



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, 2 de dezembro 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

O estudo, inserido no contexto do abastecimento de água em Curitiba, concentra-se na eficácia da previsão da demanda no Bairro Alto durante os anos de 2018 a 2020. A questão central investigada é como antecipar a demanda de água para um planejamento mais eficiente no contexto de escassez enfrentada pelos habitantes. O propósito do trabalho é contribuir para o controle eficaz dos recursos hídricos, utilizando modelos de previsão, com ênfase na melhoria do abastecimento d'água em um ambiente competitivo. São explorados modelos como ARIMA (do inglês *Auto-Regressive Integrated Moving Average*), DTR (do inglês *Decision tree regressor*), XGBoost (do inglês *eXtreme Gradient Boosting*) e RNN (do inglês *Recurrent Neural Network*) para a previsão de séries temporais, com uma análise comparativa de eficácia. A necessidade de uma solução nova ou melhor surge da escassez de água enfrentada no Bairro Alto, justificando a busca por métodos eficientes de previsão da demanda. A solução proposta, envolve a aplicação de modelos de aprendizado de máquina, como ARIMA, DTR, XGBoost e especialmente RNN, na previsão da demanda de água. A metodologia básica inclui a aplicação desses modelos aos dados coletados pela SANEPAR (Companhia de Saneamento do Paraná). As características que respondem às questões iniciais são avaliadas 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*), destacando que o modelo RNN demonstrou consistentemente os menores erros em todas as análises. Conclui-se que a abordagem proposta contribui significativamente para a previsão da demanda de água, proporcionando um planejamento eficiente e sustentável do abastecimento hídrico no Bairro Alto.

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

Abstract

The study, situated in the context of water supply in Curitiba, focuses on the effectiveness of forecasting water demand in Bairro Alto from 2018 to 2020. The central question investigated is how to anticipate water demand for more efficient planning in the context of scarcity faced by the residents. The purpose of the work is to contribute to the effective control of water resources, utilizing forecasting models, with an emphasis on improving water supply in a competitive environment. Models such as ARIMA (Auto-Regressive Integrated Moving Average), DTR (Decision Tree Regressor), XGBoost (eXtreme Gradient Boosting), and RNN (Recurrent Neural Network) are explored for time series forecasting, with a comparative analysis of effectiveness. The need for a new or better solution arises from the water scarcity in Bairro Alto, justifying the search for more efficient methods of demand forecasting. The proposed solution involves the application of machine learning models such as ARIMA, DTR, XGBoost, and especially RNN in forecasting water demand. The basic methodology includes applying these models to the data collected by SANEPAR (Sanitation Company of Paraná). The features responding to the initial questions are assessed through metrics such as sMAPE (Symmetric Mean Absolute Percentage Error), MAE (Mean Absolute Error), and RRMSE (Root Relative Mean Square Error), highlighting that the RNN model consistently demonstrated the lowest errors in all analyses. It is concluded that the proposed approach significantly contributes to water demand forecasting, providing a more efficient and sustainable planning of water supply in Bairro Alto.

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	20
3	Estrutura da dissertação	23
4	Fluxograma do problema da revisão	24
5	Etapas da revisão	25
6	Modelos de series temporais mais populares na Scopus e WoS	26
7	Significado do modelo SARIMAX	37
8	Fluxograma da árvore de decisão	40
9	Esquema da floresta aleatória	41
10	Impulsionando gradiente com XGBoost e LightGBM	42
11	Compara-se o crescimento em folha com o crescimento em nível	43
12	RNN - <i>recurrent neural network</i>	45
13	Diagrama ilustrativo do funcionamento de uma unidade recorrente gated (GRU)	48
14	RNN vs LSTM vs GRU	49
15	Arquitetura do Transformer	50
16	Modelo de uma Rede Neural Artificial MLP	52
17	A equação da figura realiza o somatório ponderado entre as sinapses de cada neurônio	53
18	Modelo de uma Rede Neural Convolucional	54
19	Correlação de Pearson	62
20	Regressão linear LT01 vs PT01 correlação 97%	62
21	Autocorrelação	65
22	Autocorrelação parcial	65
23	Ruído branco	66
24	Violino no nível do reservatório	66
25	Violino da vazão de recalque	67
26	Decomposição STL aditiva dos dados coletados	67
27	Comparação dos modelos AR, ARX e MA	72
28	Comparação do modelos ARIMAX, SARIMA e SARIMAX	72
29	Comparação dos modelos ARMA e ARIMA	73
30	Comparação dos modelos DTR, RFR, XGBoost, Light GBM	73
31	Modelo RNN e os vários horizontes	73
32	Previsões do modelo Prophet para o reservatório LT01	73
33	Comparação dos modelos ARIMA	81
34	Comparação de modelos de regressão	81

35	Comparação dos modelos na métrica RRMSE	81
36	Comparação dos modelos nas métricas sMAPE, MAE e RRMSE	82
37	Demanda média das variáveis de fluxo	85

Lista de Tabelas

1	Cruzamento de palavras-chave por meio da aplicação de filtros de área . . .	27
2	Resumo dos dados	27
3	Fator de impacto	28
4	Os autores que mais publicam em relação ao tema de pesquisa	29
5	Países com maior número de publicações	30
6	Modelos baseado na literatura e nos artigos	31
7	Descrição estatística dos dados com o filtro aplicado das 18h às 21h	63
8	Teste de Dickey-Fuller Aumentado	64
9	Parâmetros utilizados nos modelos ARIMA e seus antecessores obtidos pelo “autoARIMA” do Python.	69
10	SARIMAX(7, 0, 0) \times (2, 1, [1], 12) Results	69
11	Hiperparâmetros dos modelos	70
12	Resumo dos Hiperparâmetros dos Modelos de Redes Neurais	70
13	Comparação dos modelos de previsão com as métricas de desempenho treino	74
14	Comparação dos modelos de previsão com as métricas de desempenho teste	75
15	Comparação dos modelos de previsão com as métricas de desempenho va- lidação	76
16	Comparação dos modelos de previsão com as métricas de desempenho inteiro	77
17	Teste Nemenyi	78
18	Métricas de avaliação dos modelos	79
19	Comparação dos modelos Ljung Box: Modelos ARIMA com defasagem de 10 para previsão de longo prazo na demanda de água	83
20	Demanda de água	85

Lista de Abreviaturas e Siglas

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>)
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>)
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>)
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	18
1.2	Objetivo Geral	19
1.2.1	Objetivos Específicos	19
1.3	Procedimentos Metodológicos	20
1.3.1	Etapas da Pesquisa	20
1.4	Justificativa da Pesquisa	22
1.4.1	Contribuições	22
1.5	Estrutura da Dissertação	23
2	Revisão da Literatura	24
3	Fundamentos dos Modelos de Previsão de Séries Temporais	32
3.1	Descrição do Problema	32
3.2	Modelos de Séries Temporais Univariados	32
3.2.1	Componente Autorregressivo	33
3.2.2	Média Móvel	34
3.2.3	Modelos ARMA e ARIMA	35
3.3	Modelos de Série Temporal Multivariada	36
3.4	Modelos de Aprendizado de Máquina	37
3.4.1	Prophet	37
3.4.2	Regressão Linear (LR)	38
3.4.3	Árvore de Decisão	39
3.4.4	Floresta Aleatória	40
3.4.5	Gradient Boosting	41
3.5	Redes Neurais Artificiais	43
3.5.1	Rede Neural Recorrente	44
3.5.2	Compreendendo Redes de Memória de Curto e Longo Prazo (LSTM)	45
3.5.3	GRU (Unidade Recorrente Fechada)	46
3.5.4	Análise dos Modelos RNN, LSTM e GRU	48
3.6	Aprendizado Profundo	49
3.6.1	Explorando o Transformer: Além dos Bits e Bytes	50
3.6.2	Rede Neural Artificial	51
3.7	Rede Neural Convolucional	53
3.8	Métricas de Avaliação de Modelos	54

3.8.1	Raiz do Erro Médio Quadrático Relativo	54
3.8.2	Erro Absoluto Médio	55
3.8.3	Erro Percentual Absoluto Médio Simétrico (sMAPE)	55
3.9	Correlação de Pearson	56
3.10	Decomposição STL	56
3.11	Dickey-Fuller	56
3.12	Teste de Ljung-Box	57
3.13	Teste de Significância	59
4	Resultados	61
4.1	Descrição das Etapas	61
4.1.1	Comparação dos Modelos	80
4.2	Aplicação	84
4.2.1	Estudo de Caso 1	84
4.2.2	Estudo de Caso 2	84
5	Conclusões	86
5.1	Propostas Futuras	86
	Referências	87

1 Introdução

No âmbito da demanda d'água, desempenha-se um papel crucial nesse contexto. A água está presente em todos os aspectos do cotidiano, sendo essencial tanto para atividades básicas, como o banho, quanto para o consumo. A compreensão da importância desse recurso é vital, uma vez que ele influencia diretamente a qualidade de vida em várias esferas. Existem três tipos principais de Aprendizado de Máquina: Supervisionado, Não Supervisionado e por Reforço. No Aprendizado Supervisionado, é necessário apresentar a resposta desejada para cada exemplo. No Aprendizado Não Supervisionado, os exemplos são fornecidos sem rótulos, sendo agrupados por similaridades. No Aprendizado por Reforço, o algoritmo recebe sinais de reforço, sem a resposta correta, e determina a eficácia de suas hipóteses (SILVA; GOMES, 2021).

Nesse caso, os modelos usados nessa dissertação são os modelos de aprendizado de máquina supervisionados. Esses modelos para série temporal são usados na literatura. 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 abordagem desse estudo envolve o problema de abastecimento d'água que afetou a cidade de Curitiba no Bairro Alto entre os anos de 2018 e 2020. Durante esse período, os habitantes enfrentaram a escassez de água, sendo necessário implementar rodízios, alternando períodos com e sem fornecimento. Os dados utilizados foram coletados pela companhia de saneamento do Paraná (SANEPAR).

A previsão da demanda de água ao longo do tempo é essencial para um planejamento sustentável e eficiente do abastecimento hídrico, especialmente no contexto 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 o modelo clássico ARIMA (*Auto-Regressive Integrated Moving Average*) para realizar previsões diárias da demanda de água ao longo do tempo.

Os modelos de aprendizado de máquina foram aplicados na previsão de séries temporais com dados coletados pela SANEPAR. Cada modelo tem suas particularidades, mas os modelos de aprendizado de máquina podem ser otimizados da mesma forma que os modelos clássicos do tipo ARIMA e RNN (*Recurrent Neural Network*). Os dados coletados pela SANEPAR, utilizados para previsão, referem-se ao abastecimento d'água no Bairro Alto durante o período de 2018 a 2019.

Por meio da utilização de métodos e modelos de séries temporais, realiza-se a previsão do nível do reservatório (LT01), incorporando diversos modelos nesse processo. Dentre esses modelos, incluem-se os clássicos, como ARIMA e suas variantes, tais como AR (do

inglês *Auto-Regressive*), ARX (do inglês *Auto-Regressive with Exogenous input*), MA (do inglês *Moving Average*), ARMA (do inglês *Auto-Regressive Moving Average*), SARIMA (do inglês *Seasonal Auto-Regressive Integrated Moving Average*), ARIMAX (do inglês *Auto-Regressive Integrated Moving Average with Exogenous input*), SARIMAX (do inglês *Seasonal Auto-Regressive Integrated Moving Average with Exogenous input*), Prophet, além dos modelos de aprendizado de máquina, como árvore de decisão, floresta aleatória, XGBoost (do inglês *Extreme Gradient Boosting*), Light GBM (do inglês *Light Gradient Boosting Machine*), e redes neurais como LSTM (do inglês *Long Short-Term Memory*), GRU (do inglês *Gated Recurrent Unit*), RNN, Transformer, CNN (do inglês *Convolutional Neural Network*), ANN (do inglês *Artificial Neural Network*). A diversidade de abordagens busca otimizar a precisão das previsões, destacando a abrangência e sofisticação desse trabalho.

1.1 Contexto da Pesquisa

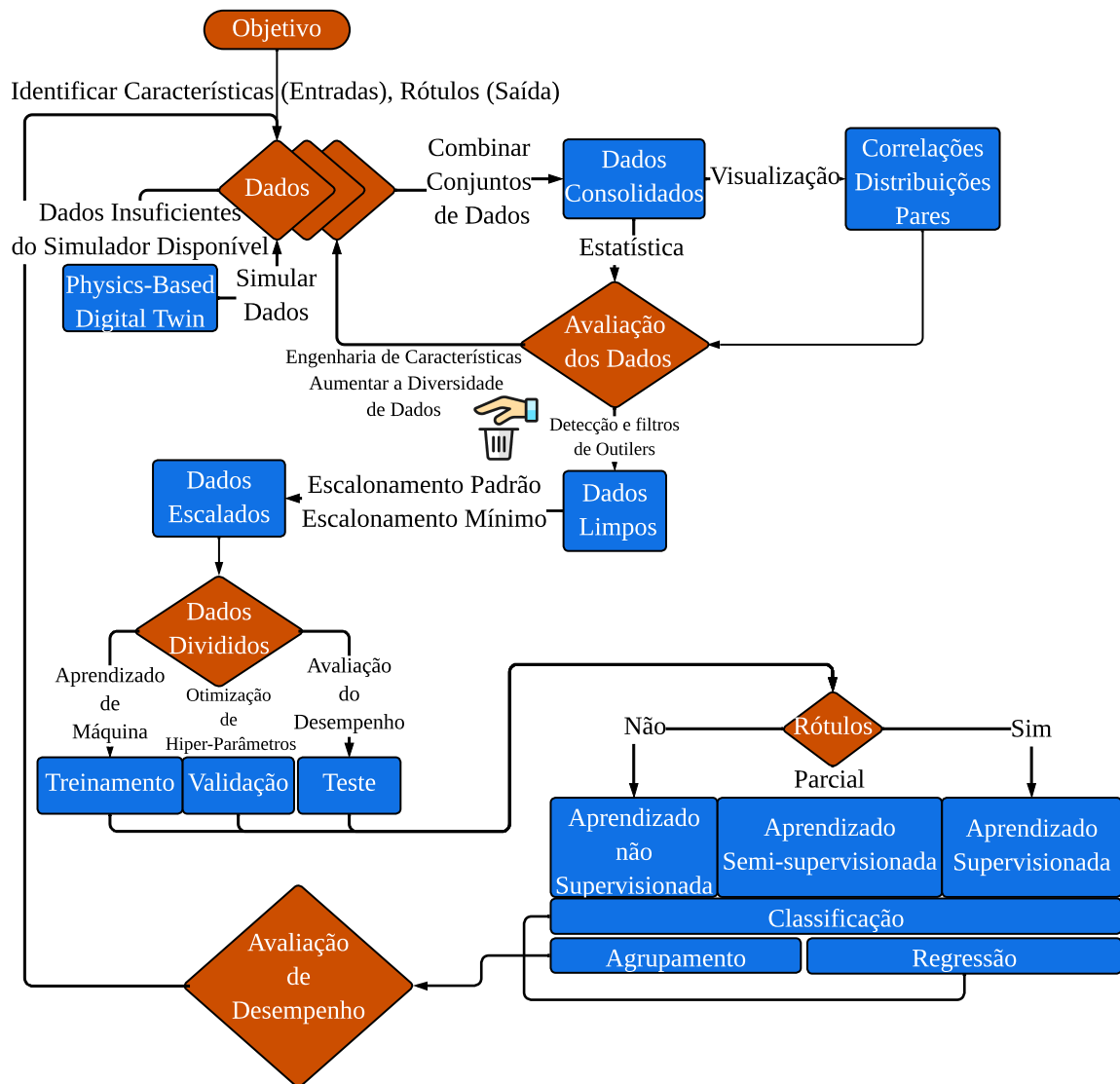
Torna-se evidente que a análise de séries temporais e previsões são ferramentas valiosas para apoiar o processo de tomada de decisão em curto, médio e longo prazo. Devido às não linearidades, sazonalidades e tendências que podem ocorrer nos dados temporais de abastecimento de água, o desenvolvimento de modelos de previsão eficientes torna-se uma tarefa desafiadora (RIBEIRO et al., 2021).

Na Figura 1, as etapas de como a análise de dados e a seleção dos modelos devem ocorrer são apresentadas. Essa escolha pode ser conduzida de modo a determinar o que deve ser previsto na variável. Feito isso, a primeira etapa envolve a preparação dos dados, garantindo que cada um tenha sido identificado com seus rótulos de entrada e saída. É imperativo que os dados não contenham NaN (do inglês *not a number*) ou dados ausentes, evitando assim falsos positivos.

Ao realizar essa etapa, deve visualizar os dados para garantir que estejam carregados corretamente e em um tamanho tolerável, o que é conhecido como avaliação dos dados. Uma vez que os dados estejam limpos e devidamente carregados, sem falsos positivos, a divisão dos dados pode ser efetuada. A otimização dos dados para os modelos pode ocorrer de diversas maneiras, como a utilização da biblioteca Optuna em Python, que emprega a otimização Bayesiana para cada modelo pré-listado, reduzindo assim o tempo de processamento.

A validação é uma prática comum em conjuntos de dados extensos, permitindo que os modelos interajam mais eficientemente com os dados e proporcionem resultados mais precisos. Após essa etapa, na escolha dos modelos, há a possibilidade de determinar se o modelo é de série temporal, classificação, agrupamento ou regressão. Posteriormente, ao

Figura 1: Paradigma de aprendizado de máquina



Fonte: Adaptado de (HEDENGREN, 2023)

listar os modelos, cada um deles deve passar por uma avaliação com métricas específicas para verificar a precisão de seus resultados.

1.1.1 Motivação da Pesquisa

A motivação desta pesquisa baseia-se na situação enfrentada por Curitiba e região metropolitana, conforme destacado 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. 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, em 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, 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.

1.2 Objetivo Geral

O objetivo desta dissertação de mestrado é desenvolver séries temporais para auxiliar na tomada de decisões em situações de escassez de água no Bairro Alto, em Curitiba. A ideia é utilizar modelos de séries temporais como suporte para a melhor gestão desse problema enfrentado pela cidade.

Diversos modelos de regressão serão avaliados, com foco especial em modelos de redes neurais e o Prophet. Destaca-se a ênfase em modelos de *gradient boosting*, redes neurais artificiais, além do ARIMA e suas variações contemporâneas.

1.2.1 Objetivos Específicos

1. Comparação de Modelos e Técnicas de Otimização

Compara modelos de séries temporais. Avalia técnicas de otimização baseadas em otimização bayesiana utilizando o algoritmo de TPE (do inglês *Tree-structured Parzen Estimator*).

2. Desempenho dos Modelos de Séries Temporais

Avalia o desempenho dos diferentes modelos de séries temporais. Analisa a precisão, eficiência e capacidade de previsão desses modelos em conjuntos de dados específicos.

3. Implementação de Estratégias de Otimização

Explora estratégias de otimização baseadas em otimização bayesiana utilizando o algoritmo TPE. Implementa técnicas de otimização para ajustar hiperparâmetros dos modelos de séries temporais.

4. Identificação da Melhor Combinação de Modelo e Otimização

Identifica a combinação mais eficaz de modelo de séries temporais e configurações de otimização. Aprimora a precisão das previsões e otimiza o processo de modelagem.

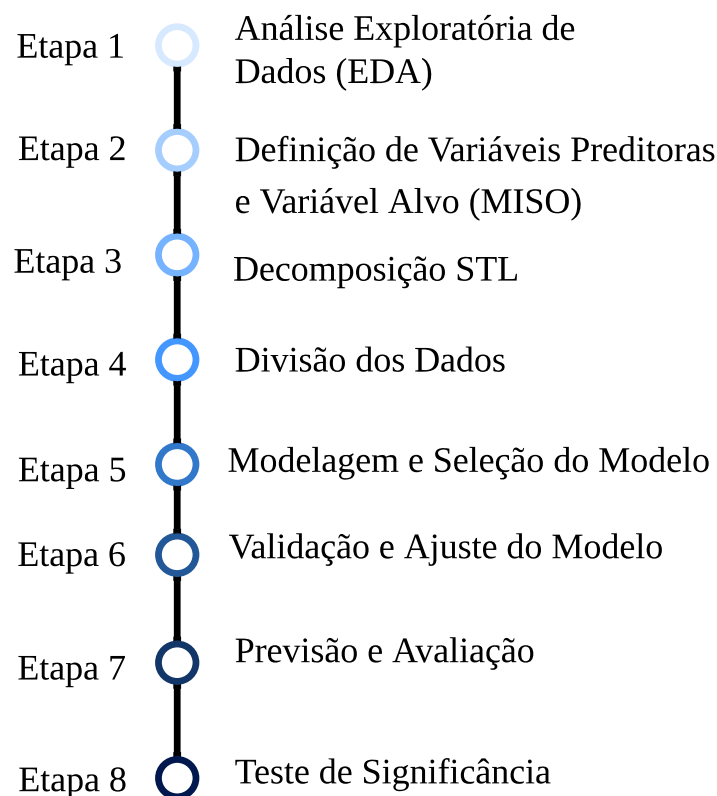
1.3 Procedimentos Metodológicos

Com o objetivo de realizar previsões e fazer comparações entre os modelos obtidos na revisão sistemática da literatura, a pesquisa adotará um processo metodológico bem definido. Esse processo é detalhado na subseção 1.3.1 desta seção, onde são 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, conforme apresentado na Figura 2.

Figura 2: Mapa das Etapas



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 uma série temporal em três componentes: sazonalidade, tendência e resíduo. Essa decomposição permite. Decompor séries temporais em sazonal captura 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, AR, MA, ARIMA, SARIMA, SARIMAX e modelos de aprendizado de máquina como RNN, LSTM, GRU, Transformer (Transformador), DTR (do inglês *Decision tree regressor*), LR (do inglês *Linear Regression*), XGBoost (do inglês *eXtreme Gradient Boosting*), Light GBM 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

A dissertação fundamenta-se em modelos previamente não explorados neste contexto, como GRU, LSTM, XGBOOST, LGBM, Transformer, RNN, ANN e CNN. A primeira contribuição aborda a previsão da demanda de água em Curitiba, considerando elementos como consumo e consumo de energia durante picos.

Vários desses modelos na literatura não foram aplicados neste tema de demanda de água, utilizando-os para comparação entre si e em relação aos modelos que já foram trabalhados nesse contexto de demanda d'água, a fim de certificar se esses modelos podem ou não ser utilizados nesse contexto.

Os modelos do tipo ARIMA e suas variantes foram aplicados neste tema, como demonstrado por (BUYUKSAHIN; ERTEKIN, 2019a; BHANGU; SANDHU; SAPRA,

2022). Alguns outros modelos, apesar de suas vantagens, ainda não foram devidamente aplicados, como é o caso do modelo de RNN (SHIH; SUN; LEE, 2019a), que se mostrou significativamente superior aos outros 19 modelos listados ao longo deste trabalho.

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.

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

Finalizando, o Capítulo 5 apresenta os resultados da pesquisa.

Figura 3: Estrutura da dissertação



2 Revisão da Literatura

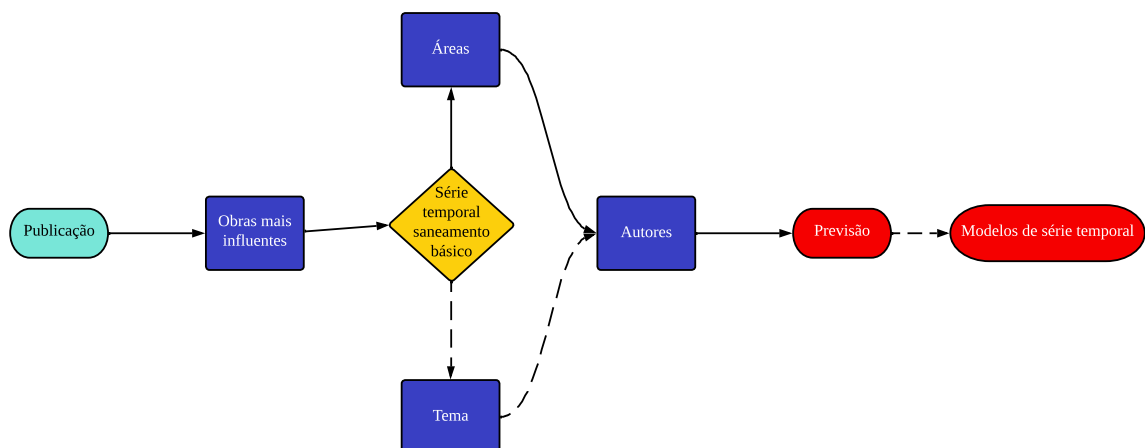
Este capítulo apresenta uma revisão sistemática da literatura nos temas relacionados a previsão de séries temporais e aplicações em hidrologia e mais especificamente em abastecimento de água. A revisão bibliográfica realizada consiste em uma análise abrangente e crítica das principais fontes de literatura. 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.

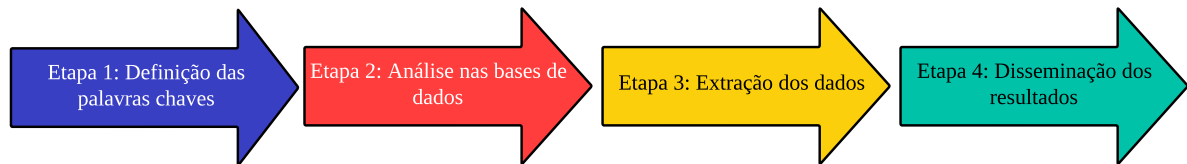
Figura 4: Fluxograma do problema da revisão



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*” .

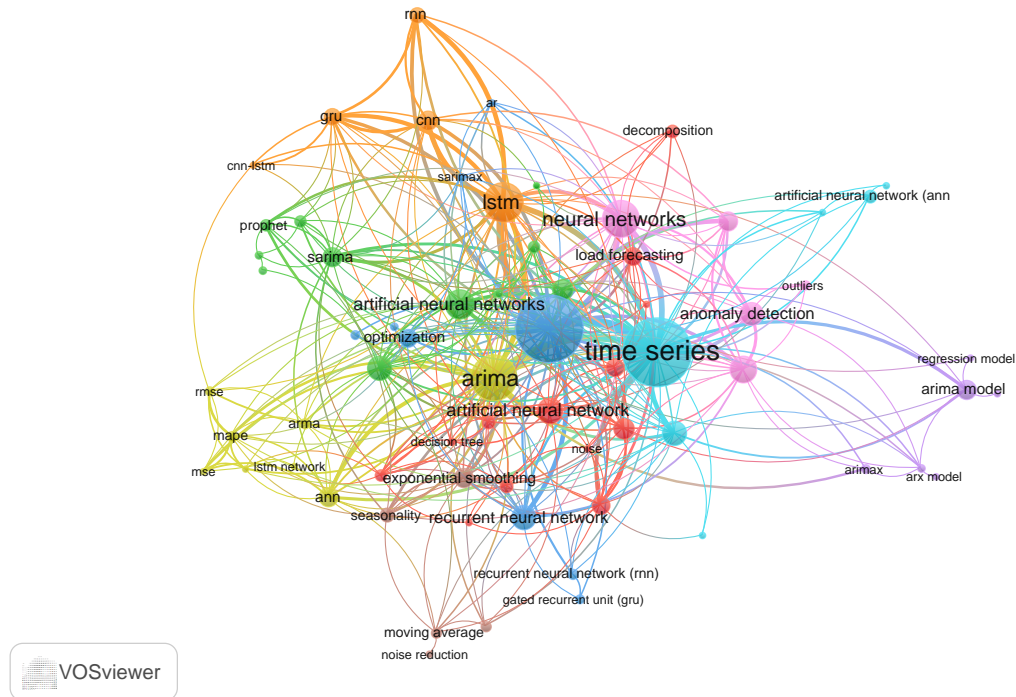
Na etapa seguinte, é realizada uma avaliação preliminar de cada artigo obtido, sem aplicar nenhum filtro anual nas buscas. Analisar todos os artigos dessa maneira resultaria em um número elevado, por exemplo, no banco de dados Scopus, existem 831 artigos, enquanto na WoS, são encontrados 98 artigos, 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 da condução da RSL, esses artigos foram excluídos.

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%

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 o número de documentos publicados em um ano e o número de documentos publicados no

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

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.

Foi realizada uma investigação dos artigos na RSL. Esses artigos retratam alguns

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

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), Sadai et al. (2019b), Yang et al. (2019), Buyuksahin e Ertekin (2019b), Carvalho Jr. e Costa Jr. (2019), Liu e Chen (2019), Shih, Sun e Lee (2019b), 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 (2019a)	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 (2019a)	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, Tyralis 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.

3.1 Descrição do Problema

A descrição do problema, centrada no abastecimento de água, é crucial. Ela apresenta variáveis-chave, como Bombas de Sucção (B1, B2 e B3), cuja frequência máxima é de 60 Hz, Nível do Reservatório (Câmara 1) representado por LT01 (m^3), Vazão de entrada (FT01) em (m^3/h), Vazão de gravidade (FT02) em (m^3/h), Vazão de recalque (FT03) em (m^3/h), Pressão de Sucção (PT01SU) medida em metros de coluna d'água (mca) e Pressão de Recalque (PT02RBAL) também em metros de coluna d'água (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 obtidos da SANEPAR sejam de 2018 a 2020, sendo que o ano de 2020 causou muitas irregularidades. É possível remover esse ano para melhor trabalhar com os dados.

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 ruído na série temporal. Com isso em mente, foram retiradas as variáveis B3 e FT02 restando assim as variáveis de previsão com as variáveis que tiveram correlação significativa.

A dimensão dos dados fornecidos pela SANEPAR foi de 26.306 linhas e 9 colunas. Essas colunas representam as variáveis listadas anteriormente, com a exclusão das duas variáveis B3 e FT02, resultando em apenas 6 variáveis no formato de variáveis exógenas e uma variável para previsão. Também é relevante observar que o ano de 2020, devido às muitas anomalias nos dados, foi removido para mitigar a variação nos dados ao longo do tempo. Com essa abordagem, restam 17.522 observações, com o intervalo temporal compreendido entre 2018 e 2019. Essa decisão foi tomada para evitar que o modelo sofra excessivamente com variações temporais.

3.2 Modelos de Séries Temporais Univariados

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 consi-

deração 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.

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).

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

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

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.

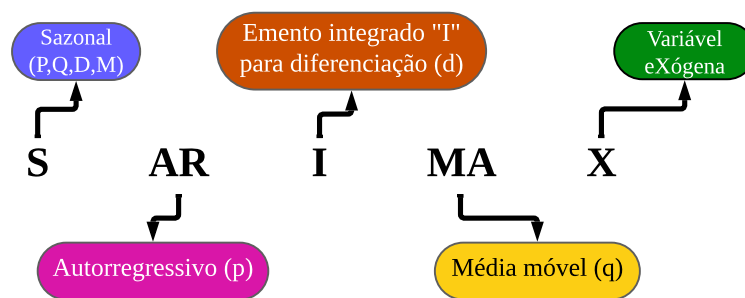
d : Diferença de tendência ordem— d é o número de diferenças não sazonais necessárias para estacionariedade. Intuitivamente, isso seria semelhante a afirmar que é provável que seja a mesma temperatura amanhã se a diferença de temperatura nos últimos três dias tiver sido muito pequena.

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 M, senão $P = 0$, $Q \geq 1$ se a ACF for negativa na defasagem M, 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 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 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).

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.

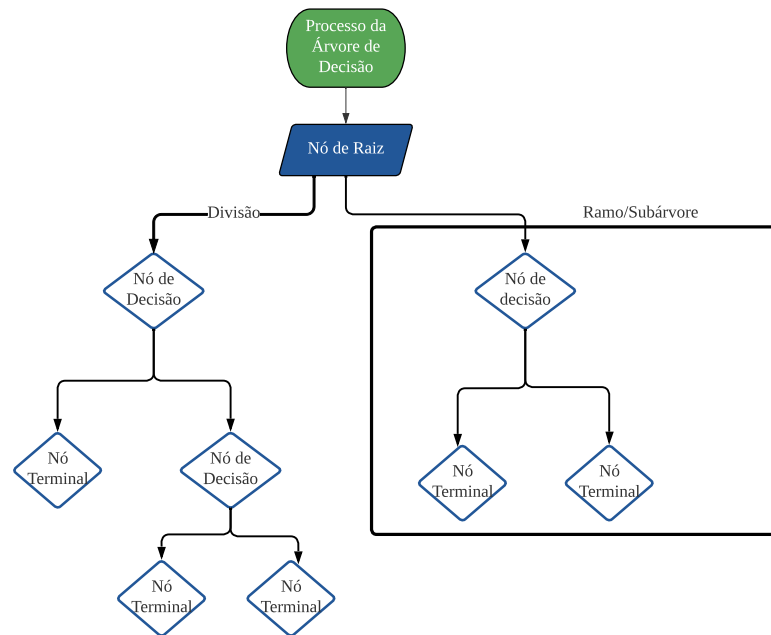
A equação (3.8) mencionada, conhecida como **OLS**, é amplamente utilizada na regressão linear (KORSTANJE, 2021). Esse método é conhecido por ser rápido de ajustar, pois requer apenas cálculos matriciais para estimar os coeficientes β . No entanto, ele é mais adequado para processos lineares e pode ser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

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 8, 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)

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.

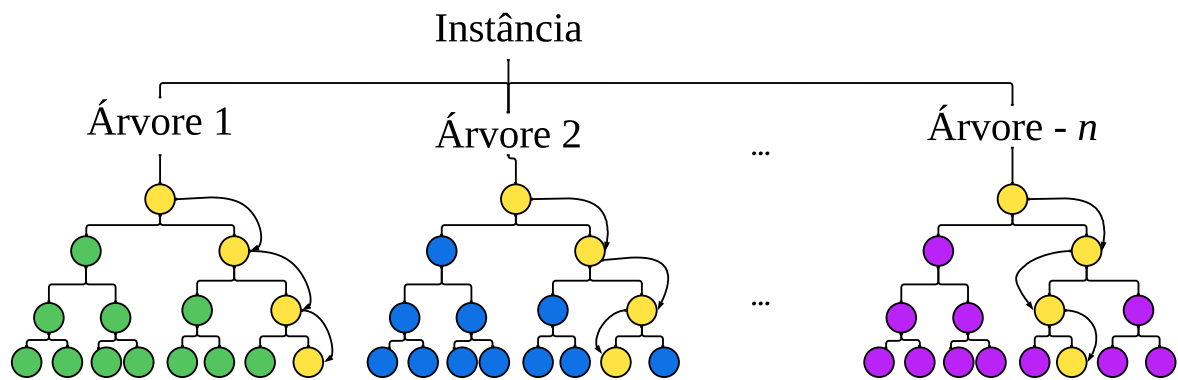
3.4.4 Floresta Aleatória

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 9, 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.

Figura 9: Esquema da floresta aleatória



3.4.5 Gradient Boosting

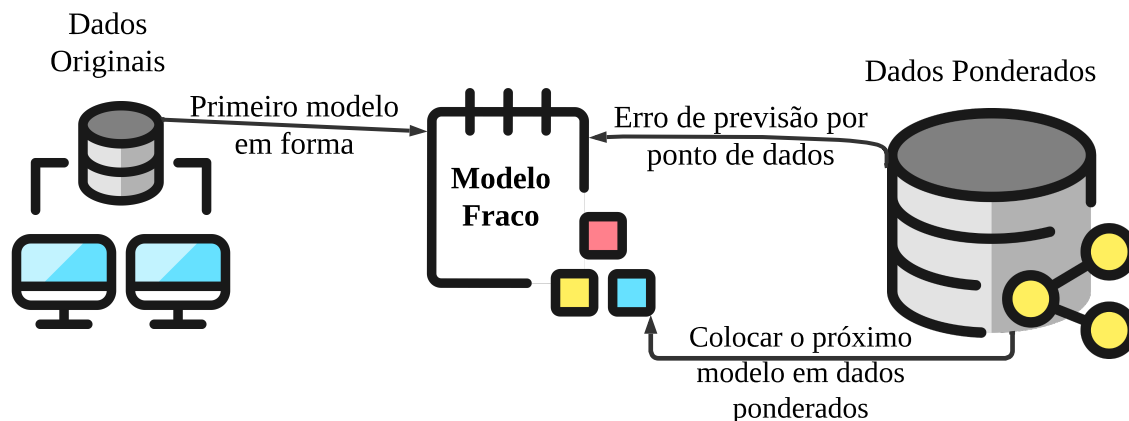
O aumento de gradiente (do inglês *gradient boosting*) é um método que combina vários modelos de árvore de decisão para realizar previsões. Cada uma dessas árvores de decisão é única, pois a diversidade é um elemento importante nesse processo. A diversidade é alcançada através de um processo chamado *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 10 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 10: Impulsionando gradiente com XGBoost e LightGBM



Fonte: Adaptação de Korstanje (2021)

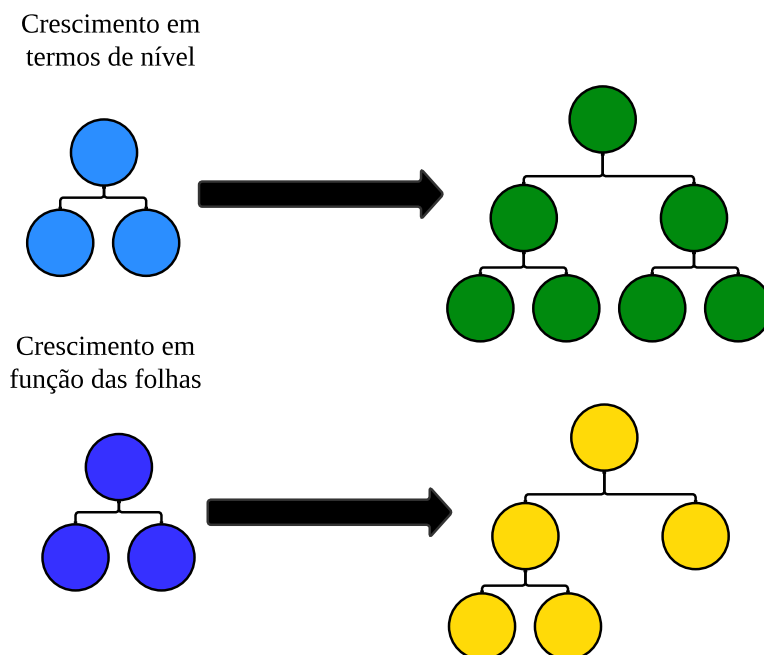
A diferença entre XGBoost e LightGBM: Uma alternativa proposta pelo XGBoost é a segmentação baseada em histograma. Nesse caso, em vez de iterar por todas as partições possíveis, o modelo constrói um histograma para cada variável e utiliza-os para encontrar a melhor divisão geral entre as variáveis. O 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 11. 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.

Na Figura 11, 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 11, apresenta um diagrama que ilustra o crescimento de uma árvore em termos de níveis e folhas. O diagrama possui duas partes.

No crescimento de árvore em folhas, como no LightGBM, novas folhas são adiciona-

Figura 11: Compara-se o crescimento em folha com o crescimento em nível



Fonte: Adaptação de Korstanje (2021)

das 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. 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.

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. 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.

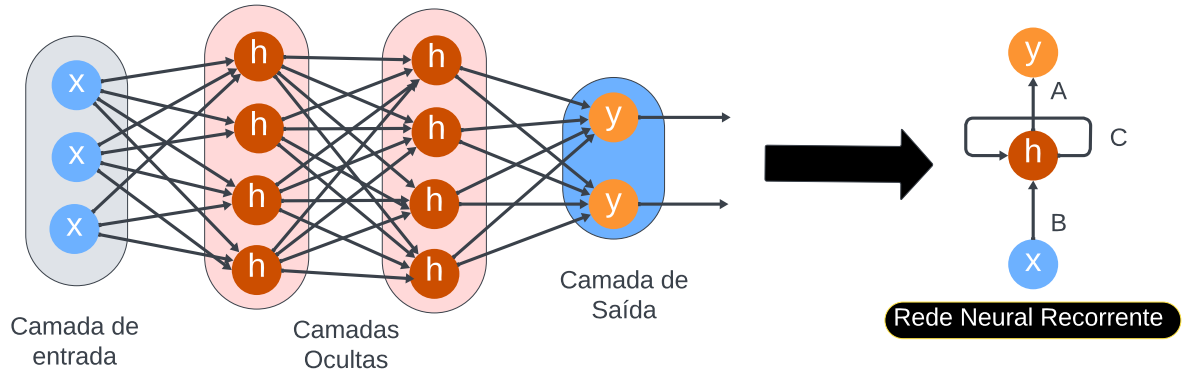
A característica principal das RNNs é que elas contêm laços em sua estrutura, permitindo que informações anteriores influenciem o processamento de informações subsequentes. Isso significa que a saída em um determinado passo de tempo não depende apenas da entrada atual, mas também das entradas anteriores 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 (3.9) 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 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 12, 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.

Figura 12: 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.

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

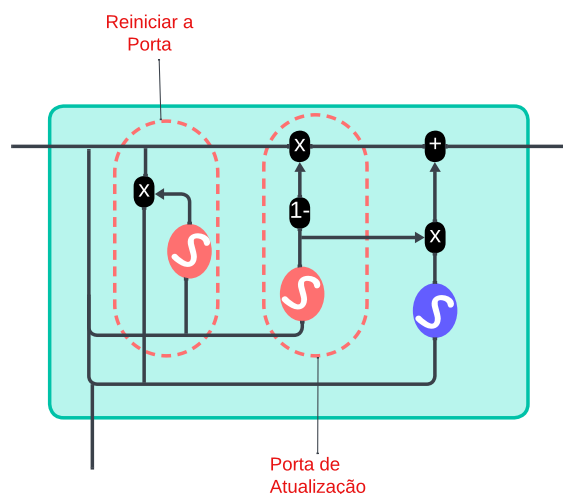
Novo Estado Oculto (h_t) : Combinação ponderada do estado anterior e do novo candidato.

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

Foi inventada uma camada RNN mais avançada, designada GRU. A célula GRU possui mais parâmetros, conforme mostrado na Figura 13. 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 13 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 13 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 13: 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

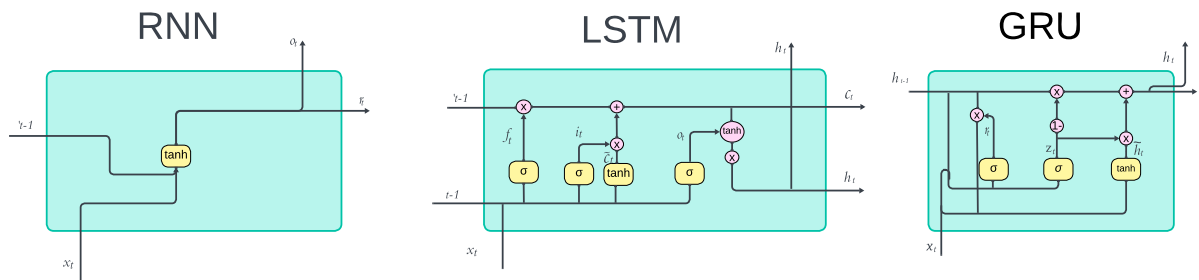
As GRUs, as LSTMs e as RNNs são variações das arquiteturas de redes neurais, todas projetadas para abordar a dificuldade de capturar dependências temporais em sequências de dados. Cada uma dessas abordagens tem características distintas que influenciam sua capacidade de lidar com esse desafio.

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 14, há um esquema que ilustra as arquiteturas das RNNs, LSTMs e GRUs, permitindo uma visualização das diferenças entre essas abordagens.

Na Figura 14, 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. A Figura 14 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 14: 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 14 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

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

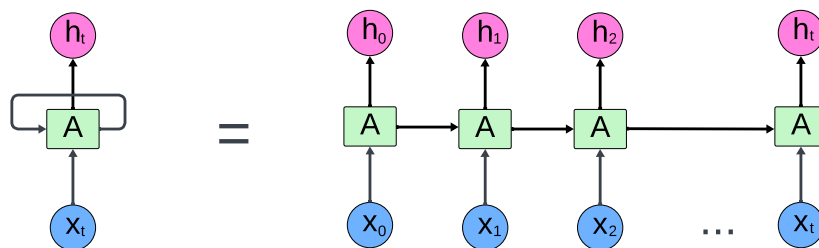
A arquitetura de rede neural Transformer representa um avanço significativo nas tarefas relacionadas. Foi introduzida por (VASWANI et al., 2017) e revolucionou a maneira como as redes neurais lidam com sequências de dados, superando limitações anteriores, como a dependência sequencial e a complexidade 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 autoatencão, 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 autoatencão 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 de séries temporais. Na Figura 15 tem o esquema de como a rede neural Transformer é abordada.

Figura 15: Arquitetura do Transformer



Fonte: Adaptado de Esposito (2021)

3.6.2 Rede Neural Artificial

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 inglês *backpropagation*) 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 17. 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.

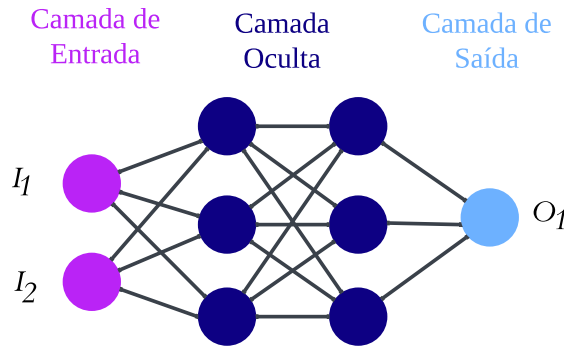
A Figura 17 é um diagrama de um modelo ANN, que é um modelo computacional inspirado no cérebro humano. Ele consiste de um grande número de nós conectados, cada um realizando uma operação matemática simples. O diagrama consiste de três camadas: uma camada de entrada, uma camada oculta e uma camada de 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.

$$I = [I_1, I_2] = \text{Vetor de Entrada}$$

$$O = [O_1] = \text{Vetor de Saída}$$

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

Figura 16: Modelo de uma Rede Neural Artificial MLP



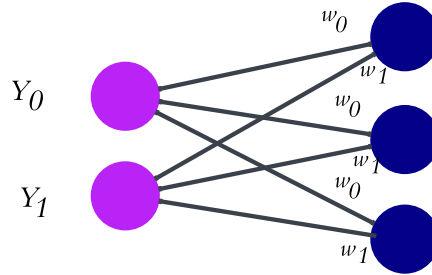
Fonte: Adaptado de Grübler (2018)

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

o funcionamento geral de uma rede MLP está representada na Figura 16. 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 17 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 17: A equação da figura realiza o somatório ponderado entre as sinapses de cada neurônio



Fonte: Adaptado de Grübler (2018)

3.7 Rede Neural Convolucional

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, no caso de séries temporais, que são sequências unidimensionais de dados amostrados em intervalos de tempo regulares (SILVA, 2021).

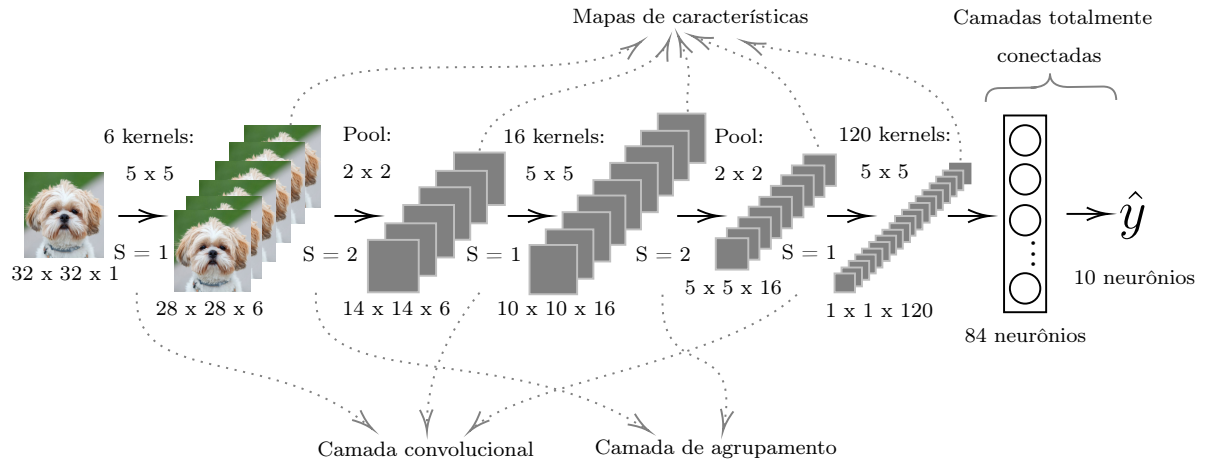
A camada convolucional tem como objetivo extrair as características mais importantes da entrada. Dessa forma, sua saída é um mapa de características obtido a partir da convolução da entrada com um *kernel* aprendido, seguido da aplicação de uma função de ativação não linear (LUCAS, 2019). Os mapas de 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 18 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).

Uma vantagem das camadas de convolução é o compartilhamento do vetor de

Figura 18: Modelo de uma Rede Neural Convolucional



Fonte: Lucas (2019)

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 18, 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

As métricas de erro são utilizadas na análise de séries temporais para avaliar se o modelo está se comportando conforme desejado. Quanto menor for o erro de cada métrica, melhor será o modelo trabalhado.

3.8.1 Raiz do Erro Médio Quadrático Relativo

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,

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

onde n número total de observações ou amostras no conjunto de dados, y_i valor real da observação i , \hat{y}_i valor previsto ou estimado da observação i pelo modelo, $\sum_{i=1}^n$ soma sobre todas as observações no conjunto de dados.

3.8.2 Erro Absoluto Médio

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:

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}_i| \quad (3.25)$$

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

3.8.3 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:

$$sMAPE = \frac{1}{n} \sum_{i=1}^n \frac{2|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)} \times 100 \quad (3.26)$$

3.9 Correlação de Pearson

A equação do coeficiente de correlação de Pearson é dada por:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum (x_i - \bar{x})^2)(\sum (y_i - \bar{y})^2)}} \quad (3.27)$$

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.

3.10 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).

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 clara dos padrões de variação presentes nos dados.

A decomposição STL é formalmente definida como:

$$y_t = f(S_t, T_t, R_t) = \begin{cases} y_t = S_t + T_t + R_t & \text{modelo aditivo} \\ y_t = S_t T_t R_t & \text{modelo multiplicativo} \end{cases} \quad (3.28)$$

3.11 Dickey-Fuller

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

$$z_t = y_t + \theta \beta_t, \quad t = 1, \dots, T, \quad (3.29)$$

$$\hat{\rho}_{\text{DF}} - 1 = \frac{\sum_{t=1}^T z_{t-1} \Delta z_t}{\sum_{t=1}^T z_{t-1}^2} \quad (3.30)$$

De (3.30) 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:

$$T(\hat{\rho}_{\text{DF}} - 1) = T \frac{\sum_{t=1}^T z_{t-1} \Delta z_t}{\sum_{t=1}^T z_{t-1}^2} \quad (3.31)$$

e

$$\hat{\tau}_{\text{DF}} = \frac{\hat{\rho}_{\text{DF}} - 1}{\hat{\sigma}_{\text{DF}} \left(\sum_{t=1}^T z_{t-1}^2 \right)^{-1/2}} \quad (3.32)$$

de (3.32) 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 (3.29), então quando $\rho = 1$,

$$T(\hat{\rho}_{\text{DF}} - 1) \xrightarrow{d} \frac{W(1)^2 - 1}{2 \int_0^1 W(r)^2 dr} - \left(\frac{\theta}{\sigma} \right)^2 \frac{\pi}{\int_0^1 W(r)^2 dr}, \text{ como } T \rightarrow \infty \quad (3.33)$$

$$\hat{\tau}_{\text{DF}} \xrightarrow{d} [1 + 2(\theta/\sigma)^2 \pi]^{-1/2} \left\{ \frac{W(1)^2 - 1}{2 \left(\int_0^1 W(r)^2 dr \right)^{1/2}} - \frac{(\theta/\sigma)^2 \pi}{\left(\int_0^1 W(r)^2 dr \right)^{1/2}} \right\} \quad (3.34)$$

$$\text{como } T \rightarrow \infty \quad (3.35)$$

a partir da equação (3.35), onde \xrightarrow{d} denota convergência na distribuição e onde $\{W(r), r \in [0,1]\}$ denota o movimento Browniano padrão.

3.12 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.

Este teste é às vezes conhecido como o teste de Ljung-Box Q, e está intimamente

ligado ao teste Box-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).

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):

$$Q = n(n+2) \sum_{k=1}^h \frac{\hat{\rho}_k^2}{n-k} \quad (3.36)$$

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:

$$Q > \chi_{1-\alpha, h}^2 \quad (3.37)$$

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)

$$Q_{BP} = n \sum_{k=1}^h \hat{\rho}_k^2 \quad (3.38)$$

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.

3.13 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 (3.39) e (3.40) 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).

$$\chi_F^2 = \frac{12N}{K(K+1)} \left[\sum_{j=1}^K R_j^2 - \frac{K(K+1)^2}{4} \right] \quad (3.39)$$

$$R_j = \frac{1}{N} \sum_{i=1}^N r_{ei}^j \quad (3.40)$$

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1)\chi_F^2} \quad (3.41)$$

As estatísticas F_F mostrados na equação (3.41) obedecem à distribuição F com graus de liberdade $K - 1$ e $(K - 1)(N - 1)$. 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 (3.41), 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:

$$CD = q_{\alpha} \sqrt{\frac{k(k+1)}{6N}} \quad (3.42)$$

na equação do CD, q_{α} representa o valor crítico obtido da Tabela 17 de teste de Nemenyi, k é o número de classificadores e N é o número total de amostras (LIU; XU, 2022).

4 Resultados

Neste capítulo, é fornecida uma síntese dos resultados obtidos até o momento. Apresenta-se um resumo sucinto da análise de resultados das principais realizações e descobertas alcançadas.

4.1 Descrição das Etapas

Análise exploratória dos dados (EDA) a partir da **Etapa 1**, foi realizado o EDA (do inglês *Exploratory Data Analysis*) para processar os dados obtidos até o momento. A análise exploratória de dados foi promovida por John Tukey (BANDARA; HYNDMAN; BERGMEIR, 2021) 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. Nessa análise da EDA, serão abordadas várias análises, como a correlação de Pearson, para iniciar e verificar quais variáveis podem ser excluídas como ruído e não têm correlação com a variável LT01. Nesse caso, as variáveis retiradas são consideradas negativas e têm pouca correlação com o LT01, as únicas variáveis que apresentam correlação negativa são B3 e FT02, mesmo sendo uma correlação negativa, essas variáveis possuem uma correlação inviável ou próxima de 0, levando à decisão de removê-las.

Uma análise bem feita do EDA mostra tudo que os dados podem ter. Esses dados fornecidos pela companhia SANEPAR são coletados em campo por hora, como por exemplo, a cada hora possui um valor esperado. Nisso, pode haver os famosos NaN, que representam a falta de dado coletado, como em um dia em que as bombas podem ter ido para manutenção. Esses NaNs também podem ser registrados como uma anomalia nos dados.

A Figura 19 mostra 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 0,97 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.

Nesse conjunto de dados que está sendo trabalhado, há uma forte correlação da variável PT01 nos modelos de LR e DTR (do inglês *Decision Tree Regressor*). Esses modelos foram escolhidos para trabalhar com o LT01 como entrada e o PT01 como saída. A Figura 20 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 19: Correlação de Pearson

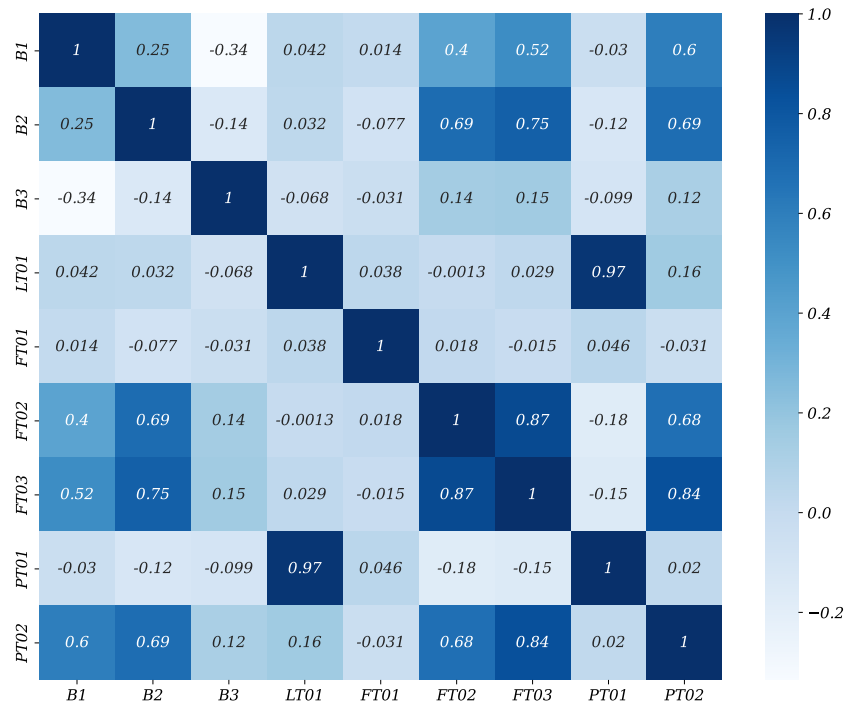
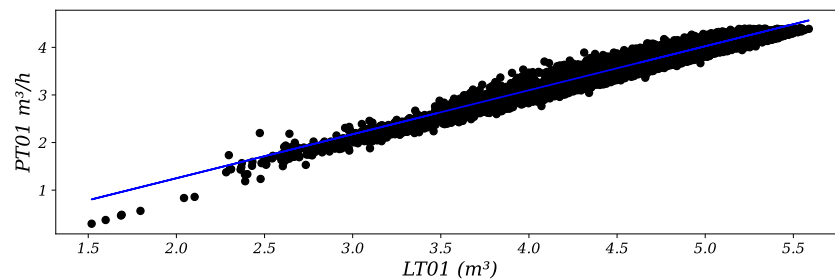


Figura 20: Regressão linear LT01 vs PT01 correlação 97%



A Tabela 7, o desvio padrão é representado pela sigla STD, que corresponde à expressão em inglês *standard deviation*. 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 máximo que o tanque pode alcançar é de $5,26m^3$ (equivalente a 5264.56 litros). As bombas são ativadas em sua potência máxima para evitar que sejam acionadas quando o nível do tanque estiver dentro dessa faixa. No entanto, a bomba 1 ainda estaria operando para completar o nível do tanque caso esteja dentro dessa faixa. Nessa Tabela 7, foram filtrados os horários de pico nos quais pode ocorrer a maior demanda d’água.

A realização da EDA consegue mostrar que os dados estão sendo coletados de maneira significativa, ao observar as correlações e possibilita trabalhar com as variáveis que apresentam correlação maior ou não negativa. Nos modelos ARIMA e seus antecesso-

Tabela 7: Descrição estatística dos dados com o filtro aplicado das 18h às 21h

18h a 21h	B1	B2	B3	LT01	FT01	FT02	FT03	PT01	PT02
Contagem	2921	2921	2921	2921	2921	2921	2921	2921	2921
Média	55,98	30,60	5,26	3,18	86,52	132,87	117,62	4,04	22,55
STD	10,00	15,63	15,57	0,67	124,41	18,48	12,83	0,71	2,92
Min.	0	0	0	0,29	0	0	0,03	0,88	0
25%	57,99	31,63	0	2,77	0	123,40	112,52	3,61	21,98
50%	57,99	35,20	0	3,24	0,12	132,88	118,18	4,10	22,87
75%	57,99	38,17	0	3,71	258,48	142,89	124,48	4,59	23,05
Max.	59,99	59,99	59,99	4,39	370,35	277,94	167,78	5,31	28,08

res, ele utiliza o ACF (do inglês *Auto-Correlation Function*) e o PACF (do inglês *Partial Auto-correlation Function*) para analisar esses métodos. A análise destes métodos para o modelo ARIMA permite otimizar os parâmetros do modelo.

O ACF 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 $-12,515$ 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 de p, aproximadamente de $0,000000000000000000000000262$, expresso de forma mais concisa como $2,62 \times 10^{-23}$ usando a notação científica, está associado ao teste ADF. Este valor-p 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 contexto do teste ADF, a hipótese nula é a presença de raiz unitária na série temporal, indicando não estacionariedade. Portanto, um valor de p baixo, geralmente abaixo de um nível de significância predefinido, como 0,05, sugere que a série temporal é estacionária, enquanto um valor de p alto sugere não estacionariedade. Dado o valor de p de $2,62 \times 10^{-23}$, evidencia-se uma probabilidade muito baixa, indicando forte suporte contra a hipótese nula e sugerindo que a série temporal é estacionária. Na Tabela 8, são apresentados todos os dados do teste para estacionalidade. Os resultados indicam fortes evidências contra a hipótese nula. Com um teste ADF de $-12,515$ e um valor de p extremamente baixo

de $2,62 \times 10^{-23}$, rejeita-se a hipótese nula de presença de raiz unitária. Os 44 atrasos utilizados e as 17.477 observações corroboram a análise estatística.

Tabela 8: Teste de Dickey-Fuller Aumentado

Teste ADF	$-12,515$
Valor de p	$2,62 \times 10^{-23}$
Atrasos utilizados	44
Número de observações	17.477
Valor crítico (1%)	$-3,431$
Valor crítico (5%)	$-2,862$
Valor crítico (10%)	$-2,567$

Ao comparar a estatística de teste ADF com os valores críticos, observa-se que está significativamente abaixo deles em todos os níveis de significância (1%, 5%, 10%). Portanto, a conclusão é de que os dados não possuem raiz unitária, indicando que são estacionários.

Na Figura 21, pode-se observar a diferença entre a autocorrelação (ACF) exibida na Figura 21 e a autocorrelação parcial (PACF) exibida na Figura 22. 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. O intervalo de confiança padrão de 95% é representado pela marca azul nas Figuras 21 e 22. 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 21 é fundamental para a interpretação do teste ADF. Em uma série de ruído branco, os valores são completamente aleatórios e não apresentam correlação significativa. Portanto, quando há correlação presente na série, isso indica a existência de padrões ou dependências entre os valores, o que pode ser explorado para a modelagem e previsão da série temporal.

Demonstrar que uma série temporal tem ou pode ter um ruído branco também é conveniente para a análise da EDA. Na Figura 23, 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.

Nesse exemplo, ao utilizar os dados da SANEPAR, a série temporal trabalhada é

Figura 21: Autocorrelação

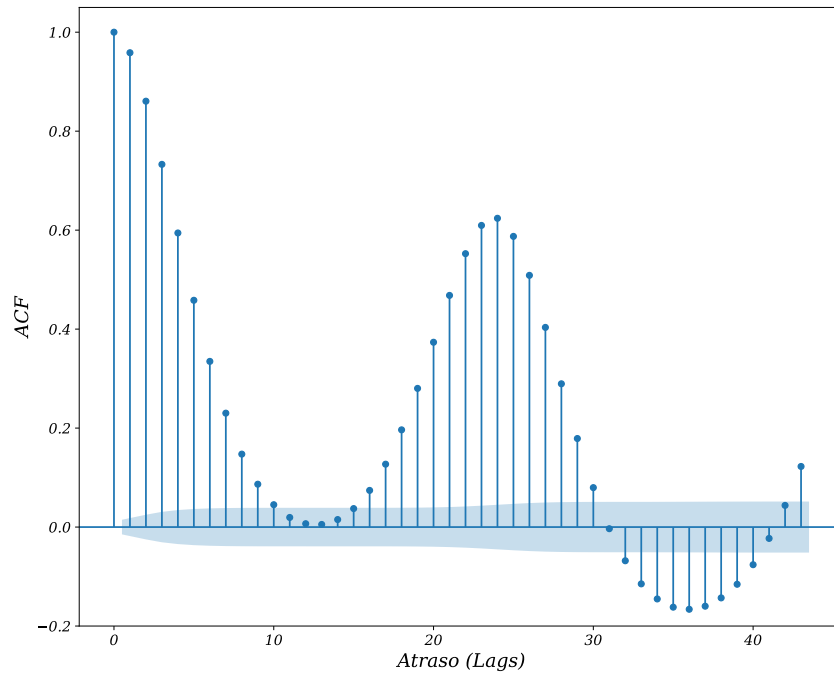
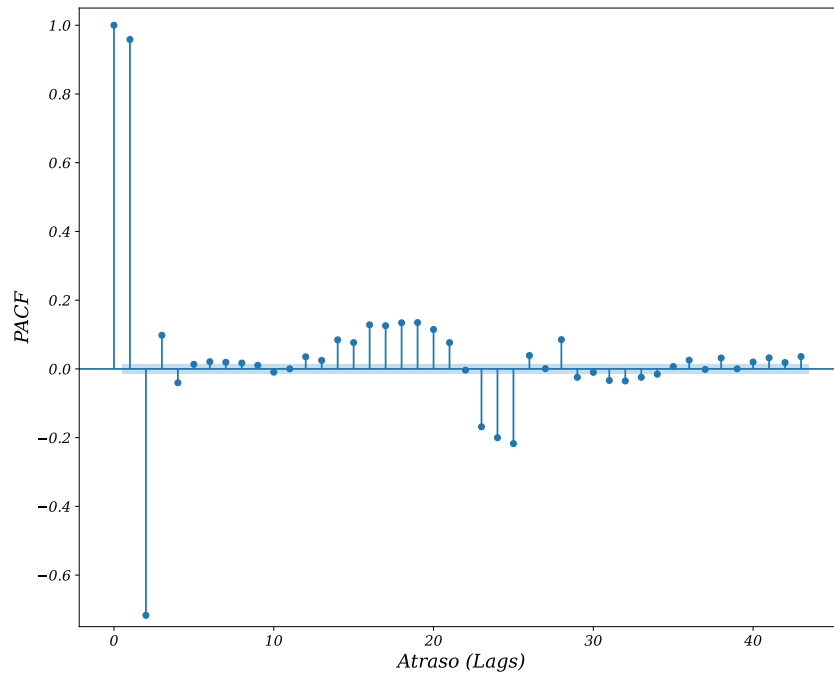


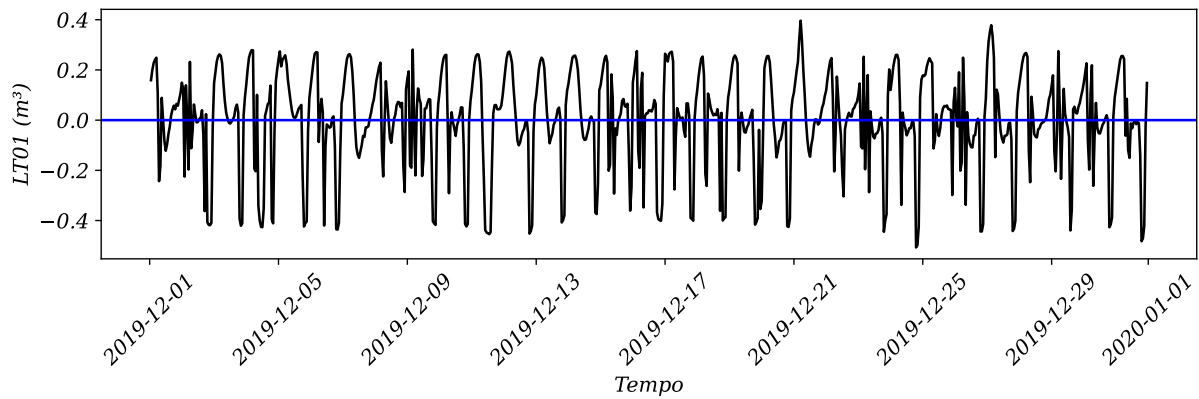
Figura 22: Autocorrelação parcial



estacionária e também apresenta ruído branco (do inglês *white noise*).

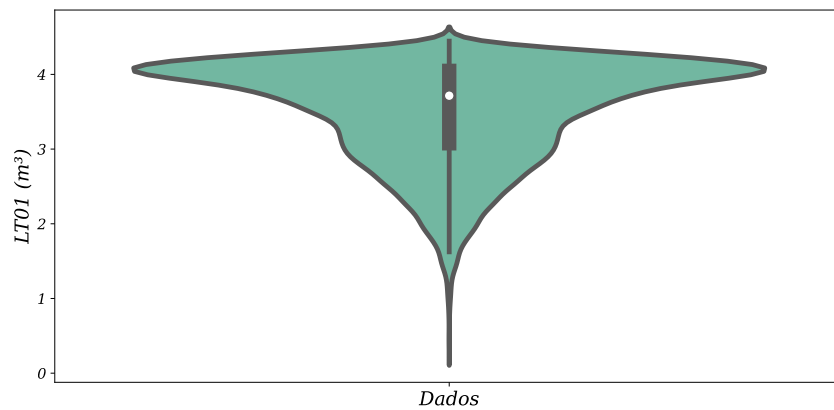
Com base na forte evidência contra a hipótese nula, podemos rejeitar a hipótese nula. A Figura 24, podemos notar um aumento na demanda durante essas horas durante o ano de 2019. Conforme mencionado na subseção 1.1.1, as anomalias climáticas ocorridas em 2020, especialmente a falta de chuvas e devido ao COVID-19, tiveram um impacto significativo nos resultados. Isso contribuiu para as mudanças observadas na demanda de

Figura 23: Ruído branco



água ao longo desse período.

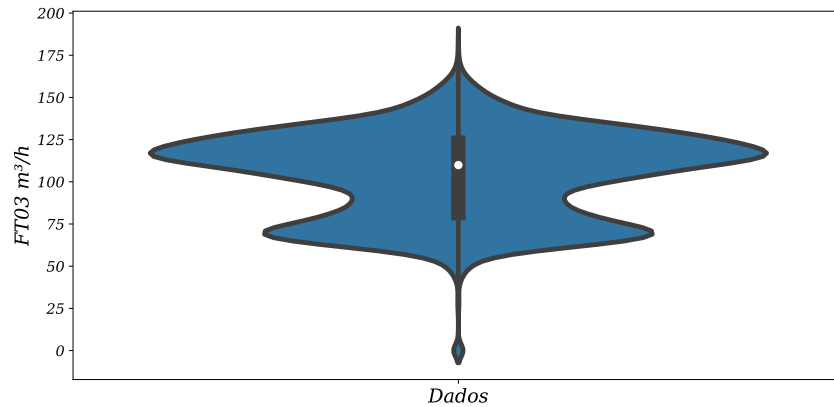
Figura 24: Violino no nível do reservatório



A Figura 25 mostra 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.

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

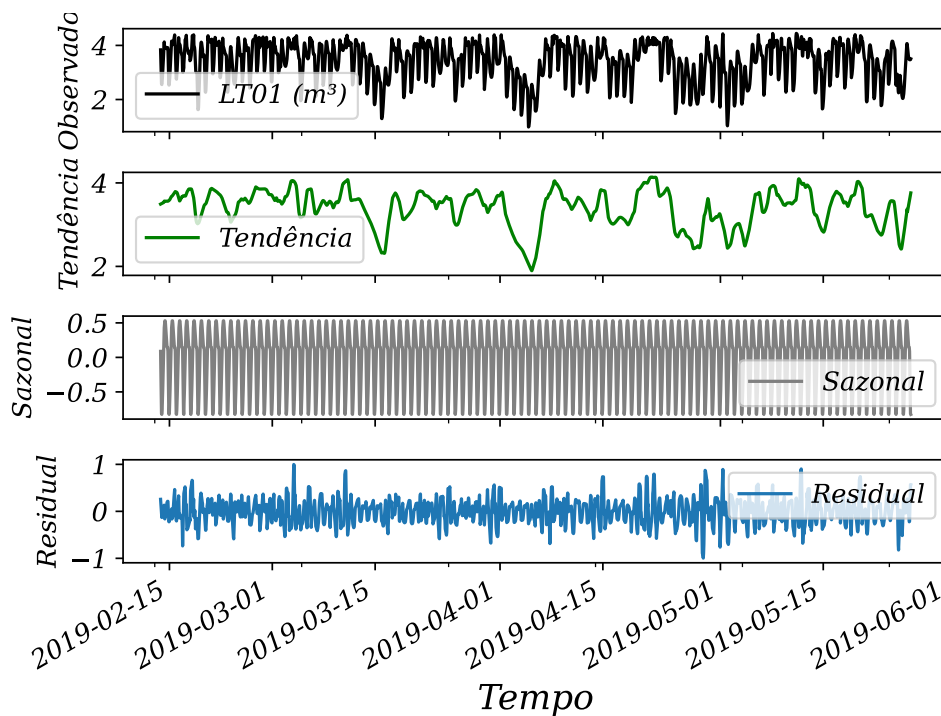
Figura 25: Violino da vazão de recalque



de várias variáveis para prever a variável de interesse.

Decomposição STL através da decomposição, é possível analisar se a série apresenta tendência, sazonalidade e resíduos. Ao observar a Figura 26, é evidente que os dados exibem ambos os padrões. Isso indica que a série é estacionária, como confirmado pelo seguinte teste ADF anterior.

Figura 26: Decomposição STL aditiva dos dados coletados



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 (RASCHKA, 2015; GÉRON, 2017).

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.

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 uma hora, seis horas, doze horas e um dia. 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 dia. 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 DTR, 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.

Validação e ajuste do modelo 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: uma hora, seis horas, doze horas e um dia. 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.

Foram utilizados os parâmetros obtidos pelo autoARIMA, que são $(p = 7, d = 0, q = 0)(P = 2, D = 1, Q = 1)_{M=12}$, mas foram ajustados para obter um melhor resultado, sendo $(p = 7, d = 1, q = 7)(P = 2, D = 1, Q = 1)_{M=12}$. Na Tabela 9, 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 10 mostra como a biblioteca do Python autoARIMA obteve os resultados dos parâmetros, exibindo o STD e os intervalos de confiança nos quais o modelo alcançou o melhor desempenho. O leve ajuste realizado não altera significativamente os parâmetros obtidos nesta biblioteca, permitindo que cada modelo seja trabalhado de maneira eficiente.

Tabela 9: Parâmetros utilizados nos modelos ARIMA e seus antecessores obtidos pelo “autoARIMA” do Python.

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

Tabela 10: SARIMAX(7, 0, 0) \times (2, 1, [1], 12) Results

	Coef	STD Err	z	P > z	[0,025	0,975]
Intercept	0,0003	0,000	1,053	0,292	0,000	0,001
ar.L1	1,6149	0,011	141,865	0,000	1,593	1,637
ar.L2	-0,8879	0,021	-42,045	0,000	-0,929	-0,847
ar.L3	0,3167	0,024	13,033	0,000	0,269	0,364
ar.L4	-0,1056	0,027	-3,961	0,000	-0,158	-0,053
ar.L5	-0,1099	0,028	-3,928	0,000	-0,165	-0,055
ar.L6	0,1431	0,027	5,368	0,000	0,091	0,195
ar.L7	-0,0673	0,015	-4,583	0,000	-0,096	-0,039
ar.S.L12	-0,1222	0,016	-7,705	0,000	-0,153	-0,091
ar.S.L24	0,1692	0,014	12,244	0,000	0,142	0,196
ma.S.L12	-0,8728	0,012	-74,569	0,000	-0,896	-0,850
sigma2	0,0157	0,000	60,022	0,000	0,015	0,016

Para os modelos de gradiente *boosting* e redes neurais artificiais, os hiperparâmetros foram otimizados usando a biblioteca Optuna do Python. Nesse contexto, são empregadas técnicas bayesianas, especificamente o algoritmo TPE, visando uma otimização mais eficiente.

Os modelos XGBoost e LightGBM tem como parâmetros e hiperparâmetros mostrado na Tabela 11 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.

Tabela 11: Hiperparâmetros dos modelos

Modelo	Estimadores	Profund. Máxima	Min. Amos- tras Divisão	Min. Amos- tras por Folha	Máx. Recursos	Taxa de Aprendi- zado
XGB Regressor	503	5	7	2	“sqrt”	0,034
LGBM Regressor	820	10	3	5	“auto”	0,014
Random Forest Regressor	135	10	4	2	None	N/A
Decision Tree Regressor	N/A	229	32	20	None	N/A

Os modelos de rede neural artificial, como RNN, ANN, CNN, GRU, LSTM e Transformer, obtidos na otimização do Optuna do Python, tiveram seus hiperparâmetros melhorados, conforme exibido na Tabela 12. Esses modelos, por serem modelos de rede neural artificial, são melhores para otimizar do que os outros.

Tabela 12: Resumo dos Hiperparâmetros dos Modelos de Redes Neurais

Modelo	Unidades/ Layers	Heads/ Dimen- sões	Tamanho do Batch	Épocas	Dropout/ Lear- ning Rate	Outros Parâmetros
LSTM	128	–	32	77	–	–
GRU	–	–	32	50	–	–
Transformers	–	8 heads, 217; 433	–	50	–	2 camadas
RNN	79	–	16	50	0,0008612	–
CNN	–	–	61	10	0,2799; 0,00052	Kernel: 7, Densas: 1, Verbosidade: 1
ANN	125	–	27	96	0,4135, 0,0004057	Densas: 1, Verbosidade: 0

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, conforme detalhado na seção 3.8.

Na análise dos modelos desenvolvidos, observou-se que o modelo DTR obteve o melhor desempenho, tanto para previsões de curto prazo, durante as horas de pico entre 18h e 21h, quanto para outros períodos. 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 âmbito das previsões de longo prazo, abrangendo casos de um dia, os modelos ARMA, AR, MA, ARIMA, ARIMAX, ARX, SARIMA, SARIMA, XGBRegressor, RFR, LGBMRegressor, DTR, RNN, ANN, CNN, GRU, LSTM, Prophet e Transformer foram avaliados. Uma observação recorrente foi a superioridade dos modelos que incorporam variáveis exógenas em termos de capacidade de previsão, evidenciada nas Figuras de 27 a 32 e nas Tabelas de 13 a 16, onde os valores menores foram destacados em **negrito** para facilitar a análise. O modelo RNN destacou-se tanto nos conjuntos de treinamento quanto na avaliação global, consolidando-se como o modelo mais eficaz nas previsões realizadas.

Cada figura, desde a 27 até a 32, ilustra cenários distintos de previsão e comparação entre modelos semelhantes. Os modelos Prophet e RNN, sendo este último a escolha superior, são apresentados de forma isolada. A decisão de não incluir o modelo LR na comparação baseou-se na constância observada em suas previsões a longo prazo.

Ao avaliar os modelos de previsão, tanto nas tabelas quanto nas imagens, o modelo RNN destaca-se como a opção mais eficaz. No caso específico da SANEPAR, esse modelo demonstra um desempenho superior em comparação com os demais modelos de previsão adotados. partir da etapa **Etap** 7, três métricas amplamente utilizadas na literatura foram empregadas para avaliar e comparar os modelos ARIMA e os modelos de regressão, conforme detalhado na seção 3.8.

Na análise dos modelos desenvolvidos, verificou-se que o modelo DTR alcançou o melhor desempenho, tanto para previsões de curto prazo, durante as horas de pico entre 18h e 21h, quanto para outros períodos. Adicionalmente, 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 das previsões de longo prazo, abrangendo períodos de um dia, os modelos ARMA, AR, MA, ARIMA, ARIMAX, ARX, SARIMA, SARIMA, XGBRegressor, RFR, LGBMRegressor, DTR, RNN, ANN, CNN, GRU, LSTM, Prophet e Transformer foram avaliados. Destacou-se a superioridade dos modelos que incorporam variáveis exógenas em termos de capacidade de previsão, evidenciada nas Figuras de 27 a 32 e nas Tabelas de 13 a 16, onde os valores menores foram destacados em **negrito** e *itálico* para

facilitar a análise. O modelo RNN destacou-se tanto nos conjuntos de treinamento quanto na avaliação global, consolidando-se como o modelo mais eficaz nas previsões realizadas.

Cada Figura, desde a 27 até a 32, ilustra cenários distintos de previsão e comparação entre modelos semelhantes. Os modelos Prophet e RNN, sendo este último a escolha superior, são apresentados de forma isolada. A decisão de não incluir o modelo LR na comparação baseou-se na constância observada em suas previsões a longo prazo.

Ao avaliar os modelos de previsão, tanto nas tabelas quanto nas imagens, o modelo RNN destaca-se como a opção mais eficaz. No caso específico da SANEPAR, esse modelo demonstra um desempenho superior em comparação com os demais modelos de previsão adotados.

Figura 27: Comparação dos modelos AR, ARX e MA

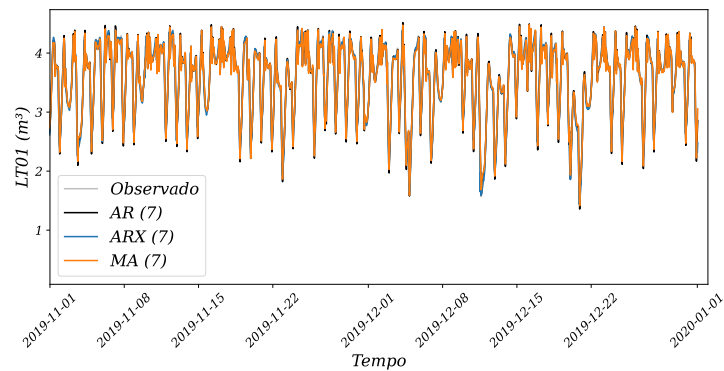


Figura 28: Comparação dos modelos ARIMAX, SARIMA e SARIMAX

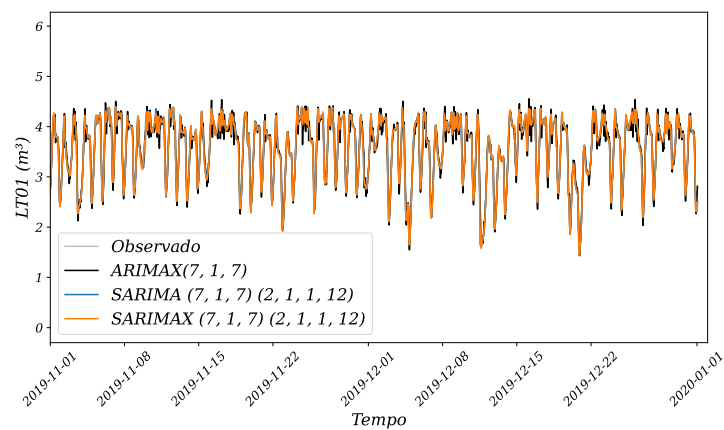


Figura 29: Comparação dos modelos ARMA e ARIMA

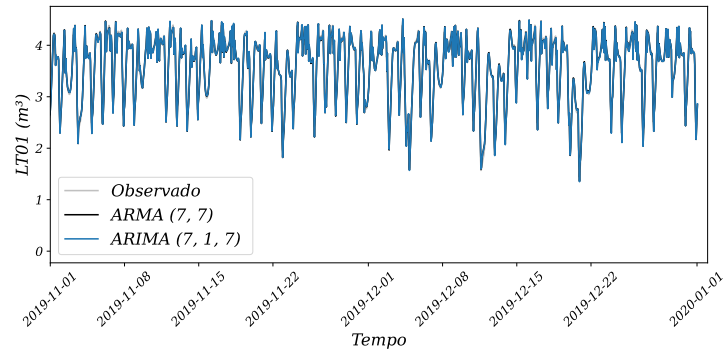


Figura 30: Comparação dos modelos DTR, RFR, XGBoost, Light GBM

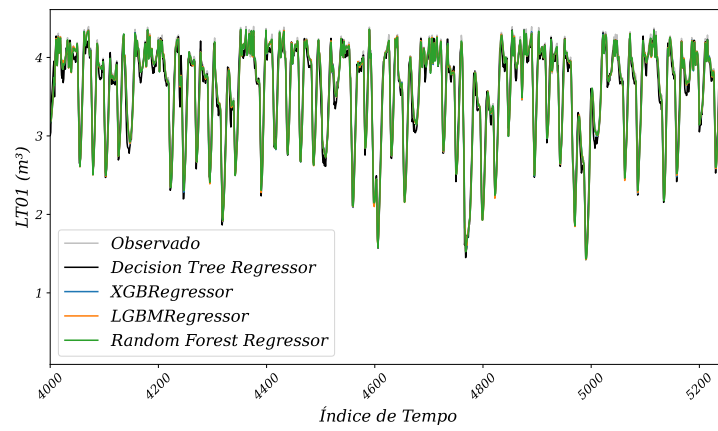


Figura 31: Modelo RNN e os vários horizontes

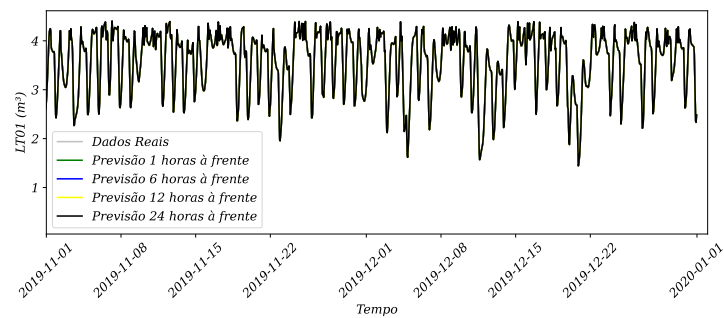


Figura 32: Previsões do modelo Prophet para o reservatório LT01

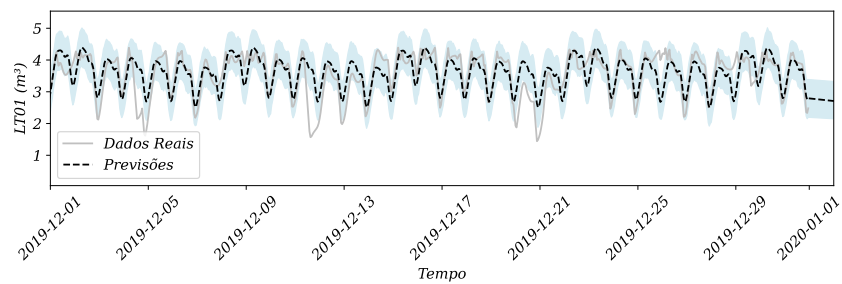


Tabela 13: Comparação dos modelos de previsão com as métricas de desempenho **treino**

		Modelos Treino																		
Horizontes	Métricas	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1 hora à frente	sMAPE	3,91	4,01	4,03	3,91	3,92	3,89	3,82	3,86	8,85	9,31	9,52	9,37	35,4	35,8	9	0,0665	16,8	23	23
	MAE	0,25	0,25	0,26	0,25	0,25	0,25	0,24	0,25	0,36	0,65	0,67	0,65	1,42	1,44	0,2	<i>0,0023</i>	0,55	0,83	0,83
	RRMSE	0,09	0,10	0,09	0,09	0,09	0,09	0,09	0,09	0,21	0,21	0,21	0,21	2,3	0,65	0,2	<i>0,0008</i>	0,31	0,48	0,48
6 horas à frente	sMAPE	9,97	10,1	9,7	9,98	9,97	10	10,1	9,99	6,99	12,4	12,7	9,369	66,2	83,9	20	<i>0,0230</i>	16,7	20,6	20,6
	MAE	0,64	0,65	0,62	0,64	0,64	0,64	0,65	0,64	0,59	0,9	0,93	0,651	3,37	4,95	0,6	<i>0,0007</i>	0,55	0,72	0,72
	RRMSE	0,23	0,23	0,23	0,23	0,23	0,23	0,23	0,23	0,16	0,32	0,33	0,209	5,02	1,71	0,6	<i>0,0006</i>	0,31	0,45	0,45
12 horas à frente	sMAPE	11,6	11,6	11,3	11,6	11,5	11,7	11,8	11,6	6,99	12,4	12,7	9,369	72	98,6	25	0,0683	16,8	29,2	29,2
	MAE	0,75	0,75	0,74	0,75	0,75	0,76	0,77	0,75	0,59	0,9	0,93	0,651	3,83	6,69	0,8	<i>0,0022</i>	0,55	1,11	1,11
	RRMSE	0,27	0,27	0,26	0,27	0,26	0,27	0,27	0,27	0,16	0,32	0,33	0,209	5,69	2,25	0,9	<i>0,0009</i>	0,31	0,55	0,55
24 horas à frente	sMAPE	6,77	6,85	6,67	6,77	6,69	6,82	6,86	6,82	6,99	12,4	12,7	9,369	74,4	104	26	0,2328	16,8	26,8	26,8
	MAE	0,43	0,44	0,43	0,43	0,43	0,44	0,44	0,43	0,59	0,9	0,93	0,651	4,04	7,5	0,8	<i>0,0079</i>	0,55	1	1
	RRMSE	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,16	0,32	0,33	0,209	5,99	2,5	1	<i>0,0024</i>	0,31	0,52	0,52

Legenda para os Modelos de Previsão: 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 14: 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 hora à frente	sMAPE	3,93	4,15	3,99	3,93	3,92	3,91	4,16	4,16	7,76	8,46	8,68	8,45	15,6	15,9	9	0,0744	15,1	20,6	20,6
	MAE	0,26	0,27	0,26	0,26	0,26	0,26	0,27	0,27	0,40	0,61	0,63	0,61	0,53	0,54	0,2	<i>0,0024</i>	0,52	0,76	0,76
	RRMSE	0,09	0,10	0,09	0,09	0,09	0,09	0,10	0,10	0,18	0,19	0,20	0,19	1,01	0,33	0,2	<i>0,0029</i>	0,34	0,5	0,5
6 horas à frente	sMAPE	9,74	9,94	9,44	9,74	9,71	9,76	9,96	9,96	6,36	10,7	11	8,446	59,5	72,7	20	<i>0,0308</i>	15,1	17,3	17,3
	MAE	0,65	0,66	0,63	0,65	0,65	0,65	0,66	0,66	0,56	0,8	0,82	0,609	2,97	4,04	0,6	<i>0,0007</i>	0,51	0,62	0,62
	RRMSE	0,23	0,23	0,22	0,23	0,23	0,23	0,23	0,23	0,14	0,28	0,29	0,191	4,9	1,42	0,6	<i>0,0033</i>	0,34	0,46	0,46
12 horas à frente	sMAPE	11,1	11,2	10,9	11,1	11,1	11,2	11,2	11,3	6,36	10,8	11	8,446	68,4	94,1	25	0,0745	15,1	18,8	18,8
	MAE	0,74	0,75	0,73	0,74	0,74	0,75	0,75	0,75	0,56	0,8	0,82	0,609	3,67	6,31	0,8	<i>0,0023</i>	0,52	0,68	0,68
	RRMSE	0,26	0,26	0,25	0,26	0,26	0,26	0,26	0,26	0,14	0,28	0,29	0,191	6,01	2,11	0,9	<i>0,0032</i>	0,34	0,48	0,48
24 horas à frente	sMAPE	6,15	6,34	6,08	6,15	6,14	6,24	6,36	6,37	6,36	10,7	11	8,446	71,5	102	26	0,2385	15,1	18,1	18,1
	MAE	0,4	0,41	0,4	0,4	0,4	0,41	0,42	0,42	0,56	0,8	0,83	0,609	3,92	7,36	0,8	<i>0,0081</i>	0,52	0,65	0,65
	RRMSE	0,16	0,16	0,16	0,16	0,16	0,16	0,16	0,16	0,16	0,14	0,28	0,29	0,191	6,42	2,43	1	<i>0,0041</i>	0,34	0,47

Legenda para os Modelos de Previsão: 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 15: 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 hora à frente	sMAPE	4,08	4,28	4,20	4,09	4,10	4,20	4,26	4,29	8,54	10,47	10,66	10,45	29,8	29,4	9	0,0675	17,4	18,3	18,3
	MAE	0,25	0,26	0,26	0,25	0,25	0,26	0,26	0,26	0,32	0,72	0,74	0,72	1,1	1,08	0,2	<i>0,0023</i>	0,56	0,6	0,6
	RRMSE	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,20	0,23	0,24	0,23	1,87	0,56	0,2	<i>0,0008</i>	0,33	0,39	0,39
6 horas à frente	sMAPE	10,9	11,1	10,6	10,9	10,9	11	11,1	11,1	6,8	13,9	14,2	10,45	67,9	84	20	<i>0,0229</i>	17,4	20,5	20,5
	MAE	0,68	0,69	0,66	0,68	0,68	0,69	0,69	0,69	0,57	1,01	1,04	0,721	3,39	4,81	0,6	<i>0,0007</i>	0,56	0,69	0,69
	RRMSE	0,25	0,25	0,24	0,25	0,25	0,25	0,25	0,25	0,16	0,36	0,37	0,233	4,98	1,72	0,6	<i>0,0005</i>	0,34	0,44	0,44
12 horas à frente	sMAPE	12,7	12,8	12,4	12,7	12,6	12,8	12,8	12,8	6,8	13,9	14,2	10,45	74,4	100	25	0,0689	17,4	22,9	22,9
	MAE	0,8	0,81	0,79	0,8	0,8	0,81	0,81	0,81	0,57	1,01	1,04	0,721	3,92	6,71	0,8	<i>0,0022</i>	0,56	0,79	0,79
	RRMSE	0,29	0,29	0,28	0,29	0,29	0,29	0,29	0,29	0,16	0,36	0,37	0,233	5,73	2,33	0,9	<i>0,0008</i>	0,33	0,48	0,48
24 horas à frente	sMAPE	7,3	7,45	7,19	7,3	7,27	7,37	7,43	7,46	6,8	13,9	14,2	10,45	76,9	106	26	0,2342	17,4	22,9	22,9
	MAE	0,46	0,46	0,45	0,46	0,45	0,46	0,46	0,46	0,57	1,01	1,04	0,721	4,14	7,59	0,8	<i>0,0077</i>	0,56	0,79	0,79
	RRMSE	0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,16	0,36	0,37	0,233	6,04	2,61	1	<i>0,0024</i>	0,33	0,48	0,48

Legenda para os Modelos de Previsão: 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 16: 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 hora à frente	sMAPE	3,94	4,08	4,05	3,93	3,95	3,91	4,05	4,05	8,51	9,22	9,43	9,244	17,1	17,4	9	0,0690	16,4	22,5	22,5
	MAE	0,25	0,26	0,26	0,25	0,25	0,25	0,26	0,26	0,36	0,65	0,67	0,648	0,57	0,58	0,2	<i>0,0023</i>	0,54	0,81	0,81
	RRMSE	0,09	0,1	0,09	0,09	0,09	0,09	0,1	0,1	0,2	0,21	0,21	0,207	1,01	0,31	0,2	<i>0,0017</i>	0,32	0,49	0,49
6 horas à frente	sMAPE	10	10,2	9,75	10	10	10,1	10,2	10,1	6,77	12,1	12,4	12,07	61,7	74,6	20	<i>0,0253</i>	16,3	20	20
	MAE	0,65	0,66	0,63	0,65	0,65	0,65	0,66	0,65	0,58	0,89	0,91	0,885	3,04	4,08	0,6	<i>0,0007</i>	0,54	0,7	0,7
	RRMSE	0,23	0,23	0,23	0,23	0,23	0,23	0,23	0,23	0,16	0,32	0,32	0,316	4,65	1,45	0,6	<i>0,0019</i>	0,33	0,46	0,46
12 horas à frente	sMAPE	11,6	11,7	11,4	11,6	11,6	11,7	11,8	11,7	6,77	12,1	12,4	12,12	70,7	96	25	0,0703	16,4	28,7	28,7
	MAE	0,76	0,76	0,74	0,76	0,75	0,76	0,77	0,76	0,58	0,89	0,91	0,889	3,75	6,38	0,8	<i>0,0023</i>	0,54	1,09	1,09
	RRMSE	0,27	0,27	0,26	0,27	0,27	0,27	0,27	0,27	0,16	0,32	0,32	0,317	5,69	2,16	0,9	<i>0,0019</i>	0,32	0,56	0,56
24 horas à frente	sMAPE	6,66	6,79	6,57	6,66	6,6	6,71	6,82	6,8	6,77	12,1	12,4	12,21	73,8	104	26	0,2347	16,4	26,2	26,2
	MAE	0,43	0,43	0,42	0,43	0,42	0,43	0,44	0,43	0,58	0,89	0,92	0,897	4,01	7,44	0,8	<i>0,0080</i>	0,54	0,98	0,98
	RRMSE	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,16	0,32	0,32	0,319	6,07	2,49	1	<i>0,0030</i>	0,32	0,53	0,53

Legenda para os Modelos de Previsão: 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.

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 17, exibe os valores de comparação múltipla de Nemenyi, onde as entradas evidenciam as diferenças significativas entre os pares de classificadores.

A Tabela 17 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 .

Tabela 17: Teste Nemenyi

Nemenyi	B1	B2	B3	LT01	FT01	FT02	FT03	PT01	PT02
B1	1.000	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
B2	0.001	1.000	0.001	0.001	0.001	0.001	0.001	0.001	0.001
B3	0.001	0.001	1.000	0.001	0.001	0.001	0.001	0.001	0.001
LT01	0.001	0.001	0.001	1.000	0.001	0.001	0.001	0.001	0.001
FT01	0.001	0.001	0.001	0.001	1.000	0.001	0.131	0.001	0.001
FT02	0.001	0.001	0.001	0.001	0.001	1.000	0.001	0.001	0.001
FT03	0.001	0.001	0.001	0.001	0.131	0.001	1.000	0.001	0.001
PT01	0.001	0.001	0.001	0.001	0.001	0.001	0.001	1.000	0.001
PT02	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	1.000

No contexto do estudo, os resultados da análise comparativa revelaram diferenças 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 17 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 18 é mostrado como cada modelo foi comparado entre si em 24 passos à frente.

Tabela 18: Métricas de avaliação dos modelos

Modelo	sMAPE	MAE	RRMSE
Prophet	25,67	0,844	0,975
Transformer	16,39	0,544	0,324
ANN	26,22	0,980	0,531
CNN	26,22	0,980	0,531
RNN	0,235	0,008	0,003
LSTM	73,75	4,010	6,068
GRU	103,57	7,443	2,485
AR	6,66	0,428	0,169
ARX	6,79	0,434	0,173
MA	6,57	0,423	0,166
ARMA	6,66	0,428	0,169
ARIMA	6,60	0,424	0,167
SARIMA	6,71	0,432	0,170
ARIMAX	6,82	0,436	0,173
SARIMAX	6,80	0,435	0,173
DTR	6,77	0,577	0,158
RFR	12,09	0,886	0,316
XGBRegressor	12,41	0,916	0,323
LGBMRegressor	12,21	0,897	0,319

Modelo com menor valor em cada métrica: Primeiramente, os diversos modelos de previsão de séries temporais foram avaliados para um horizonte de previsão de um dia. 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 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 um dia, utilizando os dados completos.

4.1.1 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 e de barra. Dessa forma, pôde-se observar qual dos modelos apresentava o melhor desempenho.

Ao examinar os modelos representados nas Figuras 33 e 34, identifiquei os modelos que se destacam em relação à natureza dos dados. Na Figura 36, 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 de bayesiana usando o método TPE.

Na Figura 35, 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.

Na Figura 34, 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.

Na Figura 35, 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 33: Comparação dos modelos ARIMA

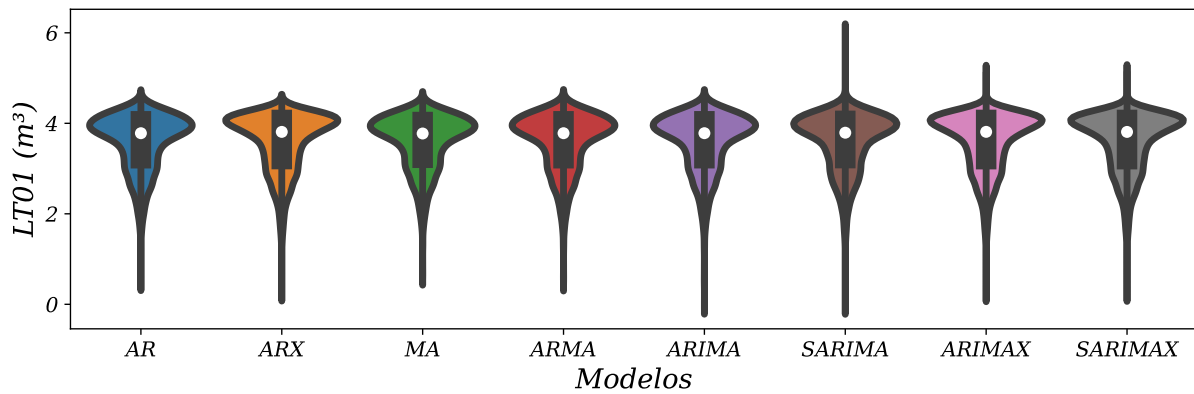


Figura 34: Comparação de modelos de regressão

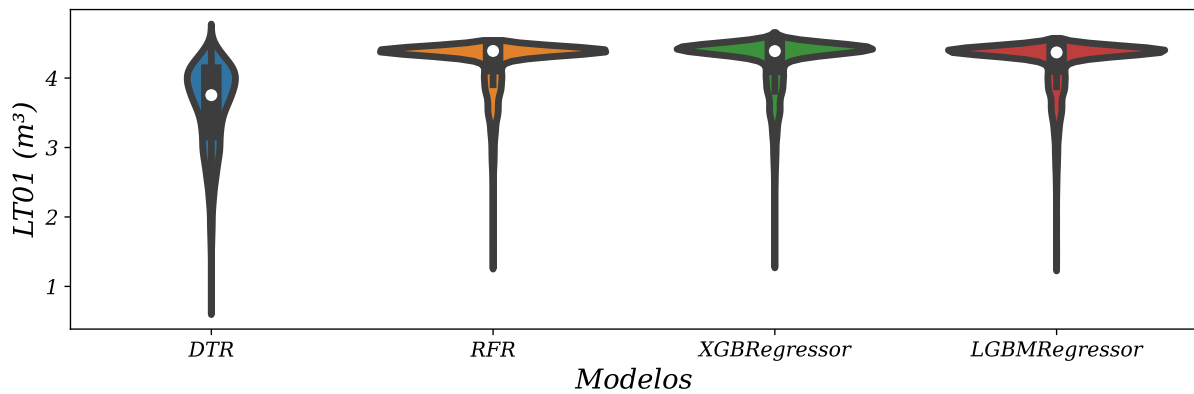
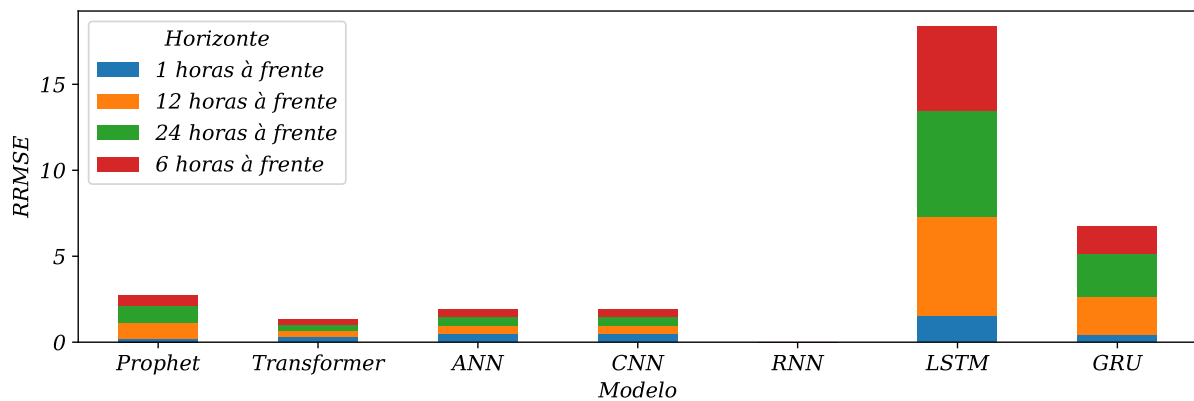
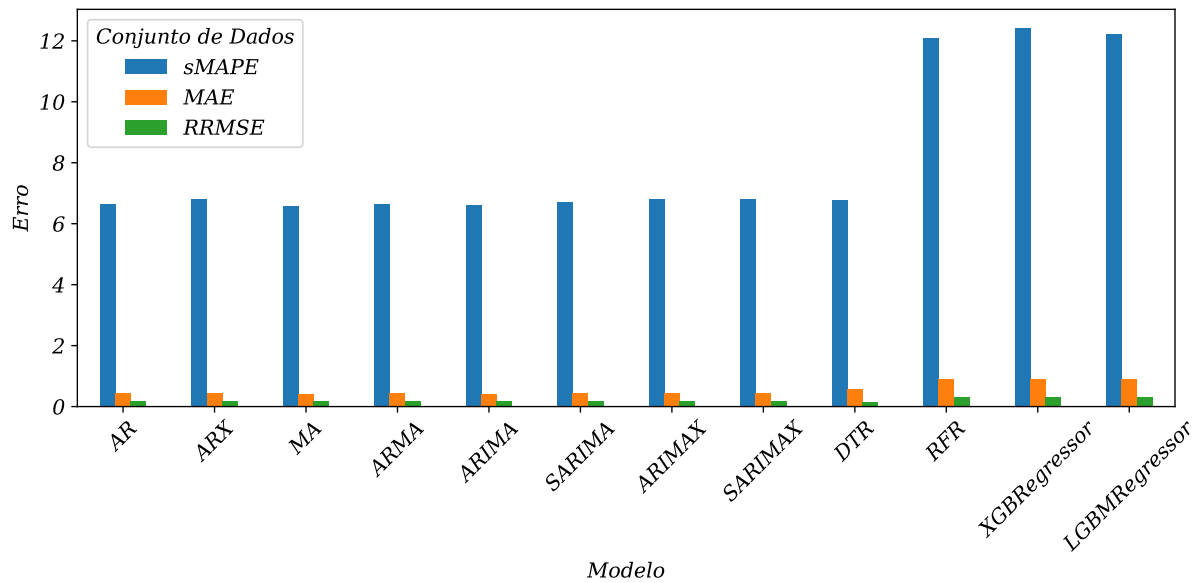


Figura 35: Comparação dos modelos na métrica RRMSE



A avaliação da eficácia dos modelos ARIMA em previsões de longo prazo emprega o teste de Ljung-Box, conforme detalhado nas Tabelas 19a a 19d ilustram a acurácia dos modelos ARIMA ao longo do tempo, com valores menores sendo destacados em **negrito** 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

Figura 36: Comparação dos modelos nas métricas sMAPE, MAE e RRMSE



temporais mais longos, diferenciando-se positivamente dos outros modelos ARIMA. Na Figura 33, 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.

Tabela 19: 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
ARX	59,677	0,000	ARX	47,177	0,000
AR	52,312	0,265	AR	49,965	0,444
MA	57,268	0,000	MA	77,884	0,000
ARMA	6,945	0,731	ARMA	1,545	0,999
ARIMA	16,724	0,081	ARIMA	5,354	0,866
SARIMA	48,505	0,000	SARIMA	24,663	0,006
ARIMAX	89,931	0,000	ARIMAX	36,738	0,000
SARIMAX	29,093	0,000	SARIMAX	21,236	0,020
(c) Validação			(d) Inteiro		
Ljung Box	Estatística de Teste	Valor De p	Ljung Box	Estatística de Teste	Valor De p
ARX	5,108	0,884	ARX	48,870	0,000
AR	4,360	0,930	AR	49,432	0,035
MA	46,252	0,000	MA	57,629	0,000
ARMA	7,515	0,676	ARMA	10,053	0,436
ARIMA	7,738	0,654	ARIMA	10,053	0,436
SARIMA	28,998	0,001	SARIMA	10,053	0,436
ARIMAX	6,115	0,000	ARIMAX	70,458	0,000
SARIMAX	4,443	0,925	SARIMAX	2,897	0,000

4.2 Aplicação

A previsão da demanda d'água é uma preocupação fundamental para muitas organizações e autoridades responsáveis pelo abastecimento de água. Neste estudo de caso, explorou-se como a análise de séries temporais pode ser aplicada para prever a demanda d'água ao longo do tempo.

4.2.1 Estudo de Caso 1

Confirmou-se que a ativação das bombas de sucção durante o período de 18h às 21h resulta em um maior custo energético para a SANEPAR. Portanto, é recomendado evitar o acionamento das bombas durante esse período, utilizando estratégias de armazenamento e gerenciamento eficientes.

Com base nos resultados obtidos, conclui-se que as pressões atuais das variáveis **PRESSÃO DE SUCÇÃO - PT01** e **PRESSÃO DE RECALQUE - PT02** são adequadas para atender à demanda diária. O percentil 10 das pressões de sucção (3,48 mca) indica que apenas 10% dos valores estão abaixo desse limite, o que sugere que a pressão de sucção geralmente se mantém em níveis adequados para o funcionamento adequado do sistema. Da mesma forma, o percentil 90 das pressões de recalque (24,02 mca) indica que apenas 10% dos valores estão acima desse limite, evidenciando que a pressão de recalque também se mantém dentro dos padrões necessários para atender à demanda diária.

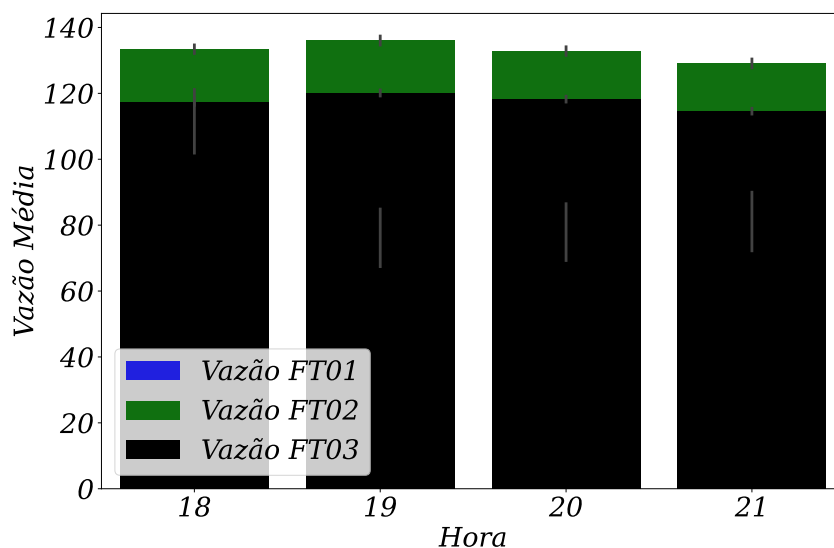
Com base na frequência de funcionamento das bombas e na demanda durante o horário de pico, determinou-se que é necessário manter um volume máximo d'água no reservatório, correspondente a 5285,90 litros, para evitar o acionamento das bombas nesse período.

4.2.2 Estudo de Caso 2

Ao analisar os dados dos últimos 3 anos do Bairro Alto, identificou-se a presença de tendências sazonais e padrões de consumo de água. Essas informações são valiosas para compreender os padrões de demanda e planejar o abastecimento de forma eficiente.

O gráfico de barras apresentado na Figura 37 mostra a demanda média das variáveis de fluxo (Vazão de Entrada – FT01, Vazão de Gravidade – FT02 e Vazão de Recalque – FT03) durante o intervalo das 18h às 21h. Cada barra representa a média da demanda para cada variável em um horário específico dentro desse intervalo. A altura de cada barra indica a magnitude da demanda média para a respectiva variável. Essa visualização permite que sejam identificados os horários em que as variáveis de fluxo apresentaram maior demanda, o que é útil para o planejamento e gerenciamento adequado do sistema.

Figura 37: Demanda média das variáveis de fluxo



A Tabela 20 apresenta os resultados para as três variáveis estudadas: vazão de entrada – FT01, vazão de gravidade – FT02 e vazão de recalque – FT03. Os resultados destacam os horários específicos em que cada variável apresentou maior demanda dentro do intervalo das 18h às 21h, fornecendo importantes para o planejamento e gerenciamento adequado do sistema. A Tabela 20 resume essas informações.

Tabela 20: Demanda de água

Vazões	Horário de Maior Demanda	Demanda
Entrada – FT01	2020/10/08 21:00:00	383,87 m^3/h
Gravidade – FT02	2020/10/20 18:00:00	326,17 m^3/h
Recalque – FT03	2020/11/26 19:00:00	194,35 m^3/h

Durante as horas de pico, é necessário que o nível do reservatório esteja mantido dentro na média de $3.9005 m^3$ para evitar o acionamento das bombas. Manter o nível do reservatório dentro dessa faixa permitirá que o sistema opere de forma eficiente, atendendo à demanda de água sem a necessidade de acionar as bombas.

É importante destacar que a vazão de recalque exerce um impacto significativo no nível do reservatório em comparação com as outras vazões. Essa diferença se deve ao fato de que a vazão de recalque está diretamente relacionada à injeção de água no reservatório por meio da bomba localizada próxima à sua base. Em contraste, as demais vazões possuem alguns valores ausentes, o que limita sua influência na análise geral do sistema.

5 Conclusões

Na dissertação realizada, foi conduzido um estudo abrangente sobre a previsão da demanda d'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. Ao longo do estudo, foram empregados os modelos ARIMA, DTR e XGBoost para validar o estudo de caso da SANEPAR.

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. 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 d'água. Nesse sentido, foram comparados 19 tipos diferentes de modelos.

Com base nos resultados obtidos, conclui-se que a abordagem de séries temporais é uma ferramenta eficaz para prever a demanda futura d'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 progressos obtidos nesta pesquisa, é crucial destacar algumas limitações a serem consideradas. Primeiramente, a análise fundamentou-se em dados históricos de demanda d'água de uma única região, especificamente o maior bairro de Curitiba. O estudo não considerou fatores externos, como mudanças climáticas ou eventos imprevistos, que poderiam impactar a demanda d'água.

5.1 Propostas Futuras

Apesar dos resultados promissores evidenciados por esta pesquisa, é essencial que se reconheçam suas limitações e que se instigue a exploração de novos horizontes em pesquisas subsequentes. Uma análise mais profunda e abrangente pode ser realizada, investigando modelos de redes neurais mais avançados. Além disso, a implementação de técnicas de otimização matemática mais refinadas, como o uso do método CMAES (do inglês *Covariance Matrix Adaptation Evolution Strategy*), pode ser considerada. Seria prudente incluir cuidadosamente variáveis exógenas em todos os modelos pertinentes, como o uso de variáveis climáticas e dados de precipitação do tempo. Implementa modelos que utilizam lógica *fuzzy* para aprimorar a previsão do tanque. Usa essa previsão juntamente com modelos existentes na literatura, como a otimização BOA, que não foi abordada neste contexto.

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.
- 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.
- 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.
- 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.
- 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.

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>>.

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.
- GÉRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow**. 2nd. ed. O'Reilly Media, 2017. Disponível em: <<https://www.oreilly.com/library/view/hands-on-machine-learning/9781491962282/>>.
- 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.
- 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.

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.

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>>.

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.

RASCHKA, S. A practical guide to machine learning in python. **Machine Learning with Python**, 2015. Disponível em: <https://sebastianraschka.com/pdf/books/machine_learning_with_python/mlwp.pdf>.

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>>.

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, p. 1421–1441, 2019. ISSN 08856125.

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.

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.

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%2FTNNLS.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-cacaf61ae863>>.

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.

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.

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.