

Previsão de Séries Temporais: Temperatura de Porto Alegre

1st Guilherme José da Silva

Centro de Informática - UFPE

Recife, Brazil

gjs@cin.ufpe.br

2nd Luis Eduardo Ribeiro Freitas

Centro de Informática - UFPE

Recife, Brazil

lerf@cin.ufpe.br

3rd Renan Santana Costa

Centro de Informática - UFPE

Recife, Brazil

rsc8@cin.ufpe.br

Abstract—Este trabalho aborda o problema de previsão de temperatura de curto prazo a partir de dados meteorológicos históricos do Instituto Nacional de Meteorologia (INMET), utilizando registros dos anos de 2010 a 2012. O objetivo é prever a temperatura máxima da próxima observação em uma série temporal. Foram realizadas análises exploratórias e testes estatísticos, incluindo ADF, KPSS, ACF e PACF, para caracterização das propriedades da série.

Diferentes modelos foram avaliados, contemplando abordagens estatísticas, de aprendizado de máquina e de aprendizado profundo, especificamente SARIMAX, Random Forest, Gradient Boosting e LSTM. Os dados foram normalizados com StandardScaler e divididos temporalmente em conjuntos de treino, validação e teste. A avaliação foi conduzida por meio das métricas MAE, RMSE e MAPE.

Os resultados indicam bom desempenho na previsão one-step à frente, com destaque para os modelos LSTM e SARIMAX. Além disso, foi explorada uma abordagem baseada em intervalos de confiança como tentativa de previsão de múltiplos passos à frente, cuja implementação completa foi limitada por aspectos de complexidade e organização da série, apontando direções para trabalhos futuros.

I. INTRODUÇÃO

A previsão de temperatura é um problema clássico e de grande importância tanto na meteorologia quanto em aplicações práticas de ciência de dados e aprendizado de máquina. Neste trabalho, investigamos a capacidade de modelos preditivos em estimar a temperatura da próxima observação a partir de séries temporais de dados meteorológicos históricos. Utilizamos como base os conjuntos de dados disponibilizados pelo **Instituto Nacional de Meteorologia (INMET)** referentes aos anos de 2010, 2011 e 2012, que contêm informações observacionais como temperatura, umidade, precipitação e outras variáveis climáticas.

A relevância prática desse problema reside na sua ampla aplicabilidade: previsões precisas de temperatura são fundamentais para planejamento agrícola, gestão de recursos hídricos, energia, saúde pública e outras atividades econômicas que dependem de condições climáticas.

II. ANÁLISE DE DADOS E FEATURE ENGINEERING

A. Análise Exploratória dos Dados

Os dados utilizados para a realização deste trabalho são dados obtidos a partir da base de dados do **INMET** que continha registros dos dados em intervalos de uma hora. Foram

considerados os anos de 2010, 2011 e 2012, que posteriormente foram concatenados em um único conjunto de dados. As variáveis disponíveis incluem informações de precipitação, pressão, radiação, temperatura e vento, coletadas de forma horária. O objetivo desta etapa é compreender a estrutura dos dados, identificar possíveis problemas (possíveis anomalias, encontrar valores faltantes) e extrair padrões relevantes para a etapa de modelagem preditiva.

1) *Análise Exploratória Estrutural*: Durante a inspeção estrutural, foi observado que diversas colunas numéricas estavam armazenadas como strings, em especial devido ao uso de vírgula como separador decimal. Essas colunas foram convertidas para o tipo numérico adequado. Além disso, foi identificada uma coluna sem nome (Unnamed: 19), que não continha informações válidas e foi removida.

As colunas de data e hora foram combinadas para formar um índice do tipo `DatetimeIndex`, facilitando a análise temporal e o uso posterior em modelos de séries temporais.

As tabelas I e II ilustram informações do dataset e estatísticas descritivas sobre o atributo alvo.

TABLE I
LISTAGEM DE ATRIBUTOS

Nome	Tipo
DATA (YYYY-MM-DD)	object
HORA (UTC)	object
PRECIPITAÇÃO TOTAL, HORÁRIO (mm)	object
PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO, HORARIA (mB)	object
PRESSÃO ATMOSFERICA MAX. NA HORA ANT. (AUT) (mB)	object
PRESSÃO ATMOSFERICA MIN. NA HORA ANT. (AUT) (mB)	object
RADIACAO GLOBAL (kJ/m ²)	int64
TEMPERATURA DO AR - BULBO SECO, HORARIA (°C)	object
TEMPERATURA DO PONTO DE ORVALHO (°C)	object
TEMPERATURA MÁXIMA NA HORA ANT. (AUT) (°C)	object
TEMPERATURA MÍNIMA NA HORA ANT. (AUT) (°C)	object
TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT) (°C)	object
TEMPERATURA ORVALHO MIN. NA HORA ANT. (AUT) (°C)	object
UMIDADE REL. MAX. NA HORA ANT. (AUT) (%)	int64
UMIDADE REL. MIN. NA HORA ANT. (AUT) (%)	int64
UMIDADE RELATIVA DO AR, HORARIA (%)	int64
VENTO, DIREÇÃO HORARIA (°)	int64
VENTO, RAJADA MAXIMA (m/s)	object
VENTO, VELOCIDADE HORARIA (m/s)	object
Unnamed: 19	float64

2) *Análise de Valores Faltantes e Outliers*: A base original apresentou uma quantidade significativa de valores que

aparentam ser outliers, representados principalmente pelo valor -9999, por ser um valor incoerente com a realidade, interpretamos esse valor como uma falha no sensor, dessa forma são considerados dados faltantes.

A análise de Outliers foi feita por meio da observação de gráficos boxplot para todas as variáveis numéricas. Por meio da análise desses gráficos, fomos capazes de identificar Outliers presentes em atributos como precipitação, vento, pressão atmosférica e temperatura máxima. Ainda que fossem valores extremos, não é necessário realizar o tratamento, uma vez que é esperado encontrar discrepâncias em dados meteorológicos, podendo até mesmo representar eventos importantes.

Na Fig. 1 temos o boxplot dos valores do atributo alvo temperatura máxima.

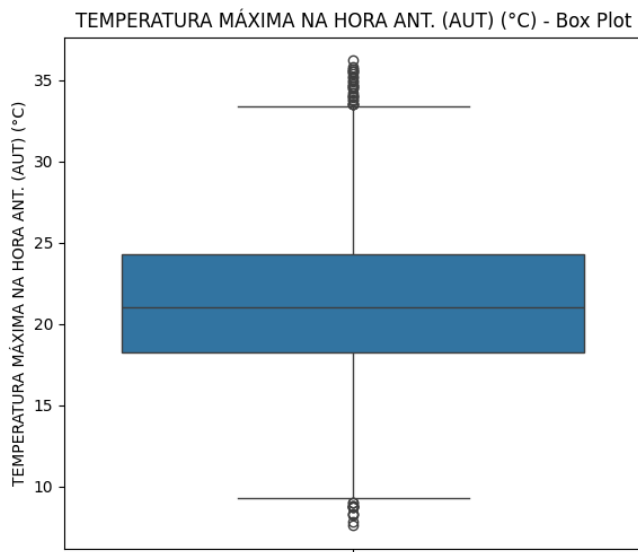


Fig. 1. Boxplot da Temperatura Máxima.

3) *Análise Univariada:* A análise univariada concentrou-se na avaliação estatística individual dos atributos, com ênfase na variável alvo: **temperatura máxima**.

Foram analisadas medidas descritivas como média, mediana, desvio padrão, valores mínimos e máximos, além da distribuição dos dados ao longo do tempo.

TABLE II
ESTATÍSTICAS DESCRITIVAS DA TEMPERATURA MÁXIMA

Ano	Média	Desvio Padrão	Mín.	Máx.
2010	20.0431	5.7169	3.8	38.5

4) *Análise das Propriedades da Série Temporal:* Com o objetivo de avaliar formalmente a estacionariedade da variável alvo, foi aplicado o teste de **Dickey-Fuller Aumentado (ADF)** sobre a série de temperatura máxima no conjunto de treino. O teste resultou em um valor estatístico de -6.10 e um p -value de aproximadamente 1×10^{-8} , inferior ao nível de significância de 5%. Dessa forma, rejeita-se a hipótese nula de presença de raiz unitária, indicando evidência estatística de estacionariedade da série no período analisado.

Adicionalmente, foi aplicado o teste **Kwiatkowski-Phillips-Schmidt-Shin (KPSS)**, cuja hipótese nula assume que a série é estacionária. O teste apresentou estatística de 4.44 e p -value de 0.01, levando à rejeição da hipótese nula ao nível de 5%. Esse resultado sugere que a série apresenta componentes não estacionários, possivelmente associados a tendência e/ou sazonalidade.

A aplicação conjunta dos testes ADF e KPSS indica que, embora a série não apresente raiz unitária, ela não pode ser considerada estritamente estacionária em todos os seus componentes. Esse comportamento é comum em séries meteorológicas, que frequentemente apresentam estacionariedade em termos de variância, mas mantêm padrões sazonais persistentes.

Complementarmente aos testes estatísticos, foi realizada a **análise das funções de autocorrelação (ACF)** e **autocorrelação parcial (PACF)** da série de temperatura máxima, com o objetivo de investigar sua estrutura de dependência temporal. Os gráficos evidenciam forte autocorrelação nas primeiras defasagens e padrões periódicos consistentes, reforçando a presença de sazonalidade e dependência temporal de curto prazo.

O gráfico de ACF apresenta autocorrelações elevadas mesmo para defasagens mais longas, com um decaimento lento e comportamento oscilatório. Esse padrão é característico de séries temporais com forte dependência temporal e presença de sazonalidade, indicando que a série não é estacionária em nível.

Por sua vez, a análise da PACF evidencia picos significativos nas primeiras defasagens, sugerindo que a temperatura atual depende fortemente de um número reduzido de observações anteriores. Além disso, observa-se a persistência de efeitos associados a ciclos sazonais, reforçando a existência de padrões periódicos na série.

Analisando ambos os resultados é possível observar incongruência entre os resultados do teste de hipótese e ACF, isso acontece devido a forte sazonalidade presente na série. Retirando a sazonalidade da série é possível observar que a série possui sim estacionariedade.

Esses resultados indicam que a série de temperatura máxima é estacionária e possui forte sazonalidade, características esperadas em dados meteorológicos.

Para observar melhor a forma como a tendência, sazonalidade e ruído estão presentes na série temporal foi utilizado o método **Seasonal and Trend decomposition using LOESS (STL)** que nos permite decompor a série temporal em tendência, sazonalidade e ruído.

Nas Figs. 2 3, 4 e 5 são mostrados os resultados dos testes de ACF/PACF e gráficos de tendência, sazonalidade e ruído.

B. Pré-processamento dos dados

1) *Tratamento de Valores Faltantes:* Os valores faltantes encontrados foram corretamente tratados e convertidos para **NaN (Not a Number)**.

A análise de valores faltantes revelou que algumas variáveis, como radiação global, possuíam uma proporção muito elevada

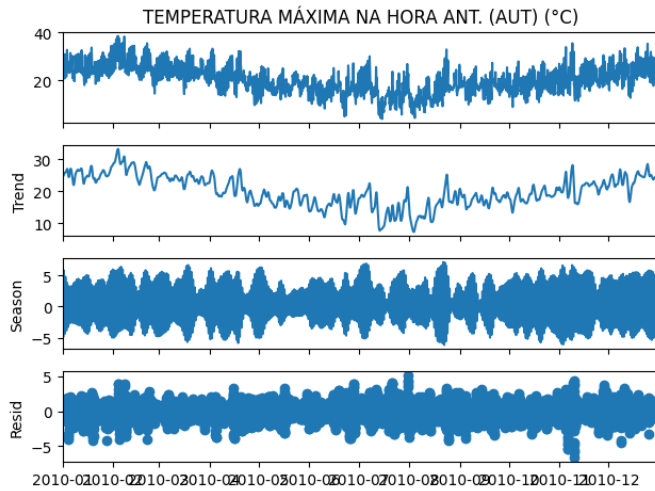


Fig. 2. STL da série temporal

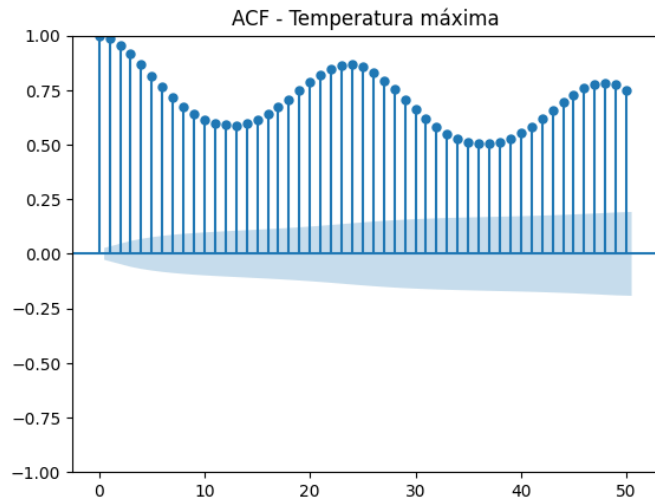


Fig. 3. ACF da temperatura máxima com sazonalidade.

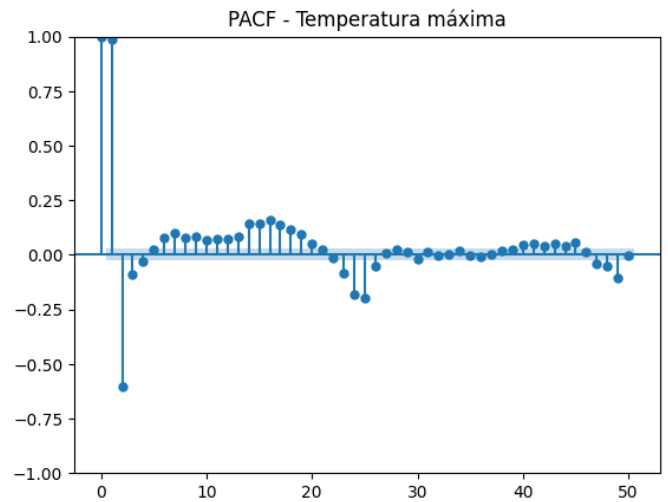


Fig. 4. PACF da temperatura máxima.

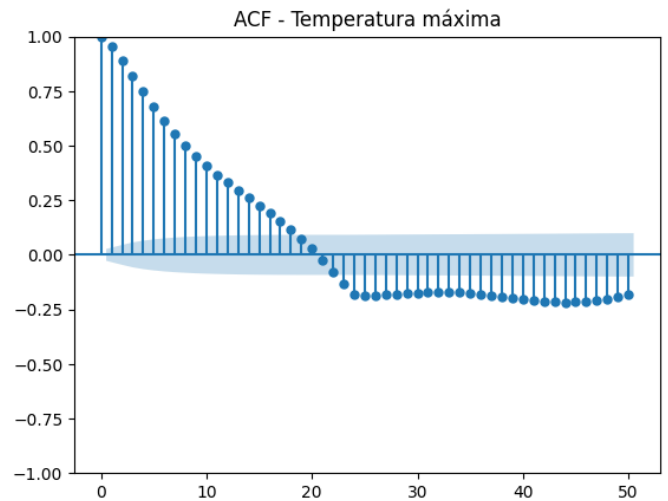


Fig. 5. ACF da temperatura máxima sem sazonalidade.

de dados ausentes, o que motivou sua remoção do conjunto de dados para evitar impactos negativos na modelagem.

Para as demais variáveis, foi aplicado o método de *forward fill* (*ffill*), uma estratégia adequada ao contexto de séries temporais, assumindo que a variação entre observações consecutivas é gradual.

2) *Tratamento de Outliers*: Não foi realizado nenhum tratamento para valores extremos, visto que os Outliers encontrados em variáveis como precipitação, vento e pressão atmosférica são Outliers esperados devido à natureza de dados meteorológicos.

3) *Deteção e Tratamento de Duplicatas*: A verificação de observações duplicadas foi realizada considerando o índice temporal da base de dados. Como cada registro corresponde a uma observação horária única, indexada por data e hora, não foram identificadas entradas duplicadas no conjunto de dados. Dessa forma, não foi necessária a aplicação de técnicas adicionais de remoção ou agregação de duplicatas.

4) *Feature Scaling*: Com o objetivo de garantir que todas as variáveis numéricas estivessem na mesma escala, foi aplicada a técnica de padronização utilizando o *StandardScaler*. Essa abordagem transforma os dados para que apresentem média zero e desvio padrão unitário.

5) *Encoding de Variáveis Categóricas*: A base de dados utilizada é composta exclusivamente por variáveis numéricas e temporais, por isso não foi necessário a utilização de nenhuma técnica de encoding de variáveis categóricas.

C. Divisão dos Dados

A divisão dos dados foi realizada de forma temporal, respeitando a ordem cronológica da série, a fim de evitar vazamento de informação (data leakage). Os dados foram separados da seguinte forma:

- Treino: dados referentes ao ano de 2010
- Validação: dados referentes ao ano de 2011

- Teste: dados referentes ao ano de 2012

Essa estratégia reflete um cenário realista de previsão, no qual o modelo é treinado com dados passados e avaliado em períodos futuros. Além disso, a separação por anos distintos permite avaliar a capacidade de generalização do modelo diante de possíveis variações sazonais e mudanças nas condições climáticas ao longo do tempo.

D. Feature Engineering

A etapa de feature engineering concentrou-se na criação e seleção de atributos relevantes para a tarefa de previsão da temperatura da próxima observação. Considerando a natureza temporal do problema, foram criadas **variáveis defasadas (lag features) da temperatura**, permitindo que os modelos capturassem dependências temporais e padrões históricos.

III. MODELAGEM

A. SARIMAX

1) *Conceitos Básicos*: O modelo **SARIMAX** (*Seasonal AutoRegressive Integrated Moving Average with eXogenous variables*) é uma extensão do modelo **ARIMA** tradicional, projetada para lidar com séries temporais que apresentam padrões sazonais bem definidos mais variáveis exógenas. Ele incorpora termos autorregressivos (AR), de média móvel (MA) e de diferenciação (I), tanto na componente não sazonal quanto na componente sazonal, permitindo capturar dependências temporais de curto e longo prazo.

A sazonalidade é representada por um período fixo s , que define a frequência com que o padrão se repete. No contexto de séries temporais de temperatura horária, a sazonalidade diária ($s = 24$) é particularmente relevante, já que após a diferenciação com esse valor a série possui a propriedade de estacionariedade. O modelo pode ser representado de forma geral por:

$$SARIMAX(p, d, q) \times (P, D, Q)_s$$

onde (p, d, q) são os parâmetros não sazonais e (P, D, Q) os parâmetros sazonais.

2) *Justificativa*: Durante a análise exploratória e os testes de estacionariedade realizados na série temporal de temperatura, observou-se a presença de uma forte componente sazonal associada ao ciclo diário, característica comum em dados meteorológicos horários. Essa sazonalidade inviabiliza o uso de modelos ARIMA simples, que não são capazes de capturar adequadamente padrões periódicos.

Dessa forma, o modelo SARIMAX foi escolhido por permitir a modelagem explícita da sazonalidade, além de capturar dependências temporais nos resíduos e permitir o uso de variáveis exógenas para aprimorar o modelo. Para garantir uma avaliação adequada do desempenho do modelo ao longo do tempo, foi adotada uma divisão temporal dos dados, utilizando o ano de 2010 para treinamento, o ano de 2011 para validação e o ano de 2012 para teste, respeitando a ordem cronológica da série e evitando vazamento de informação.

3) *Espaço de Busca*: A seleção dos hiperparâmetros do modelo foi realizada por meio de um procedimento de *grid search*, no qual diferentes combinações de parâmetros não sazonais e sazonais foram avaliadas, além de diferentes variáveis exógenas (lags). O critério utilizado para comparação dos modelos foi o Akaike Information Criterion (AIC), que equilibra qualidade de ajuste e complexidade do modelo.

O espaço de busca considerou valores baixos para os parâmetros autorregressivos e de média móvel, conforme prática comum na literatura para séries temporais meteorológicas, bem como a inclusão de diferenciação sazonal para tratar a não estacionariedade associada ao ciclo diário. O período sazonal foi fixado em $s = 24$, correspondente à frequência horária dos dados. Para as variáveis exógenas, foram usados os lags 1, 2, 3, .

4) *Hiperparâmetros Selecionados*: Após a execução do *grid search*, a combinação de hiperparâmetros que apresentou o menor valor de AIC foi:

$$SARIMAX(1, 0, 1) \times (1, 1, 1)_{24, lag1, lag2}$$

onde:

- $p = 1$: ordem autorregressiva não sazonal;
- $d = 0$: grau de diferenciação não sazonal;
- $q = 1$: ordem da média móvel não sazonal;
- $P = 1$: ordem autorregressiva sazonal;
- $D = 1$: grau de diferenciação sazonal;
- $Q = 1$: ordem da média móvel sazonal;
- $s = 24$: período sazonal correspondente a um ciclo diário.

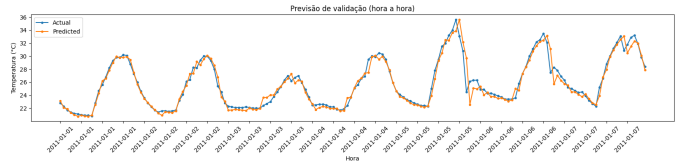


Fig. 6. Previsão usando SARIMAX sobre o conjunto de validação.

B. Machine learning

1) *Conceitos Básicos*: Nessa seção iremos discorrer sobre a aplicação de algoritmos de *Machine Learning* baseados em árvores de decisão para a tarefa de previsão da variável alvo. Foram considerados dois métodos amplamente utilizados na literatura: o **Random Forest** e o **Gradient Boost**. Ambos os modelos são capazes de capturar relações não lineares e interações complexas entre as variáveis explicativas, sendo especialmente adequados para dados com padrões sazonais e não estacionários.

O Random Forest é um método de *ensemble* que combina múltiplas árvores de decisão treinadas sobre subconjuntos aleatórios dos dados e das variáveis, reduzindo a variância e aumentando a robustez do modelo. Já o Gradient Boosting é um método de aprendizado em conjunto que constrói modelos fracos, tipicamente árvores de decisão rasas, de forma sequencial. A cada iteração, um novo modelo é ajustado para aproximar o gradiente negativo da função de perda em relação

às previsões atuais, permitindo a correção progressiva dos erros cometidos pelos modelos anteriores. Essa abordagem possibilita a modelagem de relações não lineares complexas e interações entre variáveis, o método de Gradient Boost é amplamente utilizado em problemas de regressão devido à sua capacidade preditiva e flexibilidade.

Ao longo deste trabalho, dois tipos de modelagem foram avaliados para ambos os algoritmos:

- Modelos que utilizam **lags da série temporal**, incorporando informações históricas diretas da variável alvo;
- Modelos que **não utilizam lags**, considerando apenas variáveis explicativas relacionadas ao tempo, como hora do dia, dia da semana e indicadores sazonais.

2) *Justificativa*: A escolha do Random Forest e do Gradient Boost se justifica pela capacidade desses algoritmos em modelar padrões complexos sem a necessidade de pressupostos fortes sobre a distribuição dos dados. Diferentemente dos modelos estatísticos clássicos, esses métodos não exigem estacionariedade da série e conseguem capturar efeitos não lineares entre as variáveis temporais.

A comparação entre modelos com e sem lags permite avaliar o impacto da dependência temporal explícita no desempenho preditivo. Enquanto os lags incorporam diretamente a dinâmica passada da série, as variáveis temporais permitem verificar se padrões sazonais e cíclicos são suficientes para explicar a variação da variável alvo. Essa abordagem fornece uma análise mais abrangente do comportamento da série e da capacidade de generalização dos modelos.

3) *Espaço de Busca*: A tunagem de hiperparâmetros foi realizada por meio de um random search em um espaço previamente definido. O espaço de busca adotado para cada modelo é descrito a seguir.

a) *Random Forest*:

- `n_estimators`: { `randint(50, 1000)` }
- `max_depth`: { `5, 10, 15, 20, 30, 40, 50, None` }
- `min_samples_split`: { `randint(2, 100)` }
- `min_samples_leaf`: { `randint(1, 100)` }
- `max_features`: { `ex.: sqrt, log2, none` }

b) *Gradient Boost*:

- `model_n_estimators`: { `randint(50, 1000)` }
- `model_learning_rate`: { `uniform(0.01, 0.3)` }
- `max_depth`: { `randuint(2, 100)` }
- `model_min_samples_split`: { `randint(2, 100)` }
- `model_min_samples_leaf`: { `randint(1, 100)` }
- `model_subsample`: { `uniform(0.5, 0.5)` }

4) *Hiperparâmetros Seleccionados*: Após o processo de tunagem, foram seleccionados os hiperparâmetros que apresentaram melhor desempenho no conjunto de validação, considerando métricas de erro apropriadas ao problema e o modelo de treinamento k-fold.

a) *Random Forest*:

- `n_split`: { `3` }
- `n_estimators`: { `439` }
- `max_depth`: { `15` }
- `min_samples_split`: { `3` }

- `min_samples_leaf`: { `4` }
- `max_features`: { `log2` }

b) *Gradient Boost*:

- `n_split`: { `3` }
- `model_n_estimators`: { `819` }
- `model_learning_rate`: { `0.05286004537658223` }
- `max_depth`: { `4` }
- `model_min_samples_split`: { `54` }
- `model_min_samples_leaf`: { `22` }
- `model_subsample`: { `0.8609993861334124` }

C. Long Short-Term Memory (LSTM)

1) *Conceitos Básicos*: As redes **Long Short-Term Memory (LSTM)** são uma variação especial das Redes Neurais Recorrentes (RNNs), desenvolvidas especificamente para aprender dependências de longo prazo. Enquanto RNNs tradicionais são eficazes em processar dados sequenciais, elas sofrem com o problema do desvanecimento do gradiente (*vanishing gradient*), que dificulta o aprendizado de correlações temporais distantes.

A arquitetura LSTM supera essa limitação introduzindo uma célula de memória que pode manter informações por longos períodos, regulada por estruturas chamadas *gates*: o *forget gate* (portão de esquecimento), o *input gate* (portão de entrada) e o *output gate* (portão de saída). Esses mecanismos permitem que a rede decida seletivamente quais informações descartar, quais armazenar e quais utilizar para a saída atual, tornando-a ideal para séries temporais meteorológicas onde o contexto histórico e a sequencialidade são cruciais.

2) *Justificativa*: A escolha do modelo LSTM para este trabalho baseia-se na natureza complexa e não linear dos dados meteorológicos. Diferentemente de modelos estatísticos lineares (como o SARIMAX), as redes neurais profundas possuem a capacidade de modelar interações não lineares entre a variável alvo (temperatura) e as variáveis exógenas.

Optou-se pela LSTM em detrimento de outras arquiteturas de *Deep Learning* (como MLPs ou CNNs puras) devido à sua habilidade intrínseca de preservar o estado temporal. Considerando que a temperatura em uma determinada hora é fortemente influenciada não apenas pelas condições imediatas, mas pela evolução das horas anteriores (inércia térmica), a capacidade da LSTM de manter um “estado” interno a torna superior para capturar a dinâmica temporal da série de Porto Alegre.

3) *Pré-processamento e Engenharia de Atributos*: Para garantir a estabilidade numérica e a convergência eficiente da rede neural, foram aplicadas técnicas distintas de normalização e codificação, respeitando a natureza de cada variável. É fundamental destacar que, para evitar o vazamento de dados (*data leakage*), todos os transformadores (*scalers*) foram ajustados (*.fit*) exclusivamente no conjunto de treinamento e apenas aplicados (*.transform*) nos conjuntos de validação e teste.

a) *Normalização de Variáveis Contínuas*: A variável alvo (temperatura) e as variáveis de pressão, umidade e vento foram submetidas à padronização utilizando o *Standard-Scaler*. Este método centraliza os dados removendo a média e escalonando-os para variância unitária, conforme a equação:

$$z = \frac{x - \mu}{\sigma}$$

onde μ e σ são a média e o desvio padrão calculados sobre os dados de treino.

b) *Codificação Cíclica Temporal*: As variáveis temporais “Hora do Dia” e “Dia do Ano” possuem natureza cíclica, mas representações numéricas lineares falham em capturar essa continuidade (por exemplo, a distância numérica entre 23h e 00h é grande, embora sejam temporalmente adjacentes).

Para corrigir isso e permitir que o modelo compreenda a sazonalidade, aplicou-se uma transformação trigonométrica, convertendo cada variável temporal em dois componentes (seno e cosseno):

$$x_{sin} = \sin\left(\frac{2\pi \cdot t}{T}\right), \quad x_{cos} = \cos\left(\frac{2\pi \cdot t}{T}\right)$$

onde t é o valor temporal atual e T é o período máximo correspondente (24 para horas e 365 para dias do ano). Essa abordagem adiciona 4 novas *features* ao modelo, mapeando o tempo em um círculo unitário.

c) *Janelamento (Windowing) e Causalidade*: A estrutura de entrada da LSTM foi gerada através de uma janela deslizante (*sliding window*) de tamanho L (lag). Para garantir a integridade causal do modelo e impedir o vazamento de informações futuras, a matriz de entrada X_t para prever y_t foi construída estritamente com observações passadas:

$$X_t = [x_{t-L}, x_{t-L+1}, \dots, x_{t-1}]$$

Dessa forma, o modelo utiliza apenas as informações disponíveis até o instante $t - 1$ para realizar a previsão do instante t .

4) *Espaço de Busca e Seleção de Modelos*: A definição da arquitetura final foi conduzida em duas etapas distintas: primeiramente, a determinação da janela de observação temporal (Lag) e, em seguida, a otimização dos hiperparâmetros internos da rede.

a) *Definição da Janela de Lag*: Antes da execução da busca aleatória para a rede, foram realizados testes preliminares para fixar o tamanho da janela de entrada (L). Utilizando uma arquitetura de referência, foram avaliados os valores $L \in \{4, 6, 8, 12, 24\}$. Esses candidatos foram escolhidos por representarem divisores lógicos do ciclo diário de 24 horas.

Os experimentos revelaram que, no cenário multivariado, o desempenho convergiu satisfatoriamente com $L = 8$. Esse resultado sugere que o conjunto de variáveis exógenas e temporais fornece contexto suficiente para compensar a necessidade de um histórico temporal mais longo da temperatura. Portanto, fixou-se $L = 8$.

b) *Otimização de Hiperparâmetros*: Com o lag definido, utilizou-se a biblioteca *Keras Tuner* para realizar uma busca aleatória (*Random Search*) avaliando 50 configurações de modelos. O espaço de busca compreendeu:

- **Batch Size**: {16, 32, 64, 128}
- **Camadas LSTM**: {1, 2, 3}
- **Unidades**: 32 a 256 (com passo de 32)
- **Dropout**: 0.0 a 0.5 (com passo de 0.1)
- **Learning Rate**: $\{10^{-2}, 10^{-3}, 10^{-4}\}$

c) *Estratégia de Treinamento e Parada Antecipada*: Para otimizar o custo computacional, adotou-se uma estratégia de dois estágios baseada no callback de *Early Stopping* (parada antecipada), monitorando a perda na validação (*val_loss*).

Durante a etapa de busca (*tuning*), os modelos foram treinados por no máximo 50 épocas com uma paciência de 5 épocas, permitindo o descarte rápido de arquiteturas promissoras que não convergiam. Já para o treinamento final da melhor arquitetura selecionada, o limite foi estendido para 100 épocas com uma paciência de 10 épocas, garantindo a convergência plena e a estabilidade dos pesos do modelo.

5) *Hiperparâmetros Selecionados*: Para a configuração final do modelo, o vetor de entrada foi composto por 8 variáveis: as 4 variáveis físicas (temperatura, pressão, umidade e vento) e as 4 variáveis temporais cíclicas (seno e cosseno de hora e dia). A inclusão dessas variáveis permitiu enriquecer o contexto do modelo, capturando tanto a dinâmica atmosférica quanto a sazonalidade exata.

Após a execução do *Random Search* e análise das métricas, a configuração final selecionada (utilizando variáveis exógenas, temporais e lag 8) foi:

- **Features de Entrada**: 8 (4 Físicas + 4 Temporais)
- **Janela de Lag**: 8 horas
- **Camadas LSTM**: 1
- **Unidades (Neurônios)**: 160
- **Taxa de Dropout**: 0.1
- **Batch Size**: 128
- **Learning Rate**: 0.001
- **Otimizador**: Adam

Essa configuração apresentou o melhor equilíbrio entre capacidade de generalização e precisão. Com essa arquitetura, o modelo alcançou um desempenho robusto no conjunto de teste (MSE: 0.3238, MAE: 0.4085), validando a eficácia da abordagem proposta para a previsão da temperatura.

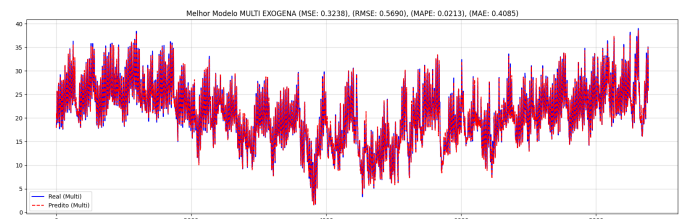


Fig. 7. Resultados obtidos com LSTM.

IV. ANÁLISE E COMPARAÇÃO DE RESULTADOS

Nesta seção são apresentados e comparados os resultados obtidos pelos modelos treinados para a tarefa de previsão da temperatura da próxima observação. A avaliação do desempenho foi conduzida por meio de métricas adequadas a problemas de regressão aplicados a séries temporais, permitindo uma análise quantitativa da acurácia e da robustez dos modelos.

A. Métricas de Avaliação

Os modelos foram avaliados utilizando as métricas Erro Médio Absoluto (MAE), Raiz do Erro Quadrático Médio (RMSE) e Erro Percentual Absoluto Médio (MAPE).

O MAE mede o erro médio absoluto entre os valores previstos e os valores reais, fornecendo uma interpretação direta do erro em unidades da variável alvo. Essa métrica é menos sensível a valores extremos e permite uma análise clara da magnitude média dos erros.

O RMSE penaliza erros maiores de forma mais severa devido à elevação ao quadrado, sendo especialmente útil para identificar modelos que apresentam grandes desvios pontuais. Dessa forma, essa métrica complementa o MAE ao fornecer uma visão mais rigorosa do comportamento do erro.

O MAPE expressa o erro médio em termos percentuais, permitindo uma interpretação relativa do desempenho do modelo em relação aos valores reais observados. Essa métrica facilita a comparação entre modelos, embora seja sensível a valores próximos de zero, aspecto considerado na interpretação dos resultados.

B. Comparação entre Modelos

A comparação dos resultados indicou desempenho semelhante entre os modelos avaliados, com pequenas variações nas métricas consideradas. Observou-se que a inclusão da variável exógena *lag1*, correspondente à temperatura da observação imediatamente anterior, exerceu papel central na capacidade preditiva dos modelos SARIMAX e dos modelos baseados em aprendizado de máquina, como Random Forest e Gradient Boosting.

Foram realizados experimentos com diferentes números de defasagens temporais. No entanto, a inclusão de lags adicionais resultou em ganhos irrelevantes de desempenho, não compensando o aumento do custo computacional e da complexidade do modelo. Dessa forma, optou-se pela utilização apenas do *lag1*, que apresentou bom equilíbrio entre simplicidade e acurácia.

Em contrapartida, para o modelo LSTM, o melhor desempenho preditivo foi obtido com a utilização de uma janela temporal maior, sendo o *lag 8* o que apresentou resultados mais consistentes. Esse comportamento está alinhado com a capacidade da arquitetura LSTM de explorar dependências temporais mais longas, diferindo dos modelos tradicionais, que se beneficiaram predominantemente de informações de curto prazo.

Na tabela III temos todas as métricas obtidas ao avaliar cada modelo.

TABLE III
MÉTRICAS DE CADA MODELO

Modelo	MAE	RMSE	MAPE
SARIMAX	0.4822	0.4822	2.66%
Gradient Boost	0.7971	0.7971	4.40%
LSTM	0.4085	0.5690	2.13%

C. Teste de Significância

A aplicação de testes estatísticos de significância para comparação de modelos não se mostra adequada no contexto deste trabalho, uma vez que o problema abordado consiste na previsão de uma única observação futura (one-step à frente) em uma série temporal. Diferentemente de cenários com múltiplas amostras independentes de erro, a avaliação do desempenho dos modelos baseia-se em um conjunto temporal fixo, no qual as previsões são fortemente correlacionadas no tempo. Dado essas condições, é válido ressaltar que a avaliação foi conduzida com o objetivo de avaliar o desempenho preditivo dos modelos e não da sua capacidade de inferência estatística.

V. CONCLUSÃO E DISCUSSÃO

Neste trabalho, foi investigado o problema de previsão da temperatura da próxima observação a partir de dados meteorológicos históricos, utilizando diferentes abordagens de modelagem baseadas em séries temporais e aprendizado de máquina. Ao longo das etapas de análise exploratória, pré-processamento, engenharia de atributos e modelagem, foi possível compreender melhor as propriedades da série e avaliar o desempenho dos modelos propostos.

A. Desempenho dos Modelos

Os resultados obtidos indicam que os modelos avaliados apresentaram boa capacidade preditiva para previsões de curto prazo, especialmente para a tarefa de prever um único passo à frente. As métricas MAE, RMSE e MAPE evidenciam que todos os modelos conseguiram capturar adequadamente a dependência temporal da série, com diferenças quantitativas entre as abordagens.

O modelo SARIMAX apresentou desempenho consistente, com MAE de 0.4822 e MAPE de 2.66%, destacando-se pela capacidade de incorporar explicitamente a estrutura temporal e componentes sazonais da série. Esse resultado reforça a adequação de modelos estatísticos clássicos quando a série apresenta forte autocorrelação e sazonalidade bem definida.

Os modelos baseados em árvores, Random Forest e Gradient Boosting, apresentaram desempenho inferior em comparação aos demais, com MAE de 0.7971 e MAPE de 4.40%. Esse comportamento sugere que, embora esses modelos sejam robustos para problemas de regressão geral, sua capacidade de capturar dependências temporais é limitada quando comparada a abordagens especificamente projetadas para séries temporais.

A LSTM apresentou o melhor desempenho geral, com MAE de 0.4085 e MAPE de 2.13%, indicando maior precisão na previsão da próxima observação. Esse resultado evidencia a capacidade das redes recorrentes em modelar padrões temporais

complexos e dependências sequenciais, mesmo em cenários de previsão de curto horizonte.

B. Abordagem Alternativa: Previsão por Intervalo de Confiança

Além da previsão pontual, foi explorada uma abordagem alternativa baseada na estimativa de intervalos de confiança para as previsões. Essa estratégia permitiu avaliar não apenas o valor esperado da temperatura futura, mas também a incerteza associada à previsão.

A tentativa da exploração nasceu da necessidade de realizar previsões mais distantes, considerando um maior intervalo de tempo para criar a previsão.

Entretanto, não foi possível implementar essa abordagem de forma completa ao longo deste trabalho. A principal dificuldade esteve relacionada à complexidade da modelagem multi-step, bem como à forma de organização da série temporal e à necessidade de ajustes adicionais na estrutura dos dados e nos modelos utilizados. A propagação do erro ao longo das previsões sucessivas e o aumento significativo da incerteza também se mostraram desafios relevantes.

C. Insights

A partir dos experimentos realizados, alguns insights importantes podem ser destacados:

- A dependência temporal de curto prazo, representada principalmente pela variável lag1, mostrou-se o principal fator explicativo para a previsão da temperatura.
- Testes estatísticos como ADF e KPSS, aliados à análise de ACF e PACF, foram fundamentais para compreender a estrutura da série e orientar decisões de modelagem.
- Modelos especializados em séries temporais e redes recorrentes apresentaram desempenho superior quando comparados a modelos genéricos de regressão.
- A inclusão de lags adicionais resultou em ganhos irrelevantes, não compensando o aumento da complexidade e do custo computacional.

D. Limitações

Apesar dos resultados positivos, este trabalho apresenta algumas limitações. A previsão foi restrita a um horizonte de um passo à frente e a um único local geográfico, o que limita a generalização dos resultados. Além disso, não foram exploradas abordagens multivariadas mais complexas.

E. Potenciais Futuros

- Ampliação do horizonte de previsão.
- Avaliação do uso de modelos híbridos que combinam abordagens estatísticas e redes neurais.

REFERÊNCIAS

- [1] INMET, “Dados meteorológicos históricos,” portal.inmet.gov.br. <https://portal.inmet.gov.br/dadoshistoricos> (accessed Dec. 5, 2025).
- [2] TensorFlow, “TensorFlow: An end-to-end open source machine learning platform,” tensorflow.org. <https://www.tensorflow.org> (accessed Dec. 5, 2025).
- [3] F. Chollet et al., “Keras: Deep learning for humans,” keras.io. <https://keras.io> (accessed Dec. 5, 2025).

- [4] Statsmodels Development Team, “Statsmodels: Statistics in Python,” statsmodels.org. <https://www.statsmodels.org> (accessed Dec. 5, 2025).
- [5] Scikit-learn Developers, “Scikit-learn: Machine learning in Python,” scikit-learn.org. <https://scikit-learn.org> (accessed Dec. 5, 2025).