



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO E
SISTEMAS (PPGEPS)

FRANCESCO SANCHES DOS SANTOS

MODELOS DE PREVISÃO DE SÉRIES TEMPORAIS APLICADOS A UM CASO
DE SISTEMA DE ABASTECIMENTO DE ÁGUA

CURITIBA

2023

FRANCESCO SANCHES DOS SANTOS

MODELOS DE PREVISÃO DE SÉRIES TEMPORAIS APLICADOS A UM CASO
DE SISTEMA DE ABASTECIMENTO DE ÁGUA

Projeto de Pesquisa de Mestrado apresentado ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas (PPGEPS). Área de concentração: Automação e Controle de Sistemas, da Escola Politécnica, da Pontifícia Universidade Católica do Paraná, como requisito parcial à obtenção do título de Mestre em Engenharia de Produção e Sistemas.

Orientador: Dr. Leandro dos Santos Coelho
Coorientadora: Dra. Viviana Cocco Mariani
(PPGEM-PUCPR)

CURITIBA

2023

FRANCESCO SANCHES DOS SANTOS

**MODELOS DE PREVISÃO DE SÉRIES TEMPORAIS APLICADOS A
UM CASO DE SISTEMA DE ABASTECIMENTO DE ÁGUA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas (PPGEPS). Área de concentração: Gerência de Produção e Logística, da Escola Politécnica, da Pontifícia Universidade Católica do Paraná, como requisito parcial à obtenção do título de Mestre em Engenharia de Produção e Sistemas.

COMISSÃO EXAMINADORA

Dr. Leandro dos Santos Coelho

Orientador

Pontifícia Universidade Católica do Paraná

Dra. Viviana Cocco Mariani (PPGEM-PUCPR)

Coorientadora

Pontifícia Universidade Católica do Paraná

Dr. Helon Vicente Hultmann Ayala

Membro Interno

Pontifícia Universidade Católica do Paraná

Curitiba, 23 de outubro de 2023

*Dedico essa dissertação de mestrado à Deus, essa força maior, que me guia e ilumina meus
pensamentos para que eu desenvolva minha luz.*

Agradecimentos

Primeiramente, expresso minha gratidão a Deus por todas as bênçãos recebidas, pois foi Ele quem abriu caminhos e me deu forças para superar esse desafio, tornando-o possível.

À minha família, sou grato pelo apoio incondicional e pelo estímulo constante para seguir em frente com determinação, buscando sempre alcançar novos patamares.

Agradeço ao professor Leandro dos Santos Coelho pela oportunidade de trabalhar ao seu lado e compartilhar seus conhecimentos e experiências ao longo do meu mestrado. Sua orientação contribuiu significativamente para o meu crescimento profissional e pessoal, tornando este trabalho uma realidade.

À professora Viviana Cocco Mariani, agradeço pela disponibilidade e paciência em me auxiliar nas minhas dificuldades, utilizando seu conhecimento para contribuir com o desenvolvimento da pesquisa.

Quero expressar minha gratidão à equipe da Pontifícia Universidade Católica do Paraná (PUCPR) e aos demais professores, especialmente à secretária Denise da Mata Medeiros (PPGEPS), pela paciência, carinho e apoio prestados em diversas ocasiões, sem medir esforços.

Aos meus amigos, que sempre torceram por mim, e aos novos amigos que conquistei ao longo dessa jornada, agradeço por compartilharmos momentos de alegria nessa batalha.

Sou grato ao investimento em bolsas de estudo concedidas pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), que possibilitou a conclusão dessa etapa da minha carreira profissional e acadêmica.

*Se vi mais longe, foi por estar de pé
sobre ombros de gigantes*

- Sir Isaac Newton

Resumo

Este estudo explora a previsão de séries temporais para a tomada de decisões relacionadas à demanda de água, visando de alguma forma auxiliar no controle eficaz dos recursos hídricos em um ambiente competitivo. Relacionado ao fornecimento de água. A abordagem proposta envolve a utilização de modelos de previsão de séries temporais para melhorar a precisão das estimativas de demanda de água. Neste estudo, o abastecimento de água pela SANEPAR (Companhia de Saneamento do Paraná) para o Bairro Alto da cidade de Curitiba é explorado. Avaliando-se modelos de redes neurais artificiais, como GRU (do inglês *Gated Recurrent Unit*), LSTM (do inglês *Long Short-Term Memory*), RNN (do inglês *Recurrent Neural Network*), *Transformer* e Facebook Prophet. Além disso, modelos do tipo ARIMA (do inglês *Auto-Regressive Integrated Moving Average*), técnicas de *boosting* como XGBoost (do inglês *eXtreme Gradient Boosting*) e LightGBM (do inglês *Light Gradient Boosting Machine*), regressão linear e RFR (do inglês *Random Forest Regression*). A eficácia dos modelos é avaliada por meio de métricas como sMAPE (do inglês *Symmetric Mean Absolute Percentage Error*), MAE (do inglês *Mean Absolute Error*) e RRMSE (do inglês *Root Relative Mean Square Error*). A análise e comparação de todos os casos, ficou evidente que o modelo RNN obteve o menor erro em todas as métricas avaliadas, incluindo-se SMAPE, MAE e RRMSE. É interessante notar que o desempenho do modelo RNN foi excepcional, com erros de previsão consistentemente menor que 1% em todas as análises.

Palavras-chave: Previsão de Séries Temporais, Abastecimento de Água, Aprendizado de Máquina, Redes Neurais Artificiais, Modelos de Previsão.

Abstract

This study explores time series forecasting for decision making related to water demand, in order to help effectively control water resources in a competitive environment. Related to water supply. The proposed approach involves the use of time series forecasting models to improve the accuracy of water demand estimates. In this study, the supply of water by SANEPAR (Companhia de Saneamento do Paraná) to the Bairro Alto neighborhood in the city of Curitiba is explored. Artificial neural network models such as GRU (Gated Recurrent Unit), LSTM (Long Short-Term Memory), RNN (Recurrent Neural Network), *Transformer* and Facebook Prophet are evaluated. In addition, ARIMA (Autoregressive Integrated Moving Average) models, boosting techniques such as XGBoost (eXtreme Gradient Boosting) and LightGBM (Light Gradient Boosting Machine), linear regression and RFR (Random Forest Regression). The effectiveness of the models is assessed using metrics such as sMAPE (Symmetric Mean Absolute Percentage Error), MAE (Mean Absolute Error) and RRMSE (Relative Root Mean Square Error). The analysis and comparison of all the cases showed that the RNN model obtained the lowest error in all the metrics evaluated, including SMAPE, MAE and RRMSE. It is interesting to note that the performance of the RNN model was exceptional, with prediction errors consistently lower than 1% in all analyses.

Keywords: Time series forecasting, Water supply, Machine learning, Artificial neural networks, Forecasting models.

Lista de Figuras

1	Paradigma de aprendizado de máquina	18
2	Mapa das Etapas	21
3	Estrutura da dissertação	25
4	Fluxograma do problema de pesquisa	26
5	Etapas da revisão	27
6	Modelos de series temporais mais populares na Scopus e WoS	28
7	Significado do modelo SARIMAX	40
8	Fluxograma da árvore de decisão	43
9	Árvore de decisão mapa mental	45
10	Esquema da floresta aleatória	46
11	Impulsionando gradiente com XGBoost e LightGBM	47
12	Compara-se o crescimento em folha com o crescimento em nível	49
13	RNN - <i>recurrent neural network</i>	52
14	Diagrama ilustrativo do funcionamento de uma unidade recorrente gated (GRU)	55
15	RNN vs LSTM vs GRU	56
16	Arquitetura do Transformer	58
17	Modelo de uma Rede Neural Artificial MLP	59
18	A equação da figura realiza o somatório ponderado entre as sinapses de cada neurônio	60
19	Modelo de uma Rede Neural Convolutacional	61
20	Comparação dos modelos AR e ARX	70
21	Modelo MA(7)	71
22	ARMA (7,7)	71
23	ARIMA (7,1,7)	72
24	SARIMA $(p, q, d)(P, Q, D) = (7, 1, 7)(2, 1, 1)_{12}$	72
25	Comparação entre ARIMAX e SARIMAX	73
26	Previsões do modelo Prophet para o reservatório LT01	73
27	Regressão linear LT01 vs PT01 correlação 98%	74
28	Regressão linear (LR) um passo a frente	74
29	Regressor de Árvore de Decisão	75
30	Regressão da Floresta Aleatória (RFR)	75
31	Resultados da regressão utilizando XGBoost e LightGBM	76
32	Correlação de Pearson	83
33	Decomposição STL aditiva dos dados coletados	85
34	Decomposição STL multiplicativa dos dados coletados	85

35	Violino no nível do reservatório	86
36	Violino da vazão de recalque	86
37	Autocorrelação	87
38	Autocorrelação parcial	88
39	Ruído branco	88
40	Comparação dos modelos ARIMA	95
41	Comparação de modelos de regressão	95
42	Análise comparativa dos modelos utilizando gráfico de barras	96

Lista de Tabelas

1	Cruzamento de palavras-chave por meio da aplicação de filtros de área . . .	29
2	Resumo dos dados	29
3	Fator de impacto	30
4	Os autores que mais publicam em relação ao tema de pesquisa	31
5	Países com maior número de publicações	32
6	Modelos baseado na literatura e nos artigos	33
7	Parâmetros utilizados nos modelos AR, ARX, MA, ARMA, ARIMA, ARI- MAX, SARIMA e SARIMAX obtidos pelo “autoARIMA” do Python. . . .	70
8	Hiperparâmetros dos modelos	76
9	Resumo dos Hiperparâmetros dos Modelos de Redes Neurais	77
10	Comparação dos modelos de previsão com as métricas de desempenho treino	78
11	Comparação dos modelos de previsão com as métricas de desempenho teste	79
12	Comparação dos modelos de previsão com as métricas de desempenho va- lidação	80
13	Comparação dos modelos de previsão com as métricas de desempenho inteiro	81
14	Descrição estatística dos dados com o filtro aplicado das 18h às 21h	84
15	Teste Nemenyi	91
16	Métricas de Avaliação dos Modelos	92
17	Comparação dos modelos Ljung Box: Modelos ARIMA com defasagem de 10 para previsão de longo prazo na demanda de água	94

Lista de Abreviaturas e Siglas

AdaBoost	Impulso ou Estímulo Adaptativo (do inglês <i>Adaptive Boosting</i>)
ANN	Rede Neural Artificial (do inglês <i>Artificial Neural Network</i>)
AR	Auto-Regressivo
ARIMA	Média Móvel Integrada Auto-Regressiva (do inglês <i>Auto-Regressive Integrated Moving Average</i>)
ARIMAX	Média Móvel Integrada Auto-Regressiva com entradas eXógenas (do inglês <i>Auto-Regressive Integrated Moving Average with eXogenous inputs</i>)
ARMA	Média Móvel Auto-Regressiva (do inglês <i>Auto-Regressive Moving Average</i>)
ARX	Auto-Regressivo com Variável Exógena (do inglês <i>Auto-Regressive with Exogenous Inputs</i>)
CNN	Rede Neural Convolucional (do inglês <i>Convolutional Neural Networks</i>)
DBN	Rede de Crenças Profundas (do inglês <i>Deep Belief Network</i>)
DTR	Regressor de Árvore de Decisão (do inglês <i>Decision tree regressor</i>)
EFB	Pacote de Características Exclusivas (do inglês <i>Exclusive Feature Bundling</i>)
FT	Flow Transmitter (Transmissor de Fluxo)
GRU	Unidade Recorrente Fechada (do inglês <i>Gated Recurrent Unit</i>)
Hz	Hertz
INMET	Instituto Nacional de Meteorologia
LGBMRegressor	Regressão da Máquina de Impulso de Gradiente Leve
Light GBM	Máquina de Impulso de Gradiente Leve (do inglês <i>Light Gradient Boosting Machine</i>)
LogitBoost	Técnicas de Regressão Logística
LPBoost	Reforço da Programação Linear (do inglês <i>Linear Programming Boosting</i>)
LR	Regressão Linear (do inglês <i>Linear Regression</i>)

LSTM	Memória de Longo Curto Prazo (do inglês <i>Long Short-Term Memory</i>)
m^3	Metros Cúbicos
m^3/h	Metros Cúbicos por Hora
MA	Média Móvel (do inglês <i>Moving Average</i>)
MadaBoost	Modificando o Sistema de Ponderação do AdaBoost
MAE	Erro Médio Absoluto (do inglês <i>Mean Absolute Error</i>)
MAPE	Erro Percentual Médio Absoluto (do inglês <i>Mean Absolute Percentage Error</i>)
mca	Metros Coluna de Água
ML	Aprendizado de Máquina (do inglês <i>Machine Learning</i>)
mm	Milímetros
MSE	Erro Médio Quadrático (do inglês <i>Mean Squared Error</i>)
PR	Estado do Paraná
RBAL	Recalque Bairro Alto
RFR	Regressão de Floresta Aleatória (do inglês <i>Random Forest Regression</i>)
RMSE	Erro de Raiz Média Quadrática (do inglês <i>Root Mean Squared Error</i>)
RNN	Rede Neural Recorrente (do inglês <i>Recurrent Neural Network</i>)
RRMSE	Raiz do Erro Médio Quadrático Relativo (do inglês <i>Root of the Relative Mean Square Error</i>)
SANEPAR	Companhia de Saneamento do Paraná
SARIMA	Auto-Regressivos Integrados de Médias Móveis com Sazonalidade (do inglês <i>Seasonal Auto-Regressive Integrated Moving Averages</i>)
SARIMAX	Média Móvel Sazonal Auto-Regressiva Integrada com Entradas Exógenas (do inglês <i>Seasonal Auto-Regressive Integrated Moving Averages with Exogenous Inputs</i>)
sMAPE	Erro Percentual Absoluto Médio Simétrico (do inglês <i>Symmetric Mean Absolute Percentage Error</i>)
SVM-VAR	Máquinas de Vetor de Suporte - Vetores Auto-Regressivos
TotalBoost	Impulso Total
Transformer	Transformador
XGBRegressor	Regressão XGBoost
XGBoost	Reforço de Gradiente Extremo (do inglês <i>eXtreme Gradient Boosting</i>)

Sumário

1	Introdução	16
1.1	Contexto da Pesquisa	17
1.1.1	Motivação da Pesquisa	19
1.2	Objetivo Geral	19
1.2.1	Objetivos Específicos	20
1.3	Procedimentos Metodológicos	20
1.3.1	Etapas da Pesquisa	20
1.4	Justificativa da Pesquisa	23
1.4.1	Contribuições	23
1.5	Estrutura da Dissertação	24
2	Revisão da Literatura	26
3	Fundamentos dos Modelos de Previsão de Séries Temporais	34
3.1	Descrição do Problema	34
3.2	Modelos de Séries Temporais Univariados	35
3.2.1	Componente Autorregressivo	35
3.2.2	Média Móvel	37
3.2.3	Modelos ARMA e ARIMA	38
3.3	Modelos de Série Temporal Multivariada	39
3.4	Modelos de Aprendizado de Máquina	40
3.4.1	Prophet	41
3.4.2	Regressão Linear (LR)	41
3.4.3	Árvore de Decisão	42
3.4.4	Floresta Aleatória	45
3.4.5	Gradient Boosting	46
3.4.6	Gradiente de Boosting (Reforço)	47
3.5	Redes Neurais Artificiais	50
3.5.1	Rede Neural Recorrente	50
3.5.2	Compreendendo Redes de Memória de Curto e Longo Prazo (LSTM)	52
3.5.3	GRU (Unidade Recorrente Fechada)	53
3.5.4	Análise dos Modelos RNN, LSTM e GRU	55
3.6	Aprendizado Profundo (DL)	56
3.6.1	Explorando o Transformer: Além dos Bits e Bytes	57
3.6.2	Rede Neural Artificial (ANN)	58
3.7	Rede Neural Convolucional (CNN)	60

3.8	Métricas de Avaliação de Modelos	62
3.8.1	Erro Quadrático Médio Raiz (RMSE)	62
3.8.2	Raiz do Erro Médio Quadrático Relativo (RRMSE)	62
3.8.3	Erro Absoluto Médio (MAE)	62
3.8.4	Erro Percentual Absoluto Médio (MAPE)	63
3.8.5	Erro Percentual Absoluto Médio Simétrico (sMAPE)	63
3.9	Teste de Ljung-Box	63
3.10	Correlação de Pearson	65
3.11	Decomposição STL	66
3.12	Dickey-Fuller (DF)	66
3.13	Teste de Significância	67
4	Resultados Preliminares	69
4.1	Análise dos Resultados dos Modelos de Previsão	69
4.1.1	Análise Exploratória dos Dados (EDA)	82
4.1.2	Múltiplas Entradas e Saída Única (MISO)	84
4.1.3	Decomposição STL	84
4.1.4	Separação dos Dados	88
4.1.5	Modelagem e Seleção do Modelo	89
4.1.6	Horizonte	89
4.1.7	Previsão e Avaliação	90
4.1.8	Teste de Significância	90
4.1.9	Comparação dos Modelos	92
5	Conclusões	97
5.1	Limitações da Pesquisa	97
5.2	Propostas Futuras	98
	Referências	99

1 Introdução

As séries temporais desempenham um papel fundamental em diversos campos do conhecimento, tais como Economia (preços diários de estoques, taxa de desemprego mensal, produção industrial) (SMITH, 2012; DOE, 2020a; DOE, 2022), Medicina (eletrocardiograma, eletroencefalograma) (DOE, 2016; DOE, 2018), Epidemiologia (número mensal de novos casos de meningite) (SMITH et al., 2015), Meteorologia (chuvas, temperatura diária, velocidade do vento) (SMITH, 2016; DOE, 2017; DOE, 2020b), entre outros. Ao longo dos anos, têm sido empregadas ferramentas computacionais para tornar a previsão em séries temporais mais eficiente, especialmente com o uso de técnicas de aprendizado de máquina e aprendizado profundo.

Existem três tipos principais de Aprendizado de Máquina: Supervisionado, Não Supervisionado e por Reforço. No Aprendizado Supervisionado, para cada exemplo apresentado ao algoritmo de aprendizado é necessário apresentar a resposta desejada (ou seja, um rótulo informando a que classe o exemplo pertence, no caso de um problema de classificação de imagens, por exemplo, como distinguir imagens de gatos e de cachorros). Cada exemplo é descrito por um vetor de valores (atributos) e pelo rótulo da classe associada. O objetivo do algoritmo é construir um classificador que possa determinar corretamente a classe de novos exemplos ainda não rotulados. Para rótulos de classe discretos, esse problema é chamado de classificação e para valores contínuos como regressão. Esse método de aprendizado é o mais utilizado (SILVA; GOMES, 2021). No Aprendizado Não Supervisionado, os exemplos são fornecidos ao algoritmo sem rótulos. O algoritmo agrupa os exemplos pelas similaridades dos seus atributos. O algoritmo analisa os exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados de alguma maneira, formando agrupamentos ou *clusters*. Após a determinação dos agrupamentos, em geral, é necessária uma análise para determinar o que cada agrupamento significa no contexto problema sendo analisado (SILVA; GOMES, 2021). No Aprendizado por Reforço, o algoritmo não recebe a resposta correta mas recebe um sinal de reforço, de recompensa ou punição. O algoritmo faz uma hipótese baseado nos exemplos e determina se essa hipótese foi boa ou ruim. Aprendizado por Reforço é bastante utilizado em jogos e robótica, e foi a técnica utilizada no AlphaGo (SILVA; GOMES, 2021). Nesse caso, os modelos mais usados nessa dissertação são os modelos de aprendizado de máquina supervisionados. Esses modelos para série temporal são usados na literatura.

A IA (Inteligencia Artificial) pode gerar impactos bons e ruins. A IA pode evitar que o ser humano se exponha a tarefas perigosas, tarefas que já podem ser realizadas por máquinas. A IA pode eliminar a necessidade de tarefas automáticas serem executadas por humanos e, com isso, sobrar tempo para que os humanos lidem com tarefas

mais instigantes e prazerosas. O uso da IA vem trazendo muitos benefícios, tais como: melhorias nos serviços de saúde; Processamento de Linguagem Natural: voz para texto, tradução; melhorias na educação; energia limpa e barata; detecção de fraudes; meios de transportes mais seguros (aplicativos de transporte), rápidos (rota otimizadas) e limpos. A IA também tem impactos negativos. Perda de vagas de trabalho é um desses impactos negativos. Vagas de empregos menos qualificados que aumentam as desigualdades sociais. Serão necessários planejamento e ações de governo para minimizar os impactos negativos da IA (SILVA; GOMES, 2021).

As séries temporais são caracterizadas como processos estocásticos regidos por leis probabilísticas. Série estacionária com média é aquela que contém a média, variância e autocorrelação constantes ao longo do tempo.

A previsão da demanda de água ao longo do tempo é essencial para um planejamento sustentável e eficiente do abastecimento hídrico para o âmbito urbano como é o caso da cidade de Curitiba no estado do Paraná. Neste estudo, adotou-se alguns modelos de previsão da área de aprendizado de máquina e modelo clássico ARIMA (do inglês *Auto-Regressive Integrated Moving Average*), para fazer previsões diárias da demanda de água ao longo do tempo.

Este capítulo apresenta modelos de aprendizado de máquina (do inglês *Machine learning*) serem aplicados na previsão de séries temporais com dados coletados pela SANEPAR (Companhia de Saneamento do Paraná). Cada modelo tem suas particularidades, mas os modelos de aprendizado de máquina têm como ser otimizados da mesma forma que os modelos clássicos do tipo ARIMA, RNN (do inglês *Recurrent Neural Network*). Os dados coletados pela SANEPAR usados para previsão referem-se ao abastecimento de água no bairro Alto durante o período de 2018 a 2019.

Em termos da análise de séries temporais e tomada de decisão, são explorados modelos para aplicação considerando a variável **LT01** que é o nível do reservatório do tanque da SANEPAR. Nesta dissertação, busca-se desenvolver previsões de séries temporais confiáveis para o abastecimento de água no Bairro Alto em Curitiba, capital do estado do Paraná. Na aplicação dos modelos de previsão, espera-se obter compreensões para auxiliar na tomada de decisões estratégicas e no planejamento eficiente do abastecimento hídrico.

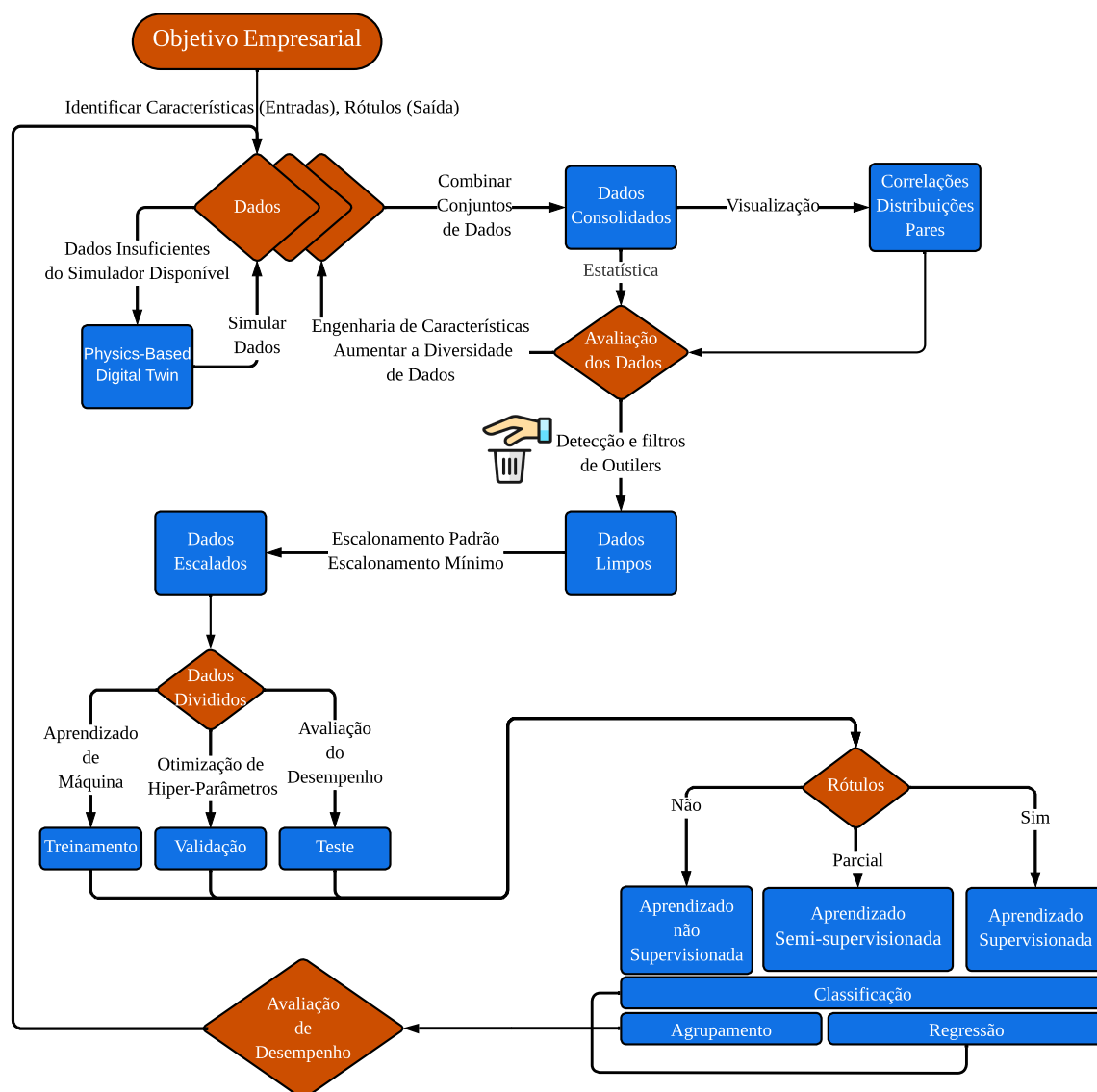
1.1 Contexto da Pesquisa

Torna a análise de séries temporais e previsões valiosas ferramentas para apoiar o processo de tomada de decisão a curto, médio e longo prazo. Devido às não linearidades, sazonalidade, tendência e que podem ocorrer em séries temporais de abastimento de água

nos dados temporais, o desenvolvimento de modelos de previsão eficientes é uma tarefa desafiadora (RIBEIRO et al., 2021).

Na Figura 1, são mostradas as etapas de como deve ocorrer a análise de dados e a seleção dos modelos. Essa seleção pode ser feita de forma que se tenha que escolher o que deve ser previsto na variável. Feito isso, temos a primeira etapa que será dos dados, depois que cada um foi identificado com seus rótulos de entrada e saída. Os dados não podem conter **NaN** (do inglês *not a number*) ou dados ausentes, o que evita falsos positivos.

Figura 1: Paradigma de aprendizado de máquina



Fonte: Adaptado de (HEDENGREN, 2023)

Fazendo isso, os dados podem ser visualizados para garantir que estejam bem carregados e que estejam em um tamanho tolerável. Isso é chamado de avaliação dos dados. Com os dados limpos e bem carregados, sem falsos positivos, a divisão dos dados

pode ser feita.

A otimização dos dados para os modelos pode ser realizada de várias formas, como o uso do Optuna para cada modelo pré-listado, reduzindo assim o tempo de processamento. Outra abordagem é aplicar a média nos dados quando eles são coletados de hora em hora, como é o nosso caso, e usá-los como conjuntos de treinamento, teste e validação.

A validação é comum em conjuntos de dados muito grandes para permitir que os modelos trabalhem mais com os dados, proporcionando resultados mais precisos. Após essa etapa, na escolha dos modelos, há a possibilidade de escolher o modelo de série temporal, se o modelo é de classificação, agrupamento ou regressão. Após listar os modelos, cada um deles deve ser avaliado em métricas para verificar a veracidade de cada um.

1.1.1 Motivação da Pesquisa

A motivação desta pesquisa é baseada na situação enfrentada por Curitiba e região metropolitana, conforme apontado por (VASCONCELOS, 2020). A região passou por um rodízio de abastecimento de água, com períodos de 36 horas com abastecimento de água seguidos por 36 horas sem abastecimento de água. A média geral dos reservatórios na região estava em torno de 27,96% de sua capacidade. Além disso, a quantidade de chuva nos anos anteriores, de 2020, foi de 1.704 mm, superando a média anual de precipitação de 1.490 mm.

Diante dessa situação, a pesquisa tem como abordagem principal a previsão do abastecimento de água, que pode ser associada a condições de seca ou decorrentes das consequências da COVID-19. A partir dos dados coletados pela SANEPAR, é possível realizar uma análise detalhada, com o objetivo de prever e evitar a ocorrência de escassez de água, que foi registrada como uma anomalia em 2020.

1.2 Objetivo Geral

O objetivo desta dissertação de mestrado é desenvolver séries temporais para que possa ser tomada a melhor decisão no problema de falta de água no Bairro Alto, na cidade de Curitiba. Com modelos de séries temporais, pode-se apoiar-se neles para tomar a melhor decisão em relação a esse problema que a cidade estava enfrentando.

Serão avaliados diversos modelos de regressão, com destaque para os modelos de redes neurais e o Prophet. É importante mencionar que serão enfatizados os modelos de *gradient boosting*, redes neurais artificiais, além do ARIMA e suas variações mais contemporâneas. Na mesma visão, foi realizadas análises de anomalias nas séries temporais avaliadas do Bairro Alto em Curitiba, buscando compreender as causas subjacentes a essas ocorrências.

1.2.1 Objetivos Específicos

O objetivo específico é comparar modelos de séries temporais e técnicas de otimização baseadas em otimização bayesiana usando o algoritmo de TPE (do inglês *Parzen Tree Estimation*). A análise se concentrará no desempenho de diferentes modelos de séries temporais em termos de precisão, eficiência e capacidade de prever padrões em conjuntos de dados específicos. Além disso, serão exploradas e implementadas estratégias de otimização baseadas em otimização bayesiana usando TPE para melhorar os hiperparâmetros desses modelos. O foco principal será identificar qual combinação de modelo de série temporal e configurações de otimização proporciona os melhores resultados, com o objetivo de aprimorar a precisão das previsões e otimizar o processo de modelagem.

1.3 Procedimentos Metodológicos

Com o intuito de realizar previsões e fazer comparações entre os modelos obtidos na revisão sistemática, será adotado um processo metodológico bem definido. Tal processo está detalhado na subseção 1.3.1 desta seção, onde foram estabelecidas as etapas a serem seguidas. Isso inclui a definição do que será previsto, bem como a seleção dos métodos a serem utilizados na Análise Exploratória de Dados (EDA).

1.3.1 Etapas da Pesquisa

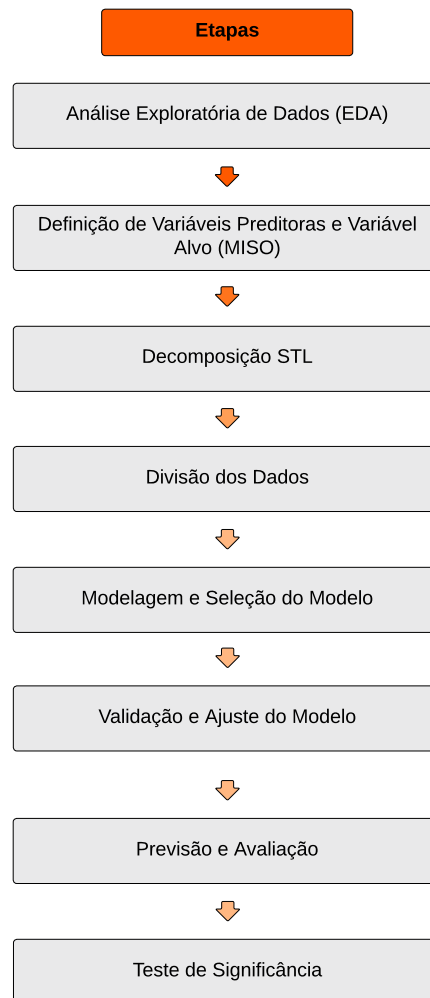
A pesquisa foi conduzida seguindo as etapas delineadas apresentado na Figura 2: As etapas da pesquisa incluem:

Etapa 1 Análise Exploratória de Dados (EDA): Nesta etapa tem-se a identificação de valores ausentes, a observação de padrões temporais e a detecção de anomalias. Gráficos de linha são comuns para visualizar a convergência dos dados (ROSTAM et al., 2021).

Etapa 2 Definição de Variáveis Predictoras e Variável Alvo (MISO): Na segunda etapa, as variáveis predictoras e a variável alvo para a previsão de Múltiplas Entradas e Uma Saída (do inglês *Multiple Inputs Single Output*, MISO) são selecionadas. Diferentes modelos, podem incorporar variáveis exógenas na modelagem. Essas variáveis exógenas aprimoram as capacidade de previsão do modelo, especialmente quando o horizonte de previsão se estende além dos dados históricos (PAWŁOWSKI et al., 2022).

Etapa 3 Decomposição STL: O método de decomposição STL (do inglês *Seasonal and Trend Decomposition Using locally estimated scatterplot smoothing (Loess)*) separa

Figura 2: Mapa das Etapas



uma série temporal em três componentes: sazonalidade, tendência e resíduo. Essa decomposição permite decompor séries temporais em sazonal para capturar variações periódicas e repetitivas. Decompor séries temporais em tendência reflete a evolução geral dos dados ao longo do tempo. Já a componente de resíduo engloba as variações não explicadas pelas anteriores (BANDARA; HYNDMAN; BERGMEIR, 2021).

Etapa 4 Divisão dos Dados: É prática comum dividir o conjunto de dados em conjuntos de treinamento, validação e teste para avaliar o desempenho do modelo. Essa divisão permite uma análise da capacidade de generalização dos modelos, evitando problemas de ajuste excessivo ou insuficiente. A proporção de alocação pode variar, mas uma abordagem é alocar 70% para treinamento e validação, e os 30% restantes para o conjunto de testes. A porção de treinamento e validação pode ser subdividida em 80% para treinamento e 20% para validação (TAO et al., 2020).

Etapa 5 Modelagem e Seleção do Modelo: Nesta etapa, diversos modelos são construídos

e avaliados. Alguns modelos comumente usados para previsão de séries temporais incluem ARX (do inglês *Auto-Regressive with Exogenous Inputs*), AR (do inglês *Auto-Regressive*), MA (do inglês *Moving Average*), ARIMA, SARIMA (do inglês *Seasonal Auto-Regressive Integrated Moving Averages*), SARIMAX (ARIMA Sazonal com variáveis exógenas) e modelos de aprendizado de máquina como RNN, LSTM (do inglês *Long Short-Term Memory*), GRU (do inglês *Gated Recurrent Unit*), Transformer (Transformador), DTR (do inglês *Decision tree regressor*), LR (do inglês *Linear Regression*), XGBoost (do inglês *eXtreme Gradient Boosting*), Light GBM (do inglês *Light Gradient Boosting Machine*) além do Prophet. A escolha do modelo baseia-se em critérios como desempenho na validação, simplicidade do modelo e interpretabilidade dos resultados.

Etapa 6 Validação e Ajuste do Modelo: Durante a construção do modelo, é importante avaliar seu desempenho usando dados de validação. Neste contexto, métricas de avaliação tais como de avaliação como MAE (Erro Médio Absoluto), sMAPE (Erro Médio Percentual Absoluto Simétrico) e RRMSE (Raiz do Erro Médio Quadrático Relativo) podem ser usadas para comparar e selecionar o melhor modelo. Além disso, técnicas de ajuste como otimização de hiperparâmetros e refinamento do modelo usando dados de treinamento e validação combinados podem melhorar o desempenho de previsão de séries temporais.

Etapa 7 Previsão e Avaliação: Com o modelo final com os dados de treinamento e validação, é possível fazer previsões para o conjunto de testes, que representa dados futuros não observados. Essas previsões são comparadas com os valores reais correspondentes para avaliar a qualidade e precisão do modelo.

Etapa 8 Teste de Significância: Aplicar os modelos de previsão e fazer comparativo baseado em testes de significância estatística (*Friedman e Nemenjy*).

O teste de Friedman é uma ferramenta estatística não paramétrica utilizada para comparar três ou mais grupos relacionados quando os dados não atendem aos pressupostos da ANOVA. Ele determina se há diferenças significativas entre os grupos. Se a diferença for confirmada, o teste de Nemenyi é frequentemente empregado para realizar comparações múltiplas entre grupos específicos, identificando quais pares são significativamente diferentes após a rejeição da hipótese nula no teste de Friedman. Assim, esses métodos são cruciais quando se deseja comparar múltiplos grupos sem fazer suposições sobre a distribuição dos dados.

Cada uma dessas etapas desempenha um papel crucial na pesquisa e no processo de modelagem de séries temporais, contribuindo para a compreensão dos dados, construção

e validação dos modelos.

1.4 Justificativa da Pesquisa

Ao longo desta dissertação, os seguintes aspectos são abordados visando a previsão e tomada de decisões.

1.4.1 Contribuições

As perguntas de pesquisa apresentadas na subseção 1.2.1, surgem duas contribuições significativas nesta dissertação. A primeira diz respeito à previsão da demanda de água na cidade de Curitiba, abordando aspectos como consumo e gasto de energia durante períodos de pico.

Segundo estudos recentes, os modelos ARIMA desempenham um papel fundamental na análise de séries temporais. Os modelos ARIMA são utilizados na previsão de séries temporais devido à sua capacidade de capturar padrões complexos e comportamentos de longo prazo (SMITH; JOHNSON, 2022).

Conforme relatos, o modelo XGBoost tem sido aplicado com sucesso em problemas de previsão de séries temporais. Estudos demonstraram que o XGBoost é uma ferramenta para lidar com desafios de previsão em séries temporais (BROWN; LEE, 2021). O LightGBM tem ganhado destaque como um modelo eficiente para previsão de séries temporais (WELLENS; UDENIO; BOUTE, 2022). De acordo com Kotsiantis (2011), o algoritmo de árvore de decisão é um dos mais populares e eficazes na área de classificação. Alguns modelos baseados em árvores possuem eficiência computacional e são capazes de lidar tanto com dados categóricos quanto numéricos. Já Anderson e Williams (2021) enfatiza a importância do uso de *Random Forest Regression* na previsão de séries temporais.

As RNNs (Redes Neurais Recorrentes) são redes neurais artificiais que possibilitam o processamento de dados sequenciais. Elas são amplamente utilizadas em mineração de dados e aprendizado de máquina devido à sua habilidade em modelar dependências temporais (LIPTON; BERKOWITZ; ELKAN, 2015). Assim como outras redes neurais artificiais, as RNNs são capazes de lidar com a detecção de anomalias de forma mais eficaz do que modelos mais básicos de séries temporais, como o modelo ARIMA e seus predecessores. Enquanto o ARIMA pode identificar anomalias nos dados, as RNNs podem ajustar e otimizar os neurônios para aumentar a eficácia na detecção dessas anomalias.

As CNNs (Redes Neurais Convolucionais) representam uma classe de redes neurais profundas amplamente utilizadas em tarefas de reconhecimento de imagens e vídeos (LECUN; BENGIO; HINTON, 2015). Embora sejam uma opção vantajosa para previ-

sões de tempo, assim como as RNNs, as CNNs são especialistas em dados bidimensionais. Isso pode resultar em erros maiores quando comparadas às RNNs em contextos de dados sequenciais.

As GRUs (Unidades Recorrentes com Portões) são um tipo de rede neural recorrente introduzido como alternativa às LSTMs (Unidades de Memória de Longo e Curto Prazo) (CHO et al., 2014). As LSTMs, por sua vez, foram desenvolvidas para resolver o problema do gradiente desvanecido em RNNs padrão (TAM, 2023). Embora sejam semelhantes às RNNs, as GRUs e LSTMs incorporam técnicas específicas para melhorar a capacidade das RNNs em lidar com dependências temporais a longo prazo. Cada uma delas possui particularidades que as tornam mais adequadas para diferentes cenários de previsão, com a GRU sendo especialmente útil para resolver problemas relacionados à dissipação de gradientes.

Constata-se que as redes GRU demonstram uma vantagem em relação às redes LSTM em termos de complexidade e desempenho. As RNN, incluindo tanto GRU quanto LSTM, são observadas como tendo uma capacidade competitiva em tarefas de aprendizagem de sequências. Portanto, os resultados sugerem que as redes GRU, pertencentes à categoria de RNN, são mais adequadas para a aprendizagem de sequências simbólicas que exigem memória seletiva e adaptativa (CAHUANTZI; CHEN; GÜTTEL, 2021).

No entanto, de acordo com as constatações dessa pesquisa, o modelo RNN especificamente o GRU é identificado como o melhor modelo para a tarefa em questão. Isso se baseia na habilidade do GRU em eficientemente memorizar e generalizar sequências, resultando em um desempenho superior em comparação a outros modelos, incluindo ANN (do inglês *Artificial Neural Network*), CNN e modelos baseados em gradientes, tais como XGBoost e LightGBM. Dessa forma, conclui-se que o modelo RNN, e mais especificamente o GRU, se destaca como a escolha mais apropriada para a aprendizagem e previsão de séries temporais.

1.5 Estrutura da Dissertação

Essa dissertação está organizada em capítulos. Cada um abordando aspectos específicos da pesquisa. O Capítulo 1, apresentou uma contextualização do estudo, destacando a motivação e os objetivos a serem alcançados.

O Capítulo 2, menciona uma visão geral das principais pesquisas e estudos relacionados às questões abordadas na pesquisa.

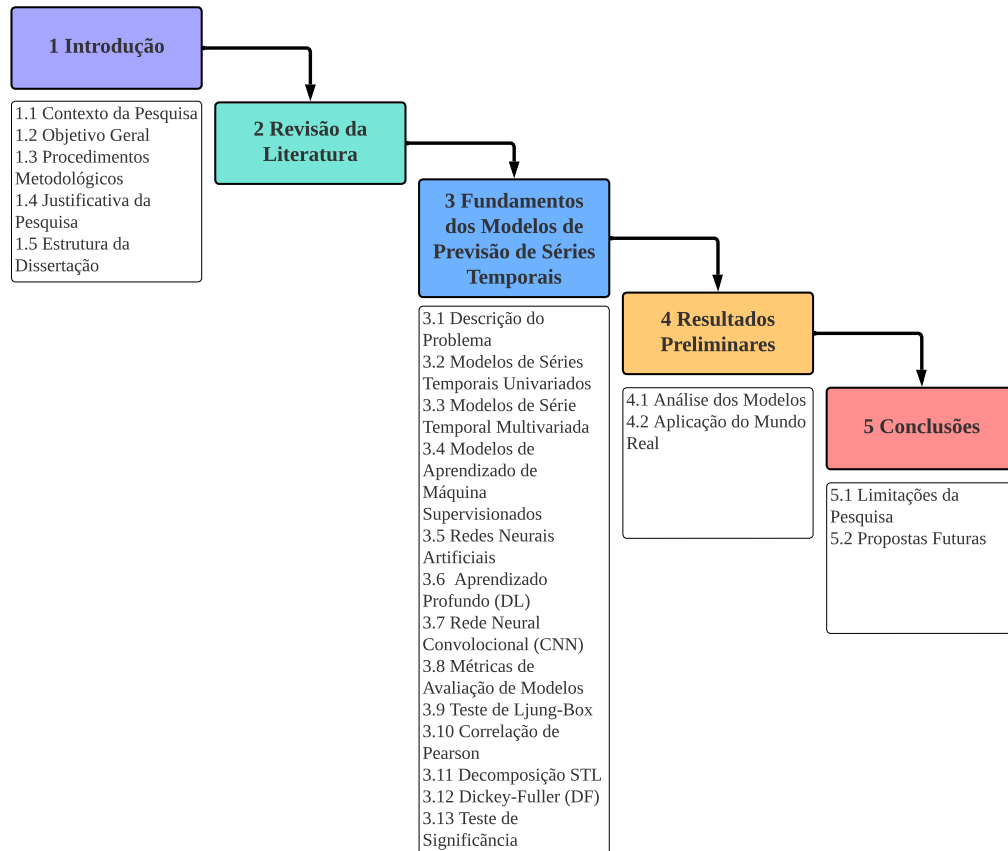
No Capítulo 3, são apresentados os modelos que serão utilização dos dados de séries temporais da SANEPAR, com os dados coletados.

O Capítulo 4, apresenta os resultados obtidos ao longo da pesquisa. Os resulta-

dos de previsão são detalhados, evidenciando análises quanto aos modelos de previsão projetados.

Finalizando, o Capítulo 5 apresenta os resultados da pesquisa e um cronograma.

Figura 3: Estrutura da dissertação



2 Revisão da Literatura

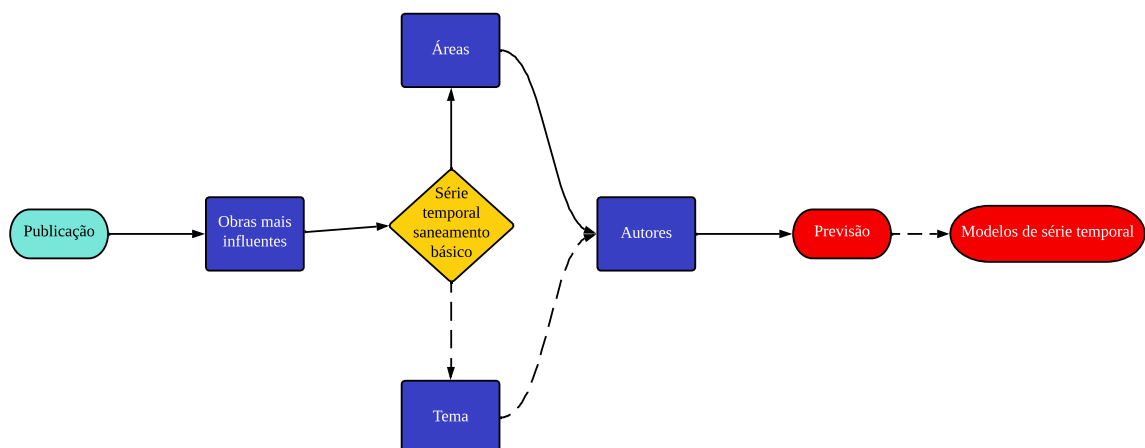
Este capítulo apresenta uma revisão sistemática da literatura nos temas relacionados a previsão de séries temporais e aplicações em hidrologia e mais especificamente em abastecimento de água. A revisão bibliográfica realizada consiste em uma análise abrangente e crítica das principais fontes de literatura. Por meio dessa revisão, busca-se obter uma compreensão aprofundada do estado atual do conhecimento na área e identificar lacunas ou oportunidades de pesquisa. As informações extraídas da literatura são fundamentais para embasar a fundamentação teórica, a metodologia e a análise dos resultados da presente dissertação.

Esta revisão sistemática da literatura (RSL) aborda o tema das séries temporais, que é relevância em diversas áreas. A seleção das referências foi baseada em critérios específicos, levando em consideração a relevância dos autores, os países com maior número de publicações e as palavras-chave mais frequentes. Também foi incluído o tema saneamento básico, que é o foco dessa dissertação.

Embora nem todos os artigos revisados tenham uma relação evidente ou mesmo acentuada aprendizado de máquina (ML), eles contribuem como material de suporte a implementação de alguns modelos avaliados para previsão este trabalho e podem servir como base para outros pesquisadores.

A Figura 4 apresenta um fluxograma de como a pesquisa foi realizada, destacando a importância dos autores como base para esta revisão da literatura.

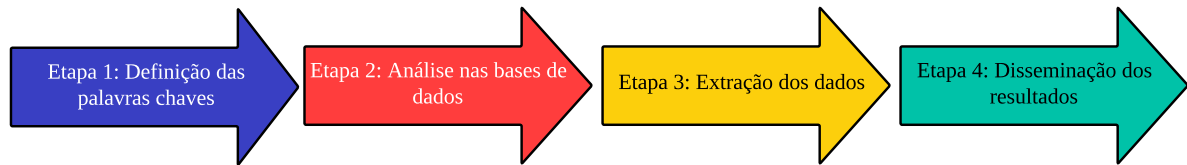
Figura 4: Fluxograma do problema de pesquisa



A Figura 5 apresenta uma adaptação da metodologia proposta por Martins e Gorschek (2016) para a realização desta RSL, foram realizadas buscas nos bancos de dados Scopus e WoS (*Web of Science*), selecionando algumas bases relevantes para o tema da

pesquisa.

Figura 5: Etapas da revisão



Fonte: Adaptado de Martins e Gorschek (2016)

Para todas as bases de busca. Foram utilizadas palavras-chave que se adequam melhor à pesquisa, como “*time series forecasting*”, “*time series analysis*”, “*sanitation*” e “*water supply*”.

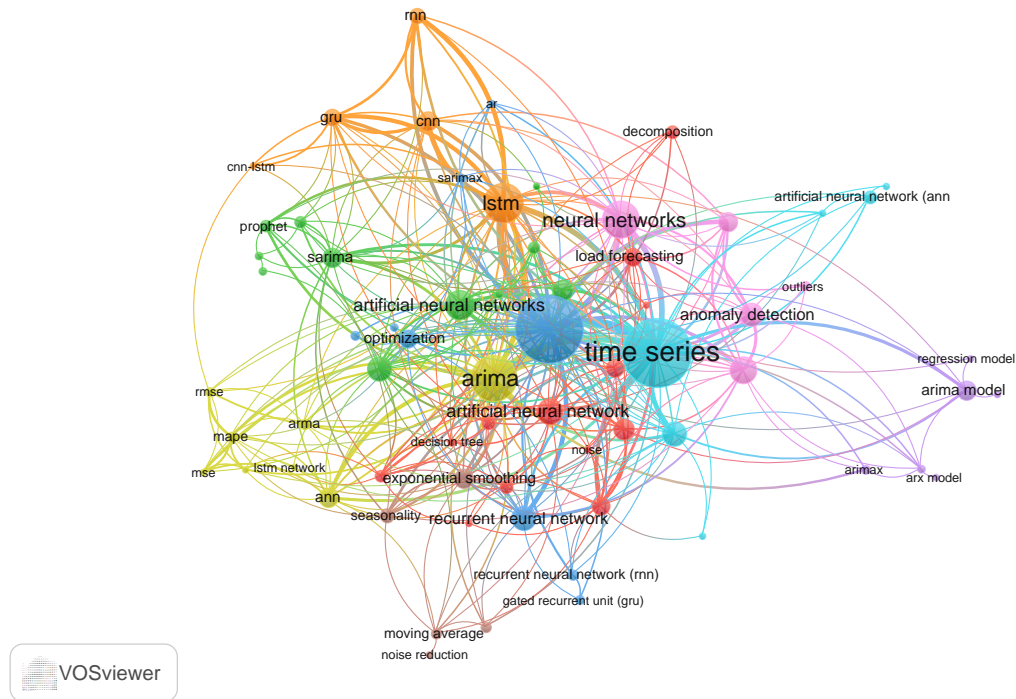
Na etapa seguinte, é realizada uma avaliação preliminar de cada artigo obtido, sem aplicar nenhum filtro anual nas buscas. Analisar todos os artigos dessa maneira resultaria em um número elevado, por exemplo, no banco de dados Scopus, existem 831 artigos, enquanto na WoS, são encontrados 98 artigos, totalizando 929 artigos sem a remoção de duplicatas. É importante ressaltar que esses artigos passaram apenas pelo filtro de idioma inglês e pela categoria de serem artigos, com o objetivo de aprimorar a busca e a tomada de decisões. Isso resulta em um número mais gerenciável de artigos para análise. Levando em consideração a diferença entre essa estimativa apresentada na Tabela 2 e a quantidade de artigos restantes após a remoção de duplicatas, tem-se menos de 929 artigos para análise. É válido lembrar que, ao remover as duplicatas, esse número pode diminuir ainda mais, chegando a 906 artigos, atingindo assim o objetivo proposto neste trabalho.

Na etapa final, é realizada uma análise aprofundada do conteúdo dos artigos selecionados, levando em consideração as áreas de especialização e correlação com séries temporais. Como esta revisão está inserida no contexto de um programa de mestrado em Engenharia de Produção e Sistemas, vale a pena analisar a correlação com áreas como Matemática. As áreas mais relevantes para a pesquisa são “**Informática**”, “**Engenharia**” e “**Matemática**”, representando 50% das publicações. Portanto, a pesquisa está alinhada com a utilização de conceitos matemáticos básicos para realizar uma estimativa do número de artigos.

São apresentados os resultados da pesquisa, utilizando um *software* para melhor aproveitamento de cada banco de dados utilizado no trabalho. Inicialmente, é realizada uma análise no software VOSviewer. A Figura 6 mostra os modelos que estão sendo usados com frequência, frequentemente utilizados como sinônimos ou em conjunto com “*time series*” nos artigos. A análise da base de dados do Scopus é feita com uma ferramenta

que exibe as palavras relacionadas em cada campo de busca, proporcionando uma visão abrangente das correlações com os modelos influentes.

Figura 6: Modelos de series temporais mais populares na Scopus e WoS



Nesse primeiro momento, foram obtidos 2.555 modelos, dos quais 83 atingiram o limite estabelecido. É importante destacar que as palavras-chave base utilizadas são “*time series forecasting*” ou “*time series analysis*” e “*water supply*” e “*sanitation*” nas bases. Esses modelos obtidos podem estar repetidos, e é por isso que resultaram em um volume significativo de modelos.

A Tabela 1 apresenta as palavras-chave utilizadas em cada base de dados, juntamente com o número de artigos encontrados inicialmente. No entanto, é importante ressaltar que esses dados ainda não foram processados para remover duplicatas. Após a utilização do software ScientoPy (RUIZ-ROSETO; RAMIREZ-GONZALEZ; VIVEROS-DELGADO, 2019) para eliminar artigos repetidos, foram selecionados 308 artigos únicos. Esses artigos representam a quantidade lida nesta RSL e são considerados relevantes para esta pesquisa.

Na Tabela 2, os dados coletados na RSL realizada no software ScientoPy Ruiz-Rosero, Ramirez-Gonzalez e Viveros-Delgado (2019) são apresentados. Nessa tabela, é exibida a quantidade de artigos coletados nas bases Scopus e WoS. Apesar de um volume considerável, nem todos os artigos foram lidos integralmente, uma vez que muitos deles não se relacionavam diretamente com o objeto de pesquisa. Consequentemente, ao longo

Tabela 1: Cruzamento de palavras-chave por meio da aplicação de filtros de área

Bases		Palavras chaves						Resultados
Scopus	time series forecasting	AND	time series analysis					798
	time series forecasting	OR	time series analysis	AND	water supply	AND	sanitation	33
WoS	time series forecasting	OR	time series analysis					79
	time series forecasting	OR	time series analysis	AND	water supply	AND	sanitation	19
Total								929

Tabela 2: Resumo dos dados

Dados Carregados	929
Artigos Omitidos	0
Total de Artigos	929
Artigos da WoS	98
Artigos do Scopus	831
Remoção de Duplicados	
Porcentagem de Duplicados Encontrados	87%
Artigos Duplicados Encontrados	23
Contagem de Artigos Original	929
Contagem de Artigos Atual	906
Porcentagem de Duplicados Removidos da WoS	19,4%
Porcentagem de Duplicados Removidos do Scopus	0,5%
Artigos Duplicados com Diferentes Citações	3
Porcentagem de Artigos Duplicados com Diferentes Citações	13%

da condução da RSL, esses artigos foram excluídos.

A Tabela 3 apresenta os periódicos que mais publicaram artigos sobre o tema em questão. Todas os periódicos listadas, incluindo aquelas com um alto fator de impacto, como a categoria **Q1**, apresentam uma correlação significativa com as áreas de **informática, engenharia e matemática**.

Essa observação ressalta a importância dessas áreas de especialização na pesquisa sobre séries temporais. Esses periódicos desempenham um papel fundamental na disseminação do conhecimento e no avanço do campo, garantindo a qualidade e o impacto dos artigos publicados. Portanto, é relevante direcionar a atenção para esses periódicos, uma vez que são reconhecidas como fontes confiáveis e respeitadas dentro da comunidade científica.

O ScientoPy encontra os principais tópicos de tendência com base na maior taxa de crescimento médio (AGR do inglês *average growth rate*). A AGR é a diferença média entre

Tabela 3: Fator de impacto

Periódicos	Quantidade de publicações	de Qualidade periódico	do <i>h-index</i>
Neurocomputing	27	Q1	143
IEEE Access	18	Q1	127
Applied Soft Computing	12	Q1	143
Energies	11	Q2	93
Energy	11	Q1	343

o número de documentos publicados em um ano e o número de documentos publicados no ano anterior (RUIZ-ROSETO; RAMIREZ-GONZALEZ; VIVEROS-DELGADO, 2019). Indica como o número de documentos publicados para um tópico cresceu (número positivo) ou diminuiu (número negativo) em média dentro de um período de tempo. Este AGR é calculado utilizando a equação (2.1):

$$\text{AGR} = \frac{\sum_{i=Y_s}^{Y_e} P_i - P_{i-1}}{(Y_e - Y_s) + 1} \quad (2.1)$$

onde AGR = taxa média de crescimento; Y_e = ano final; Y_s = ano inicial; P_i = número de publicações no ano i . Para o ano final Y_e , o ScientoPy utiliza o ano final global por defeito configurado nas opções globais ou/em parâmetros do comando ScientoPy. O ano de início Y_s é calculado a partir do ano final Y_e , conforme indicado na equação (2.2)

$$Y_s = Y_e - (\text{WindowWidth} + 1) \quad (2.2)$$

A largura da janela (do inglês *Window Width*) predefinido é de 2 anos. Assim, se o ano final for 2018, o AGR é a taxa de crescimento média entre 2017 e 2018 (RUIZ-ROSETO; RAMIREZ-GONZALEZ; VIVEROS-DELGADO, 2019).

A média de documentos por ano (ADY do inglês *average documents per year*) é um indicador absoluto que representa o número médio de documentos publicados num período de tempo para um tópico específico. O ADY é calculado utilizando a equação (2.3):

$$\text{ADY} = \frac{\sum_{i=Y_s(t)}^{Y_e(t)} P_i}{(Y_e(t) - Y_s(t)) + 1} \quad (2.3)$$

onde ADY é a média de documentos por ano; $Y_e(t)$ é o ano final; $Y_s(t)$ é o ano inicial,

calculado como descrito na equação (2.3); P_i é o número de publicações no ano i .

A percentagem de documentos nos últimos anos (PDLY do inglês *Percentage of documents in last years*) é um indicador relativo que representa a percentagem do ADY em relação ao número total de documentos para um tópico específico. Desta forma, o PDLY é calculado utilizando a equação (2.4):

$$\text{PDLY} = \frac{\sum_{i=Y_s(t)}^{Y_e(t)} P_i}{(Y_e(t) - Y_s(t) + 1) \cdot \text{TND}} \cdot 100\% \quad (2.4)$$

onde PDLY é a percentagem de documentos nos últimos anos; $Y_e(t)$ é o ano final; Y_s é o ano inicial, calculado como descrito na equação (2.4); P_i é número de publicações no ano i ; TND é o número total de documentos.

Tabela 4 para visualizar de forma mais clara os autores publicou sobre o tema em análise. Essa abordagem visa evitar a inclusão de todos os autores e destacar aquele que teve uma contribuição significativa no campo. Dessa forma, é possível identificar o principal autor que se destaca nesse tópico específico, fornecendo uma visão geral da distribuição da produção científica entre os pesquisadores.

Na Tabela 4 apresenta a taxa de crescimento médio (AGR), documentos médios por ano (ADY) e percentagem de documentos nos últimos anos (PDLY) período: 2021 - 2023.

A Tabela 5, que apresenta os países com maior número de publicações sobre o tema de saneamento básico, ordenados de forma decrescente. Os principais países que se destacam nessa análise são os seguintes: China, com 179 publicações, Estados Unidos da América com 74 publicações, Índia com 61 publicações, Brasil com 49 publicações, Espanha com 40 publicações, Reino Unido com 40 publicações, Austrália com 31 publicações, Itália com 26 publicações, Canadá com 25, Irã com 20 publicações.

Tabela 4: Os autores que mais publicam em relação ao tema de pesquisa

Pos	Author	Total	AGR	ADY	PDLY	<i>h-index</i>
1	Wang et al. (2016a)	11	-0.5	2.0	36.4	8
2	Shen e Wang (2022)	11	0.0	3.0	54.5	5
3	Xian et al. (2018)	10	1.0	2.5	50.0	5
4	Li et al. (2018)	9	-1.5	2.0	44.4	4
5	Sang et al. (2016)	7	1.5	2.0	57.1	3
6	Sadaei et al. (2019a)	7	1.0	2.0	57.1	3
7	Hao et al. (2023)	7	1.0	3.0	85.7	2
8	Guo, Pedrycz e Liu (2018)	7	1.5	3.0	85.7	3
9	O'Donncha et al. (2022)	6	0.0	1.5	50.0	4
10	Xu et al. (2019b)	6	0.0	1.5	50.0	5

Tabela 5: Países com maior número de publicações

Pos	País	Total	AGR	ADY	PDLY	<i>h-index</i>
1	China	179	18.5	48.0	53.6	31
2	Estados Unidos da América	74	3.0	16.0	43.2	21
3	Índia	61	0.0	12.0	39.3	18
4	Brasil	49	3.5	12.5	51.0	17
5	Espanha	40	1.5	8.5	42.5	12
6	Reino Unido	40	3.0	10.0	50.0	15
7	Austrália	31	3.5	7.5	48.4	14
8	Itália	26	2.0	7.0	53.8	10
9	Canadá	25	1.0	5.5	44.0	11
10	Irã	20	-1.0	3.5	35.0	11

Foi realizada uma investigação dos artigos na RSL. Esses artigos retratam alguns dos métodos utilizados em Taieb e Atiya (2016), Ursu e Pereau (2016), Wang et al. (2016b), Graff et al. (2017), Tyralis e Papacharalampous (2017a), Boroojeni et al. (2017), Coelho et al. (2017), Chou e Tran (2018), Bergmeir, Hyndman e Koo (2018), Rossi (2018), Ahmad et al. (2018), Chou e Nguyen (2018), Chen et al. (2018), Sadaei et al. (2019b), Yang et al. (2019), Buyuksahin e Ertekin (2019a), Carvalho Jr. e Costa Jr. (2019), Liu e Chen (2019), Shih, Sun e Lee (2019a), Moon et al. (2019), Yang et al. (2019), Xu et al. (2019a), Golyandina (2020), Martinović, Hunjet e Turcin (2020), Salgotra, Gandomi e Gandomi (2020), Vlachas et al. (2020), Kulshreshtha e Vijayalakshmi (2020a), Samanta et al. (2020), Shen et al. (2020), Sezer, Gudelek e Ozbayoglu (2020a), Du et al. (2020), Li e Bastos (2020), Kumar, Jain e Singh (2021), Lara-Benitez, Carranza-Garcia e Riquelme (2021), Tan et al. (2021), Liu et al. (2021). Esses artigos abordam diferentes métodos usados pelos autores para previsão de séries temporais e análise não-linear dessas previsões. Eles representam contribuições significativas para o avanço do conhecimento e aplicação prática das séries temporais, sobre abordagens eficazes nesse campo. Ao incluir esses estudos influentes na análise, obtém-se uma visão abrangente dos métodos e técnicas relevantes na previsão de séries temporais.

No estudo conduzido por Xu et al. (2019a), um modelo híbrido foi proposto, combinando o modelo linear AR e LR com o modelo não-linear ARIMA e o modelo DBN (do inglês *Dynamic Bayesian Network*). Essa abordagem permitiu capturar tanto os comportamentos lineares quanto os não-lineares de uma série temporal. Por outro lado, Li e Bastos (2020) comparou o desempenho de previsão da abordagem MAELS (Modelo Alternativo de Estação Livre Série Temporal) com outros modelos de aprendizado de máquina de última geração, como ANN, CNN, RNN, LSTM, GRU, Transformer, Prophet, ARIMA e SVM-VAR (do inglês *Support Vector Machine Variable Regression*). As abordagens ANN, CNN, RNN, GRU, Transformer e LSTM são capazes de lidar com dados

multivariados de entrada e saída, enquanto o ARIMA utiliza informações passadas para prever o futuro com base em características como autocorrelação e médias móveis. Na Tabela 6 é mostrado quantos artigos são relacionados em cada modelo que é utilizado neste trabalho e um artigo de cada modelo.

Tabela 6: Modelos baseado na literatura e nos artigos

Pos	Palavras-chave	Total	AGR	ADY	PDLY	<i>h-index</i>
1	ARIMA, Buyuksahin e Ertekin (2019b)	84	1.7	16.7	59.5	27
2	ANN, Fouilloy et al. (2018)	36	0.7	9.0	75.0	17
3	LSTM, Sezer, Gudelek e Ozbayoglu (2020b)	35	3.3	10.7	91.4	16
4	RNN, Shih, Sun e Lee (2019b)	20	0.0	4.3	65.0	11
5	Árvores de Decisão, Fouilloy et al. (2018)	12	0.7	3.0	75.0	7
6	Transformer, Peimankar et al. (2018)	10	2.3	3.0	90.0	5
7	Random Forest, Yang, Guo e Li (2022)	9	1.7	2.7	88.9	5
8	CNN, Rostamian e O'Hara (2022)	8	1.3	2.7	100.0	4
9	ARMA, Tyrallis e Papacharalampous (2017b)	7	0.3	0.7	28.6	6
10	GRU, Yang, Guo e Li (2022)	5	0.0	1.3	80.0	4
11	SARIMA, Kushwah e Wadhvani (2022)	5	1.0	1.7	100.0	4
12	ARX, Gustin, McLeod e Lomas (2018)	3	0.0	0.7	66.7	2
13	LR, Mohan et al. (2022)	3	0.0	0.7	66.7	3
14	Prophet, Kulshreshtha e Vijayalakshmi (2020b)	3	0.3	1.0	100.0	3
15	MAPE, Gupta, Singh e Jain (2020)	2	0.0	0.7	100.0	1
16	MSE, Aijaz e Agarwal (2020)	2	0.0	0.3	50.0	2
17	SARIMAX, Bhangu, Sandhu e Sapra (2022)	2	0.3	0.7	100.0	2
18	MAE, Sholtanyuk (2020)	1	0.0	0.3	100.0	1
19	XGBoost, Liu et al. (2022)	1	0.3	0.3	100.0	0

Dessa forma, por meio dessa revisão sistemática e análise de conteúdo. Além desses modelos mencionados, também será utilizada a versão atualizada do ARIMA nesta dissertação. Os modelos SARIMA e SARIMAX também serão comparados para determinar qual deles é o mais adequado. Além disso, serão empregados os modelos Light GBM e XGBoost. Os modelos de aprendizado profundo, como a RNN, ainda são considerados os melhores modelos para séries temporais no tema de saneamento básico que está sendo abordado. Quanto a modelos, tais como RNN, CNN, ANN, LSTM, Transformer, GRU, Light GBM, XGBoost, RFR, DTR e LR, não fossem encontrados na literatura relacionados a saneamento básico. Embora existam várias ramificações do modelo ARIMA, o modelo desenvolvido pelo Facebook, conhecido como Prophet, sobressai como uma opção superior em comparação com os demais. O Prophet é um modelo mais recente que simplifica significativamente muitas das tarefas que são necessárias ao lidar com o ARIMA. Enquanto o Prophet foi criado em 2017, o modelo ARIMA tem relatos de ter sido desenvolvido na década de 1960. Essa diferença temporal destaca a evolução e a modernização do campo de modelagem de séries temporais ao longo das décadas (RAMOS, 2010).

3 Fundamentos dos Modelos de Previsão de Séries Temporais

Neste capítulo, são abordados diversos aspectos relacionados a previsão de séries temporais métricas de erro, modelos de previsão e a descrição do modelo. Compreender e dominar esses conceitos é essencial para se obter resultados confiáveis e embasar as próximas etapas do trabalho de pesquisa.

3.1 Descrição do Problema

A descrição do problema, com foco no abastecimento de água, é essencial. Ela apresenta variáveis-chave, incluindo LT01, e estabelece claramente o objetivo da previsão. Sem essa clareza, o uso de modelos de previsão é difícil de justificar. Definir metas de previsão antes de aplicar modelos é fundamental.

- Bombas de sucção (B1, B2 e B3) – valor máximo da frequência 60 Hz
- Nível do Reservatório (Câmara 1) LT01 (m^3)
- Vazão de entrada (FT01) (m^3/h)
- Vazão de gravidade (FT02) (m^3/h)
- Vazão de recalque (FT03) (m^3/h)
- Pressão de Sucção (PT01SU) (mca)
- Pressão de Recalque (PT02RBAL) (mca)

A pesquisa fará uso da variável LT01, que representa o nível do reservatório e desempenha um papel de extrema importância. A separação dos dados foi feita por hora a hora, mesmo que os dados que obteve da SANEPAR foi de 2018 à 2020 sendo 2020 causando muitas irregularidade é de um forma que tem como retirar esse ano para que assim os dados seja melhor trabalhado.

Mesmo havendo 9 variáveis nesse conjunto de dados, poderia-se trabalhar com 1 para previsão e as outras 8 como variáveis exógenas. No entanto, todas as variáveis podem ter correlação com o tanque, mas nem todas são necessárias, causando assim ruído na série temporal. Com isso em mente, foram retiradas as variáveis B3, FT02, FT03 e PT02, restando assim as variáveis de previsão com as variáveis que tiveram correlação significativa.

3.2 Modelos de Séries Temporais Univariados

A previsão de séries temporais é um desafio complexo, sem uma resposta fácil. As séries temporais constituem na atualidade um tipo de estrutura de dados muito comum e frequente em diversas áreas da ciência, e sua análise continua sendo um grande desafio. As respostas às perguntas como: A bolsa de valores vai subir? Amanhã vai chover? Pode-se saber quando ocorrerá uma enchente? O valor do petróleo vai subir?, podem significar: redução de custo, maior lucro, menor prejuízo, melhor planejamento, melhoria de processos tornando seu estudo um grande atrativo. As séries temporais são conjuntos de observações de algum fenômeno, ordenados no tempo. Estas podem ser consideradas contínuas quando a coleta dos dados é realizada continuamente no tempo, e discretas quando a coleta desses dados é realizada em tempos específicos, geralmente equiespaçados. Além disso, a variável observada pode assumir valores discretos ou contínuos (MARTINHO, 2023).

As abordagens de previsão de séries temporais existentes na literatura podem ser organizadas conforme (MARTINHO, 2023): técnicas Descritivas: consistem em analisar uma ou mais séries temporais através da representação e análise gráfica dos dados sequencialmente ao longo do tempo, o que pode revelar padrões de comportamento importantes: a tendência de crescimento (ou decrescimento), padrões cíclicos, alterações estruturais, observações aberrantes, entre outros; modelos lineares: incluem modelos probabilísticos, análise espectral, métodos não-paramétricos (alisamento ou suavização), modelos de espaço de estados, séries multivariadas, estudos longitudinais e processos de longa dependência; modelos não-lineares: englobam modelos não-lineares gerais (redes neurais artificiais, sistemas nebulosos, filtro de Kalman estendido, modelos híbridos), modelos pré-definidos, modelos com volatilidade variável, entre outros.

Os clássicos modelos do tipo ARIMA são compostos de três componentes: AR (Auto-Regressão), I (Integração) e MA (Média Móvel). O componente AR leva em consideração os valores anteriores da série temporal, o componente I trata das diferenças entre os valores observados para tornar a série estacionária, e o componente MA considera os erros residuais do modelo. Esses componentes combinados tem por meta capturar os padrões e tendências presentes na série temporal.

3.2.1 Componente Autorregressivo

O componente auto-regressivo do modelo ARIMA é representado por $AR(p)$, em que o parâmetro p determina o número de defasagens ou atrasos (do inglês *lags*) a serem usados..

A equação do modelo $AR(p)$ é expressa da seguinte forma:

$$Y_t = c + \sum_{n=1}^p \alpha_n Y_{t-n} + \varepsilon_t \quad (3.1)$$

na equação (3.1), o termo ε_t representa o ruído branco que é caracterizado por um sinal com média zero e variância sigma. Essa equação pode ser entendida como uma regressão múltipla, em que os valores defasados de y_t são utilizados como preditores. Esse modelo é conhecido como modelo autorregressivo de ordem p , ou $AR(p)$.

O modelo ARX é uma extensão do modelo AR, que incorpora variáveis exógenas nos dados para tentar melhorar as previsões. Esse modelo também é multivariado, e foi incluído aqui para fins de comparação com o modelo AR simples, considerando a presença de variáveis exógenas.

Pode-se mencionar que de acordo com o valor de p tem-se alguns aspectos relevantes a citar: Se o parâmetro p for definido como zero ($AR(0)$), significa que não há termos autorregressivos no modelo. Nesse caso, a série temporal se comporta como um ruído branco.

AR(1): Caminhadas aleatórias e Oscilações: Com o parâmetro p definido como 1, o modelo AR leva em consideração o valor anterior da série temporal multiplicado por um coeficiente e, em seguida, adiciona ruído branco. Quando o coeficiente é igual a 0, há apenas ruído branco, resultando em uma série de tempo completamente aleatória, sem padrões previsíveis.

Quando o coeficiente é igual a 1, ocorre uma caminhada aleatória, onde cada valor da série é obtido somando-se o valor anterior a um termo de ruído branco. Nesse caso, os valores da série apresentam uma tendência linear, aumentando ou diminuindo ao longo do tempo sem retornar à média.

Se o coeficiente estiver na faixa $0 < \alpha < 1$, ocorre o fenômeno de reversão média. Isso significa que os valores da série tendem a oscilar em torno de uma média central e a regressar em direção a ela após se afastarem. Esse padrão indica uma tendência de retorno à média ao longo do tempo.

AR(p): Termos de ordem superior: Aumentar ainda mais o parâmetro p no modelo AR significa considerar um número crescente de medições de tempo anteriores, cada uma multiplicada pelo seu próprio coeficiente. Isso permite levar em conta uma memória mais longa da série temporal e capturar padrões de dependência complexos ao longo do tempo.

No entanto, é importante ter em mente que aumentar excessivamente o valor de p pode levar a problemas de *overfitting*, onde o modelo se ajusta muito bem aos dados de treinamento, mas tem um desempenho ruim na previsão de novos dados. Portanto, é

necessário encontrar um equilíbrio entre a complexidade do modelo e sua capacidade de generalização.

Além disso, é comum combinar o modelo AR com o modelo de média móvel (MA) para formar o modelo ARMA. O modelo MA considera os erros passados, ou seja, as diferenças entre os valores reais e as previsões anteriores, ajustadas por coeficientes. A combinação dos componentes AR e MA permite capturar tanto a dependência autorregressiva quanto a dependência na média móvel, proporcionando uma modelagem abrangente da série temporal.

Em suma, aumentar o parâmetro p no modelo AR pode melhorar a capacidade do modelo de capturar padrões complexos da série temporal, mas é necessário ter cuidado para evitar *overfitting*. A combinação com o modelo MA pode fornecer uma modelagem mais completa dos dados. A escolha adequada dos parâmetros depende da análise cuidadosa dos padrões presentes na série temporal e do equilíbrio entre a complexidade do modelo e sua capacidade de generalização.

3.2.2 Média Móvel

No modelo de média móvel (MA), o componente não é uma média móvel simples, mas sim uma combinação de termos de erro de previsão defasados. O parâmetro q no modelo MA representa o número de termos de erro de previsão que são levados em consideração na previsão.

Este componente não é uma média móvel, mas sim os atrasos no ruído branco (TRENBERTH, 1984). Em um modelo MA(1), por exemplo, a previsão é composta por um termo constante, o produto do termo de erro de previsão anterior por um multiplicador, e o termo de erro de previsão atual. Essa abordagem baseia-se em princípios estatísticos e de probabilidade, ajustando a previsão com base em termos anteriores de erro de previsão.

O modelo MA é uma alternativa ao modelo AR e é usado para capturar padrões de dependência na média móvel, ou seja, a influência de erros passados na previsão atual. Ao combinar o modelo AR e o modelo MA, como no modelo ARMA, é possível obter uma modelagem mais abrangente que considera tanto a dependência autorregressiva quanto a dependência na média móvel (VIDHYA, 2023), tal que

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} \quad (3.2)$$

onde ε_t representa o ruído branco, esse modelo é conhecido como um modelo de média móvel MA(q), em que q é a ordem da média móvel. É importante ressaltar que não se

observam diretamente os valores de ε_t , portanto, essa modelagem não se trata de uma regressão no sentido convencional.

Diferentemente de uma regressão comum em que se têm variáveis explicativas observadas, no modelo $MA(q)$, são usados os termos de ruído branco defasados para estimar e prever os valores da série temporal. O objetivo é capturar a dependência dos termos de erro passados na previsão atual (VIDHYA, 2023).

Esse modelo é útil para modelar séries temporais em que a média móvel tem um impacto significativo nas observações. Ao ajustar a série temporal com base nos termos de ruído branco defasados, pode-se obter uma estimativa dos valores futuros.

Embora o modelo $MA(q)$ seja diferente de uma regressão tradicional, é uma ferramenta estatística para modelar e prever séries temporais, levando em consideração a dependência entre os termos de erro passados.

3.2.3 Modelos ARMA e ARIMA

O modelo ARMA é uma combinação dos modelos AR e MA, onde o modelo AR é adicionado ao modelo MA. No modelo ARMA, é adicionada uma constante à soma dos termos autorregressivos multiplicados pelos seus coeficientes, juntamente com a soma dos termos de média móvel multiplicados pelos seus coeficientes, além do ruído branco. Essa estrutura é amplamente utilizada em diversos modelos de previsão em diferentes áreas. Esse modelo é bastante semelhante ao modelo ARIMA, pois calcula os termos, mas não inclui a diferenciação presente tanto no modelo ARMA quanto no modelo ARIMA, tal que

$$Y_t = \beta_2 + \omega_1 \varepsilon_{t-1} + \omega_2 \varepsilon_{t-2} + \dots + \omega_q \varepsilon_{t-q} + \varepsilon_t \quad (3.3)$$

onde Y_t representa a série temporal que foi diferenciada (possivelmente mais de uma vez). Os “preditores” no lado direito da equação incluem os valores defasados de Y_t e os erros defasados. Esse tipo de modelo é conhecido como ARIMA (p, d, q) .

O modelo ARIMA é uma extensão do modelo ARMA que incorpora uma etapa adicional de pré-processamento chamada de diferenciação. Essa etapa é representada pela notação $I(d)$, em que d denota a ordem de diferenciação, ou seja, o número de transformações necessárias para tornar a série temporal estacionária. Portanto, um modelo ARIMA é simplesmente um modelo ARMA aplicado à série temporal diferenciada. Isso permite lidar com séries temporais que possuem tendências ou padrões não estacionários.

Embora os modelos ARIMA sejam eficazes, incorporar variáveis sazonais e exógenas ao modelo pode potencializar sua capacidade de previsão. No entanto, é importante

destacar que o modelo ARIMA pressupõe que a série temporal seja estacionária. Quando lidamos com séries temporais não estacionárias, é necessário recorrer a outros modelos para a análise e previsão adequadas (VIDHYA, 2023). Um exemplo é o do modelo SARIMA gerado por

$$Y_t = c + \sum_{n=1}^p \alpha_n y_{t-n} + \sum_{n=1}^q \theta_n \epsilon_{t-n} + \sum_{n=1}^P \phi_n y_{t-sn} + \sum_{n=1}^Q \eta_n \epsilon_{t-sn} + \epsilon_t \quad (3.4)$$

O modelo proposto é uma extensão do modelo ARIMA, com a adição de componentes autorregressivos e de média móvel sazonal. Esses componentes extras são ajustados levando em consideração os padrões sazonais presentes nos dados, utilizando atrasos correspondentes à frequência sazonal (por exemplo, 12 para dados mensais). Essa abordagem permite capturar e modelar de forma mais precisa as variações sazonais e melhorar a qualidade das previsões em séries temporais com esse comportamento cíclico (PREDUM, 2021).

3.3 Modelos de Série Temporal Multivariada

Os modelos de série temporal multivariada são uma abordagem estatística utilizada para analisar e prever dados que possuem múltiplas variáveis dependentes ao longo do tempo. Nesse tipo de modelo, considera-se a interdependência entre as diferentes séries temporais, permitindo a análise conjunta e a identificação de padrões e relações entre as variáveis.

ARIMAX e SARIMAX: Nesse modelo, são consideradas variáveis exógenas, ou seja, são utilizados dados externos para a realização das previsões. É importante ressaltar que mesmo que essas variáveis exógenas sejam indiretamente modeladas no histórico de previsões do modelo, ao incluí-las diretamente, o modelo será capaz de responder de forma ágil aos efeitos dessas variáveis (PREDUM, 2021).

$$d_t = c + \sum_{n=1}^p \alpha_n d_{t-n} + \sum_{n=1}^q \theta_n \epsilon_{t-n} + \sum_{n=1}^r \beta_n x_{n_t} + \sum_{n=1}^P \phi_n d_{t-sn} + \sum_{n=1}^Q \eta_n \epsilon_{t-sn} + \epsilon_t \quad (3.5)$$

p : Ordem de autorregressão de tendência (ACF)- p é o número de termos autorregressivos (parte AR). Permite incorporar o efeito de valores passados em nosso modelo. Intuitivamente, isso seria semelhante a afirmar que é provável que esteja quente amanhã se tiver sido quente nos últimos 3 dias.

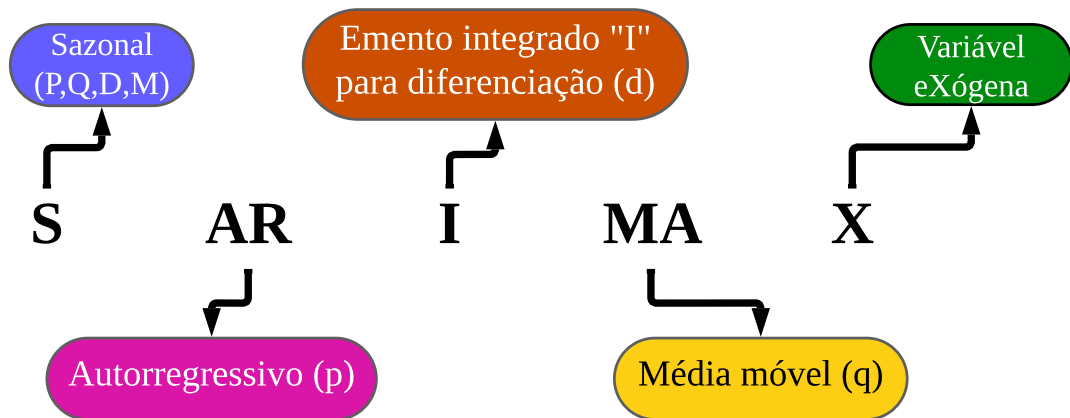
d : Diferença de tendência ordem- d é o número de diferenças não sazonais necessá-

rias para estacionariedade. Intuitivamente, isso seria semelhante a afirmar que é provável que seja a mesma temperatura amanhã se a diferença de temperatura nos últimos três dias tiver sido muito pequena.

q : Ordem da média móvel de tendência. (PCAF)- q é o número de erros de previsão defasados na equação de previsão (parte MA). Isso nos permite definir o erro do nosso modelo como uma combinação linear dos valores de erro observados em momentos anteriores no passado.

Elementos sazonais em SARIMAX: P: Ordem autorregressiva sazonal, D: Ordem das diferenças sazonais, P: Ordem de média móvel sazonal, M: O número de etapas de tempo para um único período sazonal. M é igual à defasagem ACF com o valor mais alto (normalmente em uma defasagem alta). $D = 1$ se a série tiver um padrão sazonal estável ao longo do tempo, $D = 0$ se a série tiver um padrão sazonal instável ao longo do tempo, $P \geq 1$ se a FAC for positiva na defasagem S, senão $P = 0$, $Q \geq 1$ se a ACF for negativa na defasagem S, caso contrário $Q = 0$, X variável exógena. Na Figura 7 é mostrado como o modelo SARIMAX se comporta.

Figura 7: Significado do modelo SARIMAX



3.4 Modelos de Aprendizado de Máquina

Os modelos para séries temporais têm sido amplamente reconhecidos e utilizados na literatura atual, especialmente aqueles baseados em métodos de gradiente.

Esses modelos são valorizados por sua capacidade de capturar relações complexas e não lineares nos dados, permitindo previsões mais precisas e eficientes. Sua popularidade reflete o reconhecimento da eficácia desses modelos em abordar uma ampla gama de problemas de previsão de séries temporais em diferentes áreas de estudo (AL-SHABI, 2021; SEN et al., 2022; KHEIRI; KARIMI, 2023).

A seguir são mencionados alguns dos modelos avaliados nessa dissertação.

3.4.1 Prophet

O Prophet é um modelo de previsão de séries temporais desenvolvido pelo Facebook. Foi projetado para simplificar a previsão de séries temporais que apresentam padrões sazonais, tendências e feriados. O Prophet é útil para usuários que desejam realizar previsões precisas sem requerer um profundo conhecimento em estatística ou aprendizado de máquina.

O modelo se baseia em uma abordagem aditiva que desagrega a série temporal em vários componentes individuais, como tendência de longo prazo, sazonalidade semanal e anual, e efeitos de feriados. Esses componentes são combinados para formar uma previsão geral. A equação básica do modelo Prophet pode ser representada da seguinte forma:

$$p(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad (3.6)$$

onde, $p(t)$ é o valor da série temporal no tempo t , que se deseja prever; $g(t)$ representa a tendência de longo prazo da série; $s(t)$ representa os componentes sazonais, que podem incluir padrões semanais e anuais; $h(t)$ é a representação dos efeitos de feriados ou eventos especiais.

O modelo Prophet ajusta esses componentes aos dados históricos de séries temporais para criar uma previsão futura. Ele utiliza um procedimento de ajuste automático para estimar os parâmetros desses componentes com base nos dados fornecidos. A abordagem aditiva do Prophet permite que os padrões sazonais, tendências e feriados sejam capturados separadamente e, em seguida, somados para gerar uma previsão global (KULSHRESHTHA; VIJAYALAKSHMI, 2020b).

Lembrando que essa é uma perspectiva simplificada da equação do Prophet. O modelo em sua totalidade incorpora uma gama de ajustes e considerações destinados a aprimorar a precisão das previsões, incluindo o tratamento de incertezas, a seleção automática de sazonalidades relevantes e outras otimizações.

3.4.2 Regressão Linear (LR)

A regressão linear é definida da seguinte forma:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon \quad (3.7)$$

onde há p variáveis explicativas, denotadas por x . Existe uma variável alvo, denotada por

y . O valor de y é calculado como uma constante β_0 , somada aos valores das variáveis x multiplicados por seus coeficientes β_1 a β_p .

Para utilizar a regressão linear, é necessário estimar os coeficientes (betas) com base em um conjunto de dados de treinamento. Esses coeficientes podem ser estimados por meio da seguinte fórmula, expressa em notação matricial:

$$\hat{\beta} = (X^T X)^{-1} X^T y \quad (3.8)$$

onde, $\hat{\beta}$ é um vetor de coeficientes estimados que minimiza a soma dos quadrados dos resíduos no método de mínimos quadrados ordinários (OLS) (do inglês *Ordinary Least Squares method*). Cada $\hat{\beta}_i$ representa o coeficiente estimado para a variável independente X_i ; X é a matriz de dados independentes, onde cada coluna representa uma variável independente diferente e cada linha representa uma observação separada; X^T denota a transposição da matriz X , ou seja, as linhas de X tornam-se colunas de X^T e vice-versa; y é o vetor de variável dependente, que contém os valores observados que estão sendo modelados ou previstos; $(X^T X)^{-1}$ representa a inversa da matriz resultante da multiplicação da transposta de X por X . Esta é a matriz inversa de $X^T X$; $X^T y$ denota o produto matricial de X^T com y , resultando em um vetor; $\hat{\beta} = (X^T X)^{-1} X^T y$ representa a multiplicação da inversa de $X^T X$ com $X^T y$, resultando no vetor de coeficientes estimados $\hat{\beta}$ que minimiza a soma dos quadrados dos resíduos.

A equação (3.8) mencionada, conhecida como **OLS**, é amplamente utilizada na regressão linear Korstanje (2021). Esse método é conhecido por ser rápido de ajustar, pois requer apenas cálculos matriciais para estimar os coeficientes β . No entanto, ele é mais adequado para processos lineares e pode ser menos adequado para modelos complexos que envolvam relações não-lineares. Portanto, é importante considerar suas limitações ao aplicar a regressão linear em contextos mais complexos.

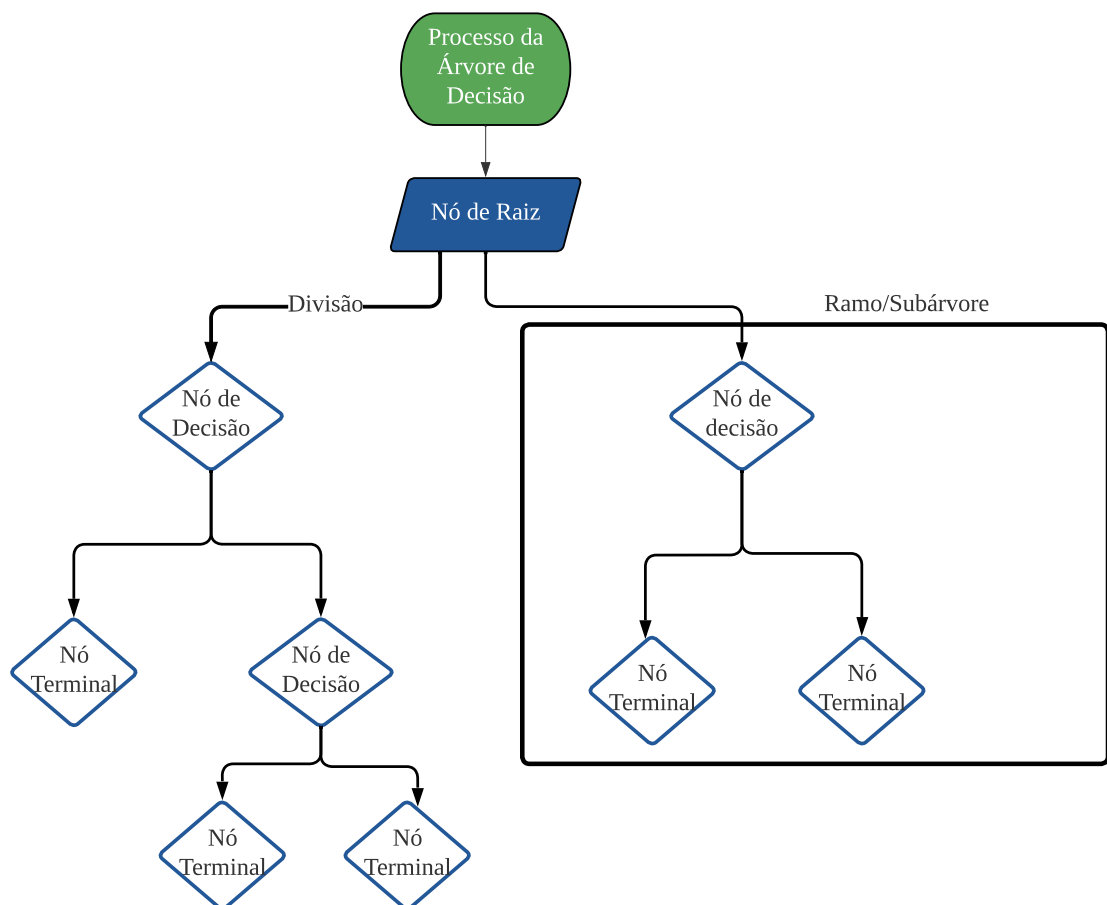
3.4.3 Árvore de Decisão

Uma árvore de decisão é um dos modelos de aprendizado de máquina mais utilizados para resolver problemas de regressão e classificação. Como o nome sugere, o algoritmo utiliza um modelo de decisões semelhante a uma árvore para prever o valor de destino (regressão) ou determinar a classe de destino (classificação). Antes de adentrar na explicação de como as árvores de decisão funcionam, é importante se familiarizar com as terminologias básicas associadas a uma árvore de decisão (READER, 2023).

Na Figura 8 trás **Nó raiz** isso representa o nó mais alto da árvore que representa todos os pontos de dados. **Divisão** refere-se à divisão de um nó em dois ou mais sub-nós.

Nó de decisão eles são os nós que são divididos em sub-nós, ou seja, esse nó que é dividido é chamado de nó de decisão. **Nó Folha / Terminal** os nós que não se dividem são chamados de nós Folha ou Terminal. Esses nós são geralmente o resultado final da árvore. **Ramo / Subárvore** uma subseção de toda a árvore é chamada de galho ou subárvore. **Nó pai e filho** um nó, que é dividido em sub-nós é chamado de um nó pai de sub-nós, enquanto sub-nós são o filho do nó pai. Na figura acima, o nó de decisão é o pai dos nós terminais (filho). **Poda** a remoção de sub-nós de um nó de decisão é chamada de poda. A poda costuma ser feita em árvores de decisão para evitar o *overfitting* (READER, 2023).

Figura 8: Fluxograma da árvore de decisão



Fonte: Adaptado de Reader (2023)

Vantagens: De fácil intuição e interpretação, sendo facilmente visualizá-las (quando não são muito profundas). Requerem pouco esforço na preparação dos dados, métodos baseados em árvores normalmente não requerem normalização dos dados, codificação e variáveis fictício. Além disso, conseguem lidar com valores faltantes, categóricos e numéri-

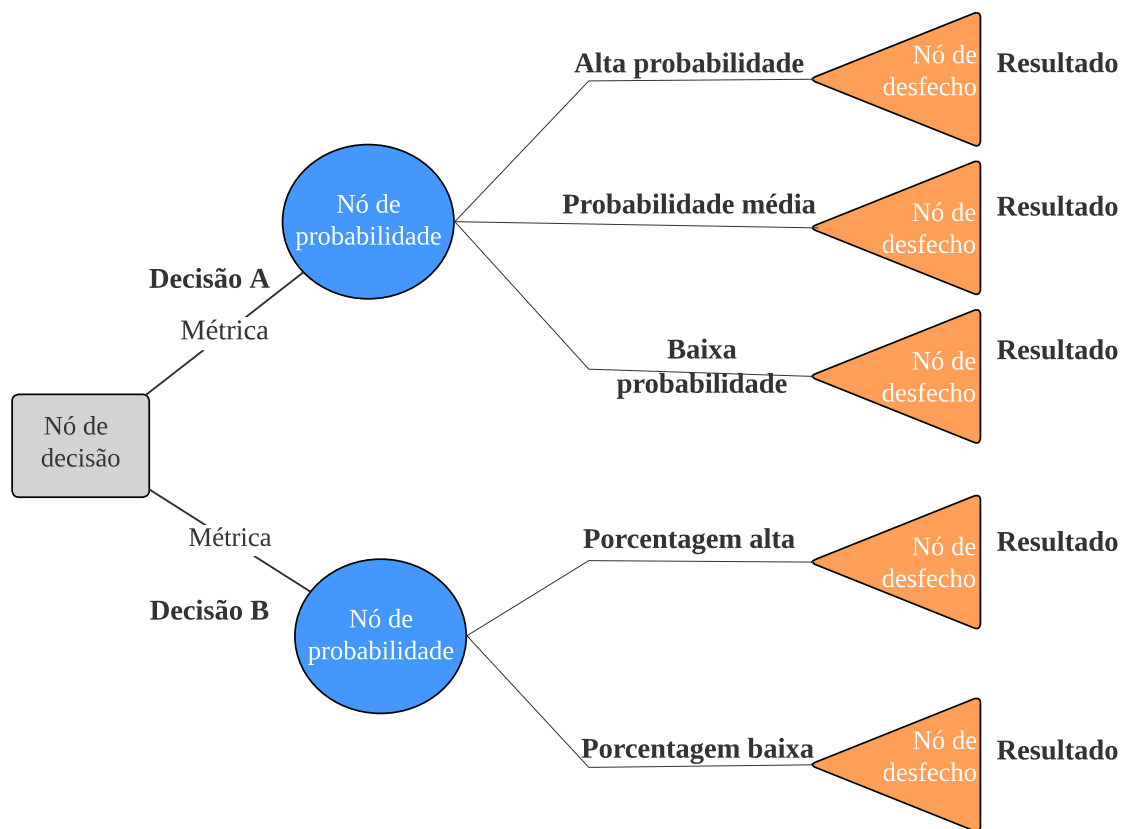
cos. Complexidade logarítmica na etapa de predição. São capazes de lidar com problemas com múltiplos rótulos. Relações não-lineares entre parâmetros não afetam o desempenho da árvore (REMIGIO, 2023).

Desvantagens: Árvore de decisão até sua profundidade máxima pode decorar o conjunto de treino (*overfitting*), o que pode degradar seu poder preditivo quando aplicado a novos dados. Isso pode ser mitigado “podando” a árvore de decisão ao atribuir uma profundidade máxima ou uma quantidade máxima de folhas. São modelos de alta variância, pequenas variações nos dados de treino podem resultar em árvores completamente distintas. Isso pode ser evitado ao treinar várias árvores distintas e agregar suas predições. O algoritmo de construção da árvore de decisão é ganancioso, ou seja, não garante a construção da melhor estrutura para o dados de treino em questão. Esse problema também pode ser mitigado ao treinarmos várias árvores distintas e agregar suas predições (REMIGIO, 2023).

Na Figura 9, é demonstrado como o processo é representado por meio de uma árvore de decisão, em relação ao mapa mental. As árvores de decisão, quando aplicadas a séries temporais relacionadas ao abastecimento de água, ajudam a prever padrões, detectar problemas, otimizar operações e apoiar a tomada de decisões específicas, como o funcionamento da variável LT01, para garantir um fornecimento confiável e eficiente de água.

Na Figura 9, o mapa é um diagrama de árvore de decisão que ajuda a visualizar a probabilidade de diferentes resultados baseados em diferentes decisões. Cada nó de decisão (círculo azul) tem três possíveis resultados: alta probabilidade, probabilidade média e baixa probabilidade. Cada resultado tem um nó de desfecho (triângulo laranja) que mostra o resultado final. Os nós são conectados por setas pretas que indicam o fluxo do processo. Esse tipo de diagrama é usado para apoiar a tomada de decisões em diversas áreas, como pesquisa operacional, análise de decisão e aprendizado de máquina.

Figura 9: Árvore de decisão mapa mental



A árvore de decisão pode ser uma opção melhor em comparação ao modelo de regressão linear. Ela tem a capacidade de otimizar os parâmetros para trabalhar com horizontes de tempo longos.

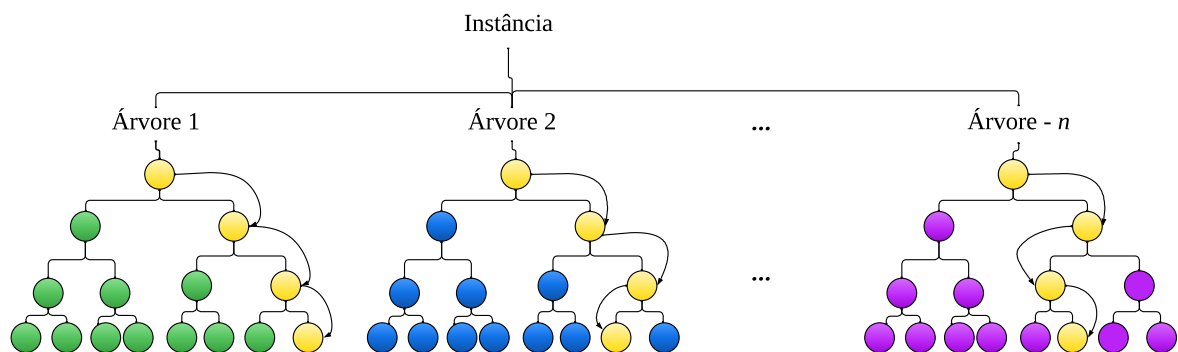
3.4.4 Floresta Aleatória

Observa-se da Figura 8 que repetir exatamente a mesma árvore de decisão várias vezes não adiciona valor significativo em comparação com o uso dessa árvore de decisão apenas uma vez. Em modelos de conjunto, é crucial que cada modelo individual apresente pequenas variações em relação aos demais. Dois métodos amplamente reconhecidos para criar conjuntos são o ensacamento (do inglês *bagging*) e o reforço (do inglês *boosting*). A floresta aleatória (do inglês *Random Forest*) utiliza o ensacamento para criar um conjunto de árvores de decisão, onde cada árvore é construída com uma amostra aleatória do conjunto de dados original. Isso assegura que as árvores sejam distintas e diversificadas, contribuindo para a robustez e eficácia do modelo.

Cada árvore em um modelo de RFR (Floresta Aleatória de Regressão) é construída por meio de um algoritmo de aprendizado individual que divide o conjunto de variáveis de

entrada em subconjuntos, com base em um teste de valor de atributo, como o coeficiente de Gini. Ao contrário das árvores de decisão clássicas, as árvores de RFR são construídas sem poda e selecionam aleatoriamente um subconjunto de variáveis de entrada em cada nó. Atualmente, o número de variáveis utilizadas para dividir um nó em uma RFR (denotado por m) corresponde à raiz quadrada do número total de variáveis de entrada. Essa abordagem ajuda a aumentar a diversidade das árvores e aprimorar o desempenho do modelo (PELLETIER et al., 2016). Na Figura 10, o esquema do modelo RFR ilustra como as árvores funcionam. Na construção da próxima árvore, os dois processos anteriores se repetirão, levando à criação de uma nova árvore. Provavelmente, essa árvore será diferente da primeira, pois tanto na seleção das amostras quanto na seleção das variáveis, o processo ocorre de maneira aleatória. Pode-se construir quantas árvores forem desejadas, e quanto mais árvores forem criadas, melhores serão os resultados do modelo, até certo ponto. No entanto, após um certo número de árvores, uma nova adição não resultará em uma melhora significativa no desempenho do modelo. É importante lembrar que, à medida que mais árvores são criadas, aumenta-se o tempo necessário para construir o modelo.

Figura 10: Esquema da floresta aleatória



3.4.5 Gradient Boosting

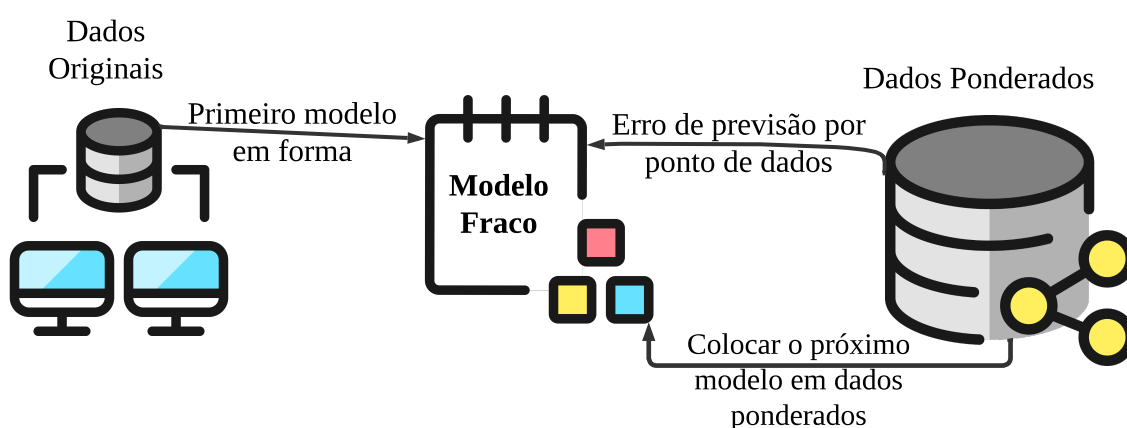
O aumento de gradiente (do inglês *gradient boosting*) é um método que combina vários modelos de árvore de decisão para realizar previsões. Cada uma dessas árvores de decisão é única, pois a diversidade é um elemento importante nesse processo. A diversidade é alcançada através de um processo chamado boosting, que é uma abordagem iterativa. O boosting adiciona modelos fracos ao conjunto de forma inteligente, dando mais peso aos pontos de dados que ainda não foram bem previstos.

O processo de boosting melhora o conjunto ao focar nas partes dos dados que ainda não são compreendidas. A Figura 11 apresenta uma visão esquemática desse processo. À medida que novos modelos fracos são adicionados, todos os modelos fracos intermediários são mantidos. O modelo final é uma combinação de todos esses modelos fracos, resultando

em um ensemble que oferece uma melhor capacidade de previsão do que um único modelo.

O boosting é apenas um dos métodos de ensemble utilizados em conjunto com o bagging. O bagging também é um método que utiliza múltiplos modelos de árvore de decisão, porém, em vez de adicionar os modelos de forma iterativa, cada modelo é treinado independentemente em subconjuntos aleatórios dos dados de treinamento. Ambos os métodos, boosting e bagging, têm como objetivo melhorar o desempenho do modelo combinando as previsões de múltiplos modelos individuais.

Figura 11: Impulsionando gradiente com XGBoost e LightGBM



Fonte: Adaptação de Korstanje (2021)

3.4.6 Gradiente de Boosting (Reforço)

O processo iterativo utilizado no aumento de gradiente, como descrito por Korstanje (2021), recebe esse nome por um motivo. O termo “gradiente” refere-se a um campo vetorial de derivadas parciais que apontam na direção da inclinação acentuada. Em termos simples, o gradiente pode ser comparado à inclinação em uma estrada. Se a inclinação é acentuada, isso indica que a estrada está subindo ou descendo abruptamente. Analogamente, nos métodos de otimização, o gradiente indica a rapidez com que uma função está aumentando ou diminuindo em um ponto específico. Quanto maior o gradiente, mais íngreme é a inclinação da função nesse ponto. Em problemas de otimização, é crucial que ele compreenda esse gradiente para encontrar os mínimos ou máximos da função de forma eficiente. Para calcular os gradientes, são realizadas derivadas ou derivadas parciais de uma função.

No aumento de gradiente, ao adicionar árvores adicionais ao modelo, o objetivo é incorporar uma árvore que explique melhor a variação que ainda não foi explicada pelas árvores anteriores. Dessa forma, a nova árvore tem como objetivo ajustar-se aos erros ou

resíduos deixados pelas árvores anteriores em explicar a variação nos dados.

Algoritmos de *boosting* de gradiente: O **XGBoost** é um dos algoritmos de aprendizado de máquina mais utilizados (KORSTANJE, 2021). É uma forma rápida de obter bom desempenho.

O **LightGBM** é outro algoritmo de aumento de gradiente que é importante conhecer. Atualmente, é um pouco menos difundido que o XGBoost, mas está ganhando popularidade rapidamente. A vantagem esperada do LightGBM em relação ao XGBoost é um ganho de velocidade e uma utilização mais eficiente de memória (KORSTANJE, 2021).

A diferença entre XGBoost e LightGBM:

Uma diferença fundamental reside na maneira como esses algoritmos identificam as melhores divisões entre os nós das árvores de decisão individuais. É crucial lembrar que uma divisão em uma árvore de decisão ocorre quando a árvore precisa encontrar a separação que mais melhora o desempenho do modelo. A abordagem intuitiva e simples para encontrar a melhor divisão é iterar por todas as possibilidades e selecionar a melhor. No entanto, essa abordagem é computacionalmente custosa, e algoritmos mais recentes apresentam alternativas mais eficientes.

Uma alternativa proposta pelo XGBoost é a segmentação baseada em histograma. Nesse caso, em vez de iterar por todas as partições possíveis, o modelo constrói um histograma para cada variável e utiliza-os para encontrar a melhor divisão geral entre as variáveis. O LightGBM, desenvolvido pela Microsoft, adota uma abordagem mais eficiente para a definição das divisões. Essa abordagem é conhecida como amostragem GOSS (do inglês *Gradient-Based One-Side Sample*). O GOSS calcula o gradiente para cada ponto de dados e utiliza-o para filtrar os pontos de dados com gradientes baixos. Afinal, os pontos de dados com gradientes baixos já são bem compreendidos, enquanto aqueles com gradientes altos precisam ser melhor aprendidos.

O LightGBM também utiliza uma abordagem chamada Exclusive EFB (do inglês *Feature Bundling*), que acelera a seleção de variáveis correlacionadas. Outra diferença é que o modelo LightGBM é adequado para o crescimento de folhas (do inglês *leaf-wise growth*), enquanto o XGBoost cultiva as árvores em níveis. Essa diferença pode ser visualizada na Figura 12. Essa diferença teoricamente favorece o LightGBM em termos de precisão, mas também apresenta um maior risco de sobre-ajuste (do inglês *overfitting*) quando há poucos dados disponíveis. Portanto, é importante que a pessoa considere essas distinções ao escolher entre os dois algoritmos de aumento de gradiente.

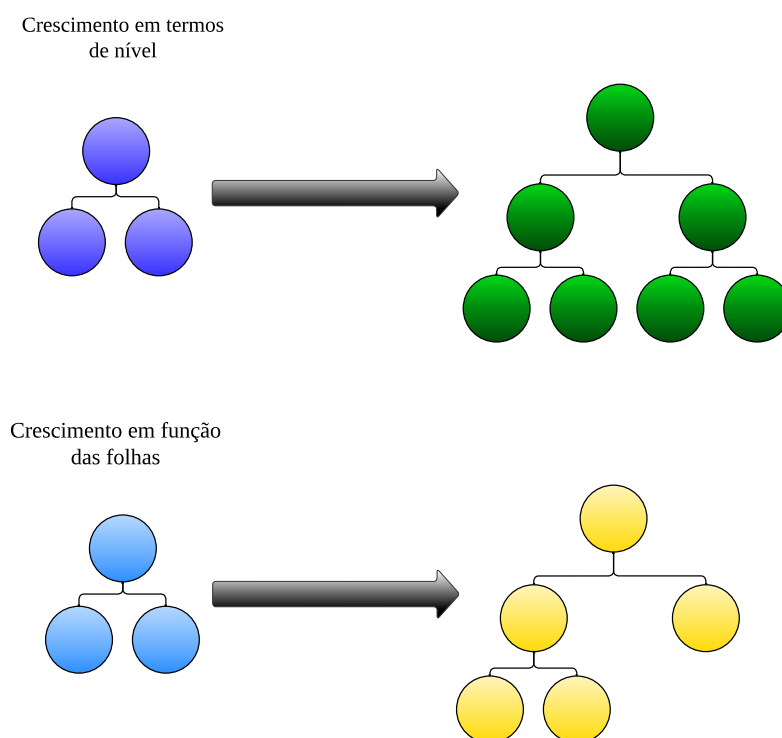
Observar a diferença na Figura 12. Essa diferença é um recurso que teoricamente favoreceria o LightGBM em termos de precisão; no entanto, há um risco maior de *overfitting* no caso de poucos dados disponíveis. Na Figura 12, é possível visualizar como cada

modelo é ajustado durante o processo de crescimento de árvore em folhas e em níveis. Essa representação gráfica oferece uma compreensão visual das diferenças entre os dois métodos. A Figura 12, apresenta um diagrama que ilustra o crescimento de uma árvore em termos de níveis e folhas. O diagrama possui duas partes.

Na parte superior, é observado que a cada nível, o número de folhas aumenta em uma unidade. Por exemplo, uma árvore com dois níveis tem duas folhas em cada nível, e uma árvore com três níveis tem três folhas em cada nível. Na parte inferior, é possível perceber que a cada nível, o número de folhas dobra. Por exemplo, uma árvore com dois níveis tem duas folhas em cada nível, e uma árvore com três níveis tem seis folhas no terceiro nível.

O diagrama também mostra que as árvores são representadas por círculos, com as folhas sendo círculos menores e os níveis sendo círculos maiores. As árvores são coloridas de azul, verde e amarelo. O texto na Figura 12 indica “Crescimento em termos de nível” e “Crescimento em função”. Isso significa que a Figura 12 demonstra como o crescimento da árvore depende do número de níveis e do número de folhas.

Figura 12: Compara-se o crescimento em folha com o crescimento em nível



Fonte: Adaptação de Korstanje (2021)

No crescimento de árvore em folhas, como no LightGBM, novas folhas são adicionadas à árvore de forma iterativa, visando maximizar a redução do erro de treinamento. Isso

significa que as árvores são expandidas adicionando folhas, uma a uma, até que o critério de parada seja alcançado. Por outro lado, no crescimento em níveis, como no XGBoost, as árvores são expandidas em profundidade de forma simultânea em todos os níveis. Ou seja, em cada nível, todas as folhas são expandidas ao mesmo tempo, resultando em um crescimento mais uniforme da árvore.

Essa distinção no modo de crescimento das árvores pode afetar o comportamento e o desempenho do modelo. Portanto, compreender essa diferença é importante ao escolher entre esses algoritmos de aumento de gradiente.

3.5 Redes Neurais Artificiais

Uma rede neural é um modelo de processamento de informações inspirado pelo funcionamento do cérebro humano. Consiste em um conjunto interconectado de unidades de processamento, conhecidas como neurônios artificiais, que trabalham em conjunto para realizar tarefas de aprendizado a partir de dados. Assim como os neurônios no cérebro estão interligados por sinapses, os neurônios artificiais são conectados por conexões ponderadas. Essas conexões permitem que a rede neural analise padrões complexos nos dados, reconhecendo relações e características importantes para executar tarefas como classificação, previsão, reconhecimento de padrões e muito mais. Conforme a rede é exposta a exemplos e informações, ela ajusta suas conexões para melhorar seu desempenho, tornando-a capaz de generalizar e lidar com novos dados (SILVA, 2003).

3.5.1 Rede Neural Recorrente

Uma Rede Neural Recorrente é um tipo de arquitetura de rede neural que pode ser utilizada para lidar com dados sequenciais ou temporais. Ao contrário das redes neurais convencionais, onde as entradas e saídas são tratadas como dados independentes, as RNNs levam em consideração a ordem e a relação entre os elementos em uma sequência, tornando-as ideais para lidar com dados como séries temporais, texto e áudio.

A característica principal das RNNs é que elas contêm laços em sua estrutura, permitindo que informações anteriores influenciem o processamento de informações subsequentes. Isso significa que a saída em um determinado passo de tempo não depende apenas da entrada atual, mas também das entradas anteriores na sequência.

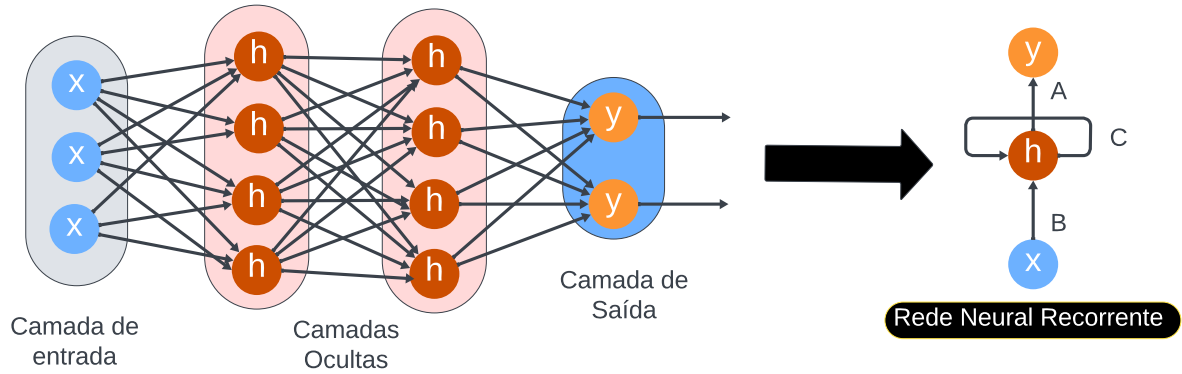
$$h_t = f(W_{hh} \cdot h_{t-1} + W_{xh} \cdot x_t + b_h) \quad (3.9)$$

onde h_t é o estado oculto (ou saída) no tempo t , h_{t-1} é o estado oculto anterior no tempo

$t - 1$, x_t é a entrada no tempo t , W_{hh} é a matriz de pesos que controla a influência do estado oculto anterior, W_{xh} é a matriz de pesos que controla a influência da entrada, b_h é o vetor de viés, f é uma função de ativação, frequentemente a função tangente hiperbólica (tanh) ou a função sigmoide (TAM, 2023).

Essa equação representa a propagação do estado oculto ao longo do tempo em uma RNN. A cada novo passo de tempo, a RNN considera a entrada atual x_t e o estado oculto anterior h_{t-1} , calculando o novo estado oculto h_t usando as matrizes de pesos e a função de ativação. No entanto, as RNNs tradicionais podem enfrentar dificuldades em capturar dependências de longo prazo, devido ao problema de dissipação do gradiente. Para lidar com isso, surgiram variações mais avançadas, como LSTM (do inglês *Long Short-Term Memory*) e GRU (do inglês *Gated Recurrent Units*), que incorporam mecanismos de aprendizado de esquecimento e controle de informação, permitindo que informações relevantes sejam mantidas por períodos mais longos de tempo.

Como pode ser visto na Figura 13, a grande diferença no bloco RNN é que há um laço de reações. Enquanto cada entrada de uma rede totalmente conectada é completamente independente, as entradas de uma RNN têm uma relação de realimentação entre si. Isso faz com que ele seja capaz de capturar padrões em dados sequenciais de uma maneira que redes neurais tradicionais não conseguem. A Figura 13 mostra um diagrama de uma RNN. Na representação, a camada de entrada é rotulada como “Camada de entrada” e consiste em três círculos laranjas rotulados como “ x ”. As camadas ocultas são identificadas como “Camadas Ocultas” e compõem-se de seis círculos rosas rotulados como “ h ”. A camada de saída é designada como “Camada de Saída” e possui dois círculos azuis rotulados como “ y ”. As conexões entre as camadas são estabelecidas por linhas pretas. Além disso, o diagrama inclui uma versão simplificada da RNN no lado direito, indicada como “Rede Neural Recorrente”, composta por três círculos laranjas rotulados como “A”, “B” e “C”. Uma Rede Neural Recorrente é um tipo de rede neural capaz de processar sequências de dados, como texto, áudio ou vídeo. Ela tem a capacidade de reter informações do passado e utilizá-las para influenciar as saídas futuras. É empregada em tarefas como reconhecimento de fala, tradução automática, geração de texto, entre outras, e nesse caso para prever o nível do tanque.

Figura 13: RNN - *recurrent neural network*

Fonte: Adaptado de Zhang (2021)

3.5.2 Compreendendo Redes de Memória de Curto e Longo Prazo (LSTM)

As LSTMs são uma evolução das RNNs, projetadas para superar desafios na captura de dependências de longo prazo em sequências de dados. Diferentemente das RNNs convencionais, as LSTMs têm a capacidade de manter informações relevantes por longos períodos, tornando-as especialmente eficazes em tarefas que envolvem padrões complexos e dependências temporais distantes (ZHANG, 2021).

Uma das principais inovações das LSTMs é a introdução de unidades de memória chamadas “células”, que possuem três componentes principais: uma porta de entrada (do inglês *input gate*), uma porta de esquecimento (do inglês *forget gate*) e uma porta de saída (do inglês *output gate*). Essas portas permitem que as LSTMs controlem o fluxo de informações através da célula, decidindo quais informações devem ser mantidas, esquecidas ou passadas para a saída (ZHANG, 2021).

$$f_t = \sigma(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f) \quad (3.10)$$

$$i_t = \sigma(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i) \quad (3.11)$$

$$\tilde{C}_t = \tanh(W_{xc} \cdot x_t + W_{hc} \cdot h_{t-1} + b_c) \quad (3.12)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (3.13)$$

$$o_t = \sigma(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + b_o) \quad (3.14)$$

$$h_t = o_t \odot \tanh(C_t) \quad (3.15)$$

onde, x_t é a entrada no tempo t , h_{t-1} é o estado oculto anterior no tempo $t - 1$, f_t é o valor da porta de esquecimento, i_t é o valor da porta de entrada, \tilde{C}_t é o candidato a novo estado de memória, C_t é o novo estado de memória, o_t é o valor da porta de saída, h_t é o

novo estado oculto (saída) no tempo t , σ é a função de ativação sigmoide, \odot representa a multiplicação elemento a elemento.

Essa estrutura permite que as LSTMs controlem o fluxo de informações e aprendam a armazenar ou descartar informações relevantes para diferentes tarefas. As portas de entrada, esquecimento e saída funcionam como mecanismos de controle, permitindo que as LSTMs aprendam a manter informações importantes, esquecer informações desnecessárias e gerar saídas precisas ao longo de sequências temporais.

3.5.3 GRU (Unidade Recorrente Fechada)

Um GRU é um tipo de arquitetura de RNN que foi projetado para lidar com o problema de dissipação de gradiente e captura de dependências de longo prazo em sequências de dados. Essa variação das RNNs tradicionais introduz mecanismos de portão para controlar o fluxo de informação por meio das unidades de tempo.

A GRU é uma alternativa vantajosa para a análise de séries temporais, devido à sua habilidade de lidar com sequências de dados de extensões variáveis e de capturar dependências de longo prazo presentes em informações sequenciais. Além disso, a GRU apresenta uma estrutura de simplicidade superior à LSTM, permitindo um processo de treinamento ágil (MIGLIATO; PONTI, 2021).

A estrutura do GRU inclui dois portões principais: o portão de atualização (do inglês *update gate*) e o portão de reinicialização (do inglês *reset gate*). Esses portões permitem que o GRU decida quais informações serão transmitidas para a próxima etapa de tempo e quais informações serão descartadas, nessas equações (3.16), (3.17), (3.18) e (3.19): h_t representa o estado oculto na etapa de tempo t , h_{t-1} é o estado oculto na etapa de tempo anterior $t - 1$, x_t é a entrada na etapa de tempo t , r_t é o valor do portão de reinicialização na etapa t , z_t é o valor do portão de atualização na etapa t , \odot denota a multiplicação elemento a elemento, σ é a função sigmoid, que retorna valores entre 0 e 1, \tanh é a função tangente hiperbólica, que retorna valores entre -1 e 1 , W_r , W_z e W_h são matrizes de pesos que o modelo aprende durante o treinamento.

Portão de Reinicialização (r_t) : Controla a quantidade de informação do passado a ser esquecida.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (3.16)$$

Portão de Atualização (z_t) : Controla a quantidade de informação do passado a ser passada para o próximo estado.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (3.17)$$

Ativação do Candidato (\tilde{h}_t) : Candidato a novo estado oculto.

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t]) \quad (3.18)$$

Novo Estado Oculto (h_t) : Combinação ponderada do estado anterior e do novo candidato.

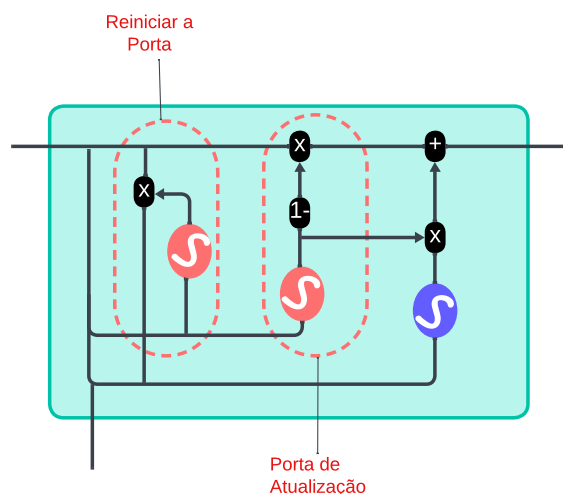
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (3.19)$$

O GRU controla como as informações são atualizadas e propagadas ao longo do tempo, permitindo a captura de dependências de longo prazo em sequências de dados. Isso o torna uma escolha popular para tarefas que envolvem processamento de linguagem natural, como tradução automática, geração de texto, entre outras.

Foi inventada uma camada RNN mais avançada, designada GRU. A célula GRU possui mais parâmetros, conforme mostrado na Figura 14. Isso demonstra que há uma passagem extra dentro da célula, permitindo que um parâmetro adicional seja estimado. Isso ajuda a aprendizagem das tendências a longo prazo.

Na Figura 14 representa um diagrama de um modelo de GRU para análise de séries temporais. O modelo GRU é um tipo de rede neural recorrente que possui dois portões: um portão de atualização e um portão de reinicialização. Esses portões controlam como a informação é armazenada e atualizada na memória oculta da rede. Um modelo GRU é capaz de aprender padrões temporais complexos e dependências de longo prazo nos dados sequenciais. A Figura 14 apresenta uma representação simplificada do modelo com três portões: o portão de reinicialização, o portão de atualização e o portão de saída. Os portões são interconectados por linhas tracejadas, representando o fluxo de informação entre eles. O diagrama está rotulado em português, com “Porta de Reinicialização”, “Porta de Atualização” e “Porta de Saída” (SARANYA; SIVAKUMAR, 2020; JORDAN; SOKÓŁ; PARK, 2021; KHAN et al., 2022).

Figura 14: Diagrama ilustrativo do funcionamento de uma unidade recorrente gated (GRU)



Fonte: Adaptado de Book (2023)

3.5.4 Análise dos Modelos RNN, LSTM e GRU

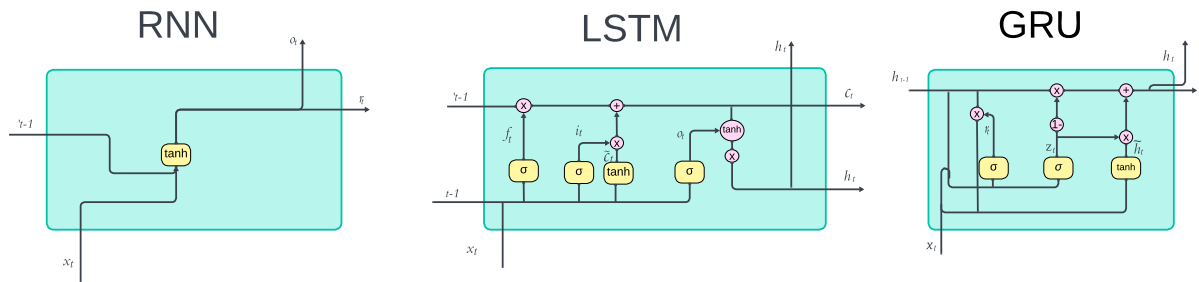
As GRUs, as LSTMs e as RNNs são variações das arquiteturas de redes neurais, todas projetadas para abordar a dificuldade de capturar dependências temporais em sequências de dados. Cada uma dessas abordagens tem características distintas que influenciam sua capacidade de lidar com esse desafio.

Enquanto as RNNs tradicionais têm uma tendência a sofrer com o desvanecimento do gradiente ao longo do tempo, as LSTMs e GRUs foram desenvolvidas para superar essa limitação. As LSTMs introduzem células de memória e portas de controle que permitem armazenar e atualizar informações relevantes ao longo das etapas temporais, sendo especialmente adequadas para capturar relações de dependência de longo prazo. As GRUs, por sua vez, simplificam a arquitetura das LSTMs, utilizando portas de atualização e reset para permitir o fluxo de informações e controle sobre o estado oculto. Na Figura 15, há um esquema que ilustra as arquiteturas das RNNs, LSTMs e GRUs, permitindo uma visualização das diferenças entre essas abordagens.

Na Figura 15, representa um diagrama de três tipos de RNNs: uma RNN regular, uma LSTM e uma GRU. Esses tipos de redes são capazes de processar dados sequenciais, como textos, áudios ou vídeos, levando em conta a ordem cronológica dos elementos. A Figura 15 é dividida em três seções, uma para cada tipo de rede. Cada seção tem uma cor de fundo diferente: a seção RNN é verde, a seção LSTM é rosa e a seção GRU é azul. Cada seção possui um diagrama da arquitetura da rede, com nós representando neurônios e arestas representando conexões entre neurônios. A seção RNN tem um único neurônio

recorrente, a seção LSTM tem vários neurônios recorrentes com conexões adicionais que formam portões e células de memória, e a seção GRU tem dois portões que controlam o fluxo de informação na memória oculta da rede. Os portões são representados por formas coloridas: o portão de reinicialização é azul, o portão de atualização é vermelho e o portão de saída é verde. O diagrama está rotulado em inglês, com “*Reset Gate*” (Portão de Reinicialização), “*Update Gate*” (Portão de Atualização) e “*Output Gate*” (Portão de Saída).

Figura 15: RNN vs LSTM vs GRU



Fonte: Adaptado de Hasan (2020)

Nas RNNs, os laços de *feedback* evidenciam a capacidade de lembrar informações passadas, fundamental para tarefas que requerem contexto temporal, como previsão de séries temporais. Nas GRUs, a estrutura modular permite controlar o fluxo de informações e o estado da memória de maneira mais eficaz. Isso ajuda a GRU a aprender padrões complexos e relações temporais em dados sequenciais. A observação direta dessas arquiteturas em ação na Figura 15 facilita a compreensão de como cada uma delas lida com as dependências temporais, sendo essencial para escolher a abordagem adequada para diferentes tipos de dados e tarefas.

As LSTMs e GRUs oferecem soluções mais sofisticadas em relação às RNNs tradicionais, apresentando mecanismos que permitem capturar dependências de longo prazo de maneira mais eficaz.

3.6 Aprendizado Profundo (DL)

Em relação ao abastecimento de água, os modelos de séries temporais no aprendizado profundo (DL do inglês *deep learning*) permitem análises detalhadas das tendências de consumo, disponibilidade e gestão dos recursos hídricos ao longo do tempo. Esses modelos também são úteis para monitorar a qualidade da água, identificando padrões de contaminação e contribuindo para a manutenção dos padrões de potabilidade.

3.6.1 Explorando o Transformer: Além dos Bits e Bytes

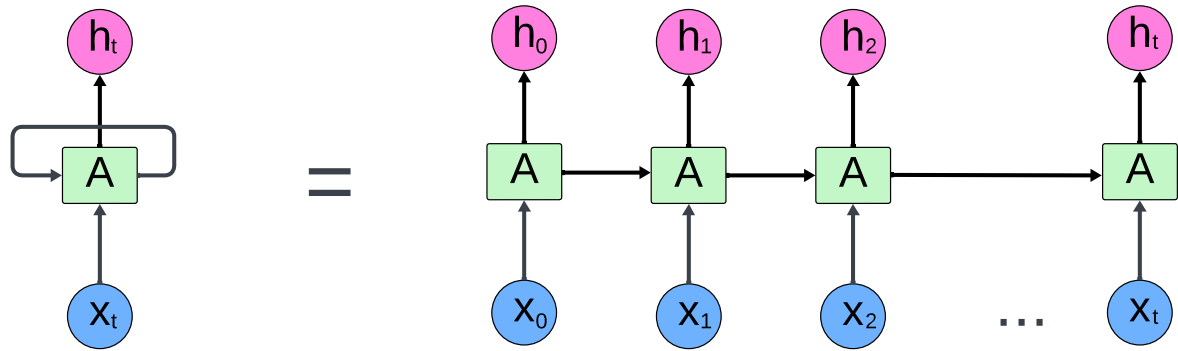
A arquitetura de rede neural Transformer representa um avanço significativo no campo do processamento de linguagem natural e tarefas relacionadas. Foi introduzida por (VASWANI et al., 2017) e revolucionou a maneira como as redes neurais lidam com sequências de dados, superando limitações anteriores, como a dependência sequencial e a complexidade computacional. A abordagem do Transformer se destaca por sua capacidade de processar simultaneamente todas as posições de uma sequência, tornando-o altamente paralelizável e eficiente.

A equação (3.20) fundamental do Transformer é a autoatencção, também conhecida como mecanismo de atenção. A atenção é um conceito-chave que permite que a rede neural “preste atenção” a diferentes partes da entrada em graus variados, capturando relações contextuais e semânticas. A equação da autoatencção é calculada ao dividir a sequência de entrada em três representações lineares: consultas (Q), chaves (K) e valores (V). A matriz de atenção é obtida multiplicando as consultas pelas chaves transpostas e aplicando uma função de softmax aos resultados, ponderando os valores de acordo com a importância atribuída pela atenção. A saída final é uma combinação linear dos valores ponderados pela matriz de atenção.

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (3.20)$$

na equação (3.20), embora simplificada, serve como base para a arquitetura do Transformer e é repetida várias vezes em diferentes camadas. Isso permite que a rede aprenda representações ricas e contextuais das sequências de entrada. A estrutura de múltiplas cabeças de atenção, presente no Transformer, aprimora a capacidade da rede em capturar diferentes tipos de relações e padrões nas sequências. Em suma, o modelo Transformer revolucionou o processamento de sequências, proporcionando melhorias notáveis em tarefas como tradução automática, resumo de texto, geração de linguagem natural e muito mais. Na Figura 16 tem o esquema de como a rede neural Transformer é abordada.

Figura 16: Arquitetura do Transformer



Fonte: Adaptado de Esposito (2021)

3.6.2 Rede Neural Artificial (ANN)

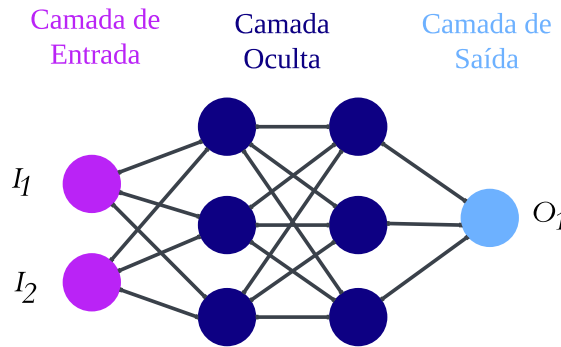
ANN pode ser definida como uma estrutura complexa interligada por elementos de processamento simples (neurônios), que possuem a capacidade de realizar operações como cálculos em paralelo, para processamento de dados e representação de conhecimento. Com a introdução de algoritmos de treinamento como a retropropagação do erro, que permite a realização de um treinamento posterior para aperfeiçoar os resultados do modelo (GRÜBLER, 2018).

Multilayer Perceptron: Com o intuito de lidar com os problemas não linearmente separáveis, foram adicionadas camadas de neurônio ocultas no modelo de *Rosenblatt*, formando então a Rede Neural Artificial Multilayer Perceptron (MLP). Essa nova topologia funciona como uma rede *feedforward* (rede progressiva, a saída de um neurônio se conecta com outro neurônio da próxima camada, no sentido esquerda/direita), formada por um conjunto de neurônios denominados “nós”, como na Figura 18. A rede possui uma camada de entrada (sem função computacional), uma ou mais camadas ocultas e uma camada de saída. A complexidade da rede MLP se dá pela quantidade de camadas ocultas que houver e a quantidade de neurônios que essas camadas possuírem.

A Figura 18 é um diagrama de um modelo ANN, que é um modelo computacional inspirado no cérebro humano. Ele consiste de um grande número de nós conectados, cada um realizando uma operação matemática simples. O diagrama consiste de três camadas: uma camada de entrada, uma camada oculta e uma camada de saída. A camada de entrada consiste de dois nós rosa rotulados I_1 e I_2 . A camada oculta consiste de três nós azuis. A camada de saída consiste de um nó azul rotulado O_1 . Os nós são conectados por linhas pretas representando as conexões entre os nós. Cada conexão tem um peso que ajusta a força do sinal entre os nós. Cada nó tem uma função de ativação que determina a saída do nó baseada na soma das entradas ponderadas. O diagrama está rotulado em

português com “Camada de Entrada” para a camada de entrada, “Camada Oculta” para a camada oculta e “Camada de Saída” para a camada de saída. Um modelo ANN pode aprender padrões e relações nos dados de entrada e produzir uma saída desejada, como uma classificação ou uma previsão. O modelo aprende ajustando os pesos das conexões através de um processo chamado treinamento, que envolve comparar a saída do modelo com a saída esperada e minimizar o erro.

Figura 17: Modelo de uma Rede Neural Artificial MLP



Fonte: Adaptado de Grübler (2018)

$I = [I_1, I_2] =$ Vetor de Entrada

$O = [O_1] =$ Vetor de Saída

O modelo de Rede Neural Artificial MLP é dado pela equação (3.21):

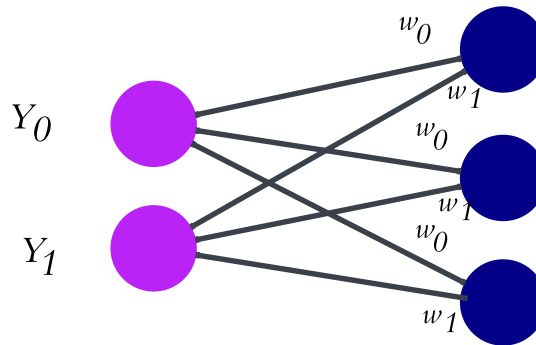
$$v_j = \sum_{i=0}^m w_i y_i + b \quad (3.21)$$

o funcionamento geral de uma rede MLP está representada na figura acima. Cada neurônio recebe todos os valores das entradas, representadas pelo símbolo y , que são multiplicadas pelos pesos sinápticos simbolizados pelo w e somadas entre si junto com uma constante chamada de polarização ou bias, representada pelo símbolo b .

A Figura 18 é um diagrama de um modelo ANN com múltiplas camadas e perceptrons, que são unidades de processamento simples que podem aprender padrões lineares nos dados. O diagrama consiste de duas camadas de perceptrons, uma com dois círculos rosa e outra com três círculos azuis. Os perceptrons são conectados por linhas pretas representando os pesos, que são os valores numéricos que ajustam a força da conexão entre os perceptrons. Os pesos são rotulados com “ w_0 ” e “ w_1 ”, indicando os valores dos pesos entre as camadas. Os círculos rosa são rotulados com “ Y_0 ” e “ Y_1 ”, indicando as saídas dos

perceptrons da primeira camada. O diagrama está rotulado em português com “Camada de Entrada” para a primeira camada de perceptrons, “Camada Oculta” para a segunda camada de perceptrons, e “Camada de Saída” para o único círculo azul que representa a saída final do modelo. Um modelo ANN com múltiplas camadas e perceptrons pode aprender padrões não lineares nos dados, usando funções de ativação não lineares nos perceptrons. O modelo é treinado usando o método de retropropagação, que consiste em ajustar os pesos das conexões de acordo com o erro entre a saída esperada e a saída obtida pelo modelo.

Figura 18: A equação da figura realiza o somatório ponderado entre as sinapses de cada neurônio



Fonte: Adaptado de Grübler (2018)

3.7 Rede Neural Convolucional (CNN)

As Redes Neurais Convolucionais (CNN) ou Redes Convolucionais são um tipo de rede neural que utiliza a operação de convolução em vez da multiplicação por matrizes em ao menos uma de suas camadas.

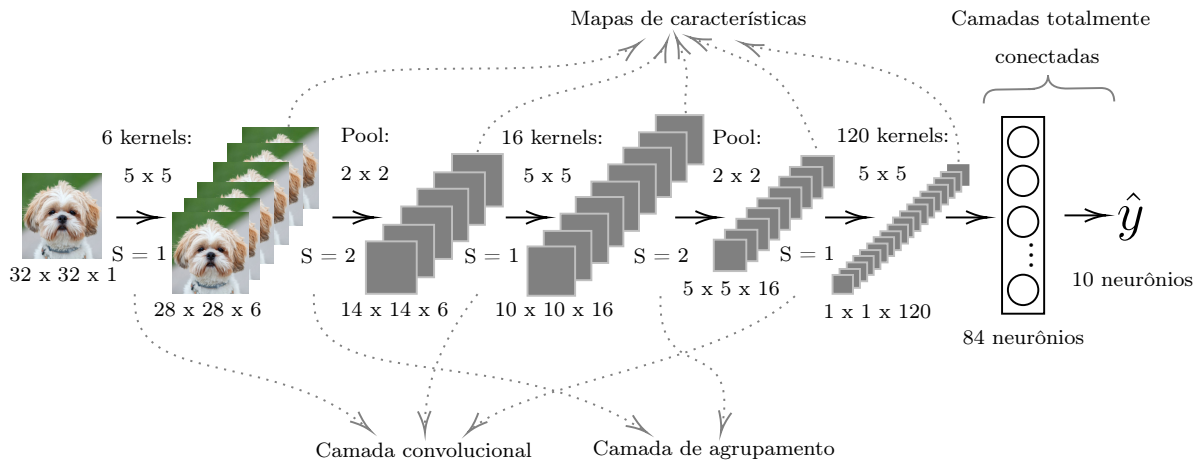
Esse tipo de rede é efetiva em aplicações (REICHMAN; MALOF; COLLINS, 2016) em que os dados são dispostos de forma que a relação de vizinhança entre os elementos é relevante, como no caso de imagens que são representadas por matrizes bidimensionais de pixels, ou no caso de séries temporais ou dados de áudio, que são sequências unidimensionais de dados amostrados em intervalos de tempo regulares (SILVA, 2021).

A camada convolucional tem como objetivo extrair as características mais importantes da entrada. Dessa forma, sua saída é um mapa de características obtido a partir da convolução da entrada com um *kernel* aprendido, seguido da aplicação de uma função de ativação não linear (LUCAS, 2019). Os mapas de características completos são obtidos pela Equação (3.22):

$$Z_{i,j,k}^L = W_k^L \cdot X_{i,j}^L + b_k^L \quad (3.22)$$

onde, $Z_{i,j,k}^L$ é o mapa de características obtido pela convolução do k -ésimo filtro da L -ésima camada com a célula de entrada centrada na localização (i, j) . W_k^L vetor de pesos do k -ésimo filtro da L -ésima camada. b_k^L termo de polarização do k -ésimo filtro da L -ésima camada. $X_{i,j}^L$ é a célula de entrada centrada na localização (i, j) da L -ésima camada. A profundidade dos mapas de características é dada pelo número de *kernels* (ou filtros) de convolução. Observe na Figura 19 que a 1ª camada de convolução com 6 *kernel* gera uma saída de profundidade 6. Isso porque, cada *kernel* possui pesos diferentes para extrair diferentes características da entrada (LUCAS, 2019).

Figura 19: Modelo de uma Rede Neural Convolucional



Fonte: Lucas (2019)

Uma vantagem das camadas de convolução é o compartilhamento do vetor de pesos para toda a circunvolução na construção de um mapa de características, pois reduz o número de parâmetros na rede, resultando em treinamento e previsões mais eficientes (LUCAS, 2019). A largura e a altura desses mapas são definidas pelo tamanho do *kernel* e do *stride* (passo da circunvolução) Equação (3.23). Voltando à Figura 19, a 1ª camada convolucional gera uma saída 28×28 , pois $\left(\frac{32-5}{1}\right) + 1 = 28$.

$$T_{\text{map}} = \left(\frac{I - F}{S + 1}\right) \quad (3.23)$$

onde T_{map} é a altura ou largura do mapa de características, I é a altura ou largura da entrada, F é a altura ou largura do *kernel* de convolução, S é o tamanho do *stride*.

3.8 Métricas de Avaliação de Modelos

A métrica de Erro Quadrático Médio (MSE) é utilizada no campo do aprendizado de máquina para avaliar a qualidade dos modelos de previsão. O MSE é regido pela seguinte equação:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.24)$$

onde n é o número de amostras, y_i é o valor medido não real correspondente à amostra i e \hat{y}_i é o valor previsto para a mesma amostra.

3.8.1 Erro Quadrático Médio Raiz (RMSE)

O RMSE é uma métrica amplamente empregada na avaliação de modelos de previsão em séries temporais. Ele é calculado tomando a raiz quadrada do MSE, conforme segue,

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3.25)$$

O RMSE penaliza os valores discrepantes.

3.8.2 Raiz do Erro Médio Quadrático Relativo (RRMSE)

O RRMSE é uma variante do RMSE. O erro quadrático médio (RMSE) é uma medida de erro quadrático médio relativo que foi escalado em relação ao valor real e depois normalizado pelo valor da raiz quadrada média. O RRMSE pode ser expressado por,

$$RRMSE = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}}{\sqrt{\sum_{i=1}^N (\hat{y}_i)^2}} \quad (3.26)$$

3.8.3 Erro Absoluto Médio (MAE)

O Erro Absoluto Médio (MAE) é utilizado como uma métrica para avaliar o desempenho de modelos de previsão. Em vez de calcular a média das diferenças entre os valores reais e previstos, o MAE calcula a média dos valores absolutos dessas diferenças,

garantindo que os erros positivos e negativos não se anulem. A equação do MAE é dada por:

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}_i| \quad (3.27)$$

sua interpretação é similar ao RMSE, em que o erro é expresso na mesma escala ou ordem de grandeza da variável estudada.

3.8.4 Erro Percentual Absoluto Médio (MAPE)

O Erro Percentual Absoluto Médio (MAPE) é uma métrica que expressa o erro de previsão como uma porcentagem relativa ao valor observado. O MAPE é calculado usando a seguinte fórmula:

$$MAPE = \frac{1}{n} \sum \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (3.28)$$

3.8.5 Erro Percentual Absoluto Médio Simétrico (sMAPE)

O sMAPE (do inglês *Symmetric Mean Absolute Percentage Error*), ou Erro Médio Percentual Absoluto Simétrico, é outra métrica comumente utilizada para avaliar a precisão de modelos de previsão.

O sMAPE é expresso como uma porcentagem, facilitando a compreensão da precisão relativa do modelo. O sMAPE é adequado para lidar com valores nulos nos dados, pois a divisão por zero é evitada no cálculo da métrica.

O sMAPE é sensível a valores extremos nos dados. Se houver valores discrepantes que não representem a tendência geral, eles podem influenciar significativamente a métrica. O sMAPE é dado por:

$$sMAPE = \frac{1}{n} \sum_{i=1}^n \frac{2|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)} \times 100 \quad (3.29)$$

3.9 Teste de Ljung-Box

O teste de Ljung-Box (nomeado em homenagem a Greta M. Ljung e George E. P. Box) é um tipo de teste estatístico de se qualquer um de um grupo de autocorrelações de uma série temporal é diferente de zero. Em vez de testar a aleatoriedade em cada lag

distinto, ele testa a aleatoriedade “geral” com base em um número de lags, e é, portanto, um teste portmanteau.

Este teste é às vezes conhecido como o teste de Ljung-Box Q , e está intimamente ligado ao teste Box-Pierce (que é nomeado após George E. P. Box e David A. Pierce). De fato, a estatística do teste de Ljung-Box foi descrita explicitamente no artigo que levou ao uso da estatística Box-Pierce, (BOX; PIERCE, 1970), (LJUNG; BOX, 1978) e da qual essa estatística leva seu nome. A estatística de teste Box-Pierce é uma versão simplificada da estatística de Ljung-Box para a qual estudos de simulação subsequentes mostraram baixo desempenho (DAVIES; NEWBOLD, 1979).

O teste de Ljung-Box é amplamente aplicado em econometria e outras aplicações da análise de séries temporais. Avaliação semelhante também pode ser realizada com o teste de Breusch-Godfrey e o teste de Durbin-Watson.

Definição Formal o teste de Ljung-Box pode ser definido como:

H_0 : Os dados são distribuídos de forma independente (ou seja, as correlações na população da qual a amostra é retirada são 0, de modo que quaisquer correlações observadas nos dados resultam da aleatoriedade do processo de amostragem).

H_a : Os dados não são distribuídos de forma independente; apresentam correlação serial. A estatística de teste é (LJUNG; BOX, 1978):

$$Q = n(n+2) \sum_{k=1}^h \frac{\hat{\rho}_k^2}{n-k} \quad (3.30)$$

onde n é o tamanho da amostra, $\hat{\rho}_k$ é a autocorrelação da amostra no lag k e h é o número de defasagens que estão sendo testadas. Debaixo H_0 a estatística Q segue assintoticamente um $\chi_{(h)}^2$. Para o nível de significância α , a região crítica para rejeição da hipótese de aleatoriedade é:

$$Q > \chi_{1-\alpha, h}^2 \quad (3.31)$$

onde $\chi_{1-\alpha, h}^2$ é o $(1 - \alpha)$ -quantil (BROCKWELL; DAVIS, 2002) da distribuição qui-quadrada com graus h de liberdade.

O teste de Ljung-Box é comumente usado na modelagem de média móvel integrada ARIMA. Note que ele é aplicado aos resíduos de um modelo ARIMA ajustado, não à série original, e em tais aplicações a hipótese que está sendo testada é que os resíduos do modelo ARIMA não têm autocorrelação. Ao testar os resíduos de um modelo ARIMA estimado, os graus de liberdade precisam ser ajustados para refletir a estimação do parâmetro. Por

exemplo, para um modelo ARIMA $(p, 0, q)$, os graus de liberdade devem ser definidos como $h - p - q$ (DAVIDSON, 2000).

Teste Box-Pierce: O teste Box-Pierce utiliza a estatística do teste, na notação descrita acima, dada por (BOX; PIERCE, 1970)

$$Q_{BP} = n \sum_{k=1}^h \hat{\rho}_k^2 \quad (3.32)$$

e usa a mesma região crítica definida acima. Estudos de simulação mostraram que a distribuição para a estatística Ljung-Box é mais próxima de um $\chi^2_{(h)}$ 6 distribuição do que é a distribuição para a estatística Box-Pierce para todos os tamanhos de amostra, incluindo os pequenos.

3.10 Correlação de Pearson

Nos modelos de aprendizado de máquina supervisionados, é feita uma tentativa de identificar as relações existentes entre diferentes variáveis (KORSTANJE, 2021):

Variável de destino: a variável que tenta prever. Variáveis explicativas: Variáveis que ajudam a prever o alvo variável

Para realizar previsões, é importante que se compreenda quais tipos de variáveis explicativas podem ser utilizadas. Neste exemplo, a variável **Pressão de Sucção (PT01SU)** será considerada como a variável x , enquanto a variável **Nível do Reservatório (Câmara 1) LT01** será considerada como a variável y . O coeficiente de correlação indica a relação entre o eixo x e y , como expresso pela seguinte fórmula. A equação do coeficiente de correlação de Pearson é dada por:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum (x_i - \bar{x})^2)(\sum (y_i - \bar{y})^2)}} \quad (3.33)$$

onde x_i e y_i representam os valores das variáveis X e Y , respectivamente. \bar{x} e \bar{y} são as médias dos valores x_i e y_i . O coeficiente de correlação de Pearson mede a força e a direção da relação linear entre as variáveis X e Y . Valores próximos a 1 indicam uma correlação positiva forte, valores próximos a -1 indicam uma correlação negativa forte, e valores próximos a 0 indicam uma ausência de correlação entre as variáveis.

3.11 Decomposição STL

A decomposição sazonal e de tendência utilizando o procedimento de suavização de diagramas de dispersão estimada localmente (Loess) (STL) é uma técnica amplamente utilizada para decompor séries temporais em seus componentes sazonais, de tendência e restantes. O método STL realiza a decomposição aditiva dos dados por meio de uma sequência de aplicações do Loess mais suave, onde regressões polinomiais ponderadas localmente são aplicadas em cada amostra do conjunto de dados, tendo como variáveis explicativas os valores próximos do amostra cuja resposta está sendo estimada (THEODOSIOU, 2011).

A decomposição STL é útil para identificar padrões sazonais e de tendência presentes nas séries temporais. Ela permite a separação dos componentes sazonais, que ocorrem em intervalos regulares ao longo do tempo, da componente de tendência, que indica a direção geral dos dados ao longo do tempo. A decomposição também resulta em uma componente restante, que representa a variação não explicada pelos componentes sazonais e de tendência.

Ao aplicar a decomposição STL, a série temporal pode ser expressa como a soma dos componentes sazonais, de tendência e restantes. Essa técnica é útil para análise e modelagem de séries temporais, pois proporciona uma compreensão mais clara dos padrões de variação presentes nos dados.

A decomposição STL é formalmente definida como:

$$y_t = f(S_t, T_t, R_t) = \begin{cases} y_t = S_t + T_t + R_t & \text{modelo aditivo} \\ y_t = S_t T_t R_t & \text{modelo multiplicativo} \end{cases} \quad (3.34)$$

3.12 Dickey-Fuller (DF)

De acordo com o Reisen et al. (2017), o teste DF tem as seguintes equações:

$$z_t = y_t + \theta \beta_t, \quad t = 1, \dots, T, \quad (3.35)$$

$$\hat{\rho}_{\text{DF}} - 1 = \frac{\sum_{t=1}^T z_{t-1} \Delta z_t}{\sum_{t=1}^T z_{t-1}^2} \quad (3.36)$$

De (3.36) onde $\Delta z_t = z_t - z_{t-1}$. Sob a hipótese nula (H_0): “ $\rho = 1$ ”, as estatísticas do teste DF e suas distribuições limitantes são dadas da seguinte forma:

$$T(\hat{\rho}_{\text{DF}} - 1) = T \frac{\sum_{t=1}^T z_{t-1} \Delta z_t}{\sum_{t=1}^T z_{t-1}^2} \quad (3.37)$$

e

$$\hat{\tau}_{\text{DF}} = \frac{\hat{\rho}_{\text{DF}} - 1}{\hat{\sigma}_{\text{DF}} \left(\sum_{t=1}^T z_{t-1}^2 \right)^{-1/2}} \quad (3.38)$$

de (3.38) onde $\hat{\sigma}_{\text{DF}}^2 = T^{-1} \sum_{t=1}^T (\Delta z_t - (\hat{\rho}_{\text{DF}} - 1) z_{t-1})^2$. Suponha que $(z_t)_{1 \leq t \leq T}$ são dadas por (3.35), então quando $\rho = 1$,

$$T(\hat{\rho}_{\text{DF}} - 1) \xrightarrow{d} \frac{W(1)^2 - 1}{2 \int_0^1 W(r)^2 dr} - \left(\frac{\theta}{\sigma} \right)^2 \frac{\pi}{\int_0^1 W(r)^2 dr}, \text{ como } T \rightarrow \infty \quad (3.39)$$

$$\hat{\tau}_{\text{DF}} \xrightarrow{d} [1 + 2(\theta/\sigma)^2 \pi]^{-1/2} \left\{ \frac{W(1)^2 - 1}{2 \left(\int_0^1 W(r)^2 dr \right)^{1/2}} - \frac{(\theta/\sigma)^2 \pi}{\left(\int_0^1 W(r)^2 dr \right)^{1/2}} \right\} \quad (3.40)$$

$$\text{como } T \rightarrow \infty \quad (3.41)$$

a partir da equação (3.41), onde \xrightarrow{d} denota convergência na distribuição e onde $\{W(r), r \in [0, 1]\}$ denota o movimento Browniano padrão.

3.13 Teste de Significância

O teste de Friedman classifica os modelos K em cada conjunto de dados em relação ao valor absoluto dos resultados dados por esses algoritmos. A classificação do algoritmo com maior desempenho é 1, e o com menor desempenho é classificado como K . Em seguida, o valor da estatística com base em todas as classificações é calculado como mostrado em equações (3.42) e (3.43) com r_{eu}^j sendo a classificação do desempenho do j -ésimo algoritmo no i -ésimo conjunto de dados. Essa estatística obedece à distribuição do quiquadrado com $K - 1$ graus de liberdade (LIU; XU, 2022).

$$\chi_F^2 = \frac{12N}{K(K+1)} \left[\sum_{j=1}^K R_j^2 - \frac{K(K+1)^2}{4} \right] \quad (3.42)$$

$$R_j = \frac{1}{N} \sum_{i=1}^N r_{ei}^j \quad (3.43)$$

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1)\chi_F^2} \quad (3.44)$$

As estatísticas FF mostrados na equação (3.44) obedecem à distribuição F com graus de liberdade $K-1$ e $(K-1)(N-1)$. Verificando-se a tabela de distribuição F, pode-se obter o valor crítico abaixo do nível de significância especificado (geralmente $\alpha = 0,05$ ou $0,01$). Ao comparar esse valor crítico com o valor calculado com a equação (3.44), a hipótese nula é rejeitada se o valor estatístico F_F é maior que o valor crítico, indicando que há diferenças significativas entre os algoritmos K . Em seguida, pode-se realizar um procedimento post hoc para analisar melhor se o algoritmo de controle é significativamente melhor do que cada algoritmo de referência nos experimentos. Ao contrário, se o valor for menor ou igual ao valor crítico, a hipótese nula é aceita, indicando que não há diferenças significativas entre os algoritmos K .

Adicionalmente, utilizou-se o valor crítico CD (do inglês *Critical Difference*) para determinar se dois classificadores eram significativamente diferentes entre si. O CD foi calculado conforme a fórmula mencionada anteriormente:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \quad (3.45)$$

na equação do CD, q_α representa o valor crítico obtido da Tabela 15 de teste de Nemenyi, k é o número de classificadores e N é o número total de amostras (LIU; XU, 2022).

4 Resultados Preliminares

Neste capítulo é fornecida uma síntese dos resultados obtidos até o momento. Além disso, é apresentado um resumo sucinto quanto a análise de resultados das principais realizações e descobertas que foram obtidos até agora.

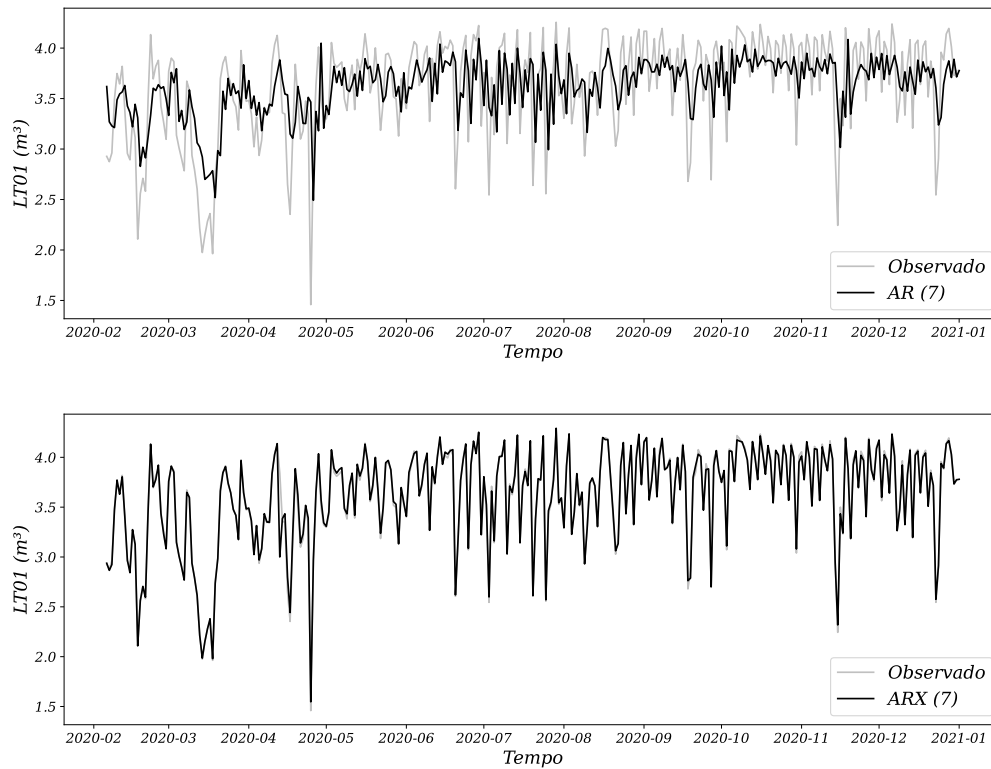
4.1 Análise dos Resultados dos Modelos de Previsão

A Figura 20 apresenta uma previsão de um passo à frente (um dia). Nos apêndices ??, pode-se observar uma comparação entre os modelos AR, MA e ARX. Todos os modelos de ARIMA e seus antecessores são obtidos pelo método da biblioteca do Python “autoARIMA”. Mesmo sendo obtidos parâmetros um pouco distintos dos apresentados aqui e usados nos dados, foi feito um ajuste para que os parâmetros tenham os melhores resultados. Foram utilizados os parâmetros obtidos pelo autoARIMA, que são $(p = 6, d = 0, q = 3)(P = 2, D = 1, Q = 0)_{M=12}$, mas foram ajustados para obter um melhor resultado, sendo $(p = 7, d = 1, q = 7)(P = 2, D = 1, Q = 1)_{M=12}$ para a média de 24 horas. Na Tabela 7, são exibidos todos os modelos obtidos por esse método do “autoARIMA” e ajustados para que obtenham o melhor resultado. p : Ordem do componente AR (*Auto-Regressivo*), d : Número de diferenciações não sazonais, q : Ordem do componente MA (*Média Móvel*), P : Ordem do componente AR sazonal, D : Número de diferenciações sazonais, Q : Ordem do componente MA sazonal, M : Período sazonal (número de observações em um ciclo sazonal). Na Tabela 7 é exibido como cada modelo ficou.

Tabela 7: Parâmetros utilizados nos modelos AR, ARX, MA, ARMA, ARIMA, ARIMAX, SARIMA e SARIMAX obtidos pelo “autoARIMA” do Python.

Modelo	Parâmetros Utilizados	Método de Estimção
AR(p)	$p = 7$	AutoARIMA do Python
ARX(p)	$p = 7$	AutoARIMA do Python
MA(q)	$q = 7$	AutoARIMA do Python
ARMA(p, q)	$p = 7, q = 7$	AutoARIMA do Python
ARIMA(p, d, q)	$p = 7, d = 1, q = 7$	AutoARIMA do Python
ARIMAX(p, d, q)	$p = 7, d = 1, q = 7$	AutoARIMA do Python
SARIMA(p, d, q)(P, D, Q)	$p = 7, d = 1, q = 7, P = 2, D = 1, Q = 1, M = 12$	AutoARIMA do Python
SARIMAX(p, d, q)(P, D, Q, M)	$p = 7, d = 1, q = 7, P = 2, D = 1, Q = 1, M = 12$	AutoARIMA do Python

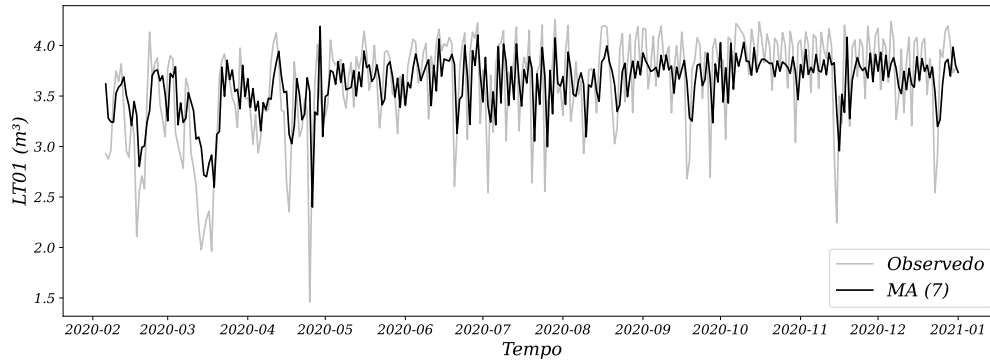
Figura 20: Comparação dos modelos AR e ARX



O modelo MA, quando comparado com um modelo AR de mesma ordem, gera uma previsão usando o método de médias móveis, enquanto o modelo AR é um modelo

regressivo que observa os dados no passado. O modelo MA, por outro lado, observa os erros que esses dados podem cometer. A Figura 21 mostra que essa previsão se assemelha ao modelo apresentado na Figura 20, embora não seja comparável ao modelo exibido na Figura 20.

Figura 21: Modelo MA(7)



A Figura 22 combina os modelos AR e MA em um modelo ARMA. Essa abordagem pode levar a uma redução significativa no erro de previsão, como observado nos apêndices ?? e ??, onde são apresentadas comparações com um número de passos de previsão de 1, 7, 14 e 30 passos à frente. Ao analisar a Figura 23, não se nota uma diferença significativa nos erros sMAPE, MAE e RRMSE, conforme indicado no apêndice ??. Pode-se notar que esta afirmação sugere que os modelos não têm muita diferenciação entre si em comparação com os outros métodos apresentados anteriormente. A análise sugere que o método ARX ainda parece ser superior aos demais.

Figura 22: ARMA (7,7)

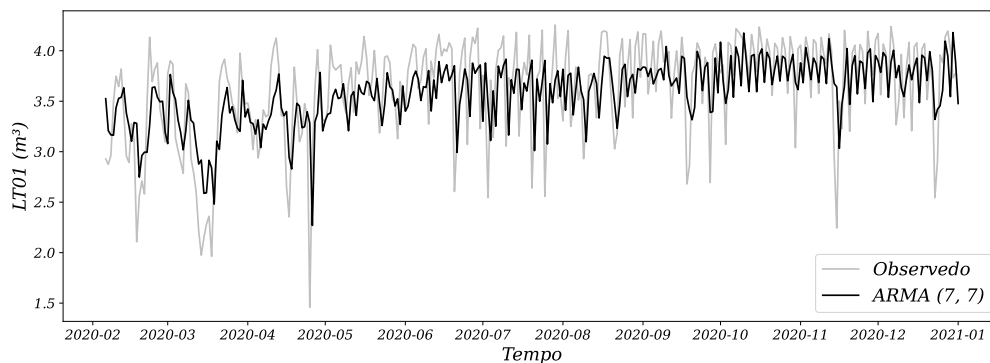
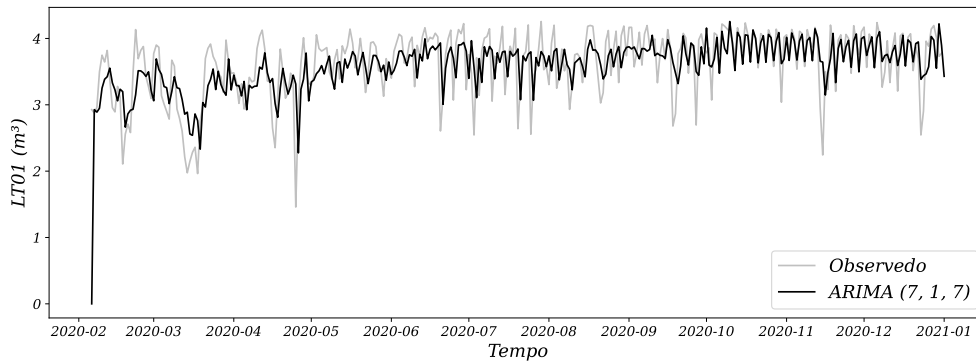
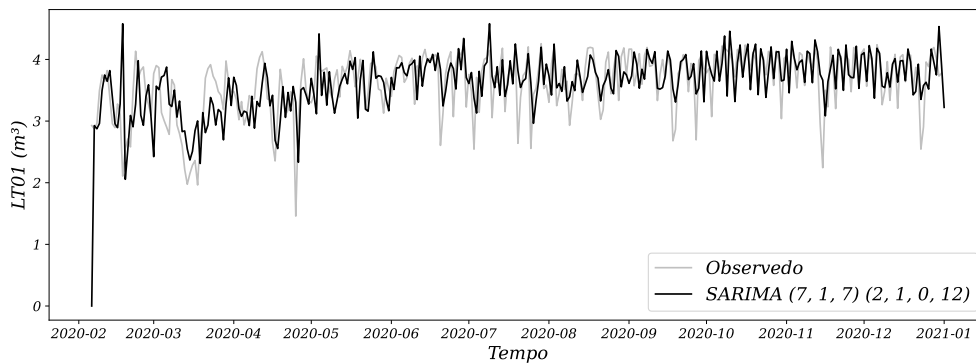


Figura 23: ARIMA (7,1,7)

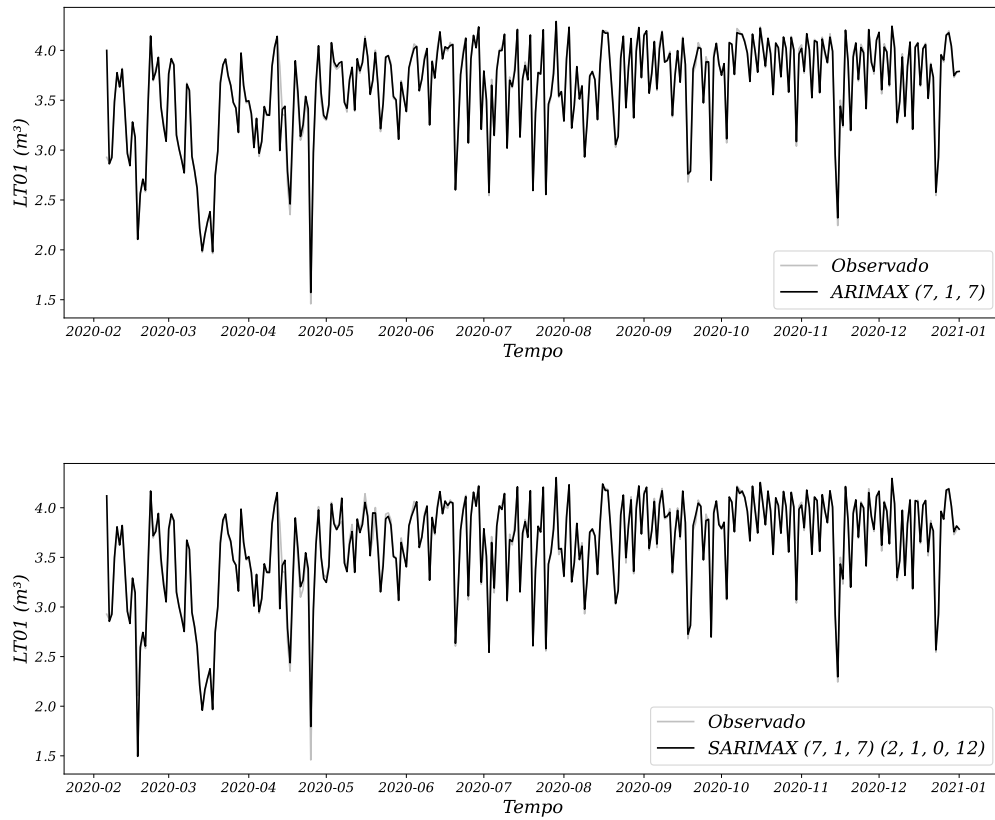


Na Figura 24, é possível observar que a previsão em vermelho está próxima dos valores observados em preto, mostrando que a inclusão do componente de sazonalidade melhora a qualidade da previsão. Os modelos SARIMA são capazes de lidar com dados que apresentam padrões sazonais, permitindo a diferenciação dos dados em termos de componentes sazonais e não sazonais. Uma abordagem útil para determinar os melhores parâmetros do modelo é utilizar uma estrutura de pesquisa automatizada de parâmetros, com a biblioteca do Python “autoARIMA”, que auxilia na identificação dos parâmetros para o modelo SARIMA.

Figura 24: SARIMA $(p, q, d)(P, Q, D) = (7, 1, 7)(2, 1, 1)_{12}$ 

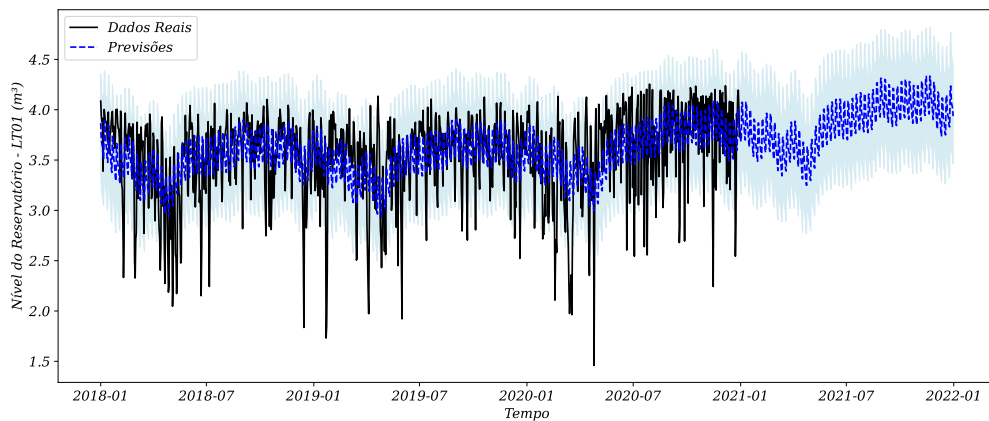
Entre os modelos com variáveis exógenas, como os modelos ARX, ARIMAX e SARIMAX mostrado na Figura 25, observa-se uma melhora na qualidade das previsões em comparação com os modelos que não incluem variáveis exógenas. A adição dessas variáveis externas permite capturar melhor a influência e os padrões presentes nos dados, resultando em previsões mais adequadas. Essa inclusão de informações adicionais contribui para uma compreensão mais abrangente do comportamento da série temporal e possibilita uma melhor adaptação do modelo aos padrões observados.

Figura 25: Comparação entre ARIMAX e SARIMAX



Na Figura 26 é apresentada a previsão da variável LT01 usando o Prophet. Este modelo exibe diversas informações, como a sazonalidade dos dados e a tendência. O Prophet é considerado um modelo proposto recentemente que pode ser comparado com ARIMA.

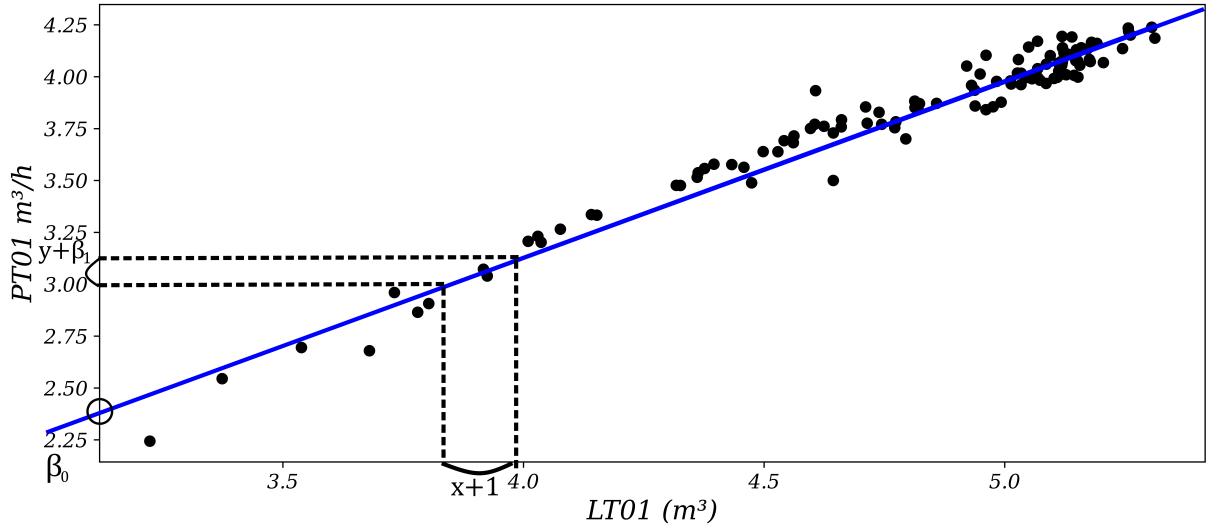
Figura 26: Previsões do modelo Prophet para o reservatório LT01



A Figura 27 fornece uma representação dos coeficientes β_0 e β_1 . Um aumento de 1

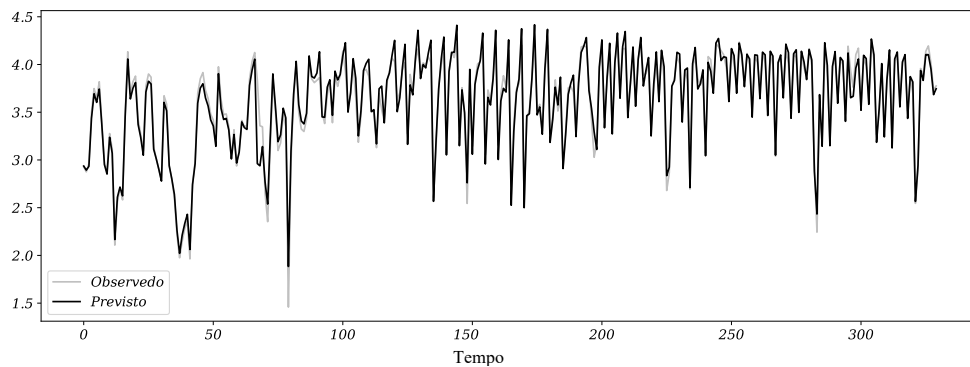
na variável x está associado a um aumento proporcional de β_1 na variável y . O valor de β_0 representa o valor de y quando x é igual a 0.

Figura 27: Regressão linear LT01 vs PT01 correlação 98%



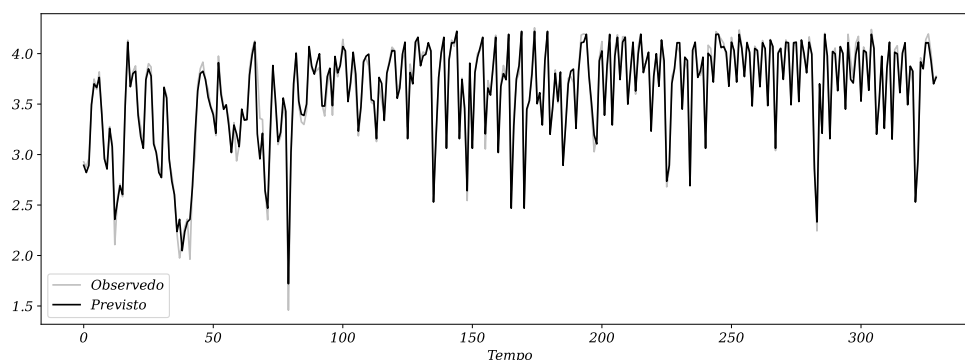
A Figura 28, um passo à frente dos dados da SANEPAR foi previsto no modelo LR. Esse modelo mais simples do que os outros modelos pode ser útil em horizonte menor, mas em horizontes de vários dias à frente o desempenho do modelo é degradado. Ele é um dos modelos que apresenta erros acentuados na análise das métricas sMAPE, MAE e RRMSE.

Figura 28: Regressão linear (LR) um passo a frente



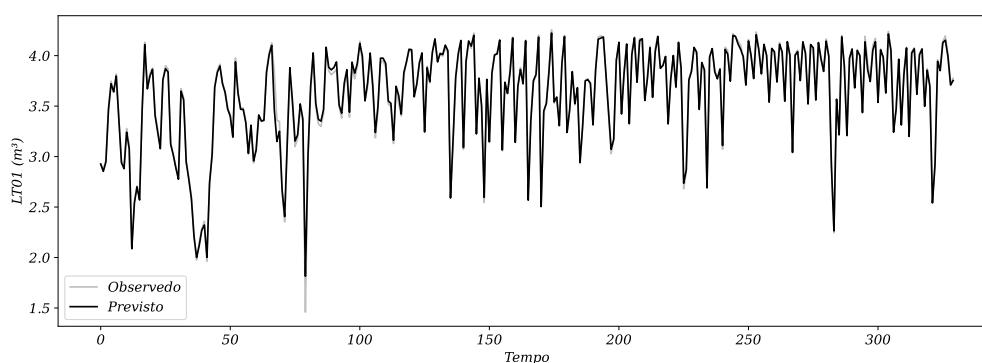
A Figura 29, é apresentado este modelo com o intuito de diminuir o erro de previsão que o LR estava tendo em prever mais de um dia à frente. Este modelo, por ser mais robusto, consegue trabalhar na otimização dos hiperparâmetros, tornando-o melhor que o LR na questão de horizontes mais longos.

Figura 29: Regressor de Árvore de Decisão



Na Figura 30, o modelo RFR apresentado em comparação com os outros modelos, e ele se destaca em relação aos modelos anteriores já analisados em um passo à frente, mas pelas Tabelas 10 à 13 pode notar que esse modelo RFR não tem grandes melhoras em relação aos outros.

Figura 30: Regressão da Floresta Aleatória (RFR)



Na Figura 31, são apresentados os modelos XGBoost e LightGBM. Esses modelos tem como parâmetros e hiperparâmetros obtidos na otimização do Optuna, uma biblioteca do Python para otimização de parâmetros e hiperparâmetros como mostrado na Tabela 8 a otimização dos parâmetros dos modelos XGBoost, LightGBM, RFR e DTR. Esses modelos, devido à sua semelhança, exibem tempos de desempenho próximos um do outro. Na Figura 31, é exibida uma previsão de um passo à frente, e observa-se que esse modelo parece ser superior aos demais modelos previamente apresentados. Por outro lado, na Figura 31, apesar de ser um modelo mais rápido em termos de tempo de computação, ele ainda não atingiu o desempenho esperado para o tema abordado.

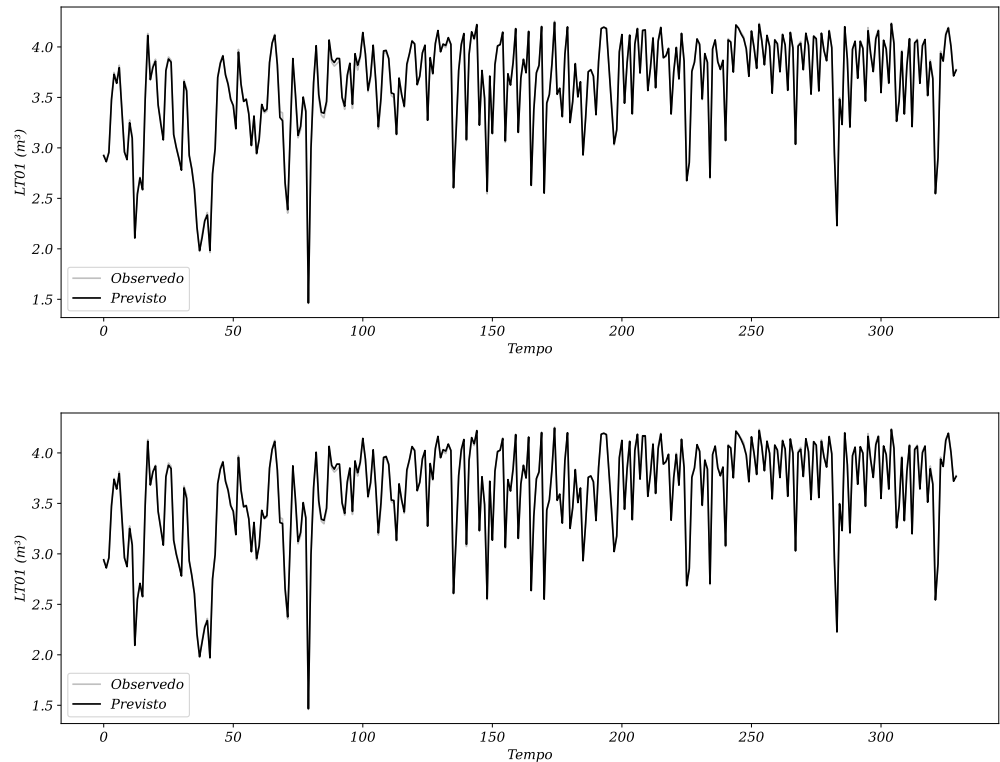
Os modelos de rede neural, como RNN, ANN, CNN, GRU, LSTM e Transformer, obtidos na otimização do Optuna do Python, tiveram seus hiperparâmetros melhorados, conforme exibido na Tabela 9. Esses modelos, por serem modelos de rede neural, são

Tabela 8: Hiperparâmetros dos modelos

Modelo		Estimadores	Profund. Máxima	Min. Amostras Divisão	Min. Amostras por Folha	Máx. Re-cursos	Taxa de Aprendizado
XGB	Re- gressor	835	4	10	2	“sqrt”	0,03495
LGBM	Re- gressor	741	4	10	1	“auto”	0,03492
Random Forest	Re- gressor	996	9	3	1	None	N/A
Decision Tree	Re- gressor	N/A	10	10	4	None	N/A

melhores para otimizar do que os outros. Estão detalhados nas Tabelas 10 à 13. Os parâmetros do modelo RNN é mostrado no Apêndice ??, onde uma comparação com os horizontes de previsão escolhidos é apresentada na Figura ??.

Figura 31: Resultados da regressão utilizando XGBoost e LightGBM



Os rótulos dos modelos foram incluídos nas tabelas, o que permite uma identificação clara e organizada das diferentes abordagens de previsão utilizadas. Esses rótulos facilitam a compreensão e a referência aos modelos ao longo do estudo, proporcionando

Tabela 9: Resumo dos Hiperparâmetros dos Modelos de Redes Neurais

Modelo	Unidades/ Layers	Cabeças/ Dimen- sões	Tamanho do Batch	Épocas	Dropout/ Lear- ning Rate	Outros Parâme- tros
LSTM	36	Não especi- ficado	32	134	Não especifi- cado	Não especi- ficado
GRU	Não especi- ficado	Não especi- ficado	32	50	Não especifi- cado	Não especi- ficado
Transformers	Não especi- ficado	8 cabeças, 217; 433	Não especi- ficado	50	Não especifi- cado	2 camadas
RNN	119	Não especi- ficado	16	50	Não especifi- cado	Não especi- ficado
CNN	Não especi- ficado	Não especi- ficado	58	63	0,267; 0,0006458	Kernel: 5, Densas: 1, Verbosi- dade: 0
ANN	109	Não especi- ficado	31	98	0,209, 0,00282	Densas: 1, Verbosi- dade: 1

uma estrutura coerente para a apresentação dos resultados.

$$(p = 7, d = 1, q = 7)(P = 2, D = 1, Q = 1)_{M=12} \text{ Média 24h}$$

- A** AR
- B** ARX
- C** MA
- D** ARMA
- E** ARIMA
- F** SARIMA
- G** ARIMAX
- H** SARIMAX
- I** Decision Tree Regressor
- J** Random Forest Regressor
- K** XGBRegressor
- L** LGBMRegressor
- M** LSTM
- N** GRU
- O** Prophet
- P** RNN
- Q** Transformer
- R** CNN
- S** ANN

Tabela 10: Comparação dos modelos de previsão com as métricas de desempenho **treino**

		Modelos Treino																		
Horizontes	Métricas	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1 dia à frente	sMAPE	4,56	5,74	4,42	4,78	4,50	5,09	5,74	5,73	6,03	8,44	8,50	9,12	25,49	30,17	2,01	0,34	9,33	17,63	17,63
	MAE	0,31	0,38	0,30	0,32	0,30	0,34	0,38	0,38	0,34	0,62	0,62	0,68	0,99	1,21	0,08	<i>0,01</i>	0,31	0,58	0,58
	RRMSE	0,12	0,15	0,12	0,12	0,12	0,13	0,15	0,15	0,14	0,19	0,19	0,20	2,55	0,43	0,08	<i>0,00</i>	0,16	0,18	0,18
7 dias à frente	sMAPE	4,46	5,74	4,95	4,84	5,13	5,40	5,76	5,76	4,98	9,56	9,77	9,12	35,84	84,93	3,43	0,08	9,34	17,73	17,73
	MAE	0,30	0,38	0,33	0,32	0,34	0,36	0,38	0,38	0,43	0,71	0,73	0,68	1,49	5,12	0,13	<i>0,00</i>	0,31	0,58	0,58
	RRMSE	0,11	0,15	0,13	0,12	0,13	0,14	0,15	0,15	0,11	0,24	0,24	0,20	3,67	1,56	0,15	<i>0,00</i>	0,16	0,18	0,18
14 dias à frente	sMAPE	5,02	6,08	5,05	5,17	5,27	5,51	6,09	6,10	4,98	9,50	9,80	9,12	36,25	99,91	9,49	0,14	9,33	16,76	16,76
	MAE	0,33	0,40	0,34	0,34	0,35	0,37	0,40	0,40	0,43	0,70	0,73	0,68	1,51	6,94	0,34	<i>0,00</i>	0,31	0,55	0,55
	RRMSE	0,13	0,16	0,13	0,13	0,14	0,14	0,16	0,16	0,11	0,23	0,24	0,20	3,72	2,10	0,49	<i>0,00</i>	0,16	0,18	0,18
30 dias à frente	sMAPE	5,73	6,58	5,67	5,71	5,92	6,08	6,60	6,62	4,98	9,34	9,54	9,12	35,38	97,81	8,45	0,09	9,31	15,85	15,85
	MAE	0,38	0,43	0,38	0,38	0,40	0,41	0,43	0,43	0,43	0,69	0,71	0,68	1,47	6,65	0,31	<i>0,00</i>	0,31	0,53	0,53
	RRMSE	0,15	0,17	0,15	0,15	0,15	0,15	0,17	0,17	0,11	0,23	0,23	0,20	3,62	2,01	0,39	<i>0,00</i>	0,16	0,17	0,17

Tabela 11: Comparação dos modelos de previsão com as métricas de desempenho teste

Modelos Teste																				
Horizontes	Métricas	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1 dia à frente	sMAPE	4,75	6,69	4,86	4,92	4,96	5,69	6,69	6,74	7,58	6,75	6,80	7,39	21,61	25,86	2,01	0,32	11,78	13,97	13,97
	MAE	0,33	0,46	0,34	0,34	0,34	0,40	0,46	0,46	0,49	0,49	0,49	0,54	0,84	1,04	0,08	<i>0,01</i>	0,41	0,48	0,48
	RRMSE	0,12	0,16	0,12	0,12	0,12	0,14	0,16	0,16	0,17	0,16	0,16	0,17	1,97	0,40	0,08	<i>0,00</i>	0,18	0,18	0,18
7 dias à frente	sMAPE	5,54	7,01	5,80	5,58	5,68	6,25	7,03	7,05	6,56	7,87	7,96	7,39	32,56	81,89	3,43	0,08	11,80	14,34	14,34
	MAE	0,38	0,48	0,40	0,38	0,39	0,43	0,48	0,48	0,58	0,58	0,59	0,54	1,37	4,98	0,13	<i>0,00</i>	0,41	0,50	0,50
	RRMSE	0,14	0,17	0,15	0,14	0,14	0,15	0,17	0,18	0,15	0,21	0,21	0,17	2,95	1,50	0,15	<i>0,00</i>	0,18	0,18	0,18
14 dias à frente	sMAPE	5,50	5,74	5,61	5,06	4,97	5,30	5,73	5,72	6,56	7,80	7,94	7,39	32,99	97,30	9,49	0,18	11,78	13,76	13,76
	MAE	0,38	0,38	0,39	0,34	0,34	0,36	0,38	0,38	0,58	0,57	0,58	0,54	1,39	6,81	0,34	<i>0,01</i>	0,41	0,48	0,48
	RRMSE	0,14	0,15	0,14	0,13	0,13	0,14	0,15	0,15	0,15	0,21	0,21	0,17	3,00	2,02	0,49	<i>0,00</i>	0,18	0,18	0,18
30 dias à frente	sMAPE	5,43	6,76	5,55	5,55	5,62	6,15	6,72	6,76	6,56	7,73	7,78	7,39	32,12	95,24	8,45	0,13	11,76	15,86	15,86
	MAE	0,37	0,46	0,38	0,38	0,39	0,43	0,46	0,46	0,58	0,57	0,57	0,54	1,34	6,54	0,31	<i>0,00</i>	0,41	0,55	0,55
	RRMSE	0,14	0,17	0,14	0,14	0,14	0,16	0,17	0,17	0,15	0,21	0,21	0,17	2,91	1,95	0,39	<i>0,00</i>	0,18	0,19	0,19

Tabela 12: Comparação dos modelos de previsão com as métricas de desempenho **validação**

		Modelos Validação																		
Horizontes	Métricas	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1 dia à frente	sMAPE	4,07	5,08	3,98	4,21	4,02	4,68	5,07	5,15	5,00	7,74	7,78	8,53	19,95	22,35	2,01	0,33	7,61	11,41	11,41
	MAE	0,28	0,35	0,28	0,30	0,28	0,33	0,35	0,36	0,34	0,58	0,59	0,65	0,77	0,88	0,08	<i>0,01</i>	0,27	0,39	0,39
	RRMSE	0,10	0,13	0,10	0,10	0,10	0,12	0,13	0,13	0,11	0,18	0,18	0,19	2,54	0,29	0,08	<i>0,00</i>	0,10	0,12	0,12
7 dias à frente	sMAPE	3,52	4,58	3,86	3,94	4,19	4,54	4,57	4,61	4,33	8,35	8,56	8,53	33,45	81,77	3,43	0,08	7,62	11,13	11,13
	MAE	0,25	0,32	0,27	0,27	0,29	0,32	0,32	0,32	0,38	0,63	0,65	0,65	1,42	4,92	0,13	<i>0,00</i>	0,27	0,39	0,39
	RRMSE	0,09	0,12	0,10	0,10	0,11	0,12	0,12	0,12	0,10	0,20	0,20	0,19	4,38	1,42	0,15	<i>0,00</i>	0,10	0,12	0,12
14 dias à frente	sMAPE	3,79	4,44	3,81	4,54	4,25	4,46	4,43	4,43	4,33	8,27	8,59	8,53	33,94	97,96	9,49	0,14	7,61	12,52	12,52
	MAE	0,26	0,31	0,27	0,32	0,30	0,31	0,31	0,31	0,38	0,63	0,65	0,65	1,44	6,83	0,34	<i>0,00</i>	0,27	0,43	0,43
	RRMSE	0,10	0,12	0,10	0,11	0,11	0,11	0,12	0,12	0,10	0,20	0,20	0,19	4,45	1,97	0,49	<i>0,00</i>	0,10	0,13	0,13
30 dias à frente	sMAPE	4,44	4,33	4,37	4,35	4,88	4,82	4,31	4,32	4,33	8,11	8,32	8,53	33,08	96,24	8,45	0,09	7,59	12,33	12,33
	MAE	0,31	0,30	0,31	0,30	0,34	0,34	0,30	0,30	0,38	0,61	0,63	0,65	1,40	6,60	0,31	<i>0,00</i>	0,27	0,42	0,42
	RRMSE	0,11	0,12	0,11	0,11	0,12	0,12	0,12	0,12	0,10	0,20	0,20	0,19	4,32	1,90	0,39	<i>0,00</i>	0,10	0,13	0,13

Tabela 13: Comparação dos modelos de previsão com as métricas de desempenho **inteiro**

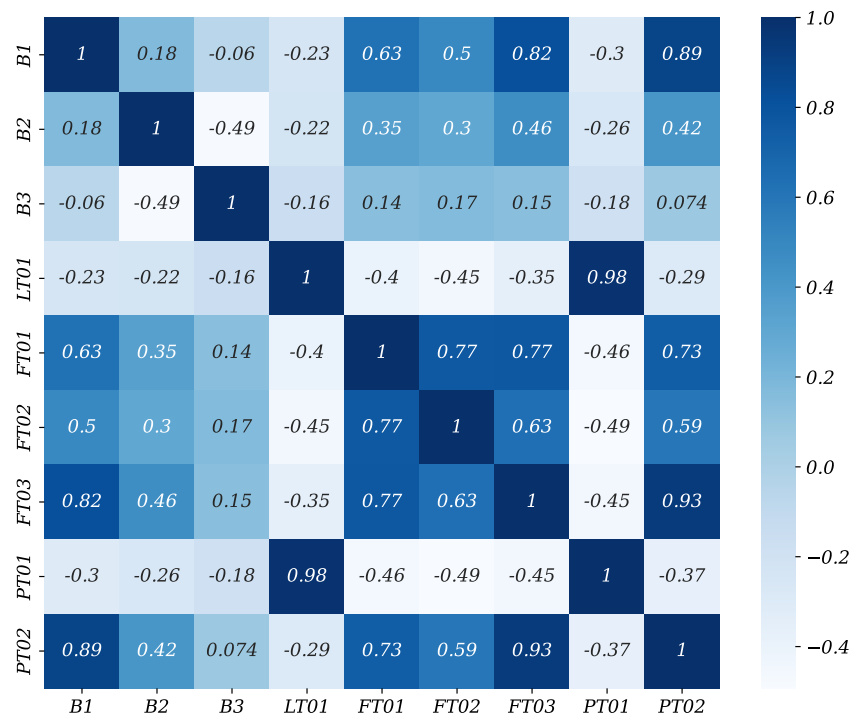
Modelos inteiros																				
Horizontes	Métricas	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1 dia à frente	sMAPE	4,44	5,98	4,43	4,63	4,51	5,11	5,87	5,95	6,36	7,83	7,89	8,51	23,38	27,41	2,01	0,33	9,83	17,63	17,63
	MAE	0,30	0,40	0,30	0,32	0,31	0,35	0,39	0,40	0,39	0,57	0,58	0,63	0,91	1,09	0,08	<i>0,01</i>	0,34	0,58	0,58
	RRMSE	0,11	0,15	0,11	0,12	0,12	0,13	0,15	0,15	0,15	0,18	0,18	0,19	2,30	0,40	0,08	<i>0,00</i>	0,16	0,18	0,18
7 dias à frente	sMAPE	4,67	6,06	5,14	4,61	4,66	5,64	5,99	6,03	5,36	8,88	9,06	9,94	34,52	83,48	3,43	0,08	9,84	17,73	17,73
	MAE	0,32	0,41	0,35	0,31	0,32	0,39	0,40	0,41	0,47	0,66	0,67	0,75	1,44	5,04	0,13	<i>0,00</i>	0,34	0,58	0,58
	RRMSE	0,12	0,16	0,13	0,12	0,12	0,14	0,15	0,16	0,12	0,22	0,23	0,25	3,45	1,52	0,15	<i>0,00</i>	0,16	0,18	0,18
14 dias à frente	sMAPE	4,95	5,92	5,00	4,62	4,74	5,37	5,86	5,90	5,36	8,81	9,07	10,17	34,94	98,84	9,49	0,15	9,83	16,76	16,76
	MAE	0,33	0,39	0,34	0,31	0,32	0,36	0,39	0,39	0,47	0,65	0,67	0,77	1,47	6,89	0,34	<i>0,01</i>	0,34	0,55	0,55
	RRMSE	0,13	0,15	0,13	0,12	0,12	0,14	0,15	0,15	0,12	0,22	0,23	0,25	3,50	2,06	0,49	<i>0,00</i>	0,16	0,18	0,18
30 dias à frente	sMAPE	5,40	6,68	5,36	5,66	5,56	5,93	6,63	6,67	5,36	8,69	8,84	9,99	34,08	96,83	8,45	0,10	9,81	15,85	15,85
	MAE	0,37	0,45	0,36	0,39	0,38	0,40	0,44	0,45	0,47	0,64	0,66	0,75	1,42	6,61	0,31	<i>0,00</i>	0,33	0,53	0,53
	RRMSE	0,14	0,17	0,14	0,14	0,14	0,15	0,17	0,17	0,12	0,22	0,22	0,25	3,40	1,98	0,39	<i>0,00</i>	0,16	0,17	0,17

4.1.1 Análise Exploratória dos Dados (EDA)

A partir do passo **Etap**a 1, foi realizado o EDA (do inglês *Exploratory Data Analysis*) para processar os dados obtidos até o momento. O EDA permite responder às questões de pesquisa levantadas. Conforme mencionado por Yu (2016), na era do *big data*, é desafiador descobrir as regras, modelos analíticos e hipóteses por trás dos volumes massivos de dados caóticos, não estruturados e multimídia coletados por meio de vários canais. A análise exploratória de dados foi promovida por John Tukey como uma abordagem para explorar os dados, resumir suas principais características e formular hipóteses que possam direcionar a coleta adicional de dados e experimentos. No contexto de análises de dados, várias técnicas de EDA têm sido adotadas.

Na Figura 32 que ambas as variáveis apresentam uma correlação quase perfeita, com um coeficiente de correlação de Pearson (r) igual a 1. Portanto, para responder a essa pergunta, basta observar a correlação de Pearson na Figura 32. Na análise de correlação de Pearson, todo o conjunto de dados da SANEPAR foi utilizado para que cada variável pudesse ser avaliada quanto à sua correlação com a variável LT01. Se houvesse correlação, a variável era usada como variável exógena para melhorar os modelos. Como o modelo de LR consegue relacionar apenas uma ou duas variáveis entre si, no modelo de LR, LT01 foi usada como variável independente e PT01 como variável dependente.

Figura 32: Correlação de Pearson



A Figura 32 ilustra a correlação entre as variáveis no conjunto de dados em questão. Essa imagem representa graficamente a relação entre as variáveis e é usada para evidenciar a existência de uma correlação forte no valor de 0,98 é considerada forte entre elas, quanto mais próximo do valor 1 a correlação é sempre forte e 0,9 para mais ou para menos indica uma correlação muito forte, 0 a 0,3 positivo ou negativo indica uma correlação desprezível.

Na Tabela 14, o desvio padrão é representado pela sigla STD, que corresponde à expressão em inglês *standard deviation*. Além disso, em resposta à pergunta ??, é importante mencionar que, assim como em qualquer empresa de tratamento de água, é utilizado um mecanismo de acionamento automático denominado “trava de segurança” para evitar que o nível do tanque se aproxime de zero e haja falta de água nos locais abastecidos por esse tanque. O nível mínimo que o tanque pode alcançar é de $5,29m^3$ (equivalente a 5.285,90 litros). As bombas são ativadas em sua potência máxima para evitar que sejam acionadas quando o nível do tanque. No entanto, a bomba 1 ainda estaria operando para completar o nível do tanque caso ele esteja dentro dessa faixa.

Em situações de demanda de pico, uma abordagem ideal, embora não necessariamente a mais econômica, seria ter um tanque de reserva adicional e instalar uma tubulação que os conecte. Durante o dia, ambos os tanques seriam abastecidos e, à noite, por meio

Tabela 14: Descrição estatística dos dados com o filtro aplicado das 18h às 21h

18 a 21h	B1	B2	B3	LT01	FT01	FT02	FT03	PT01	PT02
Contagem	4385	4385	4385	4385	4385	4385	4385	4385	4385
Média	51,94	27,81	6,41	3,24	112,68	132,93	112,41	4,11	20,80
STD	17,14	17,61	16,77	0,70	132,59	44,78	31,33	0,76	6,14
Min	0	0	0	0,29	0	0	0	0,88	0
25%	57,84	0	0	2,79	0,12	123,96	111,66	3,62	21,72
50%	57,99	34,91	0	3,30	0,12	136,00	118,82	4,15	22,05
75%	57,99	38,02	0	3,78	264,27	148,20	125,63	4,66	23,02
Max	59,99	59,99	59,99	4,40	383,87	326,17	194,35	5,68	28,08

da ação da gravidade, eles manteriam o mesmo nível até que o consumo atinja um ponto em que as bombas sejam acionadas. Essa estratégia permite um abastecimento contínuo de água.

4.1.2 Múltiplas Entradas e Saída Única (MISO)

Na etapa **Etapa 2**, foi explorado o modelo MISO (do inglês *Multiple Inputs, Single Output*) na dissertação. O modelo ARIMA, juntamente com suas variantes e extensões, foi amplamente estudado durante a pesquisa, assim como modelos regressivos que envolvem múltiplas variáveis de entrada e uma variável de saída, neste caso, a LT01. As demais variáveis foram utilizadas como suporte para melhorar o modelo do tipo ARIMAX ou modelos com variáveis exógenas. Quando aplicado sem o uso de variáveis exógenas, o modelo ARIMA apresenta apenas uma entrada, semelhante ao modelo de LR. No entanto, ao incluir variáveis exógenas, o modelo se torna MISO, permitindo uma modelagem abrangente e considerando a interação de várias variáveis para prever a variável de interesse.

4.1.3 Decomposição STL

Através da decomposição, é possível analisar se a série apresenta tendência, sazonalidade e resíduos. Ao observar as Figuras 33 e 34, é evidente que os dados exibem ambos os padrões. Isso indica que a série é estacionária, como confirmado pelo seguinte teste.

Teste de Dickey-Fuller (DF) Aumentado: Estatística de teste ADF: $-4,25$; Valor de p : $0,001$; Atrasos utilizados: 21 ; Observações: 1074 ; Valor crítico (1%): $-3,44$; Valor crítico (5%): $-2,86$; Valor crítico (10%): $-2,57$; Com base na forte evidência contra a hipótese nula, podemos rejeitar a hipótese nula. A Figura 35, podemos notar um aumento na demanda durante essas horas durante o ano de 2019.

Conforme mencionado na subseção 1.1.1, as anomalias climáticas ocorridas em

Figura 33: Decomposição STL aditiva dos dados coletados

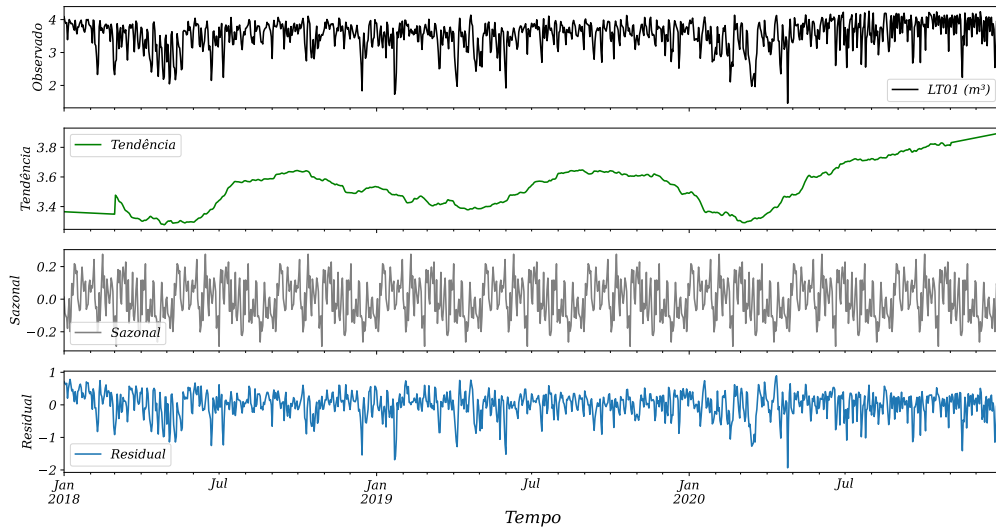
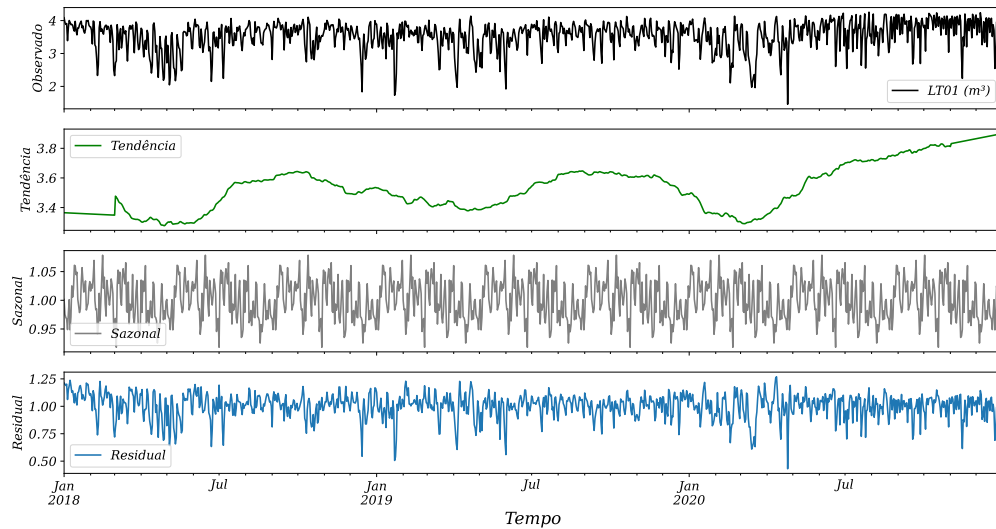


Figura 34: Decomposição STL multiplicativa dos dados coletados



2020, especialmente a falta de chuvas e devido ao COVID-19, tiveram um impacto significativo nos resultados. Isso contribuiu para as mudanças observadas na demanda de água ao longo desse período.

A Figura 36 ilustra como a vazão pode ser afetada pelo nível do tanque. É interessante observar que a vazão de recalque tem um impacto mais significativo no nível do tanque em comparação com as outras vazões. Isso ocorre porque a vazão de recalque está associada à injeção de água diretamente no tanque por meio da bomba localizada próxima à base do tanque. Por outro lado, as demais vazões apresentam alguns valores ausentes, o que limita sua influência na análise geral.

Figura 35: Violino no nível do reservatório

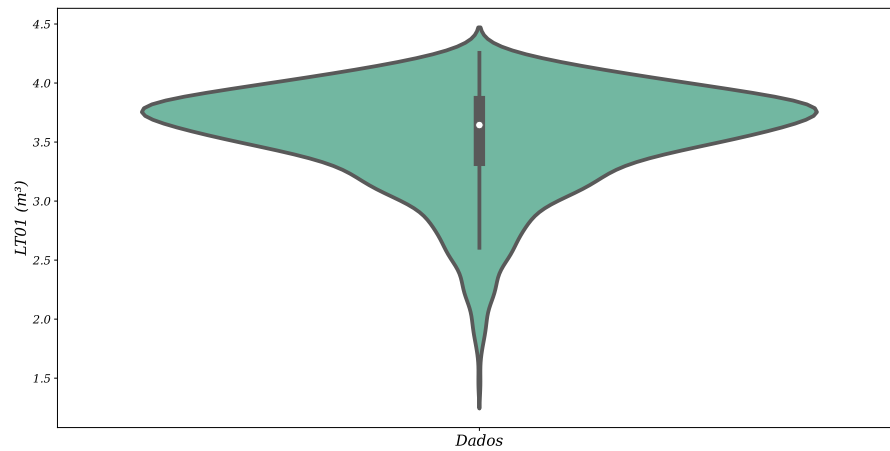
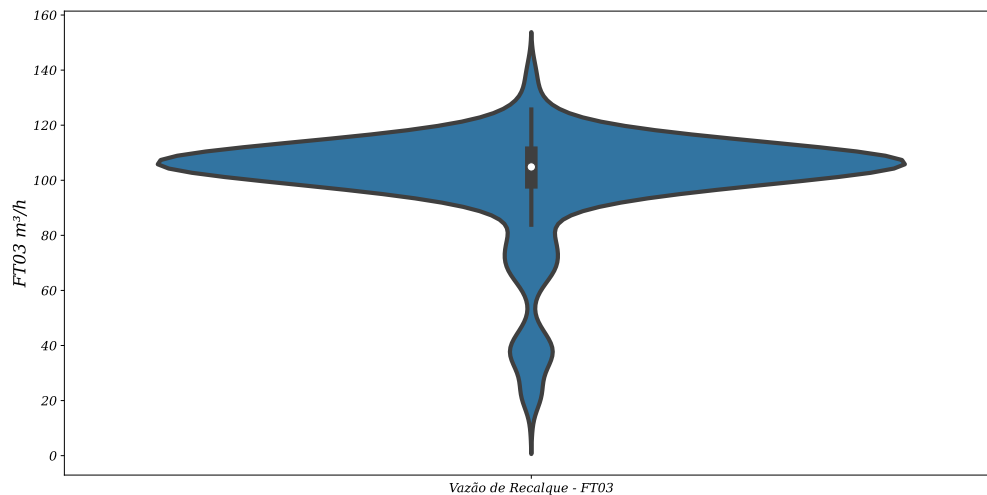


Figura 36: Violino da vazão de recalque



O ACF (do inglês *Auto-Correlation Function*) é uma medida estatística utilizada para identificar a presença de correlação serial em uma série temporal. Ele calcula a autocorrelação entre os valores da série em diferentes defasagens, ou seja, a correlação entre os valores atuais e os valores passados da série.

O ACF é útil para analisar a dependência temporal dos dados e identificar padrões de sazonalidade, tendência ou outros efeitos temporais. Por meio do ACF, é possível avaliar se a série exibe autocorrelação significativa em defasagens específicas, o que pode indicar a presença de não estacionariedade ou estrutura temporal que precisa ser considerada na análise ou modelagem da série temporal.

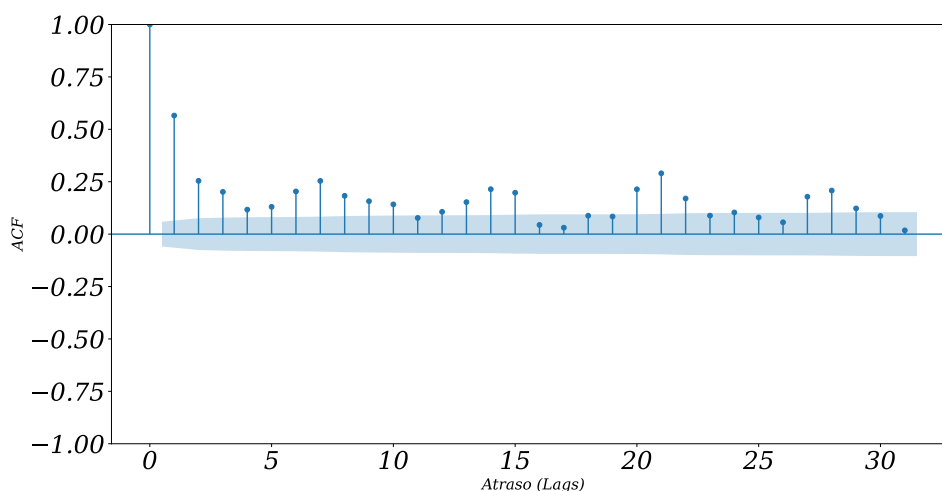
A estatística ADF (do inglês *Augmented Dickey-Fuller*) de $-4,27$ indica a evidência de estacionariedade na série temporal. Quanto mais negativo for o valor da estatística

ADF, maior é a evidência de estacionariedade nos dados.

O valor-p de 0,0005, por sua vez, está associado ao teste ADF. O valor-p é uma medida estatística que representa a probabilidade de obter um resultado igual ou mais extremo do que o observado, sob a suposição de que a hipótese nula seja verdadeira. No caso do teste ADF, a hipótese nula é a presença de raiz unitária na série temporal, o que indica não estacionariedade. Assim, um valor-p baixo (geralmente abaixo de um nível de significância predefinido, como 0,05) sugere que a série temporal é estacionária, enquanto um valor-p alto sugere que a série temporal é não estacionária. Neste caso, o valor-p de 0,0005 é bastante baixo, o que indica forte evidência contra a hipótese nula e sugere que a série temporal é estacionária.

Na Figura 37, pode-se observar a diferença entre a autocorrelação (ACF) exibida na Figura 37 e a autocorrelação parcial (PACF) exibida na Figura 38. A autocorrelação é uma medida da correlação entre os valores da série temporal em diferentes defasagens, levando em consideração tanto a correlação direta quanto a correlação indireta. Por outro lado, a autocorrelação parcial mede apenas a correlação direta entre os valores, desconsiderando a influência das defasagens intermediárias. Essas análises são úteis para identificar padrões e relações de dependência entre os valores da série temporal, fornecendo informações importantes para a modelagem e previsão desses dados.

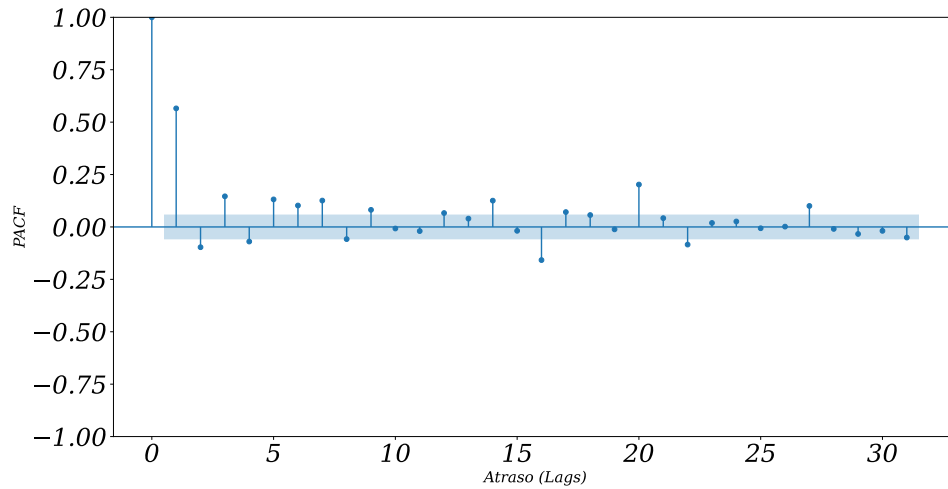
Figura 37: Autocorrelação



O intervalo de confiança padrão de 95% é representado pela marca azul na Figura. As observações que estão fora desse intervalo são consideradas estatisticamente correlacionadas, indicando a presença de padrões ou estrutura na série temporal.

A correlação visualizada na Figura 37 é fundamental para a interpretação do teste DF. Em uma série de ruído branco, os valores são completamente aleatórios e não apresentam correlação significativa. Portanto, quando há correlação presente na série, isso

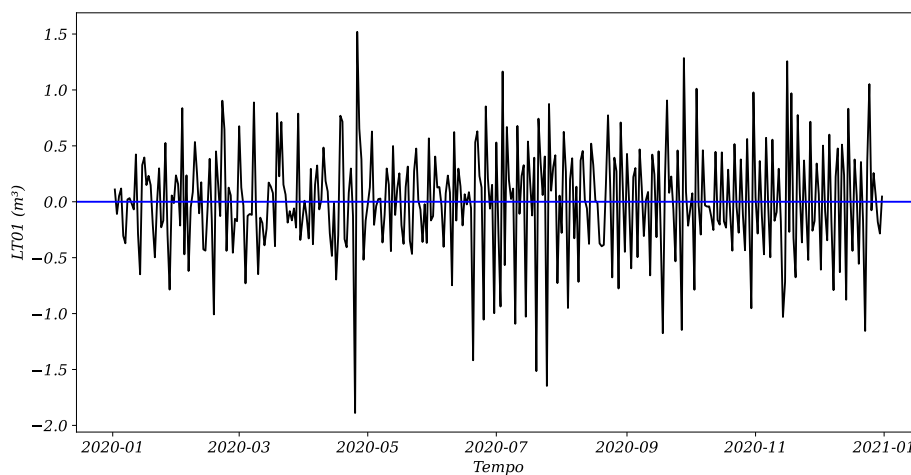
Figura 38: Autocorrelação parcial



indica a existência de padrões ou dependências entre os valores, o que pode ser explorado para a modelagem e previsão da série temporal.

Na Figura 39, é possível observar uma série temporal que pode ser caracterizada como ruído branco. Uma série temporal é considerada ruído branco se suas variáveis forem independentes e distribuídas de forma idêntica, com média zero. Isso implica que todas as variáveis possuem a mesma variância (σ^2) e que cada valor não possui correlação com os demais valores da série.

Figura 39: Ruído branco



4.1.4 Separação dos Dados

Na etapa **Etapa 4**, os dados foram divididos em conjuntos de treinamento, teste e validação. Essa prática é comum entre profissionais de aprendizado de máquina, pois

permite avaliar o desempenho do modelo em conjuntos de dados diferentes.

Quanto à divisão dos dados, foi adotada uma estratégia básica em que 70% dos dados foram destinados ao conjunto de treinamento e os 30% restantes foram reservados para o conjunto de teste. Dentro dos 70% de treinamento, foi realizada uma subdivisão em que 80% desses dados foram usados novamente para treinamento e os 20% restantes foram utilizados para validação. Essa abordagem foi implementada em linguagem de programação para facilitar o processo e evitar a necessidade de recalculá-la a cada modificação do modelo.

4.1.5 Modelagem e Seleção do Modelo

A estratégia recursiva é mencionada por Petropoulos et al. (2022) como uma abordagem eficaz na previsão de séries temporais de múltiplos passos. De acordo com o autor, essa estratégia envolve o uso de previsões anteriores como entradas para prever os próximos passos da série temporal. A abordagem recursiva tem demonstrado potencial para melhorar a acurácia das previsões de séries temporais de longo prazo.

Na Etapa **Etapa 5**, discute-se a previsão dos dados em uma janela de horizonte de previsão estendida, abrangendo diferentes períodos de tempo, como um dia, uma semana, duas semanas e um mês. Essa estratégia de previsão recorrente permite a comparação entre modelos de regressão e modelos ARIMA em diferentes horizontes temporais.

Essa abordagem é vantajosa, pois cada modelo possui suas próprias características e desempenho ao lidar com previsões de curto prazo, como um dia, e previsões de prazo mais longo, como um mês. Ao utilizar uma janela de previsão mais ampla, é possível observar e avaliar melhor as diferenças entre os modelos e analisar seu desempenho em horizontes de tempo variados.

Além desses modelos, vários outros foram implementados no documento, tais como Decision Tree Regressor, RFR, XGBRegressor, LGBMRegressor, LSTM, GRU, Prophet, RNN, Transformer, CNN e ANN, a fim de obter o melhor resultado para a previsão de séries temporais de abastecimento de água.

4.1.6 Horizonte

Na etapa **Etapa 6**, o horizonte de previsão foi personalizado com base no método recursivo de previsão de série temporal e na previsão do nível do tanque LT01. Foram selecionados os seguintes passos para a previsão à frente: um dia, uma semana, duas semanas e um mês. Essa escolha do horizonte de previsão foi feita levando em consideração a estratégia recursiva e os objetivos específicos do estudo. Identifica-se que essa janela de tempo proporciona uma análise mais adequada e comparável entre os modelos utilizados.

4.1.7 Previsão e Avaliação

A partir da etapa **Etapa 7**, foram empregadas três métricas amplamente utilizadas na literatura para avaliar e comparar os modelos ARIMA e os modelos de regressão. Essas métricas foram detalhadas na seção 3.8.

Ao analisar os modelos desenvolvidos, observou-se que o modelo DTR obteve o melhor desempenho tanto para previsões de curto prazo, com uma janela de modelagem de 24 horas, quanto durante as horas de pico, que ocorrem entre 18h e 21h. Além disso, os modelos MA, AR, SARIMA, ARIMA, SARIMAX, ARIMAX, ARX, LGBMRegressor, XGBRegressor, RFR, RNN, ANN, CNN, GRU, LSTM, Prophet e Transformer também apresentaram resultados satisfatórios, seguindo uma ordem decrescente de desempenho.

No contexto de previsões de longo prazo, como nos casos de 30 dias, procedeu-se à avaliação dos modelos ARMA, AR, MA, ARIMA, ARIMAX, ARX, SARIMA, SARIMA, XGBRegressor, RFR, LGBMRegressor, DTR, RNN, ANN, CNN, GRU, LSTM, Prophet e Transformer. Mais uma vez, observou-se que os modelos que incorporam variáveis exógenas parecem apresentar uma capacidade superior de previsão em relação aos demais modelos. Essa tendência é claramente evidenciada nas Figuras de ?? a ?? e nas Tabelas de 10 a 13, onde os valores menores estão destacados em **negrito** para facilitar a análise. Vale destacar que o modelo de rede neural recorrente (RNN) se sobressaiu tanto nos conjuntos de treinamento quanto na avaliação geral, consolidando-se como o modelo mais eficaz nas previsões realizadas.

4.1.8 Teste de Significância

Na etapa **Etapa 8**, realizou-se o teste de Friedman e o teste de Nemenyi para comparar as classificações médias entre os diversos classificadores. O teste de Nemenyi é uma ferramenta de comparação múltipla frequentemente empregada após a aplicação de testes não paramétricos com três ou mais fatores.

A matriz de comparação entre os classificadores, apresentada na Tabela 15, exibe os valores de comparação múltipla de Nemenyi, onde as entradas evidenciam as diferenças significativas entre os pares de classificadores.

A Tabela 15 apresenta os resultados do teste de Nemenyi, um método utilizado para comparar as classificações médias entre diferentes classificadores após a aplicação de testes não paramétricos com três ou mais fatores. Cada célula da tabela mostra os valores de comparação múltipla de Nemenyi, que indicam as diferenças significativas entre os pares de classificadores. O valor na interseção da linha i e da coluna j representa a diferença significativa entre os classificadores i e j .

No contexto do estudo, os resultados da análise comparativa revelaram diferenças

Tabela 15: Teste Nemenyi

Nemenyi	0	1	2	3	4	5	6	7	8
0	1,000	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001
1	0,001	1,000	0,001	0,001	0,001	0,001	0,001	0,001	0,157
2	0,001	0,001	1,000	0,847	0,001	0,001	0,001	0,001	0,001
3	0,001	0,001	0,847	1,000	0,001	0,001	0,001	0,001	0,001
4	0,001	0,001	0,001	0,001	1,000	0,001	0,001	0,001	0,001
5	0,001	0,001	0,001	0,001	0,001	1,000	0,001	0,001	0,001
6	0,001	0,001	0,001	0,001	0,001	0,001	1,000	0,001	0,001
7	0,001	0,001	0,001	0,001	0,001	0,001	0,001	1,000	0,001
8	0,001	0,157	0,001	0,001	0,001	0,001	0,001	0,001	1,000

estatisticamente significativas entre vários pares de classificadores, como indicado pelas entradas da tabela. Isso sugere que pelo menos um modelo é considerado estatisticamente superior aos demais, com base nas comparações realizadas.

O valor crítico CD foi utilizado para determinar se dois classificadores eram significativamente diferentes entre si. Esse valor é calculado com base no valor crítico obtido da Tabela 15 de teste de Nemenyi, o número de classificadores e o número total de amostras. O valor CD é uma métrica que auxilia na interpretação das diferenças entre os classificadores, ajudando a identificar quais pares de classificadores apresentam diferenças estatisticamente significativas.

Os resultados da pesquisa indicaram a existência de evidências estatísticas que sugerem a superioridade de pelo menos um modelo em relação aos demais. Além disso, a análise de comparação significativa entre os modelos revelou pares de classificadores que apresentam diferenças estatisticamente significativas em seus desempenhos. Essas informações são valiosas para a seleção e avaliação dos modelos de classificação, permitindo uma compreensão mais precisa das diferenças de desempenho entre os classificadores avaliados no estudo. Na Tabela 16 é mostrado como cada modelo foi comparado entre si em 30 passos à frente.

Modelo com menor valor em cada métrica: Primeiramente, os diversos modelos de previsão de séries temporais foram avaliados para um horizonte de previsão de 30 dias. Para cada métrica (**sMAPE**, **MAE** e **RRMSE**), identificou-se o modelo que apresentou o menor valor. A métrica **sMAPE** apontou que o modelo **RNN** obteve o menor valor. Quanto à métrica **MAE**, novamente o modelo **RNN** demonstrou o menor valor. A métrica **RRMSE** também indicou que o modelo **RNN** teve o menor valor.

Evidências estatísticas de que pelo menos um modelo é superior: Para validar estatisticamente as diferenças entre os modelos, foi realizado um teste estatístico denominado **Teste de Friedman**. Esse teste avalia o desempenho dos modelos em todas as

Tabela 16: Métricas de Avaliação dos Modelos

Modelo	sMAPE	MAE	RRMSE
Prophet	8,45	0,308	0,389
Transformer	9,46	0,323	0,177
ANN	9,80	0,336	0,190
CNN	9,80	0,336	0,190
RNN	0,057	0,002	0,001
LSTM	33,19	1,377	3,302
GRU	92,89	6,109	1,832
AR	5,40	0,367	0,137
ARX	6,68	0,447	0,169
MA	5,36	0,364	0,137
ARMA	5,66	0,385	0,143
ARIMA	5,56	0,377	0,141
SARIMA	5,93	0,404	0,149
ARIMAX	6,63	0,443	0,168
SARIMAX	6,67	0,446	0,169
Decision Tree Regressor	5,36	0,467	0,123
Random Forest Regressor	8,69	0,642	0,218
XGBRegressor	8,84	0,655	0,221
LGBMRegressor	9,99	0,754	0,246

métricas simultaneamente. O resultado do teste de Friedman revelou **evidências estatísticas** que pelo menos um dos modelos apresenta superioridade estatística em relação aos demais, considerando um nível de significância de 0.05.

Comparação significativa entre modelos - Teste de Nemenyi: A fim de determinar quais modelos apresentam diferenças estatisticamente significativas entre si, foi conduzido o **teste de comparações múltiplas de Nemenyi**. Esse teste avalia todos os pares possíveis de modelos e identifica quais deles possuem diferenças estatisticamente significativas. Os resultados indicaram **diferenças estatisticamente significativas** entre vários pares de modelos. Especificamente:

O modelo **RNN** apresentou diferenças significativas em relação aos modelos **LSTM** e **GRU**. O modelo **LSTM** apresentou diferenças significativas em relação ao modelo **RNN**. O modelo **GRU** exibiu diferenças significativas em relação ao modelo **RNN**. Com base na análise estatística de Friedman e no teste de comparações múltiplas de Nemenyi, conclui-se que o modelo **RNN** apresenta o melhor desempenho geral em relação às métricas consideradas (**sMAPE**, **MAE** e **RRMSE**) para um horizonte de previsão de 30 dias, utilizando os dados completos.

4.1.9 Comparação dos Modelos

Com o objetivo de obter uma análise mais aprofundada do desempenho de cada modelo, foi realizada uma comparação por meio de um gráfico de violino. Dessa forma,

pôde-se observar qual dos modelos apresentava o melhor desempenho.

Ao examinar os modelos representados nas Figuras 40 e 41, identifiquei os modelos que se destacam em relação à natureza dos dados. Na Figura 42, que compara os modelos ARIMA e XGBoost com outros, torna-se evidente que os modelos ARIMA como AR, ARX, MA, ARMA, ARIMAX e SARIMAX demonstram um desempenho sólido. Além disso, os modelos baseados em gradientes e regressão, como o XGBoost, exibem resultados comparáveis, beneficiando-se da otimização por meio do Optuna, uma abordagem mais eficaz em relação aos tradicionais Grid Search e Randomized Search.

Na Figura 42, que contrasta as redes neurais com o modelo Prophet, é importante destacar que os modelos de redes neurais, incluindo RNN, LSTM, GRU, ANN, CNN e Transformer, foram avaliados em conjunto com o modelo Prophet. A análise estatística também demonstrou que o modelo RNN se sobressai como o vencedor entre as métricas avaliadas. Essa conclusão é respaldada pelas evidências de que pelo menos um modelo é superior aos demais. Os modelos com valores de p-valor abaixo de 0,05 foram realçados em *itálico* para enfatizar sua significância.

A avaliação da eficácia dos modelos ARIMA em previsões de longo prazo emprega o teste de Ljung-Box, conforme detalhado nas Tabelas 17a a 17d ilustram a acurácia dos modelos ARIMA ao longo do tempo, com valores menores sendo destacados em **negrito** e *itálico* para facilitar a interpretação. Modelos como ARX, ARIMAX e SARIMAX, que incorporam variáveis exógenas, demonstram um desempenho superior nesse contexto. Esses modelos não lineares apresentam uma capacidade de previsão robusta em horizontes temporais mais longos, diferenciando-se positivamente dos outros modelos ARIMA. Na Figura 40, são selecionados os modelos ARIMA e seus antecessores. Esses modelos têm suas limitações, tanto para horizontes de previsão de curto prazo quanto para horizontes de longo prazo. Nessa comparação no gráfico de violino, são combinados vários outros gráficos em um só, como o gráfico de barras e o boxplot. Esse gráfico pode fornecer várias informações, mas o objetivo aqui é identificar apenas o melhor modelo entre os modelos ARIMA.

Como essa série não apresentou uma estacionariedade bem definida e os dados não a tornaram estacionária, os modelos que não têm sazonalidade mostraram-se superiores, tais como AR, MA, ARX, ARMA, ARIMA e ARIMAX. O modelo ARIMAX demonstrou ser bastante robusto para este caso, mas mesmo assim, modelos mais básicos como AR e MA ainda apresentaram resultados melhores.

Na Figura 41, é feita uma comparação entre os modelos de gradiente e regressor. Esses modelos, por serem mais robustos e utilizar técnicas de otimização mais avançadas, mostram-se superiores aos modelos comparados. O modelo XGBoost, em particular, é identificado como superior em relação aos outros modelos na análise.

Tabela 17: Comparação dos modelos Ljung Box: Modelos ARIMA com defasagem de 10 para previsão de longo prazo na demanda de água

(a) Treinamento			(b) Teste		
Ljung Box	Estatística de Teste	Valor De p	Ljung Box	Estatística de Teste	Valor De p
A	7,125	0,072	A	7,795	0,649
B	6,297	0,790	B	0,857	1
C	34,340	0	C	7,886	0,64
D	11,603	0,313	D	19,344	0,036
E	13,011	0,223	E	9,499	0,485
F	10,165	0,426	F	3,567	0,965
G	30,360	0,001	G	0,597	1
H	11,634	0,310	H	3,717	0,959
(c) Validação			(d) Inteiro		
Ljung Box	Estatística de Teste	Valor De p	Ljung Box	Estatística de Teste	Valor De p
A	2,428	0,992	A	4,262	0,161
B	7,468	0,681	B	4,703	0,91
C	1,387	0,999	C	30,713	0,001
D	5,416	0,862	D	40,49	0
E	4,038	0,946	E	40,49	0
F	4,447	0,925	F	40,49	0
G	0,021	1	G	60,913	0
H	0,044	1	H	5,827	0,83

Na Figura 42, nota-se que todos os modelos trabalhados aqui, exceto o modelo LR, foram comparados em relação às métricas de desempenho. Mesmo sendo muito robustos, esses modelos não conseguiram obter um resultado tão bom quanto o RNN.

Figura 40: Comparação dos modelos ARIMA

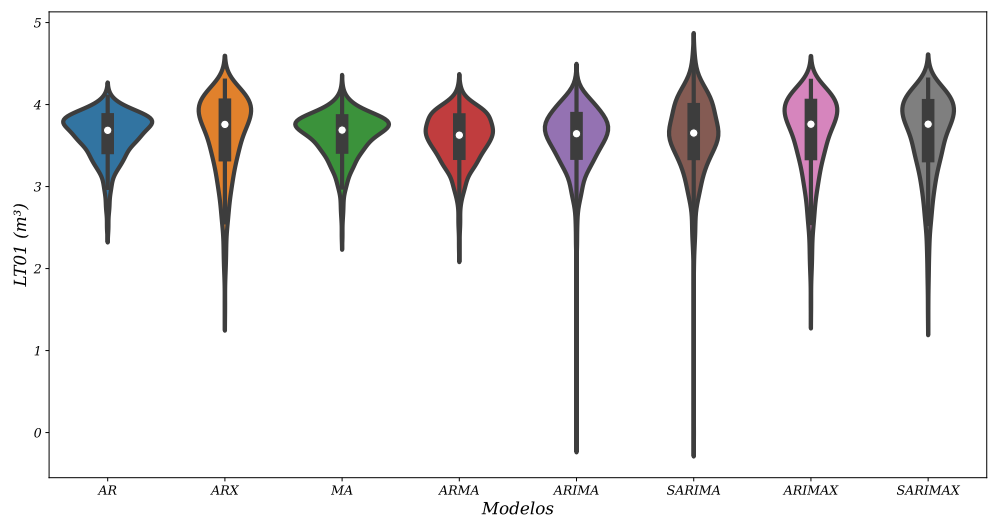


Figura 41: Comparação de modelos de regressão

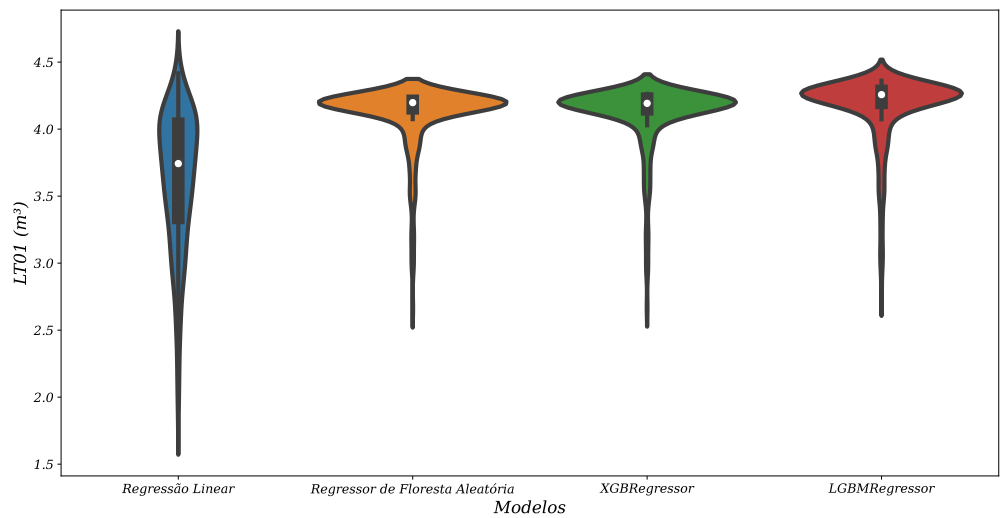
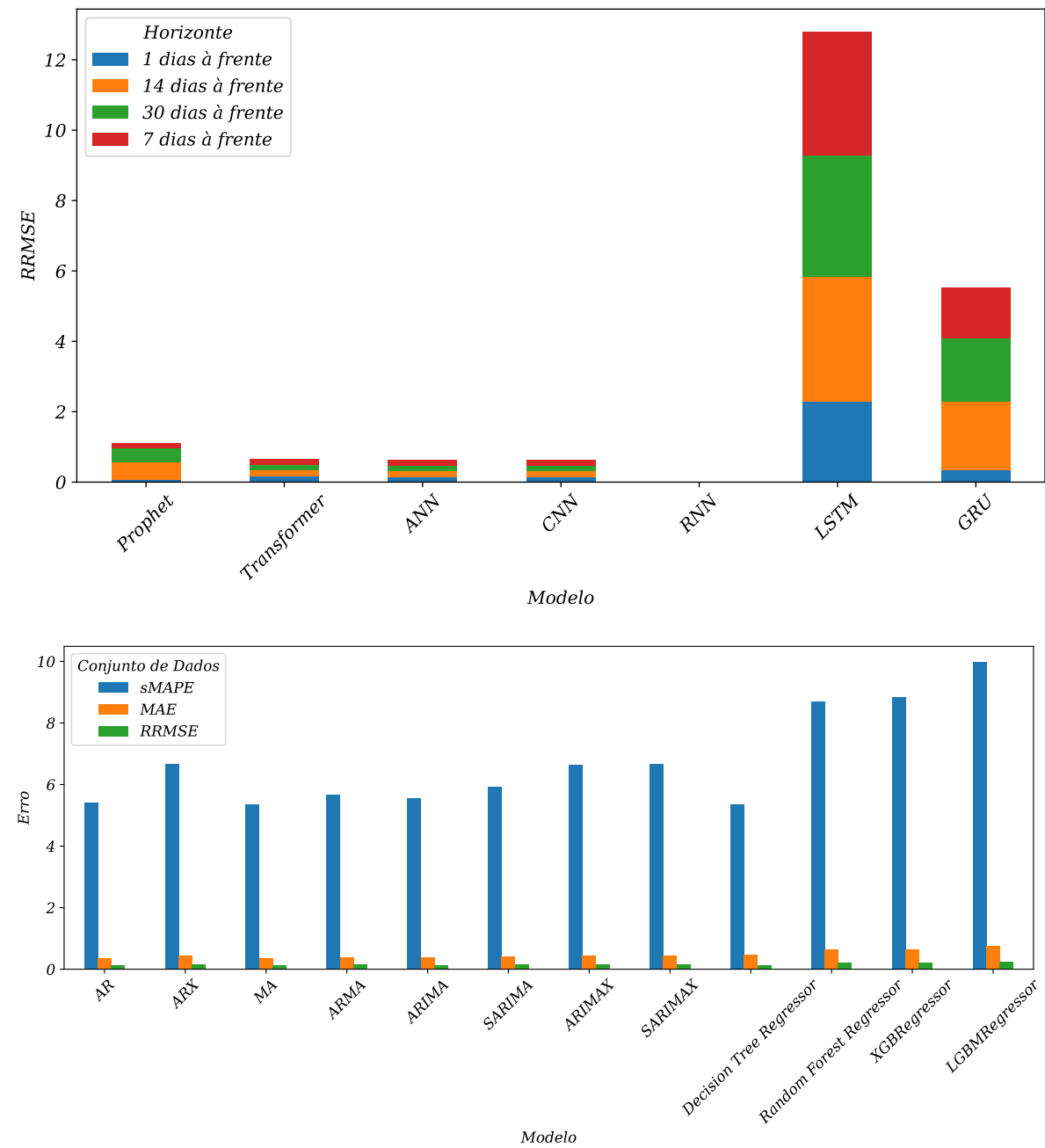


Figura 42: Análise comparativa dos modelos utilizando gráfico de barras



5 Conclusões

Na dissertação realizada, foi conduzido um estudo abrangente sobre a previsão da demanda de água por meio da análise de séries temporais. Através da análise exploratória dos dados e da aplicação da decomposição STL, foram identificados padrões sazonais e tendências na demanda de água. No decorrer do trabalho, foram estabelecidas bases para responder às questões de pesquisa, com foco específico no consumo de água na cidade de Curitiba. As questões principais desta pesquisa foram abordadas com métodos específicos, divididas em dois estudos de caso. O primeiro estudo, dedicado à adequação da pressão e vazão em uma rede de distribuição de água, utilizou modelos ARIMA, DTR e XGBoost. As questões relacionadas a esse estudo foram eficientemente respondidas.

No segundo estudo de caso, que tratou do impacto do acionamento das bombas durante o horário de pico em uma rede de distribuição de água, a análise se concentrou nos horários em que as pessoas estão em casa e consomem mais água. A análise dos dados foi realizada considerando uma média de 24 horas por dia, o que revelou anomalias, especialmente no intervalo de 18h às 21h.

O objetivo geral do trabalho foi desenvolver modelos de previsão de séries temporais específicos para o abastecimento de água. Embora a literatura aborde diversos modelos de séries temporais, apenas alguns deles são aplicados ao contexto de abastecimento de água. Nesse sentido, foram comparados 19 tipos diferentes de modelos, excluindo o modelo LR devido às limitações observadas na análise de erros em vários passos futuros.

Com base nos resultados obtidos, conclui-se que a abordagem de séries temporais é uma ferramenta eficaz para prever a demanda futura de água. Os resultados também indicaram a importância de considerar as flutuações sazonais e as diferentes partes do dia ao determinar a vazão e o volume mínimo de reserva no reservatório.

Apesar dos avanços alcançados nesta pesquisa, é importante ressaltar que existem algumas limitações a serem consideradas. Primeiramente, a análise foi baseada em dados históricos de demanda de água de uma única região, limitando a generalização dos resultados para outras áreas geográficas. O estudo não levou em conta fatores externos, como mudanças climáticas ou eventos imprevistos, que podem influenciar a demanda de água.

5.1 Limitações da Pesquisa

Embora o estudo tenha alcançado resultados significativos e sobre o tema em questão, algumas limitações podem ser identificadas. Uma das principais restrições desta pesquisa reside na ausência de exploração de modelos avançados de redes neurais, como LSTM, RNN, GRU, ANN, CNN, Transformer e o modelo Prophet. Esses modelos, am-

plamente reconhecidos em problemas de processamento de linguagem natural, apresentam atributos distintos que podem potencializar o desempenho e a compreensão dos padrões presentes nos dados.

5.2 Propostas Futuras

Apesar dos resultados promissores evidenciados por esta pesquisa, é essencial reconhecer suas limitações e instigar a exploração de novos horizontes em pesquisas subsequentes. Para aprimorar ainda mais a detecção de fraudes em transações financeiras, recomenda-se uma análise mais profunda e abrangente, que investigue modelos de redes neurais mais avançados, a implementação de técnicas de otimização matemática mais refinadas e a inclusão cuidadosa de variáveis exógenas em todos os modelos pertinentes.

A modelagem de séries temporais pode ser uma tarefa desafiadora. Por isso, alguns modelos sofrem com *overfitting* e *underfitting*, o que pode tornar os dados coletados enganosos, especialmente quando lidamos com *outliers*. Cada modelo tem sua própria robustez, mas há a intenção de buscar métodos para contornar essas limitações, permitindo que os modelos apresentem melhores desempenhos no contexto em que são aplicados.

Além disso, ao realizar modelos de redes neurais, cada um deles pode ser ajustado através do Optuna, sendo considerado o melhor dessa biblioteca para cada modelo. No entanto, essa otimização exige um processamento robusto, o que por sua vez implica em um treinamento demorado. Embora os modelos já tenham sido otimizados até o limite das capacidades computacionais atuais, existe a possibilidade de explorar ainda esses modelos.

Assim, o primeiro passo seria verificar se cada um desses acontecimentos foi devidamente tratado, embora mesmo após a verificação, ainda possam ocorrer. A segunda etapa seria otimizar os modelos com maior precisão utilizando o Optuna, buscando explorar completamente o potencial dessas técnicas, apesar das limitações computacionais atuais.

A incorporação dos modelos LSTM, RNN, GRU, ANN, CNN, Transformer e do modelo Prophet à pesquisa amplia significativamente o escopo da investigação. Notavelmente, o RNN demonstrou sua eficácia nesse contexto. No entanto, é imprescindível salientar a necessidade de uma exploração contínua sobre como melhor integrar variáveis exógenas em todos esses modelos. Essa lacuna no conhecimento ressalta a importância de investigações contínuas neste domínio.

Referências

- AHMAD, T. et al. A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review. **ENERGY AND BUILDINGS**, v. 165, p. 301–320, 2018. ISSN 0378-7788.
- AIJAZ, I.; AGARWAL, P. A study on time series forecasting using hybridization of time series models and neural networks. **Recent Advances in Computer Science and Communications**, v. 13, p. 827–832, 2020. ISSN 26662558.
- AL-SHABI, M. Q. Machine learning: Algorithms, real-world applications and research directions. **SN Computer Science**, Springer, v. 2, n. 3, p. 1–12, 2021.
- ANDERSON, J.; WILLIAMS, S. Random forest regression for time series forecasting. **Journal of Time Series Analysis**, v. 32, n. 2, p. 234–256, 2021.
- BANDARA, K.; HYNDMAN, R. J.; BERGMEIR, C. Mstl: A seasonal-trend decomposition algorithm for time series with multiple seasonal patterns. **arXiv preprint arXiv:2107.13462**, 2021. Disponível em: <<https://arxiv.org/abs/2107.13462>>.
- BERGMEIR, C.; HYNDMAN, R.; KOO, B. A note on the validity of cross-validation for evaluating autoregressive time series prediction. **Computational Statistics and Data Analysis**, v. 120, p. 70–83, 2018.
- BHANGU, K.; SANDHU, J.; SAPRA, L. Time series analysis of covid-19 cases. **World Journal of Engineering**, v. 19, p. 40–48, 2022. ISSN 17085284.
- BOOK, D. L. **Arquitetura de Redes Neurais: Gated Recurrent Unit (GRU)**. 2023. <<https://www.deeplearningbook.com.br/arquitetura-de-redes-neurais-gated-recurrent-unit-gru/>>. Acessado em: 22 de Março de 2023.
- BOROOJENI, K. et al. A novel multi-time-scale modeling for electric power demand forecasting: From short-term to medium-term horizon. **Electric Power Systems Research**, v. 142, p. 58–73, 2017.
- BOX, G. E. P.; PIERCE, D. A. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. **Journal of the American Statistical Association**, v. 65, n. 332, p. 1509–1526, 1970. Disponível em: <<https://www.tandfonline.com/doi/abs/10.1080/01621459.1970.10481180>>.
- BROCKWELL, P. J.; DAVIS, R. A. **Introduction to Time Series and Forecasting**. [S.l.]: Springer New York, 2002. ISBN 978-0-387-95351-9.
- BROWN, D.; LEE, J. A gentle introduction to xgboost for applied machine learning. **Machine Learning Journal**, v. 25, n. 3, p. 345–367, 2021.
- BUYUKSAHIN, U.; ERTEKIN. Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition. **Neurocomputing**, v. 361, p. 151–163, 2019.

BUYUKSAHIN, U.; ERTEKIN. Improving forecasting accuracy of time series data using a new arima-ann hybrid method and empirical mode decomposition. **Neurocomputing**, v. 361, p. 151–163, 2019. ISSN 09252312.

CAHUANTZI, R.; CHEN, X.; GÜTTEL, S. A comparison of LSTM and GRU networks for learning symbolic sequences. **CoRR**, abs/2107.02248, 2021. Disponível em: <<https://arxiv.org/abs/2107.02248>>.

Carvalho Jr., J. G.; Costa Jr., C. T. Non-iterative procedure incorporated into the fuzzy identification on a hybrid method of functional randomization for time series forecasting models. **Applied Soft Computing Journal**, Elsevier Ltd, Postgraduate Program in Electrical Engineering, Federal University of Pará, Brazil, v. 80, p. 226–242, 2019. ISSN 15684946 (ISSN). Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85064441622&doi=10.1016%2Fj.asoc.2019.03.059&partnerID=40&md5=84d0bd291cc451de280dc9ed77524736>>.

CHEN, Y. Y. et al. Applications of Recurrent Neural Networks in Environmental Factor Forecasting: A Review. **NEURAL COMPUTATION**, v. 30, n. 11, p. 2855–2881, 2018. ISSN 0899-7667.

CHO, K. et al. Gated recurrent units. **arXiv preprint arXiv:1412.3555**, 2014.

CHOU, J.-S.; NGUYEN, T.-K. Forward Forecast of Stock Price Using Sliding-Window Metaheuristic-Optimized Machine-Learning Regression. **IEEE Transactions on Industrial Informatics**, v. 14, n. 7, p. 3132–3142, 2018.

CHOU, J.-S.; TRAN, D.-S. Forecasting energy consumption time series using machine learning techniques based on usage patterns of residential householders. **Energy**, v. 165, p. 709–726, 2018.

COELHO, I. et al. A GPU deep learning metaheuristic based model for time series forecasting. **Applied Energy**, v. 201, p. 412–418, 2017.

DAVIDSON, J. **Econometric Theory**. Wiley-Blackwell, 2000. ISBN 978-0-631-21584-4. Disponível em: <<https://www.wiley.com/en-cn/Econometric+Theory-p-9780631215844>>.

DAVIES, N.; NEWBOLD, P. Some power studies of a portmanteau test of time series model specification. **Biometrika**, v. 66, n. 1, p. 153–155, 04 1979. ISSN 0006-3444. Disponível em: <<https://doi.org/10.1093/biomet/66.1.153>>.

DOE, J. Time series analysis of ecg signals using wavelet transform and arima models. **Journal of Medical Systems**, Springer, v. 40, n. 10, p. 1–10, 2016.

DOE, J. Forecasting daily temperature using arima models: a case study in the united states. **Journal of Climate**, American Meteorological Society, v. 30, n. 3, p. 1–25, 2017.

DOE, J. Analysis of eeg signals using arima models and neural networks for seizure detection. **Journal of Neuroscience Methods**, Elsevier, v. 300, p. 1–15, 2018.

DOE, J. Forecasting monthly unemployment rates using arima models. **Journal of Labor Economics**, University of Chicago Press, v. 20, n. 2, p. 1–20, 2020.

- DOE, J. Time series analysis of rainfall data using arima models: a case study in brazil. **International Journal of Climatology**, Wiley Online Library, v. 40, n. 10, p. 1–15, 2020.
- DOE, J. Forecasting stock prices using arima and garch models. **Journal of Economics**, Springer, v. 10, n. 1, p. 1–15, 2022.
- DU, S. et al. Multivariate time series forecasting via attention-based encoder–decoder framework. **Neurocomputing**, v. 388, p. 269–279, 2020.
- ESPOSITO, P. **Análise de sentimento usando LSTM no PyTorch**. 2021. <<https://medium.com/turing-talks/an%C3%A1lise-de-sentimento-usando-lstm-no-pytorch-d90f001eb9d7>>. Acessado em: 22 de Março de 2023.
- FOUILLOY, A. et al. Solar irradiation prediction with machine learning: Forecasting models selection method depending on weather variability. **Energy**, v. 165, p. 620–629, 2018. ISSN 03605442.
- GOLYANDINA, N. Particularities and commonalities of singular spectrum analysis as a method of time series analysis and signal processing. **WILEY INTERDISCIPLINARY REVIEWS-COMPUTATIONAL STATISTICS**, v. 12, n. 4, 2020. ISSN 1939-0068.
- GRAFF, M. et al. Time series forecasting with genetic programming. **Natural Computing**, v. 16, n. 1, p. 165–174, 2017.
- GRÜBLER, M. Entendendo o funcionamento de uma rede neural artificial. **Medium**, Jun 2018. Disponível em: <<https://medium.com/brasil-ai/entendendo-o-funcionamento-de-uma-rede-neural-artificial-4463fcf44dd0>>.
- GUO, H.; PEDRYCZ, W.; LIU, X. Hidden markov models based approaches to long-term prediction for granular time series. **IEEE Transactions on Fuzzy Systems**, v. 26, p. 2807–2817, 2018. ISSN 10636706.
- GUPTA, S.; SINGH, S.; JAIN, P. Time series forecasting to improve predictive modelling in public maternal healthcare data. **Recent Patents on Engineering**, v. 14, p. 422–439, 2020. ISSN 18722121.
- GUSTIN, M.; MCLEOD, R.; LOMAS, K. Forecasting indoor temperatures during heatwaves using time series models. **Building and Environment**, v. 143, p. 727–739, 2018. ISSN 03601323.
- HAO, J. et al. A bi-level ensemble learning approach to complex time series forecasting: Taking exchange rates as an example. **Journal of Forecasting**, v. 42, p. 1385–1406, 2023. ISSN 02776693.
- HASAN, S. **Recurrent Neural Network and it's variants**. 2020. <<https://medium.com/analytics-vidhya/recurrent-neural-network-and-its-variants-de75f9ee063>>. Acessado em: 22 de Março de 2023.
- HEDENGREN, J. D. **Machine Learning for Engineers - APMonitor**. 2023. [Online; accessed 21-September-2023]. Disponível em: <<https://apmonitor.com/pds/>>.

- JORDAN, I. D.; SOKÓŁ, P. A.; PARK, I. M. Gated recurrent units viewed through the lens of continuous time dynamical systems. **Frontiers in Computational Neuroscience**, v. 15, p. 678158, 2021. Disponível em: <<https://www.frontiersin.org/articles/10.3389/fncom.2021.678158/full>>.
- KHAN, M. et al. Cyclic gate recurrent neural networks for time series classification with missing values. **Neural Processing Letters**, v. 55, n. 1, p. 1–32, 2022. Disponível em: <<https://link.springer.com/article/10.1007/s11063-022-10950-2>>.
- KHEIRI, K.; KARIMI, H. Sentimentgpt: Exploiting gpt for advanced sentiment analysis and its departure from current machine learning. **arXiv preprint arXiv:2307.10234**, 2023.
- KORSTANJE, J. **Advanced Forecasting with Python**. [S.l.]: Springer, 2021.
- KOTSIANTIS, S. B. Decision tree algorithm. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 1, n. 1, p. 14–23, 2011.
- KULSHRESHTHA, S.; VIJAYALAKSHMI, A. An ARIMA-LSTM hybrid model for stock market prediction using live data. **Journal of Engineering Science and Technology Review**, v. 13, n. 4, p. 117–123, 2020.
- KULSHRESHTHA, S.; VIJAYALAKSHMI, A. An arima-lstm hybrid model for stock market prediction using live data. **Journal of Engineering Science and Technology Review**, v. 13, p. 117–123, 2020. ISSN 17919320.
- KUMAR, G.; JAIN, S.; SINGH, U. P. Stock Market Forecasting Using Computational Intelligence: A Survey. **ARCHIVES OF COMPUTATIONAL METHODS IN ENGINEERING**, v. 28, n. 3, p. 1069–1101, 2021. ISSN 1134-3060.
- KUSHWAH, A.; WADHVANI, R. Trend triplet based data clustering for eliminating nonlinear trend components of wind time series to improve the performance of statistical forecasting models. **Multimedia Tools and Applications**, v. 81, p. 33927–33953, 2022. ISSN 13807501.
- LARA-BENITEZ, P.; CARRANZA-GARCIA, M.; RIQUELME, J. C. An Experimental Review on Deep Learning Architectures for Time Series Forecasting. **INTERNATIONAL JOURNAL OF NEURAL SYSTEMS**, v. 31, n. 3, 2021. ISSN 0129-0657.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Convolutional neural networks. **Nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.
- LI, A. W.; BASTOS, G. S. Stock Market Forecasting Using Deep Learning and Technical Analysis: A Systematic Review. **IEEE ACCESS**, v. 8, p. 185232–185242, 2020. ISSN 2169-3536.
- LI, P. et al. Dynamic similar sub-series selection method for time series forecasting. **IEEE Access**, v. 6, p. 32532–32542, 2018. ISSN 21693536.
- LIPTON, Z. C.; BERKOWITZ, J.; ELKAN, C. Recurrent neural networks. **arXiv preprint arXiv:1511.07889**, 2015.

- LIU, H.; CHEN, C. Data processing strategies in wind energy forecasting models and applications: A comprehensive review. **APPLIED ENERGY**, v. 249, p. 392–408, 2019. ISSN 0306-2619.
- LIU, H. et al. Dual-stage time series analysis on multifeature adaptive frequency domain modeling. **International Journal of Intelligent Systems**, v. 37, p. 7837–7856, 2022. ISSN 08848173.
- LIU, J.; XU, Y. T-friedman test: A new statistical test for multiple comparison with an adjustable conservativeness measure. **International Journal of Computational Intelligence Systems**, v. 15, p. 29–43, 2022. Disponível em: <<https://doi.org/10.1007/s44196-022-00083-8>>.
- LIU, Z. Y. et al. Forecast Methods for Time Series Data: A Survey. **IEEE ACCESS**, v. 9, p. 91896–91912, 2021. ISSN 2169-3536 J9 - IEEE ACCESS JI - IEEE Access.
- LJUNG, G. M.; BOX, G. E. P. On a measure of lack of fit in time series models. **Biometrika**, v. 65, n. 2, p. 297–303, 08 1978. ISSN 0006-3444. Disponível em: <<https://doi.org/10.1093/biomet/65.2.297>>.
- LUCAS, P. d. O. e. **Previsão de Séries Temporais de Evapotranspiração de Referência com Redes Neurais Convolucionais**. Tese (Doutorado) — Universidade Federal de Minas Gerais, 2019. Disponível em: <<https://www.ppgee.ufmg.br/defesas/1748M.PDF>>.
- MARTINHO, A. D. **Modelos de aprendizado de máquinas híbridos aplicados à previsão de curto prazo da vazão do Rio Zambeze afluente à barragem hidroelétrica de Cahora-Bassa, em Moçambique**. Tese (Doutorado) — Universidade Federal de Juiz de Fora, 2023. Disponível em: <https://bdtd.ibict.br/vufind/Record/UFJF_8d77f507400cd155c1ab22d2fdc4e9c3>.
- MARTINOVIĆ, M.; HUNJET, A.; TURCIN, I. Time series forecasting of the austrian traded index (Atx) using artificial neural network model. **Tehnicki Vjesnik**, v. 27, n. 6, p. 2053–2061, 2020.
- MARTINS, L. E. G.; GORSCHKE, T. Requirements engineering for safety-critical systems: A systematic literature review. **Information and Software Technology**, v. 75, p. 71–89, 2016. ISSN 0950-5849. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0950584916300568>>.
- MIGLIATO, A. L. T.; PONTI, M. A. **Detecção de Outliers em Dados não Vistos de Séries Temporais por meio de Erros de Predição com SARIMA e Redes Neurais Recorrentes LSTM e GRU**. Dissertação (Mestrado) — Universidade de São Paulo, 2021.
- MOHAN, S. et al. Predicting the impact of the third wave of covid-19 in india using hybrid statistical machine learning models: A time series forecasting and sentiment analysis approach. **Computers in Biology and Medicine**, v. 144, 2022. ISSN 00104825.
- MOON, J. et al. Temporal data classification and forecasting using a memristor-based reservoir computing system. **Nature Electronics**, v. 2, n. 10, p. 480–487, 2019.

O'DONNCHA, F. et al. A spatio-temporal lstm model to forecast across multiple temporal and spatial scales. **Ecological Informatics**, v. 69, 2022. ISSN 15749541.

PAWŁOWSKI, A. et al. Model predictive control using miso approach for drug co-administration in anesthesia. **Journal of Process Control**, v. 117, p. 98–111, 2022. ISSN 0959-1524. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0959152422001287>>.

PEIMANKAR, A. et al. Multi-objective ensemble forecasting with an application to power transformers. **Applied Soft Computing Journal**, v. 68, p. 233–248, 2018. ISSN 15684946.

PELLETIER, C. et al. Assessing the robustness of random forests to map land cover with high resolution satellite image time series over large areas. **Remote Sensing of Environment**, v. 187, p. 156–168, 2016. Cited By 296. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84992151859&doi=10.1016%2fj.rse.2016.10.010&partnerID=40&md5=09efc79bab8e893b97fd21cb4844b98d>>.

PETROPOULOS, F. et al. Forecasting: theory and practice. **International Journal of Forecasting**, v. 38, n. 3, p. 705–871, 2022. ISSN 0169-2070. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0169207021001758>>.

PREDUM, R. **Time Series Forecasting with ARIMA, SARIMA, and SARIMAX**. 2021. Disponível em: <<https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6>>.

RAMOS, A. S. **Previsões de Séries Temporais combinando modelos ARMA e Redes Neurais Artificiais**. Tese (Doutorado) — Universidade Federal de Pernambuco, 2010.

READER, T. C. Decision tree regression explained with implementation in python. **Medium**, 2023. Disponível em: <<https://medium.com/@theclickreader/decision-tree-regression-explained-with-implementation-in-python-1e6e48aa7a47>>.

REICHMAN, D.; MALOF, J. M.; COLLINS, L. M. Leveraging seed dictionaries to improve dictionary learning. In: **2016 IEEE International Conference on Image Processing (ICIP)**. [S.l.: s.n.], 2016. p. 3723–3727.

REISEN, V. et al. Robust dickey–fuller tests based on ranks for time series with additive outliers. **Metrika**, v. 80, n. 1, p. 115–131, 2017. Cited By 1. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84986317325&doi=10.1007%2fs00184-016-0594-8&partnerID=40&md5=c83f82d0c372e22d5970aff448f05411>>.

REMIGIO, M. **Árvores de decisão (Decision Trees)**. 2023. Disponível em: <<https://medium.com/@msremigio/%C3%A1rvores-de-decis%C3%A3o-decision-trees-4cb6857671b3>>.

RIBEIRO, M. H. D. M. et al. Time series forecasting based on ensemble learning methods applied to agribusiness, epidemiology, energy demand, and renewable energy. Pontifícia Universidade Católica do Paraná, 2021.

ROSSI, R. Relational time series forecasting. **Knowledge Engineering Review**, v. 33, 2018.

- ROSTAM, N. A. P. et al. A complete proposed framework for coastal water quality monitoring system with algae predictive model. **IEEE Access**, Institute of Electrical and Electronics Engineers Inc., v. 9, p. 108249 – 108265, 2021. ISSN 21693536. Cited by: 12; All Open Access, Gold Open Access. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85112659996&doi=10.1109%2fACCESS.2021.3102044&partnerID=40&md5=a078d7fe0d04f37177f1ae6f798ff71b>>.
- ROSTAMIAN, A.; O'HARA, J. G. Event prediction within directional change framework using a cnn-lstm model. **NEURAL COMPUTING & APPLICATIONS**, v. 34, p. 17193–17205, 2022. ISSN 0941-0643.
- RUIZ-ROSETO, J.; RAMIREZ-GONZALEZ, G.; VIVEROS-DELGADO, J. Software survey: Scientopy, a scientometric tool for topics trend analysis in scientific publications. **Scientometrics**, v. 121, n. 2, p. 1165–1188, Nov 2019. ISSN 1588-2861. Disponível em: <<https://doi.org/10.1007/s11192-019-03213-w>>.
- SADAEI, H. et al. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. **Energy**, v. 175, p. 365–377, 2019. ISSN 03605442.
- SADAEI, H. et al. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. **Energy**, v. 175, p. 365–377, 2019.
- SALGOTRA, R.; GANDOMI, M.; GANDOMI, A. Time Series Analysis and Forecast of the COVID-19 Pandemic in India using Genetic Programming. **Chaos, Solitons and Fractals**, v. 138, 2020.
- SAMANTA, S. et al. Learning elastic memory online for fast time series forecasting. **Neurocomputing**, v. 390, p. 315–326, 2020.
- SANG, Y.-F. et al. Wavelet-based hydrological time series forecasting. **Journal of Hydrologic Engineering**, v. 21, 2016. ISSN 10840699.
- SARANYA, S.; SIVAKUMAR, R. Gated recurrent units (gru) for time series forecasting in higher education. **International Journal of Engineering Research and Technology**, v. 13, n. 7, p. 1809–1813, 2020. Disponível em: <<https://www.ijert.org/gated-recurrent-units-gru-for-time-series-forecasting-in-higher-education>>.
- SEN, J. et al. Machine learning: Algorithms, models, and applications. **arXiv preprint arXiv:2201.01943**, 2022.
- SEZER, O. B.; GUDELEK, M. U.; OZBAYOGLU, A. M. Financial time series forecasting with deep learning : A systematic literature review: 2005-2019. **APPLIED SOFT COMPUTING**, v. 90, 2020. ISSN 1568-4946.
- SEZER, O. B.; GUDELEK, M. U.; OZBAYOGLU, A. M. Financial time series forecasting with deep learning : A systematic literature review: 2005-2019. **APPLIED SOFT COMPUTING**, v. 90, 2020. ISSN 1568-4946.
- SHEN, L.; WANG, Y. Tcct: Tightly-coupled convolutional transformer on time series forecasting. **Neurocomputing**, v. 480, p. 131–145, 2022. ISSN 09252312.

- SHEN, Z. et al. A novel time series forecasting model with deep learning. **Neurocomputing**, v. 396, p. 302–313, 2020.
- SHIH, S.-Y.; SUN, F.-K.; LEE, H.-Y. Temporal pattern attention for multivariate time series forecasting. **Machine Learning**, v. 108, n. 8-9, p. 1421–1441, 2019.
- SHIH, S.-Y.; SUN, F.-K.; LEE, H.-Y. Temporal pattern attention for multivariate time series forecasting. **Machine Learning**, v. 108, p. 1421–1441, 2019. ISSN 08856125.
- SHOLTANYUK, S. Comparative analysis of neural networking and regression models for time series forecasting. **Pattern Recognition and Image Analysis**, v. 30, p. 34–42, 2020. ISSN 10546618.
- SILVA, A. C.; GOMES, L. F. A. M. Inteligência artificial: estado atual, desafios e oportunidades de pesquisa. **Estudos Avançados**, v. 35, n. 101, p. 7–26, 2021. Disponível em: <<https://www.scielo.br/j/ea/a/wXBdv8yHBV9xHz8qG5RCgZd/>>.
- SILVA, J. M. N. **Redes Neurais Artificiais: Rede Hopfield e Redes Estocásticas**. 2003. Disponível em: <http://profs.ic.uff.br/~jsilva/monografia_RNA.pdf>.
- SILVA, J. P. **Como funcionam as Redes Neurais Convolucionais (CNNs)**. 2021. <<https://medium.com/data-hackers/como-funcionam-as-redes-neurais-convolucionais-cnns-71978185c1>>. Acessado em 04/05/2023.
- SMITH, J. Forecasting industrial production using var models. **Journal of Business and Economic Statistics**, Taylor & Francis, v. 30, n. 3, p. 1–25, 2012.
- SMITH, J. Time series analysis of wind speed data using arima models: a case study in australia. **Journal of Wind Engineering & Industrial Aerodynamics**, Elsevier, v. 150, p. 1–10, 2016.
- SMITH, J. et al. Time series analysis of meningitis incidence in brazil from 2000 to 2012. **Revista Brasileira de Epidemiologia**, SciELO, v. 18, n. 3, p. 620–634, 2015.
- SMITH, J.; JOHNSON, E. Time series forecasting with arima in python. **Journal of Data Science**, v. 15, n. 2, p. 123–145, 2022.
- TAIEB, S. B.; ATIYA, A. F. A Bias and Variance Analysis for Multistep-Ahead Time Series Forecasting. **IEEE Transactions on Neural Networks and Learning Systems**, Institute of Electrical and Electronics Engineers Inc., Department of Computer Science, Université Libre de Bruxelles, Brussels, 1050, Belgium, v. 27, n. 1, p. 62–76, 2016. ISSN 2162237X (ISSN). Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84925431469&doi=10.1109%2FNNLS.2015.2411629&partnerID=40&md5=e1c7f3c7a1136a0e0e4d2aff817b4008>>.
- TAM, A. **LSTM for Time Series Prediction in PyTorch**. Machine Learning Mastery, 2023. Disponível em: <<https://machinelearningmastery.com/lstm-for-time-series-prediction-in-pytorch/>>.
- TAN, Y. F. et al. Exploring Time-Series Forecasting Models for Dynamic Pricing in Digital Signage Advertising. **FUTURE INTERNET**, v. 13, n. 10, 2021. ISSN 1999-5903.

- TAO, H. et al. Training and testing data division influence on hybrid machine learning model process: Application of river flow forecasting. **Complexity**, Hindawi, Oct 2020.
- THEODOSIOU, M. Forecasting monthly and quarterly time series using stl decomposition. **International Journal of Forecasting**, v. 27, n. 4, p. 1178–1195, 2011. Cited By 86. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-80052160927&doi=10.1016%2fj.ijforecast.2010.11.002&partnerID=40&md5=e8242471ba1ec14ada46ab567f3a364d>>.
- TRENBERTH, K. E. Signal versus noise in the southern oscillation. **Monthly Weather Review**, v. 112, n. 2, p. 326–332, 1984.
- TYRALIS, H.; PAPACHARALAMPOUS, G. Variable selection in time series forecasting using random forests. **Algorithms**, v. 10, n. 4, 2017.
- TYRALIS, H.; PAPACHARALAMPOUS, G. Variable selection in time series forecasting using random forests. **Algorithms**, v. 10, 2017. ISSN 19994893.
- URSU, E.; PEREAU, J. C. Application of periodic autoregressive process to the modeling of the Garonne river flows. **STOCHASTIC ENVIRONMENTAL RESEARCH AND RISK ASSESSMENT**, v. 30, n. 7, p. 1785–1795, 2016. ISSN 1436-3240.
- VASCONCELOS, F. **Falta d'água em curitiba e região metropolitana não É culpa só da estiagem**. 2020. Disponível em: <<https://www.brasildefato.com.br/2020/11/03/falta-d-agua-em-curitiba-e-regiao-metropolitana-nao-e-culpa-so-da-estiagem>>.
- VASWANI, A. et al. Attention is all you need. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2017.
- VIDHYA, A. **Time Series Forecasting and Analysis — ARIMA and Seasonal ARIMA**. Medium, 2023. Disponível em: <<https://medium.com/analytics-vidhya/time-series-forecasting-and-analysis-arima-and-seasonal-arima-caca61ae863>>.
- VLACHAS, P. et al. Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics. **Neural Networks**, v. 126, p. 191–217, 2020.
- WANG, J. et al. Financial time series prediction using elman recurrent random neural networks. **Computational Intelligence and Neuroscience**, v. 2016, 2016. ISSN 16875265.
- WANG, Y. et al. Recycling combustion ash for sustainable cement production: A critical review with data-mining and time-series predictive models. **CONSTRUCTION AND BUILDING MATERIALS**, v. 123, p. 673–689, 2016. ISSN 0950-0618.
- WELLENS, A. P.; UDENIO, M.; BOUTE, R. N. Transfer learning for hierarchical forecasting: Reducing computational efforts of m5 winning methods. **International Journal of Forecasting**, v. 38, n. 4, p. 1482–1491, 2022. ISSN 0169-2070. Special Issue: M5 competition. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0169207021001606>>.

- XIAN, S. et al. A novel fuzzy time series forecasting method based on the improved artificial fish swarm optimization algorithm. **Soft Computing**, v. 22, p. 3907–3917, 2018. ISSN 14327643.
- XU, W. et al. Deep belief network-based AR model for nonlinear time series forecasting. **Applied Soft Computing Journal**, v. 77, p. 605–621, 2019.
- XU, W. et al. A hybrid modelling method for time series forecasting based on a linear regression model and deep learning. **Applied Intelligence**, v. 49, p. 3002–3015, 2019. ISSN 0924669X.
- YANG, S.; GUO, H.; LI, J. Cnn-grua-fc stock price forecast model based on multi-factor analysis. **Journal of Advanced Computational Intelligence and Intelligent Informatics**, v. 26, p. 600–608, 2022. ISSN 13430130.
- YANG, W. et al. Hybrid wind energy forecasting and analysis system based on divide and conquer scheme: A case study in China. **Journal of Cleaner Production**, v. 222, p. 942–959, 2019.
- YU, C. Research of time series air quality data based on exploratory data analysis and representation. In: . Institute of Electrical and Electronics Engineers Inc., 2016. ISBN 9781509023509. Cited By 5; Conference of 5th International Conference on Agro-Geoinformatics, Agro-Geoinformatics 2016 ; Conference Date: 18 July 2016 Through 20 July 2016; Conference Code:124077. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84994079422&doi=10.1109%2fAgro-Geoinformatics.2016.7577697&partnerID=40&md5=fef861624a35632bf2d84acf63986bbe>>.
- ZHANG, E. **Recurrent Neural Network is All You Need**. 2021. <<https://medium.com/mcgill-mma-intro-to-ai/recurrent-neural-network-is-all-you-need-f576782c5d2>>. Acessado em: 22 de Março de 2023.