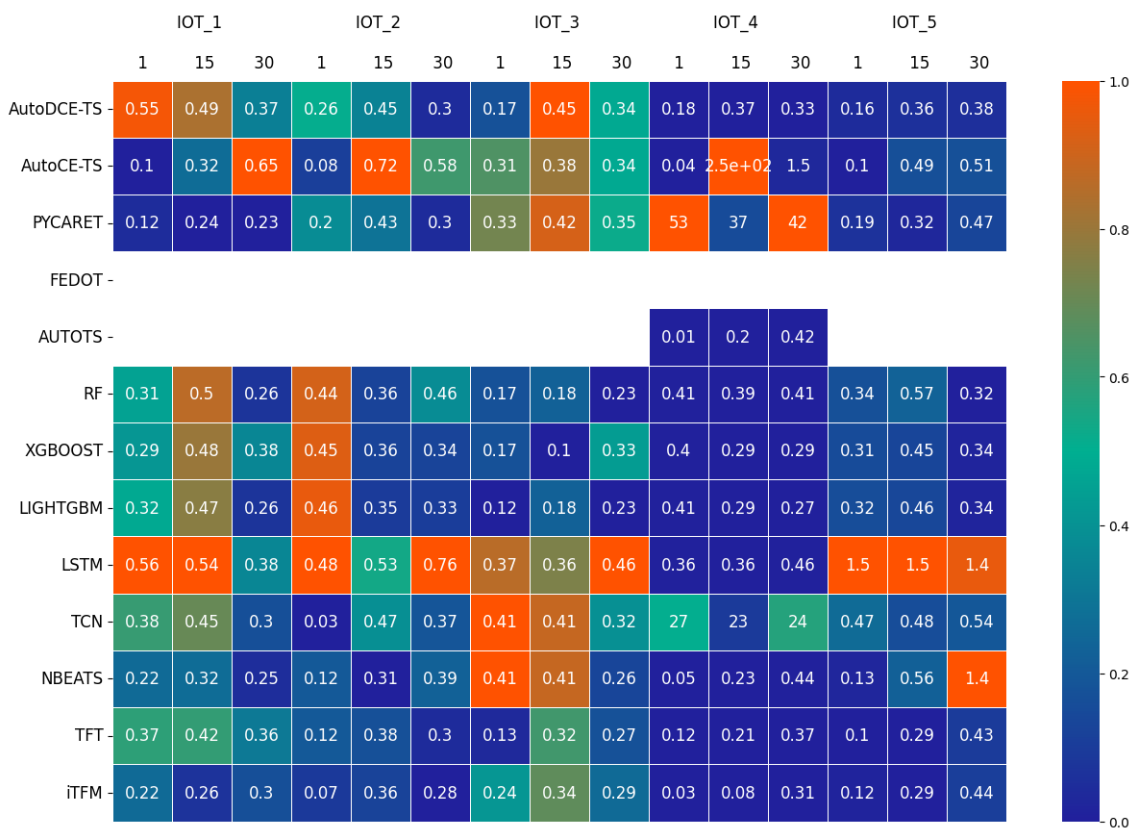


Figura 11 – Mapa de calor de NRMSE para horizontes de 1, 15 e 30 dos modelos AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT, AUTOTS, RF, XGBOOST, LIGHTGBM, LSTM, TCN, N-BEATS, TFT e iTFM nos conjuntos de dados IOT\_1, IOT\_2, IOT\_3, IOT\_4 e IOT\_5.

estatisticamente significativa na média do NRMSE para conjuntos de dados dos domínios ECONOMICO, ENERGIA e CLIMATICO, com o AutoDCE-TS alcançando um NRMSE médio menor do que o AutoCE-TS, indicando desempenho superior. No entanto, para o domínio IOT, o NRMSE médio não apresenta diferença estatisticamente significativa. Esse resultado destaca a importância de desenvolver uma estratégia que automatize a decisão de incluir ou excluir a camada de Decomposição. Essa decisão pode não apenas melhorar o desempenho preditivo, mas também gerar um pipeline mais parcimonioso, já que o AutoCE-TS reduz o número de variáveis processadas. Isso pode ser observado na redução do tempo de execução do AutoCE-TS em comparação com o AutoDCE-TS.

De modo geral, o método proposto, AutoDCE-TS ou AutoCE-TS, apresentou desempenho competitivo em comparação com os outros métodos, alcançando a melhor classificação nos conjuntos de dados dos domínios ECONOMICO e ENERGIA, bem como o terceiro lugar em CLIMATICO. No domínio IOT, embora tenha ficado em oitavo lugar, seu NRMSE médio é estatisticamente igual a todos os outros métodos.

Outro ponto interessante é que os métodos RF, XGBOOST e LIGHTGBM também estão entre os mais bem classificados, apoiando sua seleção para inclusão nos pipelines do

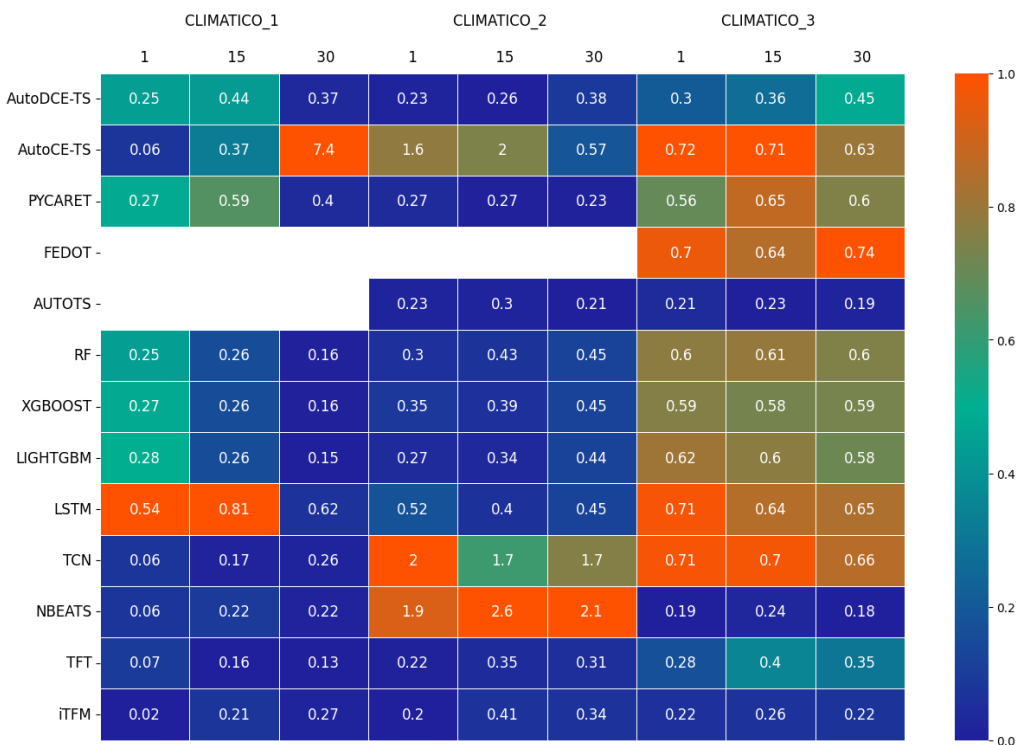


Figura 12 – Mapa de calor de NRMSE para horizontes de 1, 15 e 30 dos modelos AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT, AUTOTS, RF, XGBOOST, LIGHTGBM, LSTM, TCN, N-BEATS, TFT e iTFM nos conjuntos de dados CLIMATICO\_1, CLIMATICO\_2 e CLIMATICO\_3.

método proposto. Além disso, o tratamento individual das variáveis que compõem a série temporal e a seleção causal de defasagens não reduzem a precisão das previsões.

O método proposto obteve médias de NRMSE estatisticamente diferentes e com melhor classificação do que o FEDOT em ECONOMICO, e mesmo em conjuntos de dados não incluídos no teste estatístico, o AutoDCE-TS apresentou NRMSE visivelmente menor. Quando comparado ao AUTOTS em ECONOMICO, o método proposto alcançou melhor classificação, mas com média de NRMSE estatisticamente igual. Nos demais conjuntos de dados, o NRMSE apresentado pelo AUTOTS é visivelmente próximo aos resultados dos modelos do grupo mais bem classificado, que inclui o AutoDCE-TS. O método proposto também obteve melhor classificação, mas com média de NRMSE estatisticamente igual ao PYCARET em ECONOMICO, ENERGIA e CLIMATICO. Em IOT, o PYCARET foi melhor classificado, mas também com média de NRMSE estatisticamente igual ao método proposto.

Em relação ao tempo de execução, observa-se que o método proposto foi superado apenas pelo PYCARET. O AUTOTS apresentou tempos de execução significativamente mais longos em comparação com os outros métodos, o que impediu sua execução em conjuntos de dados com alta dimensionalidade. Devido à variedade de configurações disponíveis nos AutoMLs PYCARET e AUTOTS, este estudo optou por utilizar os modelos

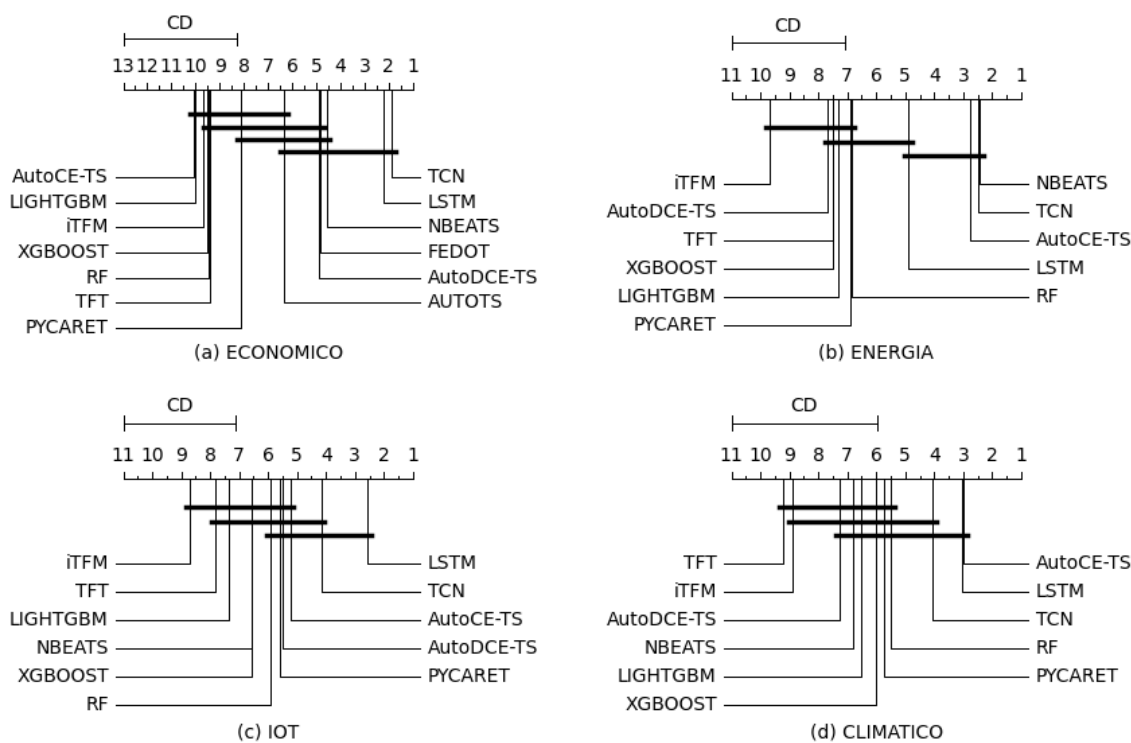


Figura 13 – Diagrama de diferenças críticas dos métodos comparados neste estudo para cada domínio dos conjuntos de dados, considerando os horizontes de previsão de 1, 15 e 30 conjuntamente. CD é a distância crítica, representada pelas barras pretas e indicam que todos os métodos dentro desse limite possuem média de NRMSE estatisticamente iguais. A régua acima do diagrama mostra a classificação média dos métodos, onde uma classificação média mais alta corresponde a uma colocação mais alta. Os modelos avaliados incluem: AutoDCE-TS, AutoCE-TS, PYCARET, FEDOT, AUTOTS, Random Forest (RF), XGBOOST, LIGHTGBM, LSTM, TCN, N-BEATS, TFT e iTFM.

padrão definidos pelos respectivos autores para problemas multivariados. É importante destacar que o tempo de execução dos AutoMLs é influenciado pela estratégia de otimização adotada, bem como pelo número e pela complexidade dos modelos implementados.

O PYCARET, que apresentou os menores tempos médios de execução, não utiliza modelos de aprendizado profundo e realiza uma pré-seleção de modelos usando seus hiperparâmetros padrão. A otimização de hiperparâmetros ocorre apenas com intervenção do usuário e é aplicada ao melhor modelo identificado durante a pré-seleção, usando Random Search. Por outro lado, os métodos AUTOTS, FEDOT e AutoDCE-TS adotam estratégias evolutivas que incorporam tanto modelos quanto hiperparâmetros no processo de otimização. Além disso, o AUTOTS implementa algoritmos de aprendizado profundo, contribuindo para seu tempo de execução significativamente maior.

Examinando os resultados dos modelos de aprendizado profundo e do método proposto, este último obteve uma diferença estatisticamente significativa na média do NRMSE e uma classificação superior em comparação ao N-BEATS e ao TCN no conjunto

de dados ENERGIA. Nos demais casos, o método proposto apresentou médias de NRMSE estatisticamente equivalentes em comparação com iTFM, TFT, LSTM, TCN e N-BEATS. Vale destacar que os métodos de aprendizado profundo e o método proposto foram executados em máquinas diferentes, pois os modelos de aprendizado profundo apresentaram tempos de execução muito lentos e insuficiência de memória quando executados na mesma máquina que rodou o método proposto. Esses resultados mostram que o método proposto, além de ser competitivo em termos de desempenho preditivo quando comparado a modelos de aprendizado profundo, oferece vantagens adicionais, como pipelines automáticos mais interpretáveis e menor custo computacional. A afirmação baseia-se em dois aspectos: (1) modelos de aprendizado profundo geralmente requerem hardware de alto desempenho, como GPUs, para atingir tempos de treinamento aceitáveis, resultando em maior sobrecarga computacional; e (2) esses modelos frequentemente envolvem um grande número de parâmetros, tornando sua interpretação consideravelmente mais complexa. Em contraste, o método proposto oferece um pipeline ao mesmo tempo automático e mais interpretável, com menores requisitos de hardware.

Para ilustrar o potencial interpretativo do AutoDCE-TS, a Figura 14 apresenta o grafo causal e a importância atribuída a cada defasagem das variáveis, conforme estimado pelo modelo selecionado (neste caso, Random Forest), utilizando o conjunto de dados CLIMATICO\_2. A figura mostra não apenas quais variáveis e defasagens foram usadas, mas também o grau de importância de cada uma na estimação da variável endógena ETO (representada pelo nó azul), conforme indicado pela escala de cores.

#### 4.4 Conclusão

Neste capítulo foram apresentados 3 experimentos para avaliar as estratégias usadas pelo método proposto AutoDCE-TS. Abaixo segue um resumo de cada experimento e suas principais conclusões:

1. No experimento 1, é feito um estudo explorando o método de descoberta causal PCMCI para seleção de características de séries temporais multivariadas, comparando-o com outros métodos de seleção de características. Os resultados indicaram que o PCMCI foi capaz de gerar um conjunto menor de características em comparação com outros métodos sem comprometer a precisão do modelo em diferentes horizontes de predição. Além disso, o número de relações espúrias criadas pela PCMCI é muito pequeno ou inexistente. Ou seja, as relações causais estabelecidas pelo método estão mais próximas do mecanismo que gerou os dados de evapotranspiração de referência. Isso demonstra a capacidade competitiva do PCMCI em comparação com abordagens bem conhecidas no campo de seleção de recursos em séries temporais, como métodos

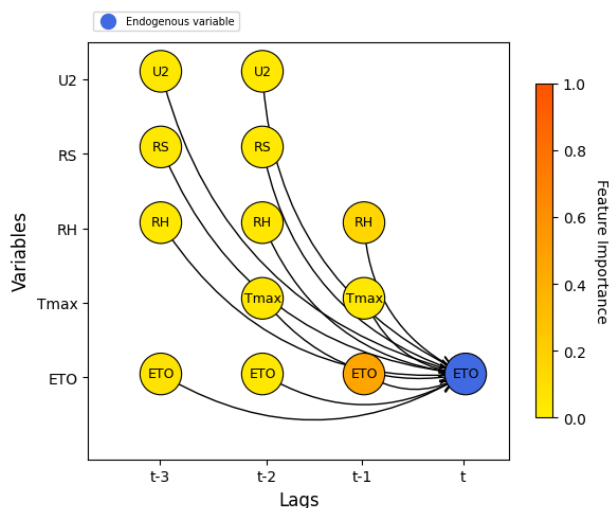


Figura 14 – Grafo causal gerado pelo AutoDCE-TS com base no modelo Random Forest aplicado ao conjunto de dados CLIMATICO\_2. Os nós representam as variáveis com suas respectivas defasagens, enquanto as setas indicam relações causais direcionadas para a variável endógena ETO (em azul). A coloração dos nós reflete a importância relativa de cada variável defasada na previsão do ETO, de acordo com a escala mostrada à direita. As variáveis envolvidas são: evapotranspiração de referência (ETO), temperatura máxima do ar (Tmax), umidade relativa (RH), radiação solar (RS) e velocidade do vento (U2).

baseados em correlação. O uso do PCMCI permite a geração de modelos mais enxutos e compreensíveis em pipelines de previsão.

2. No experimento 2, é avaliado o impacto da seleção causal de características e da EMD na metodologia AutoDCE-TS. Concluiu-se que o método causal reduziu a complexidade dos modelos implementados no AutoDCE-TS sem implicar em redução da precisão de previsão final. A inclusão da EMD apresentou ganho de desempenho e na redução da incerteza do método.
3. No Experimento 3, compara-se o desempenho do AutoDCE-TS com outros métodos de previsão de séries temporais multivariadas. Os resultados mostram que o AutoDCE-TS é competitivo nesse cenário, oferecendo um pipeline totalmente automatizado que integra seleção de variáveis, otimização de modelos e ajuste de hiperparâmetros. O método também fornece elementos de interpretabilidade, como grafos causais e árvores de regressão. Ainda que ensembles de árvores não sejam totalmente interpretáveis, eles disponibilizam mecanismos relevantes, como medidas de importância dos atributos, que auxiliam na compreensão do modelo. Além disso, o AutoDCE-TS gera pipelines mais eficientes em termos de uso de recursos computacionais, característica especialmente importante em cenários de séries temporais que variam ao longo do tempo e requerem retreinamentos periódicos. Por fim, os resultados indicam que a decisão sobre incluir a decomposição da série deve ser automatizada, dado que essa estratégia tende a melhorar o desempenho na maioria dos casos.

---

O próximo capítulo busca, por meio do uso de agentes baseados em LLMs, automatizar a escolha de parâmetros que o AutoDCE-TS não foi capaz de definir automaticamente, como o uso de decomposição e o tamanho da janela de defasagens. Além disso, esses agentes são empregados para explicar o pipeline gerado por meio de relatórios em linguagem natural.

## 5 AutoDCE-TS AGÊNTICO PARA PREVISÃO DE SÉRIES TEMPORAIS

Neste capítulo é apresentado o AutoDCE-TS-Agentic, um sistema multiagente desenvolvido para apoiar a automação e explicação dos pipelines gerados pelo AutoDCE-TS. O sistema é composto por três agentes:

1. **Executor** – responsável por automatizar a execução do pipeline de previsão, incluindo o pré-processamento dos dados, a configuração e chamada do AutoDCE-TS, a visualização dos resultados e a extração de informações sobre o processo de previsão realizado.
2. **Corretor** – encarregado de detectar erros que possam ocorrer durante as tarefas executadas pelo agente Executor.
3. **Explicador** – responsável pela geração de relatórios parciais e do relatório final, além de esclarecer dúvidas do usuário sobre todas as etapas do processo.

Todos os agentes foram implementados utilizando o modelo *anthropic/claude-4-opus*<sup>1</sup>, configurado com temperatura igual a 0 e limite máximo de 1024 tokens. Os códigos estão disponíveis em <<https://github.com/PatriciaLucas/AutoML-Agent>>.

O sistema é estruturado em quatro módulos: execução do pipeline, correção, resumo e explicação, conforme ilustrado na Figura 15. Ao receber uma solicitação do usuário, o AutoDCE-TS-Agentic implanta automaticamente todo o fluxo de trabalho, coordenando e executando os agentes de forma integrada para alcançar o objetivo definido. Esse objetivo pode corresponder tanto à execução de uma nova tarefa de previsão quanto à geração de uma explicação sobre um pipeline de previsão já realizado. O agente Explicador é responsável por interpretar a solicitação do usuário, determinar se se trata de uma nova previsão ou de uma explicação, e ativar os módulos apropriados para cada caso. As subseções a seguir detalham o funcionamento de cada módulo.

### 5.1 Execução do pipeline

O módulo de execução do pipeline é o primeiro a ser ativado após o usuário solicitar uma tarefa de previsão, devendo este fornecer a descrição da tarefa e o conjunto de dados a ser utilizado. Para operar esse módulo, o agente Executor é configurado com um *prompt* específico e equipado com ferramentas responsáveis pelo pré-processamento

<sup>1</sup> <https://www.anthropic.com/claude/opus>

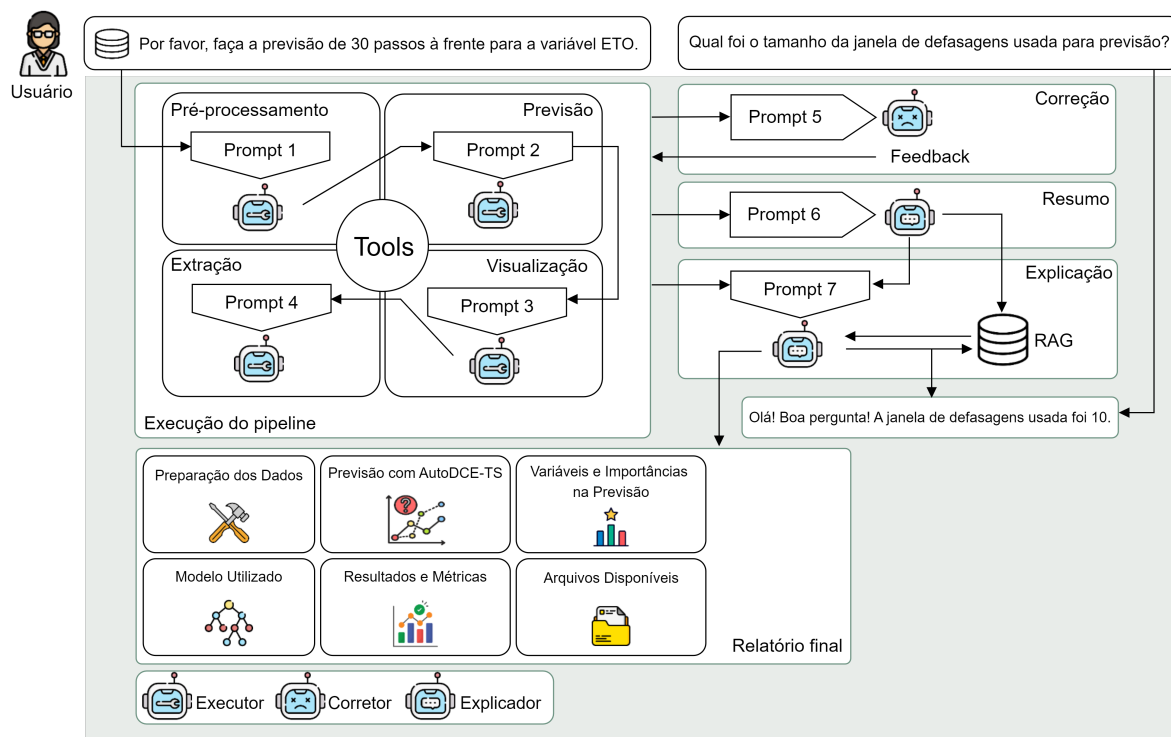


Figura 15 – Diagrama ilustrativo do AutoDCE-TS-Agentic.

dos dados, pela geração das previsões, pela visualização dos resultados e pela extração de informações do AutoDCE-TS.

Os *prompts* seguem o formato ReAct (YAO et al., 2022), o que aprimora a capacidade de raciocínio do agente e garante maior transparência na tomada de decisão em cada etapa do pipeline, por meio de ciclos iterativos de pensamento–ação–observação. Cada *prompt* é estruturado no formato padrão do ReAct, seguido da explicação da tarefa que deve ser executada. Portanto, nas descrições abaixo, cada *prompt* corresponde exclusivamente à explicação da tarefa da etapa específica do pipeline.

**Pré-processamento:** Nesta etapa, o agente é orientado, por meio do *Prompt 1* (Quadro 1), a verificar se o conjunto de dados fornecido pelo usuário contém valores faltantes. Caso existam, o agente deve selecionar automaticamente uma ferramenta apropriada para realizar a imputação.

**Previsão:** Nesta etapa, o agente é orientado, por meio do *Prompt 2* (Quadro 2), a realizar a previsão conforme a solicitação do usuário. Essa solicitação deve incluir, além do conjunto de dados, a variável a ser prevista e o horizonte de previsão desejado. Com base nessas informações, o agente configura o AutoDCE-TS, definindo o tamanho janela de defasagens a ser utilizada e decidindo se a série temporal endógena será decomposta ou não. Foi definido o limite inferior igual a 5 e superior igual a 50 para o tamanho janela de defasagens. Para isso, além do AutoDCE-TS, o agente



pode recorrer a ferramentas auxiliares para analisar as séries temporais de entrada e embasar suas decisões.

**Visualização dos resultados:** Nesta etapa, o agente é orientado, por meio do *Prompt 3* (Quadro 3), a criar um gráfico comparativo das previsões realizadas pelo AutoDCE-TS e dos valores reais.

**Extração de informações:** Nesta etapa, o agente é orientado, por meio do *Prompt 4* (Quadro 4), a extrair informações do processo de previsão gerado pelo AutoDCE-TS. Essas informações incluem os grafos causais gerados para seleção das variáveis e defasagens, modelos e hiperparâmetros de cada variável e a importância de cada uma delas para o modelo de previsão da variável endógena.

#### Prompt 1

Você é um assistente especializado em pré-processamento de séries temporais, com foco na imputação de valores faltantes.

Regras:

1. Verifique todas as colunas do dataframe para detectar valores faltantes.
2. Caso existam valores faltantes, utilize a tool mais apropriada para imputação.

Se atente a observação a seguir: *{feedback do Corretor.}*

#### Prompt 2

Você é um assistente especializado em previsão de séries temporais. Sua única tarefa é gerar previsões para o problema descrito pelo usuário: mensagem do usuário.

Regras:

1. Defina automaticamente, sempre com uma breve justificativa: - O número máximo de defasagens ('max\_lags'), escolhendo entre 5 e 50 para equilibrar precisão e custo computacional. - Se a decomposição ('decomposition') deve ser usada ou não.
2. Utilize a tool 'testar\_estacionariedade' caso seja necessário decidir sobre a estacionariedade.

Se atente a observação a seguir: *{feedback do Corretor.}*

#### Prompt 3

Você é um assistente especializado em visualização de séries temporais.

Regras:

1. Foque apenas na tarefa de visualização de dados.
2. O dataset possui duas colunas referentes aos valores reais e previstos.
2. Gere um gráfico comparando os valores reais e previstos.
3. Retorne a figura em Base64 pronta para uso, sem explicações ou texto adicional.

Se atente a observação a seguir: *{feedback do Corretor.}*

**Prompt 4**

Você é um assistente de análise de modelos.

Regras:

1. Extraia informações geradas pelo AutoML modelo com a ferramenta `extrair_informacao_automl`.
2. Responda apenas no formato de um dicionário Python válido.
3. Não inclua explicações, comentários ou texto fora do dicionário.

A tool retornará as seguintes informações técnicas dentro de um dicionário:

- O dicionário `'modelo_dict'` contém uma chave para cada variável. Cada variável foi prevista por um modelo específico, com seus hiperparâmetros associados.
- Os grafos causais gerados para as variáveis estão no dicionário `'grafo'`. Cada chave corresponde a uma variável, e o valor é um dataframe onde True indica que a variável (coluna) foi selecionada no lag correspondente (linha).

Sua tarefa:

Com base nessas informações, gere uma resposta estruturada no formato: `{{ "modelos": {{ "variável1": "Texto curto descrevendo o modelo e seus hiperparâmetros.", "variável2": "Texto curto descrevendo o modelo e seus hiperparâmetros." }}, "variaveis_selecionadas": "variável1": "Texto curto descrevendo as variáveis selecionadas e seus lags.", "variável2": "Texto curto descrevendo as variáveis selecionadas e seus lags." } } }`

Objetivo: Gerar um dicionário compacto e informativo, sem texto extra fora da estrutura solicitada.

Se atente a observação a seguir: feedback do Corretor.

As ferramentas disponíveis para o agente Executor são descritas a seguir. As ferramentas responsáveis pela imputação de dados foram definidas a partir de um levantamento das principais técnicas de imputação de valores ausentes em séries temporais, conforme destacado em (RIBEIRO; CASTRO, 2022; DOLATABADI et al., 2024). As descrições das ferramentas também são disponibilizadas ao agente, que decide qual delas utilizar em cada etapa.

**Interpolação linear:** aplica uma função linear entre pontos adjacentes para estimar os valores ausentes, assumindo uma transição uniforme entre eles. É indicada quando há uma relação aproximadamente linear entre os pontos vizinhos e as ausências são pontuais, sem longas sequências de dados faltantes.

**Interpolação spline:** utiliza funções polinomiais suaves para aproximar os dados e calcular valores ausentes, resultando em uma curva mais contínua que a interpolação linear. É recomendada para séries com comportamento não linear suave, em que se deseja preservar a forma e a suavidade da série temporal.

**Média:** preenche os valores ausentes utilizando a média global ou em janelas da própria série temporal. Essa técnica é mais adequada para séries aproximadamente estacionárias, ou seja, com média estável ao longo do tempo.

**Vizinho mais próximo:** substitui o valor ausente pelo valor de um ponto temporal

adjacente. É apropriada quando há baixa variação entre observações consecutivas e a proximidade temporal é um bom indicativo do valor faltante.

**AutoDCE-TS:** ferramenta baseada no método *Automated Machine Learning Based on Decomposition, Causality, and Evolutionary Multitask Optimization for Time Series Forecasting* (AutoDCE-TS). O AutoDCE-TS é um sistema de múltiplas entradas e múltiplas saídas (MISO) para previsão de múltiplos passos à frente, estruturado em um pipeline fixo de quatro camadas que automatiza as etapas de extração e seleção de características, seleção e geração de modelos e previsão.

**Gráfico:** apresenta, em gráfico, os valores previstos da série temporal endógena, possibilitando a análise comparativa com os dados reais.

**Extração de informações do AutoDCE-TS:** extrai os grafos causais, modelo e hiperparâmetros de todas as variáveis do problema de previsão. Para a série temporal endógena também é extraída a importância relativa de cada variável retornada pelo estimador.

**Auxiliares:** são ferramentas que podem ajudar o agente a tomar decisões. A ferramenta Teste de estacionariedade, por exemplo, avalia se a série temporal é estacionária com base nos testes estatísticos ADF e KPSS. Essa ferramenta retorna que a série é estacionária apenas se ambos os testes confirmarem essa propriedade. Além disso, o agente pode utilizar funções que exibem informações estatísticas sobre os dados, detectam valores faltantes, analisam os nomes das variáveis, entre outras.

## 5.2 Correção

O módulo de correção é ativado após cada etapa do pipeline, utilizando o agente Corretor, orientado pelo *Prompt 5* (Quadro 5). Dessa forma, cada etapa conduzida pelo agente Executor é analisada, e, caso ocorra algum erro, o Executor recebe um *feedback* incorporado ao *prompt* específico daquela etapa, permitindo a reexecução da tarefa. Esse módulo é essencial para a automação do tratamento de erros.

## 5.3 Resumo

O módulo de resumo é ativado após a aprovação de cada fase do pipeline na etapa de correção, utilizando o agente Explicador, orientado pelo *Prompt 6* (Quadro 6). Dessa forma, cada etapa conduzida pelo agente Executor é sintetizada e armazenada, de modo que o agente Explicador tenha à disposição uma base de dados resumida sobre o que foi realizado em cada etapa do pipeline. Os resumos são importantes, pois os LLMs possuem um limite de contexto, isto é, uma restrição na quantidade de texto que podem processar

**Prompt 5**

Você é um corretor de erros de um agente que utiliza as seguintes tools:  $\{lista\ de\ ferramentas\}$ . Durante a etapa nome da etapa, o agente executou as seguintes ações:  $\{log\ das\ ações\ do\ Executor\}$ , mas ocorreu um erro na última ação, interrompendo a execução.

Sua tarefa é:

1. Identificar qual foi a última ação realizada antes do erro.
2. Analisar o formato da chamada feita a essa tool e verificar se está de acordo com o formato esperado.
3. Explicar brevemente ao agente o formato da chamada para a tool.

Retorne apenas um objeto JSON válido, sem blocos de código, sem texto explicativo, e sem markdown. O formato deve ser exatamente um desses dois:

```
{ "avaliacao": "não", "feedback": "Descreva aqui o que o agente fez de errado e como corrigir a chamada da tool." }
```

Não adicione nada antes ou depois do JSON.

simultaneamente. Ao armazenar apenas os resumos, o agente Explicador consegue acessar essa base de dados na etapa seguinte (Explicação) sem a necessidade de reprocessar textos extensos.

**Prompt 6**

Você é o Agente Resumidor. Receberá dois blocos abaixo: - etapas: linhas alternadas no padrão 'THOUGHT: ...' e 'ACTION: tool(args)'

OBJETIVO Escreva um relatório em pt-BR exatamente no formato:

Nome do etapa realizada em Português: título curto em pt-BR.

Descrição da Ação: copie/sintetize fielmente o THOUGHT correspondente, sem inventar ou 'nenhum'.

Ferramenta Escolhida: nome da tool indicada em ACTION ou 'nenhuma', parâmetros indicados em Action Input ou 'nenhum'.

Resultado: resuma o output associado a esta etapa, sem inventar.

REGRAS

- Mantenha a ordem cronológica das etapas.
- Não invente ferramentas, dados ou resultados.
- Seja conciso e técnico; evite floreios.
- Se não houver THOUGHT ou ACTION para alguma etapa, preencha com '—' no campo faltante.
- Se não houver nada para resumir, responda apenas: Sem dados para resumir.

DADOS:  $\{ações\ do\ Executor\}$

**5.4 Explicação**

Em caso de solicitação de uma tarefa de previsão, o módulo de explicação é o último a ser ativado. O agente Explicador, orientado pelo *Prompt 7* (Quadro 7), utiliza a base de dados dos resumos e todas as informações retornadas pelo AutoDCE-TS para gerar um relatório completo sobre o que foi realizado, informando, por exemplo sobre

a imputação de valores, parâmetros escolhidos pelo agente, as variáveis selecionadas, os modelos escolhidos e seus respectivos hiperparâmetros.

Por outro lado, se a solicitação do usuário for uma explicação adicional sobre um pipeline de previsão já executado, esse módulo será o único a ser ativado, permitindo que o usuário interaja com o agente para obter informações sobre o processo de previsão. Para isso, toda a base de dados dos resumos, relatório final e outras informações extraídas do AutoML são armazenadas em um banco de dados vetorial, utilizando a técnica de *Retrieval-Augmented Generation* (RAG). Assim, durante o diálogo, o agente Explicador recupera informações por meio da busca semântica para responder adequadamente ao usuário.

A possibilidade de solicitar informações adicionais é essencial porque relatórios gerados por agentes nem sempre contemplam todos os detalhes necessários, isto é, não são completamente determinísticos, mesmo com o uso de prompts bem definidos. Além disso, em problemas multivariados, a grande quantidade de variáveis e resultados pode dificultar a inclusão de todas as informações no relatório final, aumentando o risco de omissões e confundindo o agente Explicador.

#### Prompt 7

Você receberá as seguintes informações técnicas:

- Etapa de pré-processamento: {resumo 1}. - Etapa de previsão: {resumo 2}. - Listas de variáveis/defasagens e suas importâncias utilizadas para prever a variável {nome da variável endógena}: {defasagens} - O dicionário {modelo.dict} possui uma chave para cada variável. Cada variável foi prevista por um modelo e seus hiperparâmetros. - As previsões feita pelo modelo: {previsoes} - As métricas calculadas com as previsões e valores reais: {metricas}.

Sua tarefa:

- Gerar uma explicação em tom amigável. - Explique o que foi feito em cada etapa de pré-processamento e previsão de forma conectada (não apenas em lista). - Informe ao usuário que o AutoML usado na previsão é o AutoDCE-TS. - Se a decomposição EMD foi aplicada, explique de maneira simples o que isso significa, em quantas IMFs foi decomposta a série e como os componentes (IMFs) foram usados. - Informe ao usuário que ele terá acesso ao arquivo modelo.pickle. Nesse arquivo ele poderá acessar o modelo escolhido para previsão de cada variável, seus hiperparâmetros acessando modelo.dict.variables['nome da variável']. As variáveis selecionadas podem ser acessadas em modelo.G\_list['nome da variável']. - Liste todas as variáveis e defasagens utilizadas na previsão da variável {nome da variável endógena}, bem como suas respectivas importâncias para o modelo. Ao invés de dizer "variável 1 (valor)", diga variável no tempo t-1: valor. Descarte a variável cuja importancia é -1. - Liste o modelo usado para previsão da variável {nome da variável endógena} e seus hiperparâmetros. - Informe as métricas calculadas. - Explique ao usuário que ele terá acesso a duas imagens: (grafico.jpg) gráfico de comparação dos valores reais e previstos e (grafo.jpg) grafo causal com as variáveis selecionadas e suas respectivas importâncias para o modelo. - Informe ao usuário as previsões feitas pelo modelo como uma lista. - Use um texto fluido, como se estivesse conversando diretamente com o usuário, sem apenas repetir os dados recebidos.

## 5.5 Estudo de caso sobre a base de dados CLIMATIC\_2

Neste estudo de caso, apresentamos como o sistema multiagente AutoDCE-TS-Agentic realiza a previsão da evapotranspiração de referência (ETO), utilizando base de dados CLIMATIC\_2. O objetivo é demonstrar o processo de decisão dos agentes, desde a preparação dos dados até a geração das explicações finais. Além disso, avalia-se de que forma as configurações adotadas pelo AutoDCE-TS-Agentic influenciam os resultados obtidos, em comparação com o sistema AutoDCE-TS. A seguir são apresentadas as descrições de solicitação do usuário, os relatórios parciais de cada etapa de execução do pipeline e o relatório final.

### Solicitação do usuário para tarefa de previsão:

Faça a previsão de 5 passos à frente para a coluna ETO. A base de dados contém dados climáticos diários. A coluna ETO representa a evapotranspiração de referência diária (mm).

### Resumo da execução da etapa Pré-processamento:

**Nome da etapa realizada:** Análise inicial do dataframe

**Descrição da Ação:** Necessidade de analisar o dataframe para verificar valores ausentes e aplicar métodos de imputação apropriados, caso necessário.

**Ferramenta Escolhida:** `python_repl_ast`, parâmetros: `df.info()`

**Resultado:** Identificou-se que a coluna Tmax possui apenas 1980 valores não nulos de um total de 2000 entradas, indicando 20 valores ausentes. Todas as demais colunas possuem dados completos.

**Nome da etapa realizada:** Contagem de valores ausentes

**Descrição da Ação:** Verificação da quantidade de valores ausentes em Tmax para auxiliar na escolha do melhor método de imputação.

**Ferramenta Escolhida:** `python_repl_ast`, parâmetros: `df['Tmax'].isnull().sum()`

**Resultado:** Confirmou-se a existência de 20 valores ausentes na coluna Tmax.

**Nome da etapa realizada:** Análise do padrão de valores ausentes

**Descrição da Ação:** Necessidade de entender o padrão dos valores ausentes em Tmax, verificando a localização dos índices ausentes e se eles ocorrem de forma consecutiva ou dispersa.

**Ferramenta Escolhida:** `python_repl_ast`, parâmetros: código para identificar índices de valores ausentes e verificar sua distribuição.

**Resultado:** Os valores ausentes encontram-se dispersos ao longo do dataset, sem ocorrência de seqüências consecutivas.

**Nome da etapa realizada:** Análise estatística da coluna Tmax

**Descrição da Ação:** Uma vez confirmada a dispersão dos valores ausentes, torna-se necessário examinar estatísticas básicas da variável Tmax para fundamentar a escolha do método de imputação, analisando desvio padrão, coeficiente de variação e o comportamento dos dados ao redor dos pontos ausentes.

**Ferramenta Escolhida:** `python_repl_ast`, parâmetros: código para examinar estatísticas descritivas, variabilidade e pontos adjacentes

**Resultado:** A coluna `Tmax` apresenta variabilidade moderada (coeficiente de variação  $\approx 13,77\%$ ), com transições suaves entre valores adjacentes.

**Nome da etapa realizada:** Aplicação da imputação linear

**Descrição da Ação:** Com base na análise: 1) `Tmax` possui 20 valores ausentes (1% do total), distribuídos de forma dispersa; 2) Os dados apresentam variabilidade moderada; 3) Há transições suaves entre valores vizinhos aos pontos ausentes; 4) Séries de temperatura tipicamente exibem mudanças graduais.

Assim, a interpolação linear é o método mais apropriado, pois preserva a tendência local e é adequada para valores ausentes dispersos em séries temporais de temperatura.

**Ferramenta Escolhida:** `impute_values_with_linear_method`, parâmetros: `Tmax`

**Resultado:** Imputação concluída.

## Resumo da execução da etapa Previsão:

**Nome da etapa realizada:** Análise inicial da estrutura dos dados

**Descrição da Ação:** Examinar a estrutura do dataframe e verificar se há valores ausentes na coluna `ETO`.

**Ferramenta Escolhida:** `python_repl_ast`, parâmetros: `df.head()`

**Resultado:** Visualização das primeiras linhas do dataframe, contendo colunas de dados climáticos (`Rs`, `u2`, `Tmax`, `Tmin`, `RH`, `pr`, `ETO`).

**Nome da etapa realizada:** Verificação de dados faltantes e dimensões.

**Descrição da Ação:** Verificar o tamanho do dataframe, a existência de valores ausentes em `ETO` e o tipo de índice temporal.

**Ferramenta Escolhida:** `python_repl_ast`, parâmetros:

```
print(f"Shape: {df.shape} \n Missing values in ETO: {df['ETO'].isna().sum()} \n\n Data type of index: {type(df.index)}")
```

**Resultado:** Shape: (2000, 7), sem valores ausentes na coluna `ETO`.

**Nome da etapa realizada:** Teste de estacionariedade

**Descrição da Ação:** Testar se a série `ETO` é estacionária, a fim de decidir se a decomposição deve ser aplicada.

**Ferramenta Escolhida:** `testar_estacionariedade`, parâmetros: `ETO`

**Resultado:** A série `ETO` foi identificada como estacionária.

**Nome da etapa realizada:** Análise de autocorrelação

**Descrição da Ação:** Analisar a autocorrelação (ACF) e autocorrelação parcial (PACF) para determinar o número adequado de defasagens (*lags*).

**Ferramenta Escolhida:** `python_repl_ast`, parâmetros: código para cálculo de ACF e PACF com 50 lags.

**Resultado:** Foram identificados 31 lags significativos no ACF, enquanto o último lag significativo no PACF ocorre entre 6 e 7, com correlações mais fortes nos primeiros lags.

**Nome da etapa realizada:** Previsão com AutoML

**Descrição da Ação:** Com base na análise de autocorrelação (PACF significativo até os lags 6–7 e ACF com 31 lags relevantes), definiu-se `max_lags = 15` para equilibrar modelagem temporal e eficiência computacional. Como a série é estacionária, definiu-se `decomposition = false`.

**Ferramenta Escolhida:** `automl`, parâmetros:

```
{target: ETO, step_ahead: 5, max_lags: 15, decomposition: false}
```

**Resultado:** Valores reais de ETO: [4.4, 4.2, 4.9, 4.8, 4.3].

Valores previstos de ETO: [4.615, 3.858, 4.312, 4.003, 3.815].

### Correção de erro durante a execução da etapa Previsão:

**Erro:** 'ETO\n'.

**Avaliação:** não.

**Feedback:** A tool 'testar\_estacionariedade' espera receber apenas o nome da coluna como input direto, mas o agente passou 'ETO' com uma quebra de linha no final. O formato correto do input deve ser apenas o nome da coluna sem espaços ou quebras de linha extras. O agente deve chamar a tool com o input: ETO (sem quebra de linha ou espaços adicionais).

### Resumo da execução da etapa Geração do gráfico de previsões:

**Nome da etapa realizada:** Geração do gráfico comparativo.

**Descrição da Ação:** Foram identificadas as colunas do dataframe: "real ETO" (valores reais) e "previsto ETO" (valores previstos). É necessário gerar um gráfico comparando esses valores utilizando a ferramenta `plot_real_vs_pred`.

**Ferramenta Escolhida:** `plot_real_vs_pred`, parâmetros: nenhum

**Resultado:** O gráfico de real versus previsto foi gerado.

### Resumo da execução da etapa Extração de informações:

**Nome da etapa realizada:** Extração de informações do AutoML.

**Descrição da Ação:** É necessário extrair as informações do modelo AutoML utilizando a ferramenta `extrair_informacao_automl` para obter detalhes sobre os modelos e as variáveis selecionadas.

**Ferramenta Escolhida:** `extrair_informacao_automl`, parâmetros: nenhum

**Resultado:** Foram extraídos 7 modelos XGBoost (Rs, u2, Tmax, Tmin, RH, pr, ETO), todos configurados com 75 estimadores, `min_samples_leaf = 2` e `max_features = 0.2`. Cada modelo possui variáveis selecionadas específicas com diferentes lags temporais.

### Relatório final:



Olá! Vou explicar todo o processo de análise e previsão que foi realizado com seus dados de evapotranspiração (ETO).

### **Análise e Preparação dos Dados**

Começamos analisando o conjunto de dados para garantir que estivesse pronto para a modelagem. Durante essa análise inicial, descobriu-se que havia alguns valores faltantes na coluna de temperatura máxima (Tmax) — especificamente, 20 valores ausentes em um total de 2000 registros, o que representa apenas 1% dos dados.

Para compreender melhor o tratamento desses valores ausentes, investigou-se sua localização e observou-se que estavam dispersos ao longo do conjunto de dados, não sendo consecutivos. Também foram analisadas as características estatísticas da temperatura máxima, constatando-se que ela apresentava variabilidade moderada (aproximadamente 13,77%) com transições suaves entre valores adjacentes.

Com base nessas características — valores ausentes dispersos, transições suaves e comportamento gradual natural da temperatura — decidiu-se aplicar a interpolação linear para preencher esses valores faltantes. Esse método é apropriado para dados de temperatura, pois preserva tendências locais e mantém a continuidade natural dos dados climáticos.

### **Processo de Previsão com AutoDCE-TS**

Para realizar as previsões, utilizou-se o **AutoDCE-TS** (Automated Machine Learning Based on Decomposition, Causality, and Evolutionary Multitask Optimization for Time Series Forecasting), um sistema avançado de aprendizado de máquina automatizado especializado em séries temporais. Antes da aplicação do modelo, foi realizada uma análise detalhada da série ETO. Primeiramente, testou-se a estacionariedade da série (isto é, se suas propriedades estatísticas permanecem constantes ao longo do tempo), e confirmou-se que a série era estacionária. Em seguida, analisou-se a autocorrelação dos dados para determinar o número de valores passados com influência sobre valores futuros. Identificou-se correlação significativa em até 31 defasagens temporais, com as mais fortes nos primeiros 6–7 períodos.

Com base nessa análise, configurou-se o AutoML para usar até 15 defasagens temporais, buscando um equilíbrio entre capturar os padrões relevantes e manter a eficiência computacional. Como a série era estacionária, não foi necessário aplicar decomposição.

### **Variáveis e Importâncias na Previsão**

O modelo selecionou automaticamente as variáveis e defasagens mais relevantes para prever a ETO. A seguir, apresentam-se as variáveis utilizadas e suas respectivas importâncias:

#### **Variáveis mais importantes (presentes em múltiplos horizontes)**

- Temperatura máxima no tempo  $t - 1$ : 0,263
- Radiação solar no tempo  $t - 1$ : 0,199
- Umidade relativa no tempo  $t - 3$ : 0,044

#### **Outras variáveis relevantes**

- Temperatura máxima no tempo  $t - 3$ : 0,073
- Temperatura máxima no tempo  $t - 2$ : 0,067
- Radiação solar no tempo  $t - 3$ : 0,062

- ETO no tempo  $t - 2$ : 0,061
- Precipitação no tempo  $t - 1$ : 0,047
- Umidade relativa no tempo  $t - 1$ : 0,044
- ETO no tempo  $t - 3$ : 0,043

Esses resultados indicam que a temperatura máxima e a radiação solar são fatores altamente influentes na previsão da evapotranspiração, o que é coerente com o comportamento físico do processo.

### Modelo Utilizado

Para a previsão da variável ETO, o sistema selecionou o modelo **XGBoost** com os seguintes hiperparâmetros otimizados:

- Número de estimadores: 75
- Mínimo de amostras por folha: 2
- Máximo de *features*: 0.2 (20% das variáveis em cada árvore)

### Resultados e Métricas

As previsões para os próximos 5 períodos foram:

Previsões = [4.615, 3.858, 4.312, 4.003, 3.815]

O desempenho do modelo foi avaliado pelas seguintes métricas:

- **RMSE**: 0.525
- **NRMSE**: 0.750 (75%)
- **MAE**: 0.485
- **MAPE**: 10.58%

Esses resultados indicam que o modelo apresenta precisão razoável, com erro médio percentual de 10,58%.

### Arquivos Disponíveis

Você possui acesso aos seguintes arquivos:

1. **modelo.pickle**: Contém o modelo completo. É possível acessar:
  - O modelo treinado para cada variável: `modelo.dict_variables['nome da variável']`
  - As variáveis selecionadas: `modelo.G_list['nome da variável']`
2. **grafico.jpg**: Gráfico comparando valores reais e valores previstos.
3. **grafo.jpg**: Grafo causal exibindo relações entre variáveis e suas importâncias.

Esperamos que esta análise tenha sido útil. Os resultados mostram que o modelo capturou de maneira satisfatória os padrões temporais e as interações entre variáveis climáticas para prever a evapotranspiração.

**Solicitação do usuário para explicações adicionais:**

Qual o tamanho da janela de defasagens usada no AutoML?

### Resposta do AutoDCE-TS-Agentic:

O tamanho da janela de defasagens (`max_lags`) usada no AutoML foi 15. Essa escolha foi baseada na análise de autocorrelação, onde o PACF mostrou valores significativos até lag 6-7 e o ACF mostrou 31 lags significativos, sendo 15 um valor que balanceia a captura de padrões temporais com eficiência computacional.

### Solicitação do usuário para explicações adicionais:

Qual o modelo e hiperparâmetros utilizados para prever  $T_{max}$ ?

### Resposta do AutoDCE-TS-Agentic:

Para prever  $T_{max}$  foi utilizado o modelo XGBoost com os seguintes hiperparâmetros: 75 estimadores, `min_samples_leaf=2`, `max_features=0.2`. As variáveis selecionadas para o modelo foram:  $R_s$  (lag 3),  $T_{max}$  (lags 1, 2, 3, 11),  $RH$  (lags 1, 3) e  $pr$  (lag 3).

#### 5.5.1 Comparação AutoDCE-TS versus AutoDCE-TS-Agentic

Os experimentos foram conduzidos seguindo a mesma metodologia descrita na Seção 4.3, e os resultados obtidos foram comparados com aqueles apresentados na Subseção 4.3.0.1. O sistema AutoDCE-TS-Agentic realizou a configuração de acordo com cada janela de dados, optando pelo uso da decomposição sempre que a série temporal endógena apresentava comportamento não estacionário.

A Tabela 8 apresenta os valores médios de NRMSE obtidos nas 10 janelas experimentais. Observou-se que o AutoDCE-TS-Agentic utilizou decomposição em três janelas e definiu tamanhos de defasagem entre 10 e 15. Os resultados indicam que a versão agêntica alcançou desempenho semelhante aos melhores resultados do AutoDCE-TS, apresentando, contudo, uma configuração mais parcimoniosa ao adotar janelas de defasagem menores e empregar a decomposição apenas quando identificou séries não estacionárias.

## 5.6 Conclusão

O objetivo deste capítulo foi apresentar o AutoDCE-TS-Agentic, um sistema multiagente capaz de automatizar a imputação de dados faltantes, configurar o AutoDCE-TS por meio da análise da série temporal endógena e gerar relatórios explicativos sobre as etapas do pipeline de previsão.

Tabela 8 – NRMSE da previsão para os horizontes de 1, 15 e 30 dos modelos AutoDCE-TS, AutoCE-TS e AutoDCE-TS-Agentic.

	<b>CLIMATICO_2</b>		
	1	15	30
<b>AutoDCE-TS</b>	0.23	0.26	0.38
<b>AutoCE-TS</b>	0.72	0.71	0.63
<b>AutoDCE-TS-Agentic</b>	0.28	0.25	0.40

Os resultados obtidos, tanto nos relatórios gerados quanto na comparação entre modelos, mostram que o AutoDCE-TS-Agentic cumpre de forma eficaz seu papel de complementar a automação oferecida pelo AutoDCE-TS. O sistema realiza o tratamento dos dados de entrada e define configurações adequadas a partir de uma análise estatística da série temporal, de maneira consistente com o que seria feito por um especialista humano. Isso é demonstrado, por exemplo, por meio do uso e de ferramentas como testes de estacionariedade e análise de correlações para decidir sobre os hiperparâmetros do AutoDCE-TS.

Além disso, o relatório final demonstra sua capacidade de produzir textos claros, estruturados e informativos, descrevendo as decisões tomadas ao longo do processo de previsão e ainda explicando as saídas do AutoDCE-TS. O agente também mantém memórias sobre esse processo, permitindo responder a perguntas adicionais que extrapolam o conteúdo presente no relatório final.

## 6 CONCLUSÕES

Nesta tese foi investigada a automação de pipelines para previsão de séries temporais multivariadas, com foco em estratégias capazes de manter baixo custo computacional, transparência e desempenho competitivo. Para isso, foi conduzida uma revisão abrangente dos principais conceitos relacionados ao AutoML, bem como das estratégias de extração de características, seleção de variáveis e modelos. Essa revisão forneceu uma visão panorâmica do estado da arte e permitiu identificar abordagens adequadas ao tratamento de dados multivariados com eficiência computacional e interpretabilidade.

A metodologia proposta, denominada AutoDCE-TS, formula o problema de previsão como um sistema MISO (múltiplas entradas e uma saída) e integra estratégias de decomposição, seleção causal de características e otimização evolutiva multitarefa.

A primeira estratégia investigada consistiu na aplicação de descoberta causal para seleção de características em séries temporais multivariadas. Os resultados mostraram que esse método é capaz de identificar um conjunto reduzido e relevante de variáveis, produzindo modelos mais simples sem perda de desempenho. Além disso, a geração de grafos causais oferece uma interpretação direta das relações entre variáveis, uma vez que estas são determinadas por testes estatísticos, eliminando a influência de estimadores no processo.

O AutoDCE-TS utiliza esses grafos causais para construir um sistema MISO para cada variável de interesse, o que demanda múltiplos modelos cujas configurações devem ser ajustadas às características de cada série. Este cenário configura um problema de otimização multitarefa, solucionado por meio do AEMOH (Algoritmo Evolutivo Multifatorial para Otimização de Hiperparâmetros). A motivação para empregar o AEMOH no contexto de previsão multivariada decorre do fato de que frequentemente esses problemas envolvem séries de um mesmo domínio e compartilham padrões. Assim, o compartilhamento de informações entre tarefas favorece o processo de otimização e reduz o custo computacional ao permitir que a busca seja realizada de forma conjunta.

Também foi analisado o uso da decomposição da variável endógena, observando-se que ela melhora a precisão das previsões do AutoDCE-TS na maior parte dos casos. No entanto, essa estratégia aumenta a complexidade do problema ao introduzir novas variáveis no pipeline. Os experimentos indicaram que a decisão de utilizar ou não a decomposição deveria igualmente ser automatizada, idealmente por meio de busca orientada por validação. Entretanto, tal procedimento duplicaria o custo computacional, uma vez que exigiria a execução completa do pipeline com e sem decomposição. Diante disso, a decomposição foi tratada como um hiperparâmetro do AutoDCE-TS. Pela mesma razão, o tamanho da

janela de defasagens também não foi automatizado.

A avaliação dos pipelines gerados pelo AutoDCE-TS envolveu comparações com outras abordagens de AutoML para séries temporais, modelos ensemble e métodos de aprendizado profundo. Os resultados demonstraram que o AutoDCE-TS é competitivo em todos os cenários avaliados, independentemente do domínio dos dados, evidenciando sua aplicabilidade geral em tarefas de previsão multivariada.

O processo experimental também evidenciou a escassez de frameworks de AutoML capazes de lidar adequadamente com problemas MISO. Alguns métodos apresentaram limitações práticas: o FEDOT registrou erros de execução e o AutoTS apresentou tempos de processamento excessivamente longos. Esses achados reforçam a necessidade de novos estudos e de frameworks mais robustos e especializados em previsão de séries temporais. Outro desafio relevante foi o desenho experimental para comparação de métodos, dado que cada abordagem adota técnicas distintas para construção e seleção de pipelines, frequentemente baseadas em múltiplas metodologias.

A contribuição do AutoDCE-TS, entretanto, vai além do desempenho preditivo. O método oferece um pipeline automatizado que integra seleção de variáveis, otimização de modelos e hiperparâmetros, e previsão de múltiplos passos à frente. Sua estrutura produz informações valiosas, como grafos causais e árvores de regressão, que aumentam a transparência do fluxo de informações e auxiliam o usuário na compreensão do modelo. Além disso, o AutoDCE-TS gera pipelines mais eficientes em termos de uso de recursos computacionais, característica especialmente importante em cenários onde as séries temporais podem variar ao longo do tempo e requerem retreinamentos periódicos. Esses aspectos são especialmente relevantes em cenários práticos, nos quais interpretabilidade, automação e facilidade de implantação são tão importantes quanto a precisão.

Apesar dessas vantagens, o AutoDCE-TS ainda possui limitações de automação. Por isso, investigou-se o uso de agentes baseados em LLM para apoiar tarefas críticas, como tratamento de valores ausentes, configuração dos hiperparâmetros do AutoDCE-TS e geração de relatórios. O sistema multiagente desenvolvido, denominado AutoDCE-TS-Agentic, demonstrou capacidade de orquestrar agentes especializados para executar, de forma sequencial e explicável, tarefas de imputação, configuração baseada na análise das séries de entrada e preparação de relatórios detalhados.

O AutoDCE-TS-Agentic também forneceu uma interface mais amigável ao usuário ao produzir relatórios que incluem a análise das séries temporais, justificativas das decisões tomadas, informações geradas pelo AutoDCE-TS (como variáveis selecionadas, modelos, hiperparâmetros e previsões) e arquivos adicionais contendo gráficos e modelos treinados. O agente ainda mantém memórias sobre o processo, o que lhe permite responder a perguntas complementares que extrapolam o conteúdo do relatório final. Essas características tornam o AutoDCE-TS mais transparente e reproduzível.

O AutoDCE-TS pode ser adaptado para incorporar novos modelos ao pipeline, sendo limitado apenas por aqueles que não suportam múltiplas defasagens não sequenciais como variáveis de entrada. Além disso, embora concebido como um sistema MISO, o AutoDCE-TS pode ser configurado para operar em um cenário MIMO (Multivariate Input–Multivariate Output), uma vez que trata cada variável de forma independente.

Por fim, ressalta-se que o AutoDCE-TS não foi avaliado em cenários que envolvem horizontes de previsão muito longos, nem em bases de dados com elevada dimensionalidade. Esses aspectos estão diretamente relacionado à sua viabilidade de escalabilidade, uma vez que o aumento do número de variáveis tende a elevar significativamente o custo computacional e a complexidade do processamento. Como perspectivas futuras, sugere-se a incorporação de mecanismos de redução de dimensionalidade para o tratamento das séries temporais, bem como a adoção de estratégias de previsão mais eficientes. Atualmente, o AutoDCE-TS utiliza um método recursivo simples, baseado em previsões de um passo à frente, o que tende a limitar sua eficiência em problemas que demandam horizontes de previsão extensos.

### 6.0.1 Trabalhos Futuros

Como trabalhos futuros, são propostas as seguintes atividades:

1. Avaliar o AutoDCE-TS em cenários mais desafiadores, incluindo previsão MIMO (Multivariate Input–Multivariate Output), permitindo prever simultaneamente todas as variáveis endógenas do sistema.
2. Investigar estratégias de redução de dimensionalidade para séries temporais de alta dimensionalidade, de forma a mitigar problemas de escalabilidade e reduzir o custo computacional do AutoDCE-TS.
3. Avaliar o desempenho do AutoDCE-TS integrando novos modelos de previsão.
4. Expandir o sistema multiagente AutoDCE-TS-Agentic para realizar correções adicionais nos dados, como detecção e tratamento de outliers.
5. Estender o AutoDCE-TS-Agentic com agentes responsáveis por supervisionar cada etapa de execução do AutoDCE-TS, identificando falhas, sugerindo configurações e garantindo maior robustez e automação ao pipeline.
6. Avaliar sistematicamente o AutoDCE-TS-Agentic em diferentes domínios de aplicação, comparando sua capacidade de análise, explicação e geração de relatórios com abordagens tradicionais e outros sistemas assistidos por LLMs.

## 6.1 Publicações

Durante o desenvolvimento desse trabalho de doutorado, foram publicados os seguintes artigos:

### 6.1.1 Publicações em revista

1. LUCAS, Patrícia Oliveira; ORANG, Omid; SILVA, Petrônio Cândido Lima; MENDES, Eduardo Mazoni Andrade Marçal; GUIMARÃES, Frederico Gadelha. A tutorial on fuzzy time series forecasting models: recent advances and challenges. *Learning and Nonlinear Models*, v. 19, n. 2, p. 29, 2022.
2. ORANG, Omid; BITENCOURT, Hugo Vinicius; SOUZA, Luiz Augusto Facury de; LUCAS, Patrícia Oliveira; SILVA, Petrônio Cândido Lima; GUIMARÃES, Frederico Gadelha. Multiple-Input Multiple-Output Randomized Fuzzy Cognitive Map Method for High-Dimensional Time Series Forecasting. *IEEE Transactions on Fuzzy Systems*, v. 32, n. 06, p. 3703 - 3715, 2024. DOI: <https://doi.org/10.1109/TFUZZ.2024.3379853>.
3. BITENCOURT, Hugo Vinicius; LUCAS, Patrícia Oliveira; ORANG, Omid; SILVA, Petrônio Cândido Lima; GUIMARÃES, Frederico Gadelha. A multistep multivariate fuzzy-based time-series forecasting on Internet of Things data. *IEEE Internet of Things Journal*, v. 12, n. 12, p. 21679–21690, jun. 2025. DOI: [10.1109/JIOT.2025.3549715](https://doi.org/10.1109/JIOT.2025.3549715).
4. LUCAS, Patrícia Oliveira; GUIMARÃES, Frederico Gadelha; MENDES, Eduardo Mazoni Andrade Marçal. Automated machine learning based on decomposition, causality and evolutionary multitask optimization for time series forecasting. *Neurocomputing*, v. 656, p. 131569, dez. 2025. DOI: <https://doi.org/10.1016/j.neucom.2025.131569>.

### 6.1.2 Publicações em congresso

1. LUCAS, Patrícia Oliveira; SILVA, Petrônio Cândido Lima; GUIMARÃES, Frederico Gadelha; ALVES, Marcos Antônio. Reference Evapotranspiration Prediction for Precision Agriculture Using Fuzzy Times Series. In: SBPO - Simpósio Brasileiro de Pesquisa Operacional, 2022, Juiz de Fora. Anais do LIV Simpósio Brasileiro de Pesquisa Operacional, 2022. v. 54.
2. LUCAS; Patrícia Oliveira; MENDES, Eduardo Mazoni Andrade Marçal; GUIMARÃES, Frederico Gadelha. Explorando causalidade na seleção de variáveis para previsão de séries temporais multivariadas. In: Anais do XVI Congresso Brasileiro de Inteligência Computacional (CBIC), 2023.



3. FERRAZ, Vinicius Claudino; GUIMARÃES, Frederico Gadelha; LUCAS, Patrícia Oliveira; CAMINHAS, Walmir Matos. Previsão de séries temporais de consumo diário de gás natural no Brasil. In: Anais do Congresso Brasileiro de Automação (CBA), 2024.
4. BITENCOURT, Hugo Vinicius; LUCAS, Patrícia Oliveira; ORANG, Omid; SILVA, Petronio Cândido Lima; GUIMARÃES, Frederico Gadelha. A weighted multivariate fuzzy time series method for multiple output high-dimensional time series forecasting in IoT applications. In: IEEE LATIN AMERICAN CONFERENCE ON COMPUTATIONAL INTELLIGENCE (LA-CCI), 2024. Anais [...]. Piscataway, NJ: IEEE, nov. 2024. p. 1–6. DOI: <https://doi.org/10.1109/LA-CCI62337.2024.10814884>.
5. ORANG, Omid; LUCAS, Patricia O.; PAIVA, Gabriel I. F.; SILVA, Petronio C. L.; SILVA, Felipe Augusto Rocha da; VELOSO, Adriano Alonso; GUIMARÃES, Frederico Gadelha. Causal graph fuzzy LLMs: a first introduction and applications in time series forecasting. arXiv preprint, 2025. DOI: <https://doi.org/10.48550/arXiv.2507.17016>. Disponível em: <https://arxiv.org/abs/2507.17016>. In: Anais do XVII Congresso Brasileiro de Inteligência Computacional (CBIC), 2025.
6. (Em revisão) LUCAS, Patrícia Oliveira; SIMAN, Cauã Ferreira Sathler; PAIVA, Gabriel Ikaro Fonseca de; VELOSO, Gabriel Wedson Mendonça de Souza; LUNA, Julia Almeida; XAVIER, Izabela Esber; ASTORE, Lucas Malacarne; SADA EI, Hossein Javedani; GUIMARÃES, Frederico Gadelha. FAE: A Multi-Agent System for Automated and Explainable Multivariate Time Series Forecasting Pipelines. Association for Computational Linguistics (ACL 2026).

## REFERÊNCIAS

- ALEXANDROV, A. et al. GluonTS: Probabilistic Time Series Models in Python. *CoRR*, abs/1906.05264, 6 2019. Disponível em: <https://arxiv.org/abs/1906.05264v2>.
- ALI, M. *PyCaret: An open source, low-code machine learning library in Python*. 2020.
- ALLEN, R. G.; PEREIRA, L. S.; RAES, D. *Evapotranspiración del cultivo. Guías para la determinación de los requerimientos de agua de los cultivos*. Roma, Itália, 2006. 298 p. Disponível em: <http://academic.uprm.edu/abe/backup2/tomas/fao56.pdf>.
- BAI, S.; KOLTER, J. Z.; KOLTUN, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *CoRR*, abs/1803.0, 2018. Disponível em: <http://arxiv.org/abs/1803.01271>.
- BARBOSH, M.; SINGH, P.; SADHU, A. Empirical mode decomposition and its variants: a review with applications in structural health monitoring. *Smart Materials and Structures*, IOP Publishing, v. 29, n. 9, p. 93001, 8 2020. Disponível em: <https://dx.doi.org/10.1088/1361-665X/aba539>.
- BELETE, D. M.; HUCHAIAH, M. D. Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. <https://doi.org/10.1080/1206212X.2021.1974663>, Taylor & Francis, v. 44, n. 9, p. 875–886, 2021. ISSN 19257074. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/1206212X.2021.1974663>.
- BILAL, M. et al. Auto-Prep: Efficient and Automated Data Preprocessing Pipeline. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 10, p. 107764–107784, 2022. ISSN 21693536. Disponível em: <https://ieeexplore.ieee.org/document/9856663>.
- BISHOP, C. M. *Pattern Recognition and Machine Learning*. 2. ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2011. ISBN 0387310738.
- BREIMAN, L. Random forests. *Machine Learning*, v. 45, p. 5–32, 10 2001. ISSN 08856125.
- CHAN, J. Y.-L. et al. Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review. *Mathematics*, v. 10, n. 8, p. 1283, 4 2022. ISSN 2227-7390.
- CHEN, B. et al. Autostacker: A Compositional Evolutionary Learning System. *CoRR*, abs/1803.00684, 2018. Disponível em: <http://arxiv.org/abs/1803.00684>.
- CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.]: Association for Computing Machinery, 2016. v. 13-17-August-2016, p. 785–794.
- CHEN, T.; GUESTRIN, C. *XGBoost Library*. 2016. Disponível em: <https://xgboost.readthedocs.io/en/stable/index.html>.
- CHEN, Y.-W.; SONG, Q.; HU, X. Techniques for Automated Machine Learning. 7 2019. Disponível em: <http://arxiv.org/abs/1907.08908>.

CHEUNG, Y.-W.; LAI, K. S. Lag Order and Critical Values of the Augmented Dickey-Fuller Test. *Journal of Business & Economic Statistics*, JSTOR, v. 13, n. 3, p. 277, 7 1995. ISSN 07350015.

COMMISSION, E. et al. *Industry 5.0 – Towards a sustainable, human-centric and resilient European industry*. [S.l.]: Publications Office of the European Union, 2021.

DEGUCHI, A. et al. What Is Society 5.0? *Society 5.0: A People-centric Super-smart Society*, Springer, Singapore, p. 1–23, 1 2020. Disponível em: <[https://link.springer.com/chapter/10.1007/978-981-15-2989-4\\_1](https://link.springer.com/chapter/10.1007/978-981-15-2989-4_1)>.

DOLATABADI, S. H. et al. Closing the Data Gap: A Comparative Study of Missing Value Imputation Algorithms in Time Series Datasets. In: . [S.l.: s.n.], 2024. p. 77–90. ISBN 9783031535512.

DUAN, K.; KEERTHI, S. S.; POO, A. N. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, Elsevier, v. 51, p. 41–59, 4 2003. ISSN 0925-2312.

DUBRAWSKI, A. Tuning neural networks with stochastic optimization. *IEEE International Conference on Intelligent Robots and Systems*, IEEE, v. 2, p. 614–621, 1997.

EBERHARDT, F. Introduction to the foundations of causal discovery. *International Journal of Data Science and Analytics*, Springer Science and Business Media Deutschland GmbH, v. 3, n. 2, p. 81–91, 3 2017. ISSN 23644168. Disponível em: <<https://link.springer.com/article/10.1007/s41060-016-0038-6>>.

ERICKSON, N. et al. AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data. 3 2020. Disponível em: <<https://arxiv.org/abs/2003.06505v1>>.

FALKNER, S.; KLEIN, A.; HUTTER, F. BOHB: Robust and Efficient Hyperparameter Optimization at Scale. *35th International Conference on Machine Learning, ICML 2018*, International Machine Learning Society (IMLS), v. 4, p. 2323–2341, 7 2018. Disponível em: <<https://arxiv.org/abs/1807.01774v1>>.

FEURER, M. et al. Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning. *CoRR*, abs/2007.04074, 7 2020. Disponível em: <<http://arxiv.org/abs/2007.04074>>.

FEURER, M. H. F. Hyperparameter Optimization. In: KOTTHOFF, L. H. F.; JOAQUIN, V. (Ed.). *Automated Machine Learning: Methods, Systems, Challenges*. Cham: Springer International Publishing, 2019. p. 3–33. ISBN 978-3-030-05318-5. Disponível em: <[https://doi.org/10.1007/978-3-030-05318-5\\_1](https://doi.org/10.1007/978-3-030-05318-5_1)>.

FISCHER, R.; SAADALLAH, A. AutoXPCR: Automated Multi-Objective Model Selection for Time Series Forecasting. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2024. p. 806–815. ISBN 9798400704901.

FORRESTER, A. I.; KEANE, A. J. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, Pergamon, v. 45, n. 1-3, p. 50–79, 1 2009. ISSN 0376-0421.

FRAZIER, P. I. A Tutorial on Bayesian Optimization. 7 2018. Disponível em: <<https://arxiv.org/abs/1807.02811v1>>.

- FREUND, Y.; SCHAPIRE, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, Academic Press, v. 55, n. 1, p. 119–139, 8 1997. ISSN 0022-0000.
- GANAIÉ, M. A. et al. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, Pergamon, v. 115, p. 105151, 10 2022. ISSN 0952-1976.
- GARNETT, R. Bayesian Optimization. *Cambridge University Press*, 2023.
- GIAGKIOZIS, I.; PURSHOUSE, R. C.; FLEMING, P. J. An overview of population-based algorithms for multi-objective optimisation. <http://dx.doi.org/10.1080/00207721.2013.823526>, Taylor & Francis, v. 46, n. 9, p. 1572–1599, 7 2013. ISSN 14645319.
- GIJSBERS, P.; VANSCHOREN, J. GAMA: a General Automated Machine learning Assistant. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, v. 12461 LNAI, p. 560–564, 7 2020. Disponível em: <<http://arxiv.org/abs/2007.04911>[http://dx.doi.org/10.1007/978-3-030-67670-4\\_39](http://dx.doi.org/10.1007/978-3-030-67670-4_39)>.
- GLYMOUR, C.; ZHANG, K.; SPIRITES, P. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, Frontiers Media S.A., v. 10, n. JUN, p. 418407, 6 2019. ISSN 16648021.
- GUO, S. et al. DS-Agent: Automated Data Science by Empowering Large Language Models with Case-Based Reasoning. In: *Proceedings of Machine Learning Research*. [S.l.]: ML Research Press, 2024. v. 235, p. 16813–16848. ISSN 26403498.
- GUPTA, A.; ONG, Y. S.; FENG, L. Multifactorial Evolution: Toward Evolutionary Multitasking. *IEEE Transactions on Evolutionary Computation*, Institute of Electrical and Electronics Engineers Inc., v. 20, n. 3, p. 343–357, 6 2016. ISSN 1089778X.
- GUYON, I. et al. *LightGBM Library*. 2017. Disponível em: <<https://lightgbm.readthedocs.io/en/latest/index.html>>.
- HE, X.; ZHAO, K.; CHU, X. AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems*, Elsevier B.V., v. 212, 8 2021. Disponível em: <<http://arxiv.org/abs/1908.00709><http://dx.doi.org/10.1016/j.knosys.2020.106622>>.
- HERBOLD, S. Autorank: A Python package for automated ranking of classifiers. *Journal of Open Source Software*, v. 5, n. 48, p. 2173, 4 2020. ISSN 2475-9066.
- HIRA, Z. M.; GILLIES, D. F. A review of feature selection and feature extraction methods applied on microarray data. *Advances in Bioinformatics*, Hindawi Publishing Corporation, v. 2015, 2015. ISSN 16878035.
- HOCHREITER, S.; SCHMIDHUBER, U. J. J. *Long Short-Term Memory*. [S.l.], 1997. v. 9, n. 8, 1735–1780 p. Disponível em: <<http://www7.informatik.tu-muenchen.de/~hochreith><http://www.idsia.ch/~juergen>>.
- HOLLMANN, N.; MÜLLER, S.; HUTTER, F. Large Language Models for Automated Data Science: Introducing CAAFE for Context-Aware Automated Feature Engineering. 5 2023. ISSN 10495258.

- HUANG, N. E. et al. The empirical mode decomposition and the Hubert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Royal Society, v. 454, n. 1971, p. 903–995, 1998. ISSN 13645021.
- JAMIESON, K.; TALWALKAR, A. Non-stochastic Best Arm Identification and Hyperparameter Optimization. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016*, PMLR, p. 240–248, 2 2015. Disponível em: <<https://arxiv.org/abs/1502.07943v1>>.
- KAUL, A.; MAHESHWARY, S.; PUDI, V. Autolearn - automated feature generation and selection. *Proceedings - IEEE International Conference on Data Mining, ICDM*, Institute of Electrical and Electronics Engineers Inc., v. 2017-November, p. 217–226, 12 2017. ISSN 15504786.
- KE, G. et al. *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. 2016.
- KHURANA, U.; SAMULOWITZ, H.; TURAGA, D. Feature Engineering for Predictive Modeling using Reinforcement Learning. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, AAAI press, p. 3407–3414, 9 2017. ISSN 2159-5399. Disponível em: <<https://arxiv.org/abs/1709.07150v1>>.
- KLEIN, A. et al. Learning Curve Prediction with Bayesian Neural Networks. In: *International Conference on Learning Representations*. [S.l.: s.n.], 2022.
- KOHAVI, R.; JOHN, G. H. Automatic Parameter Selection by Minimizing Estimated Error. *Machine Learning Proceedings 1995*, Morgan Kaufmann, p. 304–312, 1 1995.
- KOPRINSKA, I.; RANA, M.; AGELIDIS, V. G. Correlation and instance based feature selection for electricity load forecasting. *Knowledge-Based Systems*, Elsevier, v. 82, p. 29–40, 7 2015. ISSN 0950-7051. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0950705115000714>>.
- KOTTHOFF, L. et al. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *Journal of Machine Learning Research*, v. 18, p. 1–5, 2017. Disponível em: <<http://automl.org/autowekahttps://www.cs.ubc.ca/labs/algorithms/Projects/autoweka/papers/16-599.pdf>>.
- KOZIEL, S.; CIAURRI, D. E.; LEIFSSON, L. Surrogate-based methods. *Studies in Computational Intelligence*, Springer, Berlin, Heidelberg, v. 356, p. 33–59, 2011. ISSN 1860949X. Disponível em: <[https://link.springer.com/chapter/10.1007/978-3-642-20859-1\\_3](https://link.springer.com/chapter/10.1007/978-3-642-20859-1_3)>.
- KRISHNAN, S. et al. BoostClean: Automated Error Detection and Repair for Machine Learning. 11 2017. Disponível em: <<https://arxiv.org/abs/1711.01299v1>>.
- KRISHNAN, S.; WU, E. AlphaClean: Automatic Generation of Data Cleaning Pipelines. 2019. Disponível em: <<https://doi.org/10.1145/nnnnnnn.nnnnnnn>>.
- KWIATKOWSKI, D. et al. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics*, North-Holland, v. 54, n. 1-3, p. 159–178, 10 1992. ISSN 0304-4076.

- LEDELL, E.; POIRIER, S. H2O AutoML: Scalable Automatic Machine Learning. *7th ICML Workshop on Automated Machine Learning (AutoML)*, 2020.
- LEE, T. EMD and LSTM Hybrid Deep Learning Model for Predicting Sunspot Number Time Series with a Cyclic Pattern. *Solar Physics*, v. 295, n. 6, p. 82, 6 2020. ISSN 0038-0938.
- LI, L. et al. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Journal of Machine Learning Research*, Microtome Publishing, v. 18, p. 1–52, 3 2016. ISSN 15337928. Disponível em: <<https://arxiv.org/abs/1603.06560v4>>.
- LI, Z.; ZHANG, Q.; LI, Z. Sunspot time series prediction based on EMD and LSTM. In: *2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. [S.l.]: IEEE, 2023. p. 939–943. ISBN 978-1-6654-6004-0.
- LIM, B. et al. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, Elsevier B.V., v. 37, p. 1748–1764, 10 2021. ISSN 01692070.
- LIU, S.; GAO, C.; LI, Y. Large Language Model Agent for Hyper-Parameter Optimization. 2 2024.
- LIU, Y. et al. ITRANSFORMER: INVERTED TRANSFORMERS ARE EFFECTIVE FOR TIME SERIES FORECASTING. In: *12th International Conference on Learning Representations, ICLR 2024*. [S.l.]: International Conference on Learning Representations, ICLR, 2024.
- LIU, Z.; LIU, J. A robust time series prediction method based on empirical mode decomposition and high-order fuzzy cognitive maps. *Knowledge-Based Systems*, v. 203, p. 106105, 9 2020. ISSN 09507051.
- LUO, Y.; HAN, S.; LIU, J. Machine learning’s own Industrial Revolution. 11 2023. Disponível em: <<https://arxiv.org/abs/2311.02278v1>>.
- LUO, Y. et al. AutoCross: Automatic Feature Crossing for Tabular Data in Real-World Applications. *CoRR*, abs/1904.12857, 2019. Disponível em: <<http://arxiv.org/abs/1904.12857>>.
- MENÉNDEZ, M. L. et al. The Jensen-Shannon divergence. *Journal of the Franklin Institute*, Pergamon, v. 334, n. 2, p. 307–318, 3 1997. ISSN 0016-0032.
- MOTZ, M.; KRAUSS, J.; SCHMITT, R. H. Benchmarking of hyperparameter optimization techniques for machine learning applications in production. *Advances in Industrial and Manufacturing Engineering*, Elsevier, v. 5, p. 100099, 11 2022. ISSN 2666-9129.
- MOUNIR, N.; OUADI, H.; JRHILIFA, I. Short-term electric load forecasting using an EMD-BI-LSTM approach for smart grid energy management system. *Energy and Buildings*, Elsevier Ltd, v. 288, 6 2023. ISSN 03787788.
- NIKITIN, N. O. et al. Automated evolutionary approach for the design of composite machine learning pipelines. *Future Generation Computer Systems*, North-Holland, v. 127, p. 109–125, 2 2022. ISSN 0167-739X.



NIKITIN, N. O. et al. *FEDOT Package*. 2022. Disponível em: <<https://fedot.readthedocs.io/en/latest/index.html>>.

OLSON, R. S. et al. Evaluation of a tree-based pipeline optimization tool for automating data science. *GECCO 2016 - Proceedings of the 2016 Genetic and Evolutionary Computation Conference*, Association for Computing Machinery, Inc, p. 485–492, 7 2016. Disponível em: <<http://dx.doi.org/10.1145/2908812.2908918>>.

ORESHKIN, B. N. et al. N-BEATS: NEURAL BASIS EXPANSION ANALYSIS FOR INTERPRETABLE TIME SERIES FORECASTING. In: *8th International Conference on Learning Representations, ICLR 2020*. [S.l.]: International Conference on Learning Representations, ICLR, 2020.

OSABA, E. et al. Evolutionary Multitask Optimization: a Methodological Overview, Challenges, and Future Research Directions. *Cognitive Computation*, Springer, v. 14, n. 3, p. 927–954, 5 2022. ISSN 18669964. Disponível em: <<https://link.springer.com/article/10.1007/s12559-022-10012-8>>.

PALDINO, G. M. et al. Does AutoML Outperform Naive Forecasting? *Engineering Proceedings*, MDPI AG, v. 5, n. 1, p. 36, 7 2021.

PASSALACQUA, M. et al. Human-centred AI in industry 5.0: a systematic review. *International Journal of Production Research*, v. 63, n. 7, p. 2638–2669, 4 2025. ISSN 0020-7543.

PASSOS, D.; MISHRA, P. A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks. *Chemometrics and Intelligent Laboratory Systems*, Elsevier, v. 223, p. 104520, 4 2022. ISSN 0169-7439.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Disponível em: <<http://scikit-learn.sourceforge.net>>

PILYUGINA, P. et al. A Large-Scale Empirical Study of Aligned Time Series Forecasting. *IEEE Access*, v. 12, p. 131100–131121, 2024. ISSN 2169-3536.

PROBST, P.; BOULESTEIX, A. L. To tune or not to tune the number of trees in random forest? *Journal of Machine Learning Research*, Microtome Publishing, v. 18, p. 1–8, 5 2017. ISSN 15337928. Disponível em: <<https://arxiv.org/pdf/1705.05654>>.

QI, F. et al. DarwinML: A Graph-based Evolutionary Algorithm for Automated Machine Learning. *CoRR*, abs/1901.08013, 2019. Disponível em: <<http://arxiv.org/abs/1901.08013>>.

REKATSINAS, T. et al. HoloClean: Holistic Data Repairs with Probabilistic Inference. *Proceedings of the VLDB Endowment*, Association for Computing Machinery, v. 10, n. 11, p. 1190–1201, 2 2017. ISSN 21508097. Disponível em: <<https://arxiv.org/abs/1702.00820v1>>.

RIBEIRO, S. I. M.; CASTRO, C. I. Leite de. *Missing Data in Time Series: A Review of Imputation Methods and Case Study*. [S.l.], 2022. v. 20, n. 1, 31–46 p.

- RUNGE, J. et al. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science Advances*, American Association for the Advancement of Science, v. 5, n. 11, p. 4996–5023, 11 2019. ISSN 23752548. Disponível em: <<https://www.science.org/doi/10.1126/sciadv.aau4996>>.
- SHAH, S. Y. et al. AutoAI-TS: AutoAI for Time Series Forecasting. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*. [S.l.]: Association for Computing Machinery, 2021. p. 2584–2596. ISSN 07308078.
- SHCHUR, O. et al. AutoGluon-TimeSeries: AutoML for Probabilistic Time Series Forecasting. In: *International Conference on Automated Machine Learning*. [s.n.], 2023. Disponível em: <<http://arxiv.org/abs/2308.05566>>.
- SHEIKHPOUR, R. et al. A Survey on semi-supervised feature selection methods. *Pattern Recognition*, Pergamon, v. 64, p. 141–158, 4 2017. ISSN 0031-3203.
- SHEN, Y. et al. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face. In: *Advances in Neural Information Processing Systems*. [S.l.]: Neural information processing systems foundation, 2023. v. 36. ISSN 10495258.
- SILVA, P. C. L. et al. Distributed Evolutionary Hyperparameter Optimization for Fuzzy Time Series. *IEEE Transactions on Network and Service Management*, Institute of Electrical and Electronics Engineers (IEEE), p. 1–1, 3 2020.
- SWERSKY, K.; SNOEK, J.; ADAMS, R. P. Freeze-Thaw Bayesian Optimization. *CoRR*, 6 2014. Disponível em: <<https://arxiv.org/abs/1406.3896v1>>.
- TAN, K. C.; FENG, L.; JIANG, M. Evolutionary Transfer Optimization - A New Frontier in Evolutionary Computation Research. *IEEE Computational Intelligence Magazine*, Institute of Electrical and Electronics Engineers Inc., v. 16, n. 1, p. 22–33, 2 2021. ISSN 15566048.
- TAN, Z.; LUO, L.; ZHONG, J. *Knowledge transfer in evolutionary multi-task optimization: A survey*. [S.l.]: Elsevier Ltd, 2023.
- TIBSHIRANI, R. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, v. 58, n. 1, p. 267–288, 1 1996. ISSN 1369-7412.
- TRIRAT, P.; JEONG, W.; HWANG, S. J. AutoML-Agent: A Multi-Agent LLM Framework for Full-Pipeline AutoML. 10 2024.
- TUGGENER, L. et al. Automated Machine Learning in Practice: State of the Art and Recent Results. 7 2019. Disponível em: <<http://arxiv.org/abs/1907.08392><http://dx.doi.org/10.1109/SDS.2019.00-11>>.
- VANSCHOREN, J. Meta-Learning. In: KOTTHOFF, L. H. F.; JOAQUIN, V. (Ed.). *Automated Machine Learning: Methods, Systems, Challenges*. Cham: Springer International Publishing, 2019. p. 35–61. ISBN 978-3-030-05318-5. Disponível em: <[https://doi.org/10.1007/978-3-030-05318-5\\_2](https://doi.org/10.1007/978-3-030-05318-5_2)>.
- VERDONCK, T. et al. Special issue on feature engineering editorial. *Machine Learning*, Springer, p. 1–12, 8 2021. ISSN 15730565. Disponível em: <<https://link.springer.com/article/10.1007/s10994-021-06042-2>>.



- VIEIRA, A. C. et al. Improving flood forecasting through feature selection by a genetic algorithm – experiments based on real data from an Amazon rainforest river. *Earth Science Informatics*, Springer Science and Business Media Deutschland GmbH, v. 14, n. 1, p. 37–50, 3 2021. ISSN 18650481.
- VIEIRA, S.; PINAYA, W. H. L.; MECHELLI, A. Introduction to machine learning. *Machine Learning: Methods and Applications to Brain Disorders*, Academic Press, p. 1–20, 1 2020.
- WANG, C. et al. AutoTS: Automatic Time Series Forecasting Model Design Based on Two-Stage Pruning. *JOURNAL OF LATEX CLASS FILES*, v. 14, n. 8, 2022.
- WANG, C. et al. *AutoTS Package*. 2022. Disponível em: <<https://winedarksea.github.io/AutoTS/build/html/source/intro.html>>.
- WHITE, C. et al. BANANAS: Bayesian Optimization with Neural Architectures for Neural Architecture Search. *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, Association for the Advancement of Artificial Intelligence, v. 12A, p. 10293–10301, 10 2019. ISSN 2159-5399. Disponível em: <<https://arxiv.org/abs/1910.11858v3>>.
- WOLPERT, D. H.; MACREADY, W. G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, v. 1, n. 1, p. 67–82, 1997. ISSN 1089778X.
- WU, G.; MALLIPEDDI, R.; SUGANTHAN, P. N. Ensemble strategies for population-based optimization algorithms – A survey. *Swarm and Evolutionary Computation*, Elsevier, v. 44, p. 695–711, 2 2019. ISSN 2210-6502.
- XU, J. et al. Large Language Models Synergize with Automated Machine Learning. 5 2024.
- YANG, Z. et al. *CARS: Continuous Evolution for Efficient Neural Architecture Search*. 2020. 1829–1838 p.
- YAO, S. et al. ReAct: Synergizing Reasoning and Acting in Language Models. 10 2022. Disponível em: <<http://arxiv.org/abs/2210.03629>>.
- YU, K. et al. Causality-based Feature Selection. *ACM Computing Surveys (CSUR)*, ACM/PUB27New York, NY, USA, v. 53, n. 5, 9 2020. ISSN 15577341. Disponível em: <<https://dl.acm.org/doi/10.1145/3409382>>.
- YU, K.; LIU, L.; LI, J. A Unified View of Causal and Non-causal Feature Selection. 2018.
- ZANGA, A.; OZKIRIMLI, E.; STELLA, F. A Survey on Causal Discovery: Theory and Practice. *International Journal of Approximate Reasoning*, Elsevier, v. 151, p. 101–129, 12 2022. ISSN 0888-613X.
- ZEILER, A. et al. Empirical mode decomposition - An introduction. In: *Proceedings of the International Joint Conference on Neural Networks*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2010. ISBN 9781424469178.
- ZHANG, J.; FOGELMAN-SOULIÉ, F.; LARGERON, C. Towards automatic complex feature engineering. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, v. 11234 LNCS, p. 312–322, 2018. ISSN 16113349. Disponível em: <[https://link.springer.com/chapter/10.1007/978-3-030-02925-8\\_22](https://link.springer.com/chapter/10.1007/978-3-030-02925-8_22)>.

ZHAO, H. et al. What makes evolutionary multi-task optimization better: A comprehensive survey. *Applied Soft Computing*, Elsevier, v. 145, p. 110545, 9 2023. ISSN 1568-4946.

ZHAO, H. et al. TimeSeriesScientist: A General-Purpose AI Agent for Time Series Analysis. 10 2025. Disponível em: <<https://arxiv.org/pdf/2510.01538>>.

ZIMMER, L.; LINDAUER, M.; HUTTER, F. Auto-Pytorch: Multi-Fidelity MetaLearning for Efficient and Robust AutoDL. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 43, n. 9, 2021. Disponível em: <<https://github.com/automl/Auto-PyTorch>>.

ZÖLLER, M.-A.; HUBER, M. F. Benchmark and Survey of Automated Machine Learning Frameworks. *Journal of Artificial Intelligence Research*, v. 70, p. 409–474, 4 2021. Disponível em: <<http://arxiv.org/abs/1904.12054>>.

ZOPH, B.; BRAIN, Q. V. L. G. Neural Architecture Search with Reinforcement Learning. In: *International Conference on Learning Representations*. [S.l.: s.n.], 2017.

## A Conjuntos de dados

Este capítulo descreve as bases de dados utilizadas neste trabalho, nomeadas de acordo com seus respectivos domínios de aplicação: econômico, energia, Internet das coisas e clima. Para contribuir com a reprodutibilidade da pesquisa, todos os dados estão disponíveis em <https://github.com/PatriciaLucas/AutoML/tree/main/Datasets>.

### A.1 ECONOMICO\_1

O conjunto de dados ECONOMICO\_1 é usado para prever o índice médio diário que reflete as cotações de Bitcoin dos Estados Unidos. Ele é composto de amostras diárias de 2010 a 2018. As séries temporais referentes a cada variável estão descritas na Tabela 9 e a variável AVG, usada como alvo, está apresentada na Figura 16.

### A.2 ECONOMICO\_2

O conjunto de dados ECONOMICO\_2 é usado para prever o índice médio diário que reflete aproximadamente a situação do mercado financeiro dos Estados Unidos. Ele é composto por uma combinação das 30 ações mais importantes. Além do índice médio, outras variáveis neste conjunto de dados incluem os preços de abertura, máxima, mínima, fechamento e o volume de negociações para cada dia útil de 1985 a 2017. As séries temporais referentes a cada variável estão descritas na Tabela 10 e a variável AVG, usada como alvo, está apresentada na Figura 16.

### A.3 ECONOMICO\_3

O conjunto de dados ECONOMICO\_3 é usado para prever o índice médio diário que reflete aproximadamente a situação do mercado mercado FOREX (Foreign Exchange) do par de moedas EUR/GBP. Além do índice médio, outras variáveis neste conjunto de

Tabela 9 – Descrição das variáveis contidas na base de dados ECONOMICO\_1.

Variável	Tipo	Descrição	Média	Desvio padrão
Open	Real	Preço de abertura	1381.63	2910.87
High	Real	Preço máximo	1431.53	3028.51
Low	Real	Preço mínimo	1324.15	2762.81
Close	Real	Preço de fechamento	1383.97	2912.55
Volume	Real	Volume de negociações diário	$1.42 \times 10^8$	$4.17 \times 10^8$
AVG	Real	Índice médio diário	1380.32	2902.08

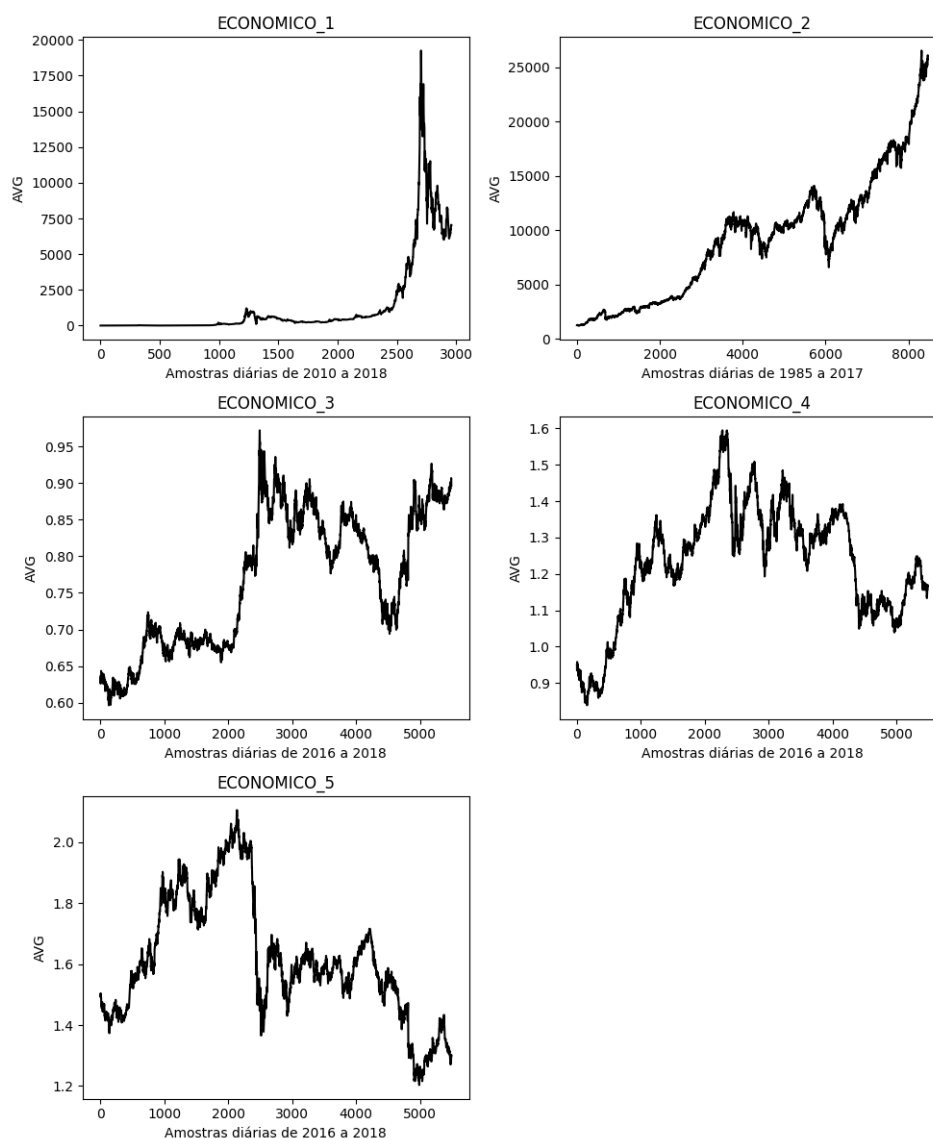


Figura 16 – Variáveis alvo das bases de dados ECONOMIC\_1, ECONOMIC\_2, ECONOMIC\_3, ECONOMIC\_4 e ECONOMIC\_5.

Tabela 10 – Descrição das variáveis contidas na base de dados ECONOMIC\_2.

Variável	Tipo	Descrição	Média	Desvio padrão
Open	Real	Preço de abertura	9182.66	5724.70
High	Real	Preço máximo	9238.91	5747.54
Low	Real	Preço mínimo	9125.19	5701.10
Close	Real	Preço de fechamento	9185.42	5725.67
Volume	Real	Volume de negociações diário	$1.32 \times 10^8$	$1.16 \times 10^8$
AVG	Real	Índice médio diário	9183.05	5724.55

Tabela 11 – Descrição das variáveis contidas na base de dados ECONOMICO\_3.

Variável	Tipo	Descrição	Média	Desvio padrão
Open	Real	Preço de abertura	0.76	0.09
High	Real	Preço máximo	0.76	0.09
Low	Real	Preço mínimo	0.76	0.09
Close	Real	Preço de fechamento	0.76	0.09
AVG	Real	Índice médio diário	0.76	0.09

Tabela 12 – Descrição das variáveis contidas na base de dados ECONOMICO\_4.

Variável	Tipo	Descrição	Média	Desvio padrão
Open	Real	Preço de abertura	1.23	0.16
High	Real	Preço máximo	1.23	0.16
Low	Real	Preço mínimo	1.22	0.15
Close	Real	Preço de fechamento	1.23	0.16
AVG	Real	Índice médio diário	1.23	0.16

dados incluem os preços de abertura, máxima, mínima e o fechamento para cada dia útil de 2016 a 2018. As séries temporais referentes a cada variável estão descritas na Tabela 11 e a variável AVG, usada como alvo, está apresentada na Figura 16.

#### A.4 ECONOMICO\_4

O conjunto de dados ECONOMICO\_4 é usado para prever o índice médio diário que reflete aproximadamente a situação do mercado mercado FOREX (Foreign Exchange) do par de moedas EUR-USD. Além do índice médio, outras variáveis neste conjunto de dados incluem os preços de abertura, máxima, mínima e o fechamento para cada dia útil de 2016 a 2018. As séries temporais referentes a cada variável estão descritas na Tabela 12 e a variável AVG, usada como alvo, está apresentada na Figura 16.

#### A.5 ECONOMICO\_5

O conjunto de dados ECONOMICO\_5 é usado para prever o índice médio diário que reflete aproximadamente a situação do mercado mercado FOREX (Foreign Exchange) do par de moedas GBP-USD. Além do índice médio, outras variáveis neste conjunto de dados incluem os preços de abertura, máxima, mínima e o fechamento para cada dia útil de 2016 a 2018. As séries temporais referentes a cada variável estão descritas na Tabela 13 e a variável AVG, usada como alvo, está apresentada na Figura 16.

Tabela 13 – Descrição das variáveis contidas na base de dados ECONOMICO\_5.

Variável	Tipo	Descrição	Média	Desvio padrão
Open	Real	Preço de abertura	1.61	0.20
High	Real	Preço máximo	1.62	0.20
Low	Real	Preço mínimo	1.60	0.19
Close	Real	Preço de fechamento	1.61	0.20
AVG	Real	Índice médio diário	1.61	0.20

Tabela 14 – Descrição das variáveis contidas na base de dados ENERGIA\_1.

Variável	Tipo	Descrição	Média	Desvio padrão
temperature <sub>2m</sub>	Real	Temperatura	47.86	19.45
relativehumidity <sub>2m</sub>	Real	Umidade relativa a 2m	72.28	16.85
dewpoint <sub>2m</sub>	Real	Ponto de orvalho a 2m	38.56	18.77
windspeed <sub>10m</sub>	Real	Velocidade do vento a 10m	3.59	1.64
windspeed <sub>100m</sub>		Velocidade do vento a 100m	6.28	2.68
winddirection <sub>10m</sub>	Real	Direção do vento a 10m	203.63	96.37
winddirection <sub>100m</sub>	Real	Direção do vento a 100m	203.34	97.95
windgusts <sub>10m</sub>	Real	Rajadas de vento	7.77	3.56
Power	Real	Potência	0.40	0.28

## A.6 ENERGIA\_1

O conjunto de dados ENERGIA\_1 é usado para prever a produção de energia eólica. O conjunto de dados representa um registro detalhado por hora, começando em 2 de janeiro de 2017. As séries temporais referentes a cada variável estão descritas na Tabela 14 e a variável Power, usada como alvo, está apresentada na Figura 17.

## A.7 ENERGIA\_2, ENERGIA\_3, ENERGIA\_4, ENERGIA\_5

Os conjunto de dados ENERGIA\_2, ENERGIA\_3, ENERGIA\_4, ENERGIA\_5 contém as mesmas variáveis. Porém, neste trabalho foram nomeados diferentes para representar o objetivo de previsão. Este conjunto de dados contém 4 anos de consumo elétrico, geração, preços e dados meteorológicos para a cidade de Valência na Espanha. ENERGIA\_2 foi usado para prever o consumo de energia elétrica. ENERGIA\_3 foi usado para prever o consumo de energia solar. ENERGIA\_4 foi usado para prever o consumo de energia eólica. ENERGIA\_5 foi usado para prever o preço da energia elétrica. As séries temporais referentes a cada variável estão descritas na Tabela 15. As variáveis usadas como alvo estão apresentadas na Figura 17.

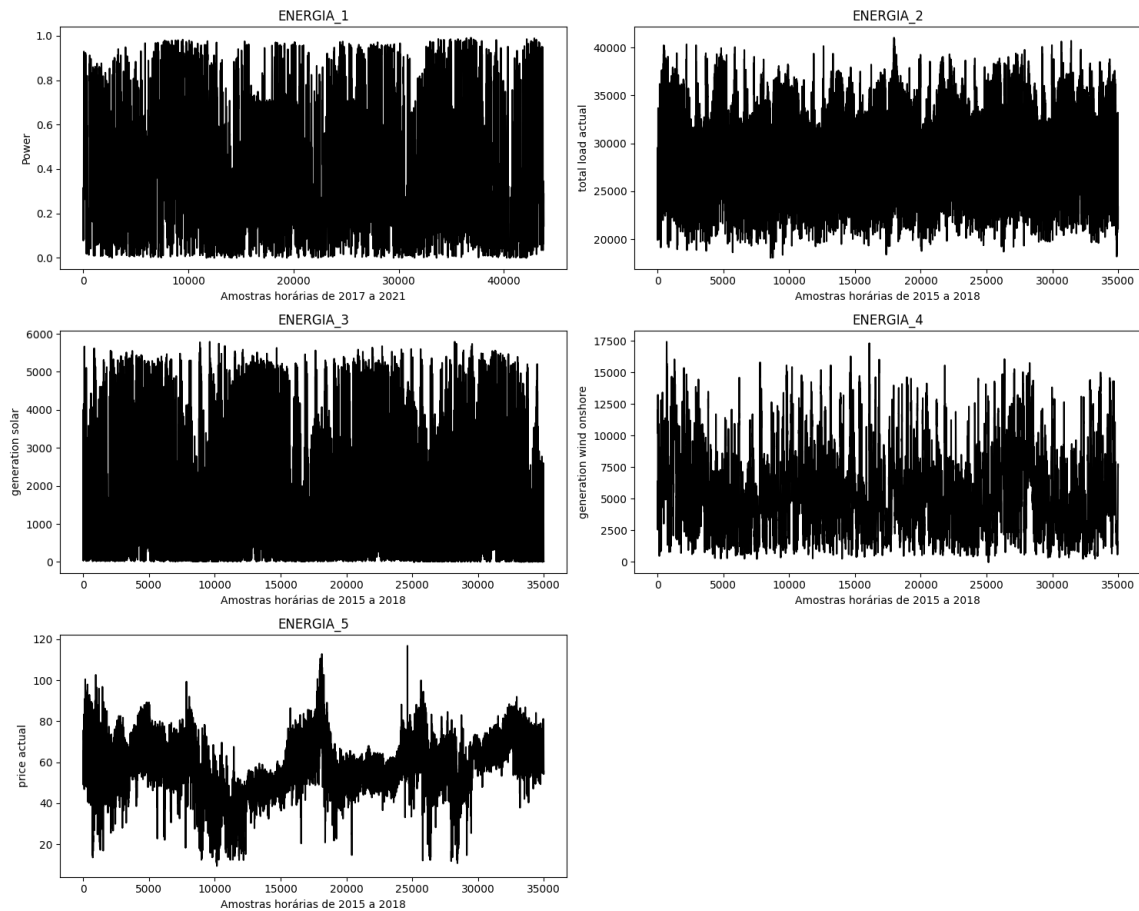


Figura 17 – Variáveis alvo das bases de dados ENERGIА\_1, ENERGIА\_2, ENERGIА\_3, ENERGIА\_4 e ENERGIА\_5.

## A.8 IOT\_1

O conjunto de dados IOT\_1 é utilizado para prever o consumo de energia elétrica numa residência. Um sistema de aquisição de dados ad-hoc foi projetado e implementado para coletar dados de energia e clima de uma casa doméstica no nordeste do México. Os dados foram coletados por 14 meses, começando em 5 de novembro de 2022 a 5 de janeiro de 2024. A série temporal referente a cada variável está descrita na Tabela 16 e a variável `active_power`, usada como alvo, está apresentada na Figura 18.

## A.9 IOT\_2

O conjunto de dados IOT\_2 é utilizado para prever o consumo de energia de eletrodomésticos em residências. Os dados consistem em medições de temperatura e umidade coletadas por uma rede de sensores sem fio (WSN) em várias áreas de uma casa de baixo consumo de energia na Bélgica, informações meteorológicas de uma estação meteorológica próxima e o consumo de energia registrado de eletrodomésticos e luminárias. Os dados de consumo de energia dos eletrodomésticos foram medidos continuamente (a

Tabela 15 – Descrição das variáveis contidas na base de dados ENERGIA\_2, ENERGIA\_3, ENERGIA\_4 e ENERGIA\_5.

Variável	Tipo	Descrição	Média	Desvio padrão
generation biomass	Real	Geração de biomassa	383.71	85.31
generation fossil brown coal/lignite	Real	Geração de carvão marrom/lignite	448.91	354.42
generation fossil gas	Real	Geração de gás fóssil	5622.76	2203.30
generation fossil hard coal	Real	Geração de carvão duro	4260.89	1961.02
generation fossil oil	Real	Geração de óleo fóssil	298.48	52.45
generation hydro pumped storage consumption	Real	Consumo de armazenamento hidroelétrico bombeado	475.83	792.61
generation hydro run-of-river and poundage	Real	Geração hidroelétrica de passagem e reservatório	972.00	401.04
generation hydro water reservoir	Real	Geração hidroelétrica por reservatório de água	2606.47	1836.27
generation nuclear	Real	Geração nuclear	6263.83	841.00
generation other	Real	Outras gerações	60.22	20.25
generation other renewable	Real	Outra geração renovável	85.61	14.08
generation solar	Real	Geração solar	1433.45	1680.33
generation waste	Real	Geração a partir de resíduos	269.35	50.23
generation wind onshore	Real	Geração eólica onshore	5466.14	3215.63
total load actual	Real	Carga total atual	28702.56	4576.76
price actual	Real	Preço atual	57.86	14.21
temp	Real	Temperatura	290.81	7.22
temp_min	Real	Temperatura mínima	290.25	7.07
temp_max	Real	Temperatura máxima	291.38	7.51
pressure	Real	Pressão	1015.92	11.92
humidity	Real	Umidade	65.09	19.67
wind_speed	Real	Velocidade do vento	2.69	2.58
wind_deg	Real	Direção do vento	160.55	120.43
rain_1h	Real	Chuva	0.03	0.26
clouds_all	Real	Nuvens	20.79	25.63

Tabela 16 – Descrição da série temporal das variáveis do conjunto de dados IOT\_1.

Variável	Tipo	Descrição	Média	Desvio padrão
active_power	Real	Potência ativa	286.01	189.54
current	Real	Corrente elétrica	2.58	1.59
voltage	Real	Tensão elétrica	125.41	4.39
apparent_power	Real	Potência aparente calculada	321.83	191.74
reactive_power	Real	Potência reativa calculada	132.53	71.03
power_factor	Real	Fator de potência calculado	0.85	0.11
temp	Real	Temperatura	19.52	6.60
feels_like	Real	Sensação de temperatura	18.71	6.72
temp_min	Real	Temperatura mínima	18.08	6.51
temp_max	Real	Temperatura máxima	20.80	6.35
pressure	Real	Pressão atmosférica	1015.22	5.99
humidity	Real	Porcentagem de umidade	47.95	19.44
speed	Real	Velocidade do vento	2.62	2.09
deg	Real	Direção do vento	152.73	127.59



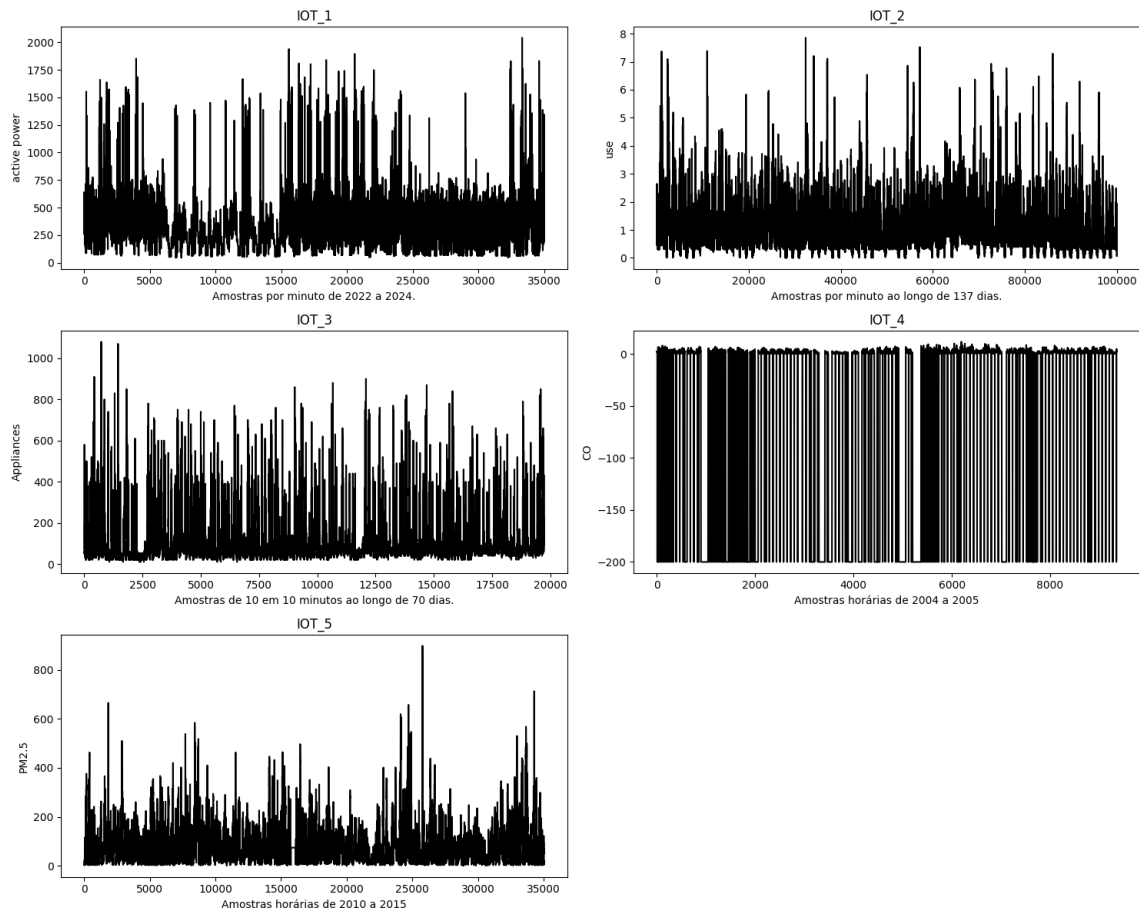


Figura 18 – Variáveis alvo das bases de dados IOT\_1, IOT\_2, IOT\_3, IOT\_4, IOT\_5 e IOT\_6.

cada minuto) ao longo de 70 dias. A série temporal referente a cada variável está descrita na Tabela 17 e a variável use, usada como alvo, está apresentada na Figura 18.

## A.10 IOT\_3

O conjunto de dados IOT\_3 é utilizado para prever o consumo de energia de aparelhos em um edifício de baixo consumo de energia na Bélgica. Os dados de energia foram registrados a cada 10 minutos com medidores de energia m-bus durante 70 dias. O clima da estação meteorológica do aeroporto mais próximo (Aeroporto de Chievres, Bélgica) foi baixado de um conjunto de dados público da Reliable Prognosis (rp5.ru) e mesclado com os conjuntos de dados experimentais usando a coluna de data e hora. A série temporal referente a cada variável está descrita na Tabela 18 e a variável Appliances, usada como alvo, está apresentada na Figura 18.

Tabela 17 – Descrição da série temporal das variáveis do conjunto de dados IOT\_2.

Variável	Tipo	Descrição	Média	Desvio padrão
use	Real	Consumo total de energia	0.98	0.69
gen	Real	Energia total gerada	0.06	0.12
Dishwasher	Real	Energia consumida pelo lava-louças	0.04	0.22
Furnace 1	Real	Energia consumida pelo forno 1	0.22	0.22
Furnace 2	Real	Energia consumida pelo forno 2	0.23	0.24
Home office	Real	Energia consumida no escritório	0.11	0.13
Fridge	Real	Energia consumida pela geladeira	0.05	0.07
Wine cellar	Real	Energia consumida pela adega	0.01	0.03
Garage door	Real	Energia consumida pela porta da garagem	0.01	0.00
Kitchen 12	Real	Energia consumida pela cozinha 1	0.00	0.00
Kitchen 14	Real	Energia consumida pela cozinha 2	0.01	0.08
Kitchen 38	Real	Energia consumida pela cozinha 3	1.04	1.03
Barn	Real	Energia consumida pelo celeiro	0.07	0.18
Well	Real	Energia consumida pelo poço	0.01	0.15
Microwave	Real	Energia consumida pelo microondas	0.01	0.10
Living room	Real	Energia consumida pela Sala de estar	0.04	0.10
Solar	Real	Geração de energia solar	0.06	0.12
temperature	Real	Temperatura	31.06	12.98
humidity	Real	Humidade	0.61	0.16
visibility	Real	Visibilidade	9.91	1.64
apparentTemperature	Real	Temperatura aparente	25.06	15.74
pressure	Real	Pressão	1015.79	9.28
windSpeed	Real	Velocidade do vento	7.70	4.28
windBearing	Real	Direção do ventor	251.66	103.37
precipIntensity	Real	Intensidade da precipitação	0.00	0.01
dewPoint	Real	Ponto de orvalho	18.56	13.95
precipProbability	Real	Probabilidade de precipitação	0.05	0.15

## A.11 IOT\_4

O conjunto de dados IOT\_4 é utilizado para prever concentração média horária real de óxido de metal incorporados em um dispositivo multissensor químico de qualidade do ar em uma cidade italiana. Os dados foram registrados de hora em hora de março de 2004 a fevereiro de 2005 (um ano). A série temporal referente a cada variável está descrita na Tabela 19 e a variável CO, usada como alvo, está apresentada na Figura 18.

## A.12 IOT\_5

O conjunto de dados IOT\_5 é utilizado para a previsão horária da poluição do ar usando a variável PM2.5, que se refere a partículas atmosféricas (PM) com um diâmetro inferior a 2,5 micrômetros. Os dados referem-se a Aot Zhongxin, China, abrangendo janeiro de 2010 a dezembro de 2015. A série temporal horária referente a cada variável está descrita na Tabela 20 e a variável PM2.5, usada como alvo, está apresentada na Figura 18.

Tabela 18 – Descrição da série temporal das variáveis do conjunto de dados IOT.3.

Variável	Tipo	Descrição	Média	Desvio padrão
Appliances	Real	Uso de energia dos eletrodomésticos	97.69	102.52
lights	Real	Uso de energia das luminárias	3.80	7.93
T1	Real	Temperatura na área da cozinha	21.68	1.60
RH_1	Real	Umidade na área da cozinha	40.25	3.97
T2	Real	Temperatura na área da sala de estar	20.34	2.19
RH_2	Real	Umidade na área da sala de estar	40.42	4.06
T3	Real	Temperatura na área da lavanderia	22.26	2.00
RH_3	Real	Umidade na área da lavanderia	39.24	3.25
T4	Real	Temperatura na sala de escritório	20.85	2.04
RH_4	Real	Umidade na sala de escritório	39.02	4.34
T5	Real	Temperatura no banheiro	19.59	1.84
RH_5	Real	Umidade no banheiro	50.94	9.02
T6	Real	Temperatura fora do prédio	7.91	6.09
RH_6	Real	Umidade fora do prédio	54.60	31.14
T7	Real	Temperatura na sala de passar roupa	20.26	2.10
RH_7	Real	Umidade na sala de passar roupa	35.38	5.11
T8	Real	Temperatura no quarto do adolescente	22.02	1.95
RH_8	Real	Umidade no quarto do adolescente	42.93	5.22
T9	Real	Temperatura no quarto dos pais	19.48	2.01
RH_9	Real	Umidade no quarto dos pais	41.55	4.15
T_out	Real	Temperatura externa	7.41	5.31
Press_mm_hg	Real	Pressão externa	755.52	7.39
RH_out	Real	Umidade externa	79.75	14.90
Windspeed	Real	Velocidade do vento	4.03	2.45
Visibility	Real	Visibilidade	38.33	11.79
Tdewpoint	Real	Temperatura do ponto de orvalho	3.76	4.19
rv1	Real	Variável aleatória 1	24.98	14.49
rv2	Real	Variável aleatória 2	24.98	14.49

Tabela 19 – Descrição da série temporal das variáveis do conjunto de dados IOT.4.

Variável	Tipo	Descrição	Média	Desvio padrão
CO	Real	Cobalto	-34.20	77.65
PT08.S1	Real	Óxido de estanho	1048.99	329.83
C6H6	Real	Hidrocarbonetos não metânicos	1.86	41.38
PT08.S2	Real	Titânio	894.59	342.33
NOx	Real	Nitrogênio totais	168.61	257.43
PT08.S3	Real	Óxido de tungstênio	794.99	321.99
NO2	Real	Dióxido de nitrogênio	58.14	126.94
PT08.S4	Real	Óxido de tungstênio	1391.47	467.21
PT08.S5	Real	Óxido de índio	975.07	456.93
T	Real	Temperatura	18.31	8.65
RH	Real	Umidade relativa	49.03	17.01
AH	Real	Umidade absoluta	1.02	0.39

Tabela 20 – Descrição da série temporal das variáveis do conjunto de dados IOT\_5.

Variável	Tipo	Descrição	Média	Desvio padrão
PM2.5	Real	Partículas atmosféricas	83.15	82.31
SO2	Real	Dióxido de enxofre	18.62	28.87
NO2	Real	Dióxido de nitrogênio	63.21	63.56
CO	Real	Monóxido de carbono	1266.88	1242.09
O3	Real	Ozônio	61.48	78.90
TEMP	Real	Temperatura	13.58	11.41
PRES	Real	Pressão	1011.84	10.40
DEWP	Real	Ponto de orvalho	3.12	13.68
RAIN	Real	Precipitação	0.06	0.90
WSPM	Real	Velocidade do vento	1.70	1.20

Tabela 21 – Descrição da série temporal das variáveis do conjunto de dados CLIMATICO\_1.

Variável	Tipo	Descrição	Média	Desvio padrão
glo_avg	Real	Radiação global	505.31	3625.30
dir_avg	Real	Radiação direta	$-2.63 \times 10^{15}$	$9.96 \times 10^{15}$
diff_avg	Real	Radiação difusa	$3.26 \times 10^{15}$	$8.72 \times 10^{15}$
lw_avg	Real	Radiação de ondas longas	$3.47 \times 10^{15}$	$7.81 \times 10^{15}$
par_avg	Real	Estimativa de iluminância e radiação global	$-2.59 \times 10^{15}$	$8.80 \times 10^{15}$
lux_avg	Real	Iluminância	$-2.96 \times 10^{15}$	$1.04 \times 10^{16}$
tp_sfc	Real	Temperatura	$3.75 \times 10^{15}$	$8.37 \times 10^{15}$
humid	Real	Humidade	$-2.44 \times 10^{15}$	$9.41 \times 10^{15}$
press	Real	Pressão	$2.52 \times 10^{15}$	$4.01 \times 10^{15}$
rain	Real	Precipitação	0.00	0.03
ws_10m	Real	Velocidade do vento	$-1.21 \times 10^{15}$	$9.78 \times 10^{15}$
wd_10m	Real	Direção do vento	$1.71 \times 10^{15}$	$8.65 \times 10^{15}$

### A.13 CLIMATICO\_1

O conjunto de dados CLIMATICO\_1 é utilizado para a previsão de radiação global e foi extraído do Sistema Nacional de Organização de Dados Ambientais (SONDA) no INPE - Instituto Nacional de Pesquisas Espaciais, Brasil. A série temporal referente a cada variável está descrita na Tabela 21 e a variável glo\_avg, usada como alvo, está apresentada na Figura 19.

### A.14 CLIMATICO\_2

O conjunto de dados CLIMATICO\_2 é utilizado para a previsão da evapotranspiração de referência. Os dados foram extraídos do POWER (Prediction Of Worldwide Energy Resources) da National Aeronautics and Space Administration (NASA) e do satélite Chirps (Climate Hazards Group InfraRed Precipitation with Stations) em três locais nas bacias hidrográficas dos rios Jequitinhonha (Lat.  $-16.1133^\circ$ ; Long.  $-42.1479^\circ$ ), Mucuri (Lat.  $-17.7031^\circ$ ; Long.  $40.7472^\circ$ ) e Doce (Lat.  $-18.8633^\circ$ ; Long.  $-41.7855^\circ$ ) no estado de Minas Gerais. A série temporal referente a cada variável está descrita na Tabela

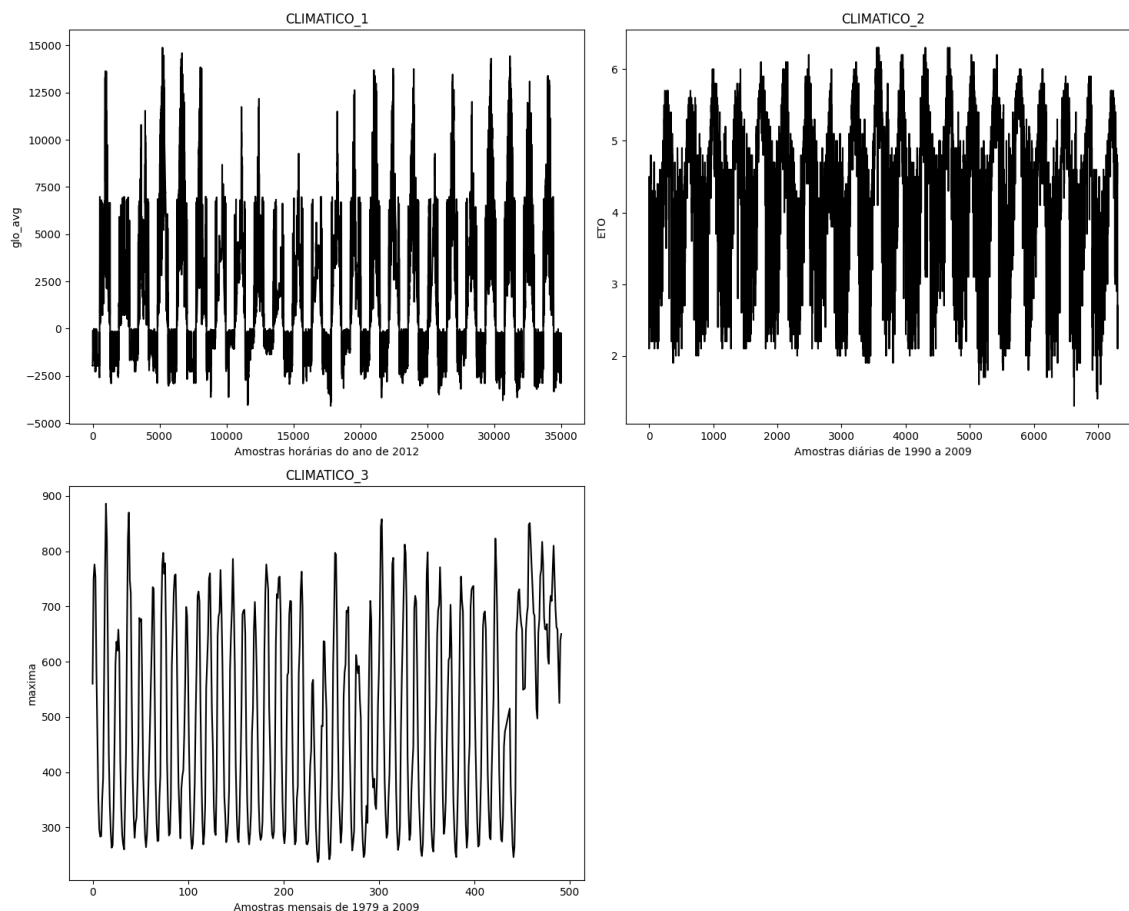


Figura 19 – Variáveis alvo das bases de dados CLIMATICO\_1, CLIMATICO\_2, CLIMATICO\_3.

Tabela 22 – Descrição da série temporal das variáveis do conjunto de dados CLIMATICO\_2.

Variável	Tipo	Descrição	Média	Desvio padrão
Rs	Real	Radiação solar	19.02	4.38
u2	Real	Velocidade do vento	1.44	0.70
Tmax	Real	Temperatura máxima	32.03	1.72
Tmin	Real	Temperatura mínima	24.09	0.82
RH	Real	Humidade realtiva	81.09	5.74
pr	Real	Precipitação	6.55	8.81
ETO	Real	Evapotranspiração de referência	4.23	1.03

22 e a variável ETO, usada como alvo, está apresentada na Figura 19.

## A.15 CLIMATICO\_3

O conjunto de dados CLIMATICO\_3 fornece um conjunto de variáveis para prever os valores máximos mensais dos níveis do Rio Xingu na área específica de Altamira. Os dados foram extraídos de (VIEIRA et al., 2021) e abrangem os anos de 1979 a 2019. A série temporal referente a cada variável está descrita na Tabela 23 e a variável maxima,

Tabela 23 – Descrição da série temporal das variáveis do conjunto de dados CLIMATICO\_3.

Variável	Tipo	Descrição	Média	Desvio padrão
máxima	Real	Nível do rio	502.39	179.71
nino1_2	Real	Temperatura da superfície do mar	23.43	2.23
nino3	Real	Temperatura da superfície do mar	26.11	28.68
nino4	Real	Temperatura da superfície do mar	28.68	0.62
nino3_4	Real	Temperatura da superfície do mar	27.19	0.93
altantico_norte	Real	Temperatura da superfície do mar	26.86	0.98
altantico_sul	Real	Temperatura da superfície do mar	24.96	1.51
atlantico_tropical	Real	Temperatura da superfície do mar	27.83	0.53
pressao_darwin	Real	Pressão	1010.17	2.69
pressao_tahiti	Real	Pressão	1012.66	1.75
prp_xingu-al01	Real	Precipitação na sub-bacia 1	138.52	128.54
prp_xingu-al02	Real	Precipitação na sub-bacia 2	137.23	126.36
prp_xingu-al03	Real	Precipitação na sub-bacia 3	135.95	124.63
prp_xingu-al04	Real	Precipitação na sub-bacia 4	143.98	127.51
prp_xingu-al05	Real	Precipitação na sub-bacia 5	155.14	125.54
prp_xingu-al06	Real	Precipitação na sub-bacia 6	165.18	120.34
prp_xingu-al07	Real	Precipitação na sub-bacia 7	149.24	113.50
prp_xingu-al08	Real	Precipitação na sub-bacia 8	149.02	108.21
prp_xingu-al09	Real	Precipitação na sub-bacia 9	159.04	128.26

usada como alvo, está apresentada na Figura 19.