UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

Caio Luchetti Cortada

PREDIÇÃO DE SÉRIES TEMPORAIS COM MODELOS TREINADOS A PARTIR DA TRANSFERÊNCIA DE APRENDIZADO

Sorocaba

2025

CAIO LUCHETTI CORTADA

PREDIÇÃO DE SÉRIES TEMPORAIS COM MODELOS TREINADOS A PARTIR DA TRANSFERÊNCIA DE APRENDIZADO

Trabalho de Conclusão de Curso apresentado à Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba, como parte dos requisitos para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Leopoldo André Dutra Lusquino Filho

Sorocaba

2025

C827p

Cortada, Caio Luchetti

Predição de séries temporais com modelos treinados a partir da transferência de aprendizado / Caio Luchetti Cortada. -- Sorocaba, 2025

59 p. : il., tabs., fotos

Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba

Orientador: Leopoldo André Dutra Lusquino Filho

1. Aprendizagem profunda (Aprendizado do computador). 2. Análise de séries temporais. 3. Meteorologia. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

CAIO LUCHETTI CORTADA

PREDIÇÃO DE SÉRIES TEMPORAIS COM MODELOS TREINADOS A PARTIR DA TRANSFERÊNCIA DE APRENDIZADO

Trabalho de Conclusão de Curso apresentado ao Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista (UNESP), como parte dos requisitos para obtenção do grau de Bacharel(a) em Engenharia de Controle e Automação.

Data da defesa: 27/05/2025

BANCA EXAMINADORA:

Prof. Dr. Leopoldo André Dutra Lusquino Filho
UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

Prof. Dr. William Dantas Vichete
UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

Prof. Dr. Fabrício Leonardo Silva
UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

AGRADECIMENTOS

Gostaria de agradecer, em primeiro lugar, à minha família, pelo apoio incondicional em todos os momentos, desde o incentivo no colégio, educação de qualidade, no apoio durante meus anos de universidade. Um agradecimento em especial, ao meu pai, cujo esforço e superação após o AVC me inspiraram a manter o foco e a determinação.

Agradeço também ao meu orientador, Prof. Dr. Leopoldo André Dutra Lusquino Filho, pela orientação, paciência e incentivo ao longo do desenvolvimento deste trabalho.

À minha namorada, Beatriz, por todo o apoio durante o periodo de graduação e incentivo para eu buscar meus objetivos.

Por fim, gostaria de agradecer aos meus amigos Giuliano, Thiago, Santiago, Ricardo, Guilherme, João Paulo, Christian, André e João Pedro por toda a amizade, vivência universitária e colaborações em todos os trabalhos e estudos durante esses cinco anos e meio de faculdade.

RESUMO

A previsão de séries temporais climáticas é um desafio relevante para diversas áreas, como agricultura, energia e planejamento urbano, especialmente diante da complexidade dos dados meteorológicos e da escassez de informações históricas confiáveis. Este trabalho propõe uma análise comparativa entre modelos tradicionais e modernos de previsão, com foco em redes neurais profundas como LSTM, RNN, GRU e CNN, além da técnica de Transfer Learning para melhorar a generalização dos modelos e reduzir o custo computacional. Foram utilizados dados climáticos extraídos do portal POWER LARC da NASA, referentes às cidades de São Paulo (Brasil) e Calgary (Canadá), abrangendo variáveis como temperatura, umidade, pressão e velocidade do vento. A LSTM apresentou o melhor desempenho entre os modelos testados, com métricas superiores de erro e alinhamento temporal. A aplicação do Transfer Learning demonstrou ser eficaz tanto dentro do mesmo domínio quanto em domínios distintos, mantendo ou melhorando os resultados preditivos. Os experimentos reforçam a viabilidade do uso de aprendizado profundo aliado ao Transfer Learning na previsão meteorológica, especialmente em cenários com dados limitados. O trabalho contribui para o avanço de soluções eficientes e transferíveis para aplicações em séries temporais multivariadas no contexto climático.

Palavras-chave: séries temporais; deep learning; LSTM; previsão climática; transfer learning.

ABSTRACT

Forecasting climate time series is a significant challenge across various sectors, such as agriculture, energy, and urban planning, especially given the complexity of meteorological data and the scarcity of reliable historical records. This study presents a comparative analysis between traditional and modern forecasting models, focusing on deep learning architectures such as LSTM, RNN, GRU, and CNN, as well as the application of Transfer Learning to enhance model generalization and reduce computational cost. Climate data was sourced from NASA's POWER LARC portal for the cities of São Paulo (Brazil) and Calgary (Canada), including variables such as temperature, humidity, pressure, and wind speed. Among the tested models, the LSTM network delivered the best performance, achieving superior accuracy and temporal alignment metrics. The application of Transfer Learning proved effective both within the same domain and across different domains, maintaining or even improving predictive performance. The experiments support the viability of deep learning combined with Transfer Learning for meteorological forecasting, particularly in scenarios with limited data availability. This work contributes to the advancement of efficient and transferable solutions for multivariate time series applications in climate-related contexts.

Keywords: time series; deep learning; LSTM; climate forecasting; transfer learning.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

Sumário

1 - Introdução

A previsão climática desempenha um papel essencial em várias áreas da sociedade, como agricultura, geração de energia, prevenção de desastres e até no planejamento do dia a dia das pessoas. A análise de séries temporais dos elementos climáticos é um desafio devido à complexidade dos fatores envolvidos, como tendências e sazonalidade (IPCC, 2021). Além disso, fenômenos meteorológicos extremos são difíceis de prever devido à sua natureza não linear (IPCC, 2021).

Outra dificuldade é a obtenção dos dados meteorológicos que serão utilizados na análise, cálculos e previsão. Dados históricos são muitas vezes ruidosos e com valores discrepantes da realidade, seja por falha na captura ou mau tratamento pelos envolvidos nos projetos. Portanto, é fundamental recorrer a fontes de dados confiáveis, obtidos por meio de metodologias adequadas e validadas.

Modelos como ARIMA e Holt-Winters, têm sido muito utilizados para esse propósito, mas com dificuldades de capturar padrões não lineares e interdependências complexas dos dados climáticos e fazendo necessário a utilização de modelos mais modernos para suas tarefas. (HEWAMALAGE; BERGMEIR; BANDARA, 2021)

Outros dois grandes desafios enfrentados são a escala de tempo em que será feita a previsão, seja ela curta ou longa, e o ambiente geográfico em que seu modelo será modelado. Previsões de diferentes janelas no tempo exigem diferentes tamanhos de entrada nos seus modelos, levando a um treinamento que pode exigir grandes bases de dados e poder computacional grande e custoso, tanto energeticamente quanto financeiramente. O local geográfico também é um ponto muito importante a ser levado em consideração, pois a modelagem pode ficar enviesada em detrimento do local dos dados obtidos, fazendo com que a sua previsão se torne ineficaz.

Com o rápido avanço das redes neurais e modelos de aprendizado de máquina, os modelos preditivos tiveram uma melhora significativa. Modelos baseados em aprendizado profundo, como a LSTM (Long Short-Term Memory) e a RNN (Recurrent Neural Network), demonstram capacidade superior na identificação de padrões temporais complexos e não lineares, especialmente em grandes bases de dados. Apesar dessa vantagem, a aplicação de tais modelos apresenta desafios consideráveis, incluindo um elevado custo computacional e a exigência de grandes volumes de dados para o treinamento. Soma-se a isso a dificuldade na interpretação dos resultados, conhecida como o problema da "caixa-preta", e a necessidade de

um minucioso ajuste de hiperparâmetros para garantir que os modelos convirjam de forma adequada (HEWAMALAGE; BERGMEIR; BANDARA, 2021).

Uma estratégia eficaz para otimizar a criação de modelos de aprendizado profundo é a aplicação da Aprendizagem por Transferência (Transfer Learning). Essa técnica consiste no reaproveitamento do conhecimento de modelos previamente treinados em grandes bases de dados para a aplicação em contextos específicos. Em vez de treinar uma rede neural a partir do zero, a abordagem utiliza a estrutura e os pesos de uma rede pré-treinada como ponto de partida, realizando apenas um ajuste fino (fine-tuning) para o novo problema. Isso não apenas reduz significativamente o custo computacional e o tempo de treinamento, mas também se mostra uma solução robusta para cenários com escassez de dados, melhorando a capacidade de generalização do modelo preditivo (PAN; YANG, 2010).

Dessa forma, este trabalho tem como motivação demonstrar a eficácia da utilização de redes neurais para predição de séries temporais climáticas. Para isso, são comparadas as capacidades preditivas e eficiência de diferentes modelos, incluindo abordagens tradicionais e técnicas de Transfer Learning. Com o objetivo de identificar uma arquitetura que ofereça o melhor equilíbrio entre custo computacional e precisão, considerando a grande preocupação recente com a limitação dos recursos energéticos e computacionais disponíveis.

Dessa forma, este trabalho visa demonstrar a eficácia de redes neurais na predição de séries temporais climáticas. Para isso, o estudo se propõe a, primeiramente, analisar comparativamente modelos tradicionais (SARIMA) e modernos (LSTM, RNN, GRU, CNN). Em seguida, busca-se avaliar a técnica de Transfer Learning como um método para melhorar a generalização e reduzir o custo computacional. Por fim, o objetivo é identificar uma arquitetura com bom equilíbrio entre custo computacional e precisão, considerando a crescente preocupação com a sustentabilidade e a limitação de recursos energéticos.

2 - Revisão de literatura

Este capítulo apresenta a fundamentação teórica que alicerça o presente trabalho. A revisão de literatura abordará os conceitos essenciais, partindo da definição e dos componentes de séries temporais, em seguida, serão exploradas as arquiteturas de aprendizado profundo, como as Redes Neurais Recorrentes (RNN) e Long Short-Term Memory (LSTM), que constituem o cerne das abordagens modernas, após isso, será discutida a técnica de Aprendizagem por Transferência (Transfer Learning) como uma estratégia para otimização de modelos. Por fim, são abordados os modelos estatísticos tradicionais de previsão, como o SARIMA. A estrutura segue uma progressão lógica do fundamental ao específico, iniciando pela base conceitual das séries temporais.

2.1 - Séries temporais

Uma série temporal constitui um dos principais objetos de estudo em análises quantitativas que envolvem a dimensão do tempo. Formalmente, ela é definida como um conjunto de observações de uma variável, coletadas em intervalos de tempo regulares (EVERITT; SKRONDAL, 2010). Essa sequência de dados cronologicamente ordenados carrega informações sobre tendências, ciclos e sazonalidades, sendo, portanto, fundamental para a modelagem e previsão em domínios como econometria, finanças e climatologia. A análise de seus componentes e a aplicação de modelos preditivos são abordagens centrais em estudos estatísticos, como detalhado na obra de Morettin e Toloi (2018).

As séries temporais podem apresentar diversos comportamentos que afetam diretamente sua análise e modelagem. Entre os principais componentes estão a tendência, a sazonalidade e a estacionariedade, que desempenham um papel essencial na visualização da dinâmica dos dados temporais.

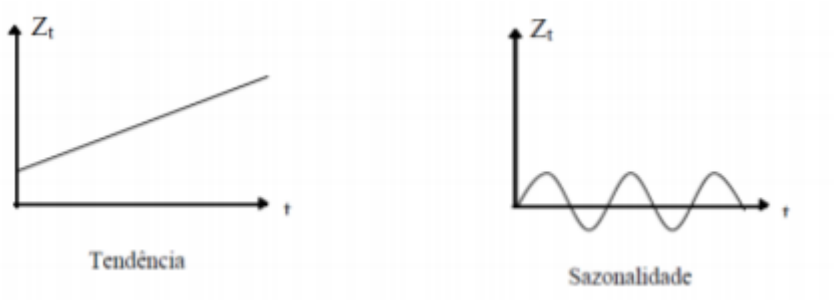
A tendência é um dos componentes fundamentais de uma série temporal e representa seu movimento de longo prazo, caracterizado por uma mudança persistente que não se repete de forma periódica. A presença de uma tendência, seja ela linear ou de outra natureza, viola uma premissa essencial para muitos modelos de previsão: a estacionariedade. Isso ocorre porque a média da série não se mantém constante ao longo do tempo, o que, por definição, a caracteriza como não estacionária (HAMILTON, 1994). Consequentemente, para que a modelagem estatística e a previsão sejam aplicadas de forma robusta, torna-se necessário um

pré-processamento para remover essa tendência, geralmente por meio de técnicas como a diferenciação da série. Tal comportamento é frequentemente observado em séries econômicas, como o Produto Interno Bruto (PIB) de um país, cuja análise exige o tratamento da tendência de longo prazo para a correta aplicação de modelos preditivos (MORETTIN; TOLOI, 2018).

A sazonalidade, por outro lado, se caracteriza por padrões que se repetem em ciclos ou períodos específicos, como meses, anos ou estações ano. Como afirma Brockwell e Davis (2016), a sazonalidade é identificada pela correlação significativa em períodos específicos da série temporal, o que pode indicar um padrão cíclico que se repete. Por exemplo, em regiões tropicais, chover durante o verão é algo que acontece repetidamente todos os anos. A identificação de sazonalidade é essencial, pois ajuda a melhorar a precisão das previsões dos modelos (BROCKWELL; DAVIS, 2016).

A Figura 1 ilustra esquematicamente a tendência e a sazonalidade, que são as principais fontes de não estacionariedade em uma série temporal.

Figura 1- Componentes de tendência e Sazonalidade



Fonte: MIRANDA (2007)

Assim, retomando, ao se deparar com as características de tendência e sazonalidade conforme os outros textos descreveram, é necessário as transformar em estacionárias para que seja possível realizar análises e a modelagem de modelos estatísticos em cima da série temporal. Modelos como ARIMA (Autoregressive Integrated Moving Average) partem do pressuposto que a série é estacionária para oferecer previsões confiáveis (BOX; JENKINS; REINSEL, 2008)

Outros aspectos importantes das séries temporais são o ruído e a aleatoriedade. O ruído pode ser definido como flutuações imprevisíveis e irregulares nos dados, que podem obscurecer os padrões sistemáticos de tendência e sazonalidade. Vários autores, como Chatfield (2003) e Morettin e Toloi (2018), descrevem técnicas de suavização ou alisamento, como, por exemplo,

a aplicação de médias móveis e filtros, como um método eficaz para reduzir o impacto do ruído e tornar a estrutura subjacente da série temporal mais clara para análise.

A aleatoriedade é o componente de uma série temporal que não pode ser explicado por modelos determinísticos. Ela está associada ao conceito de processo estocástico, que é um processo governado, em parte, pelo acaso e cuja evolução futura só pode ser descrita em termos de probabilidades. Como apontam Hyndman e Athanasopoulos (2018), algumas séries apresentam um comportamento altamente estocástico, em que a componente aleatória é tão significativa que a previsibilidade se torna limitada. Nesses cenários, a utilização de métodos mais avançados, como redes neurais ou modelos de aprendizado de máquina, torna-se necessária para tentar capturar possíveis padrões complexos e não lineares que estão ocultos no ruído (HYNDMAN; ATHANASOPOULOS, 2018).

Dessa maneira, o entendimento aprofundado dos componentes de uma série temporal, como tendência, sazonalidade e estacionariedade, é um pré-requisito para a modelagem preditiva. Conforme consolidado nas obras de autores de referência como Hamilton (1994), Box, Jenkins e Reinsel (2008) e Hyndman e Athanasopoulos (2018), uma análise criteriosa desses aspectos permite a aplicação de técnicas adequadas para obter previsões mais acuradas.

Neste contexto, modelos estatísticos clássicos como o ARIMA são frequentemente empregados como um ponto de referência fundamental, ou baseline. A inclusão do ARIMA neste trabalho tem, portanto, o propósito de estabelecer um benchmark de desempenho robusto. O resultado deste modelo servirá como base de comparação para avaliar a eficácia das arquiteturas de aprendizado profundo, que são o foco principal desta pesquisa. Essa análise comparativa permitirá quantificar objetivamente o ganho de performance obtido pelas redes neurais ao lidar com as não linearidades e interdependências complexas dos dados climáticos.

2.2 - Deep Learning

As redes neurais têm sido muito utilizadas na análise de séries temporais por conta de sua capacidade de capturar padrões complexos nos dados. Dentre os modelos mais convencionais, foram selecionados quatro para abordar neste trabalho: as Redes Neurais Convolucionais (CNNs), RNNs, Gated Recurrent Units (GRUs) e as LSTMs.

As Redes Neurais Convolucionais (CNNs) são arquiteturas eficazes na identificação de padrões locais e hierárquicos, filtrando informações relevantes através de operações de convolução (LECUN et al., 1998). Em seu artigo seminal, os autores demonstram que essa capacidade de aprender representações hierárquicas dos dados torna as CNNs aptas para

identificar padrões em séries temporais multivariadas. Mais recentemente, Borovykh, Bohte e Oosterlee (2017) destacam a eficiência das CNNs na previsão de séries temporais financeiras, aplicando camadas convolucionais para identificar dependências temporais de curto prazo de maneira eficiente. No entanto, apesar de seu bom desempenho em alguns cenários, as CNNs não foram originalmente projetadas para dados sequenciais. Sua limitação em capturar relações de longo prazo ao longo da sequência pode prejudicar a modelagem de séries temporais com fortes dependências temporais, tornando modelos recorrentes, como RNNs, GRUs e LSTMs, alternativas mais adequadas para esse tipo de tarefa.

Na modelagem de modelos de predição, LSTMs são muito utilizadas por conta de sua capacidade de aprender padrões complexos. Graves (2013) demonstrou que LSTMs superam as RNNs em tarefas que envolvem previsões a longo prazo e de padrões em janelas maiores. Recentemente, trabalhos como o de Smyl (2020), paper vencedor do desafio M4 de previsão de séries temporais, mostraram que arquiteturas híbridas que combinam LSTMs com outras abordagens estatísticas convencionais, podem melhorar significativamente a precisão preditiva.

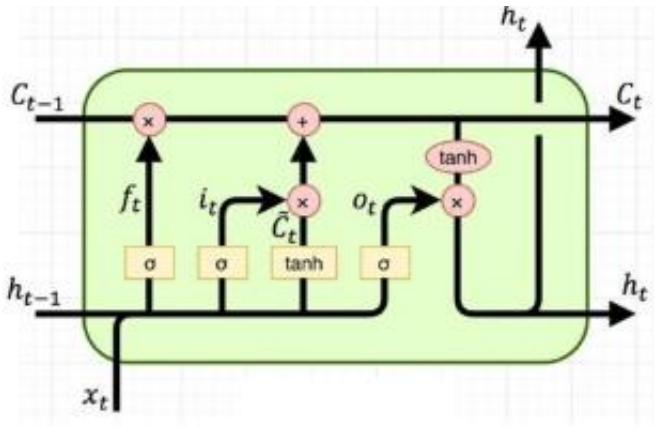
A topologia das redes neurais difere entre elas, afetando diretamente em como é feita a inferência dos modelos. As CNNs utilizam camadas convolucionais para extrair os padrões relevantes, seguidas por camadas totalmente conectadas que processam as características aprendidas. No processo de inferência, os dados de entrada percorrem a estrutura toda de maneira sequencial, e no caso de uma previsão de série temporal, produz previsões baseadas nos padrões identificados nos dados de treinamento (LI et al., 2022).

As RNNs e LSTMs, por outro lado, possuem uma estrutura diferente. Ambas possuem conexões recorrentes que permitem que os dados sejam analisados ao longo do tempo, capturando as dependências temporais. Durante esse processo, os modelos utilizam estados ocultos para armazenar informações do passado e ajustam de forma dinâmica a predição baseada na sequência analisada. No caso das LSTMs, são utilizadas as células de memória e mecanismos de portas, possibilitando um controle mais eficiente de que informações devem ser armazenadas ou descartadas, garantindo que os padrões de longo prazo sejam mais robustos (HOCHREITER; SCHIMDHUBER, 1997).

A arquitetura LSTM (Long Short-Term Memory) foi projetada para superar as limitações das RNNs no aprendizado de dependências de longo prazo (HOCHREITER; SCHMIDHUBER, 1997). Sua unidade básica, a célula LSTM, controla o fluxo de informações por meio de um sistema de três portões (gates), cuja estrutura é detalhada na Figura 2. Conforme explica Olah (2015), esses portões são: a porta de esquecimento (forget gate), que decide quais informações devem ser descartadas; a porta de entrada (input gate), que determina quais novas

informações serão armazenadas; e a porta de saída (output gate), que controla a geração da saída da célula.

Figura 2- Célula de LSTM



Fonte: OLAH 2015

O primeiro passo dentro de uma célula de LSTM é a decisão de quais informações serão descartadas. Essa decisão é tomada pela *forget gate*, que recebe como entrada o estado oculto anterior (h_{t-1}) e a entrada atual (x_t), calculando um valor entre 0 e 1 para cada elemento do estado da célula anterior (C_{t-1}). Um valor próximo de 1 indica que a informação será mantida enquanto um valor próximo de 0 indica que será descartada. A equação 1 que rege esse gate é descrita da seguinte forma:

$$\boldsymbol{f_t} = \boldsymbol{\sigma}\,[\,\boldsymbol{W_f} \cdot (\boldsymbol{h_{t-1}}, \boldsymbol{x_t}) + \boldsymbol{b_f}\,] \textbf{ (1)}$$

Com $\boldsymbol{W_f}$ sendo o peso atribuído no cálculo dessa saída e $\boldsymbol{b_f}$ é o fator de viés. No exemplo da previsão do tempo, essa função poderia descartar um estado em que a célula contiver algum dado sobre umidade relativa do ar, no entanto, se no período em que a rede está treinando o parâmetro for irrelevante para a predição, a célula pode esquecer essa informação.

Na *input gate*, é determinado quais novas informações serão armazenadas. Nessa porta temos duas etapas, a seleção da informação (i_t), como apresentado na equação 2, e após isso temos a vetorização dos novos valores que serão adicionados ao estado $\tilde{C}_t$, como apresentado na equação 3. Para a seleção é realizado mais um cálculo nos mesmos moldes do *forget gate* e

na sequência é aplicada uma função de tangente hiperbólica para a criação do vetor do estado. Os cálculos seguem as fórmulas a seguir:

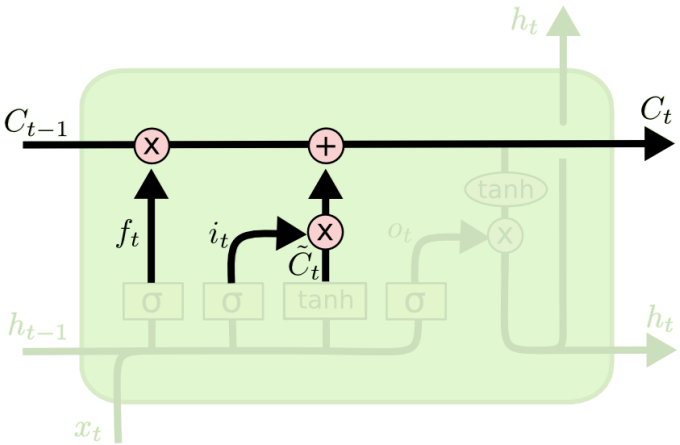
$$\boldsymbol{i_t} \;=\; \boldsymbol{\sigma}\,[\,\boldsymbol{W_i}\cdot(\boldsymbol{h_{t-1}},\boldsymbol{x_t})\;+\;\boldsymbol{b_i}\,] \textbf{ (2)}$$

$$\tilde{\boldsymbol{C}}_t \;=\; \boldsymbol{tanh}\,[\,\boldsymbol{W_C}\cdot(\boldsymbol{h_{t-1}},\boldsymbol{x_t})\;+\;\boldsymbol{b_C}\,] \textbf{ (3)}$$

Por fim, utilizamos o estado da antiga célula C_{t-1} para calcular o novo estado C_t. Então juntando todos os cálculos, primeiro multiplicamos o estado antigo por f_t, esquecendo o que decidimos esquecer anteriormente. Após isso, adicionamos $I_t * \tilde{C}_t$. Assim, encontramos o valor para a próxima célula candidata, obtendo a equação 4, que rege o mostrado no esquemático da figura 2:

$$C_t = f_t \,.\, C_{t-1} + i_t \,.\, \tilde{C}_t \;\textbf{ (4)}$$

Figura 3- Representação dos cálculos finais para o próximo estado.



Fonte: OLAH 2015

O treinamento das redes difere de modelo para modelo, para as LSTMs e RNNs. Na RNN o treinamento é mais simplificado e ocorre por meio da atualização de um estado oculto h_t, que armazena informações sobre os elementos anteriores da sequência dos dados.

O estado oculto é atualizado a cada novo passo temporal com a equação 5:

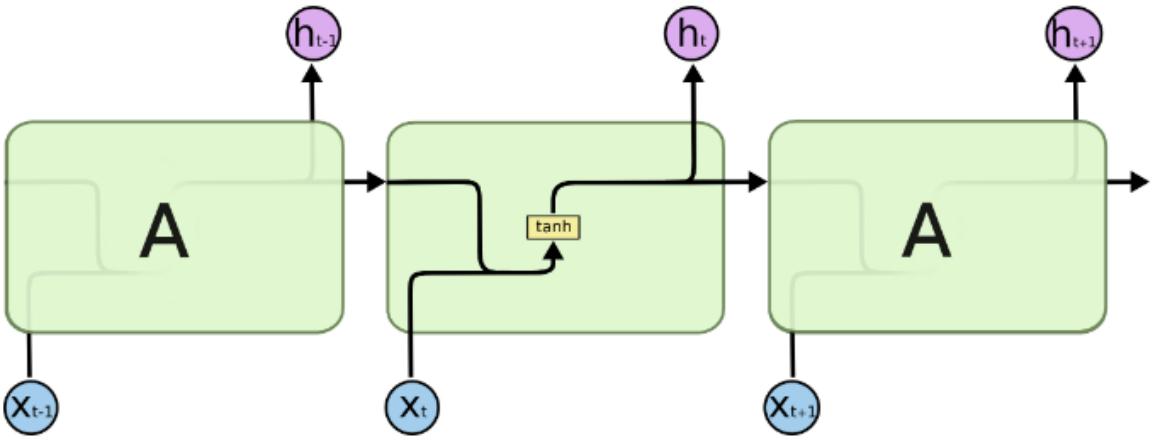
$$h_t \;=\; a \;\; (\,W_h h_{t-1} \;+\; W_x x_t + b\,) \textbf{ (5)}$$

Onde h_t é o novo estado oculto, h_{t-1} é o estado oculto do tempo anterior, x_t é a entrada atual, W_h e W_x são os pesos e treináveis, b é o viés e a é a função de ativação que pode ser tanh, ReLu, sigmoid, etc. E ao processar uma sequência, a RNN gera uma saída y_t, com a equação 6:

$$y_t = W_y h_t + b_y \quad \textbf{(6)}$$

onde W_y e b_y são os pesos e parâmetros treináveis da rede.

Figura 4- Representação de uma rede RNN com ativação tanh

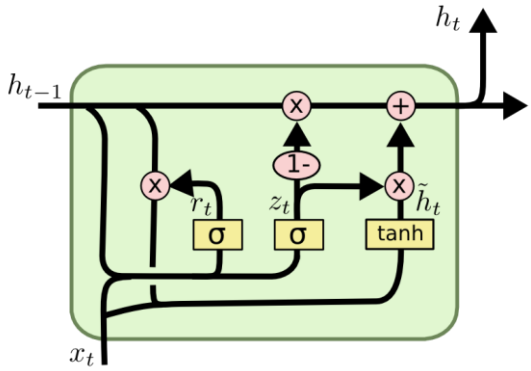


Fonte: OLAH 2015

A RNN é treinada usando o *backpropagation through time* (BPTT), uma adaptação do algoritmo de retro propagação. Esse processo ajusta os pesos, minimizando um erro. Porém uma deficiência desse modelo é exatamente o que é prevenido com a utilização das LSTMs, ela passa a perder as dependências de longo termo, que as LSTMs são capazes de manter.

Uma outra variação das RNNs apresentada por Cho et. al (2014), também para lidar com o problema do desaparecimento dos gradientes é a GRU, uma arquitetura semelhante à LSTM, porém mais simples e mais eficiente em termos computacionais. Elas utilizam mecanismos de gates semelhantes ao das LSTMs, mas combinam as funcionalidades de atualização e esquecimento em um único gate, reduzindo o número de parâmetros para o treinamento. Essa simplificação torna o modelo naturalmente mais rápido para treinar, ainda mantendo dependências de longo prazo. No estudo apresentado por Chung et. al. (2014), é mostrado que as GRUs frequentemente alcançam resultados próximos as LSTMs em tarefas de previsão sequencial, sendo uma alternativa competitiva para a modelagem de séries temporais.

Figura 5- Representação de uma célula de GRU



Fonte: OLAH 2015

Como representado na figura 5, a célula de uma GRU possui alguns cálculos assim como em outros modelos. A atualização do estado oculto de uma GRU é feita em algumas etapas, o gate de atualização, gate de reinicialização e o cálculo do novo estado candidato.

O gate de reinicialização (*reset gate)* é o primeiro gate a ser calculado, é regido pela equação 7:

$$r_t = \sigma\,(\,W_r\,.\,[h_{t-1}, x_t]\,)\ \textbf{(7)}$$

O reset gate determina o quanto do estado oculto anterior deverá ser esquecido. Ele recebe a informação do estado anterior e a atual e calcula um vetor com valores entre 0 e 1 que controla o quanto dos dados deverá ser reinicializado no momento atual.

O gate de atualização *(update gate)* é o segundo passo, é determinar o quanto do vetor de ativação deverá ser incorporado no novo estado oculto. Esse *gate* recebe o estado oculto anterior e a nova informação e produz um vetor entre 0 e 1 que controla o grau em qual o vetor de ativação candidato será incorporado no novo estado oculto, com cálculo descrito pela equação 8:

$$Z_t = \sigma\,(\,W_z\,.\,[h_{t-1}, x_t]\,)\ \textbf{(8)}$$

A partir dos dois gates, tem-se os pesos que são aprendidos durante o treinamento W_z e W_r. O cálculo do novo estado candidato segue a seguinte fórmula:

$$\tilde{h}_t = tanh\,(\,W\,.\,[r_t\,.\,h_{t-1}, x_t]\,)\ \textbf{(9)}$$

O *reset gate* é aplicado ao estado anterior, gerando uma nova representação do estado a partir da seleção da informação do estado anterior, sendo aplicada em uma função de ativação *tanh* que faz com que a saída seja entre 0 ou 1, sendo aplicada sobre os pesos que são obtidos nas etapas anteriores.

Por fim, o estado oculto de saída h_t é obtido depois de todos os cálculos anteriores, conforme a equação 10:

$$h_t = (1 - z_t) \, . \, h_{t-1} + z_t \, . \, \tilde{h}_t \quad \textbf{(10)}$$

A atualização final do estado oculto h_t ocorre como uma interpolação entre o estado anterior h_{t-1} e o novo estado candidato $\tilde{h}_t$, ponderada pelo *gate* de atualização z_t. Quando $z_t$está próximo de 1, a nova informação $\tilde{h}_t$ tem mais influência; quando está próximo de 0, o modelo mantém mais informações do estado anterior.

Segundo Lathuilière et al. (2020), a eficácia dos modelos de aprendizado profundo está diretamente relacionada à otimização dos hiperparâmetros e da escolha das técnicas de regularização, como *batch normalization* e *dropout.* Essas técnicas auxiliam o modelo a não ter *overfitting*, fenômeno que ocorre quando o modelo aprende excessivamente os detalhes presentes nos dados de treinamento, dessa forma, permitindo que as redes generalizem melhor para novos dados e melhorem sua robustez em previsões futuras.

O treinamento dessas redes exige a utilização de técnicas de otimização como o uso do gradiente descendente e variações como Adam e RMSprop, que são algoritmos que ajustam de forma dinâmica a taxa de aprendizado para acelerar a convergência, estabilizando o modelo (GOODFELLOW; BENGIO; COURVILLE, 2016).

Conforme discutido por Lathuilière et al. (2020), abordagens como o *deep learning* combinadas a técnicas de *fine-tuning* de redes pré-treinadas podem gerar ganhos significativos nos parâmetros avaliativos em tarefas de séries temporais, reduzindo o tempo de treinamento e melhorando a generalização.

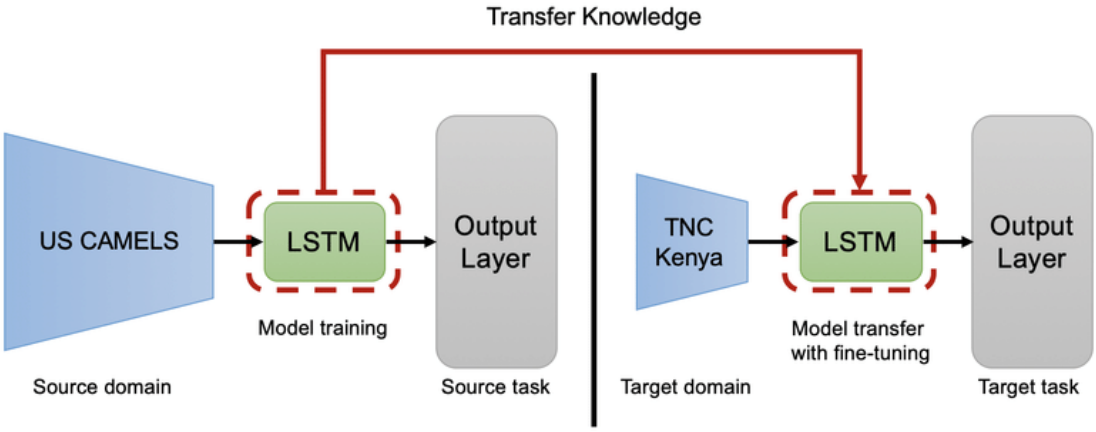
2.3 - Transfer Learning

O Transfer Learning, ou Aprendizado por Transferência, é uma técnica de aprendizado de máquina em que um modelo treinado para uma tarefa é reutilizado como ponto de partida para um segundo problema. Essa abordagem tem sido adotada com grande sucesso em diversas áreas, notadamente no Processamento de Linguagem Natural (PLN), com modelos como BERT e GPT, e na Visão Computacional, onde redes pré-treinadas em grandes bases de imagens,

como a ImageNet, são adaptadas para tarefas específicas como o diagnóstico médico. No contexto deste trabalho, a técnica também é explorada para a previsão de séries temporais, onde modelos pré-treinados são refinados para cenários específicos, visando acelerar o treinamento e melhorar a generalização (GOODFELLOW; BENGIO; COURVILLE, 2016).

Um exemplo de *transfer learning* é representado pela figura 6 abaixo:

Figura 6 - Esquemático exemplo de Transfer Learning com LSTM



Fonte: ORUCHE et al. (2023)

A principal vantagem do TL é explorar a capacidade de reduzir as necessidades dos modelos de redes neurais de utilizar grandes volumes de dados, permitindo que os modelos já ajustados a padrões sejam refinados com conjuntos de dados menores. Isso se mostra importante em contextos onde a captura de dados é limitada.

No contexto deste trabalho, a aplicação de TL em séries temporais pode ser aplicada em séries temporais. Uma rede neural pode ser treinada em qualquer domínio, desde análise climática até variações do mercado financeiro. Permitindo, então, melhor desempenho preditivo em novas tarefas ou tarefas semelhantes sem a necessidade de um treinamento do início.

Kim et al. (2023) exploram a aplicação de um modelo baseado em LSTM com TL para previsão da demanda energética em edifícios. Essa aplicação demonstrou que o uso de uma topologia mista de LSTM e TL melhora significativamente a precisão da previsão, em especial nos cenários nos quais os dados disponíveis para treinamento são limitados. Além disso, os resultados indicam que diferentes estratégias de Transfer Learning impactam diretamente no desempenho do modelo, alterando a quantidade de camadas congeladas e parâmetros no ajuste fino das camadas de treinamento.

Por fim, a utilização de TL para a previsão de séries temporais é um assunto que tem se tornado mais popular por conta de sua capacidade de lidar com os problemas de escassez de

dados e de generalização dos modelos. Estratégias como o ajuste fino e reutilização dos pesos treinados anteriormente permitem que os modelos se adaptem de forma a performar com excelência em diferentes contextos, tornando essa abordagem eficiente em termos de coleta de dados e tempo de treinamento. (KIM et al., 2023).

2.4 – SARIMA (Seasonal Autoregressive Integrated Moving Average)

O modelo SARIMA, também conhecido como modelo ARIMA sazonal, é uma extensão do modelo ARIMA que incorpora componentes sazonais. É amplamente utilizado para modelar séries temporais que apresentam padrões cíclicos regulares ao longo do tempo, como dados climáticos, econômicos e de demanda (HYNDMAN & ATHANASOPOULOS, 2018).

O modelo ARIMA combina três componentes: Auto-regressivo (AR): uma regressão linear dos valores passados da série; Integração (I): diferenciações aplicadas à série para torna-la estacionária; Média móvel (MA): modela os erros residuais como uma combinação linear dos erros passados.

Com isso, o modelo ARIMA(p, d, q) é representado pela equação 11:

$$\Phi(\mathrm{B})(1-\mathrm{B})^{d}\mathrm{y}_{t} = \Theta(\mathrm{B})\varepsilon\mathrm{t} \quad \textbf{(11)}$$

Em que y_t representa o valor da série temporal no tempo t, e ε_t é o termo de erro aleatório, que se assume como tendo média zero, variância constante e ausência de autocorrelação. O operador B, conhecido como operador de defasagem ou *backshift*, atua deslocando a série no tempo, de modo que $By_t=y_{t-1}$.

Os polinômios $\Phi(B)$ e $\Theta(B)$ são, respectivamente, o polinômio autorregressivo de ordem p e o polinômio de média móvel de ordem q. Especificamente, o polinômio autorregressivo é expresso como descrito na equação 12:

$$\Phi(\mathrm{B}) = 1 - \phi_1\mathrm{B} - \phi_2\mathrm{B}^2 - \cdots - \phi_\mathrm{p}\mathrm{B}^\mathrm{p} \textbf{(12)}$$

Enquanto o polinômio de média móvel assume a forma apresentada na equação 13:

$$\Theta(\mathrm{B}) = 1 + \theta_1\mathrm{B} + \theta_2\mathrm{B}^2 + \cdots + \theta_\mathrm{q}\mathrm{B}^\mathrm{q} \textbf{(13)}$$

O termo $(1-B)^d$ representa a operação de diferenciação aplicada d vezes à série temporal, a fim de remover a tendência e torná-la estacionária.

O modelo SARIMA, por sua vez, estende essa formulação ao incluir componentes sazonais, representados pelos parâmetros (P, D, Q, s). Nessa notação, *P* indica a ordem do componente autoregressivo sazonal, *D* representa o número de diferenciações sazonais necessárias para remover padrões periódicos, *Q* corresponde à ordem do componente de média móvel sazonal, e *s* refere-se ao período da sazonalidade, como, por exemplo, 12 em séries mensais com ciclos anuais.

A equação que define o modelo SARIMA completo, considerando os componentes sazonais e não sazonais, é dada pela equação 14:

$$\Phi_{\mathrm{P}}(\mathrm{B}^{\mathrm{s}})\,\phi(\mathrm{B})\,(1-\mathrm{B})^{\mathrm{d}}\,(1-\mathrm{B}^{\mathrm{s}})^{\mathrm{D}}\,\mathrm{y}_{\mathrm{t}} = \Theta_{\mathrm{Q}}(\mathrm{B}^{\mathrm{s}})\,\theta(\mathrm{B})\,\varepsilon_{\mathrm{t}} \quad \textbf{(14)}$$

Nessa equação, $\Phi(B)$ e $\Theta(B)$ mantêm os mesmos significados que no ARIMA tradicional, são os polinômios não sazonais de ordem *p* e *q*. Já $\Phi_p(B^s)\ e\ \Theta_Q(B^s)$ representam os polinômios sazonais autoregressivo e de média móvel, respectivamente, com defasagens múltiplas de *s*, sendo expressos pelas equações 15 e 16:

$$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_P B^{Ps} \quad \textbf{(15)}$$

$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \cdots + \Theta_Q B^{Qs} \quad \textbf{(16)}$$

O termo $(1 - B^s)^D$ realiza a diferenciação sazonal da série D vezes, removendo efeitos cíclicos com frequência *s*.

Dessa forma, o modelo SARIMA(p, d, q)(P, D, Q)s oferece uma estrutura robusta para capturar tanto as características dinâmicas locais quanto os padrões sazonais periódicos presentes em séries temporais reais (BOX; JENKINS; REINSEL, 2008), sendo amplamente empregado em aplicações como previsão de temperatura, vendas sazonais, demanda energética, entre outras.

3 - Materiais e Métodos

Nesta seção, serão apresentadas as etapas aplicadas no desenvolvimento do projeto, detalhando os processos e ferramentas utilizadas para garantir que os objetivos estabelecidos fossem atingidos.

Inicialmente, são apresentados o ambiente de desenvolvimento, em seguida são apresentados os dados, de onde foram extraídos e de que maneira, sua estrutura e processamentos aplicados.

Em seguida, são detalhados os modelos aplicados às topologias da LSTM, da RNN e da CNN treinadas para comparação no trabalho.

Depois da apresentação dos modelos, é abordado a aplicação do TL, e como foi feito para melhorar o treinamento e generalização do modelo escolhido. Por fim, são apresentadas as ferramentas e o ambiente computacional empregado, também a abordagem metodológica utilizada para comparação dos modelos e de seleção da melhor abordagem.

3.1 - Ambiente de Desenvolvimento

O projeto foi desenvolvido em um computador equipado com um processador Intel Core i5-11400, uma GPU NVIDIA RTX 3060 com 12GB de VRAM e 32GB de memória RAM. Esse hardware permitiu a execução eficiente dos modelos de aprendizado profundo, garantindo um treinamento mais rápido e facilitando o processamento de grandes volumes de dados.

As implementações foram realizadas em Python, utilizando bibliotecas como TensorFlow, Keras, statsmodels e scikit-learn para modelagem e treinamento dos algoritmos. Além disso, o ambiente de desenvolvimento incluiu o uso do Jupyter Notebook e VS Code para experimentação e depuração do código.

3.2 - Bibliotecas Utilizadas

Para o desenvolvimento do trabalho, algumas bibliotecas foram utilizadas para a manipulação de dados, para a modelagem dos sistemas preditivos e avaliação dos modelos.

A manipulação dos dados foi realizada utilizando as bibliotecas *Numpy* e *Pandas*, que facilitam operações matemáticas e a organização de grandes tabelas e conjuntos de dados, auxiliando no tratamento das séries temporais. Para a normalização das variáveis, utilizamos o *MinMaxScaler*, da biblioteca *Scikit-learn*, fazendo com que os dados estivessem na mesma escala antes da modelagem dos algoritmos.

Para a modelagem preditiva, foi utilizado aprendizado de máquina e estatística. As redes neurais, LSTM, RNN e CNN, foram implementadas utilizando *TensorFlow* e *Keras*, aproveitando as funcionalidades de facilitação para a construção e treinamento dos modelos. Além disso, para as análises estatísticas das séries temporais, o modelo SARIMAX foi feito utilizando a biblioteca *statsmodels*.

Por fim, para a visualização dos resultados, foi utilizada a *matplotlib*, que permite a criação de gráficos para interpretar as previsões dos modelos, e visualizar as séries temporais.

Essa combinação de bibliotecas possibilitou que o experimento fosse realizado com sucesso, possibilitando realizar todas as análises e modelagem com êxito.

3.3 – Métricas

A avaliação do desempenho dos modelos preditivos foi realizada por meio de quatro métricas: Erro Quadrático Médio (MSE), Erro Absoluto Médio (MAE), Coeficiente de Determinação (R^2) e Dynamic Time Warping (DTW). As três primeiras são amplamente utilizadas para avaliação de regressão. Já o DTW foi aplicado na validação dos resultados das redes em relação à série original, devido à sua capacidade de medir similaridade entre séries temporais com variações de fase ou ritmo.

3.3.1 Erro Quadrático Médio (MSE)

O MSE mede a média dos quadrados das diferenças entre os valores observados e previstos. Ele penaliza erros maiores de forma mais intensa, conforme a equação 17:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \tilde{y}_i)^2 \quad \textbf{(17)}$$

Onde y_i é o valor real, $\hat{y}_i$ é o valor previsto, e n é o número de observações.

3.3.2 Erro Absoluto Médio (MAE)

O MAE, descrito na equação 18, calcula a média dos valores absolutos das diferenças entre as observações e as previsões. É menos sensível a outliers do que o MSE:

$$MAE = \frac{1}{n}\sum_{i}^{n} |y_i - \hat{y}_i| \quad \textbf{(18)}$$

3.3.3 Coeficiente de Determinação (R^2)

O R^2 quantifica a proporção da variância total dos dados que é explicada pelo modelo. Sua fórmula é regida pela equação 19:

$$R^2 = 1 - \frac{[\sum_{i}^{n}(y_i - \hat{y}_i)^2]}{\sum_{i}^{n}(y_i - \bar{y})^2} \quad \textbf{(19)}$$

Onde $\bar{y}$ representa a média dos valores reais. Valores próximos de 1 indicam forte correlação entre os valores previstos e observados.

3.3.4 Dynamic Time Warping (DTW)

O DTW é uma métrica que mede a similaridade entre duas sequências temporais que podem não estar perfeitamente alinhadas no tempo. É útil em cenários onde atrasos ou variações temporais ocorrem entre as previsões e os valores reais.

A distância DTW é definida como:

$$DTW(A,B) = \min\{w \in W\}\left\{\sum_{(i,j)\in w} d(a_i, b_j)\right\} \textbf{(20)}$$

Sejam $A=(a_1,a_2,...,a_n)$ e $B=(b_1,b_2,...,b_n)$ as séries temporais a serem comparadas. A função $d(a_i,b_j)$ representa a distância entre os pontos a_i e b_j, geralmente calculada por meio da distância euclidiana. O caminho de alinhamento ótimo é representado por w, e W é o conjunto de todos os caminhos válidos possíveis. Essa métrica permite comparar séries mesmo quando apresentam deslocamentos temporais locais, oferecendo uma visão complementar às demais métricas tradicionais de erro.

3.4 - Coleta dos dados

Para a realização do trabalho proposto com o intuito de modelar um sistema de predição real, foi procurada uma fonte boa para um dataset “sujo” com dados de qualidade. Um dataset sujo é um dataset no qual os dados vão ter imperfeições e que exigem processamento antes de serem levados a treinamento, se aproximando mais de dados que poderiam ter sido coletados por sensores em caso de uma aplicação própria.

Para a obtenção dos dados meteorológicos, foi selecionada a base de dados do projeto Prediction Of Worldwide Energy Resources (POWER), uma iniciativa mantida pelo Langley Research Center (LaRC) da NASA. A plataforma disponibiliza séries temporais de diversas variáveis, como temperatura, umidade e radiação solar, derivadas de observações de satélite e modelos de reanálise que são ajustados com dados de estações de campo.

É fundamental destacar que os dados do POWER são fornecidos em formato de grade (grid), com uma resolução espacial de 0.5° x 0.5°, o que equivale a uma área de aproximadamente 55 km por 55 km na linha do Equador. Por essa razão, os valores representam uma média sobre uma área considerável e podem apresentar desvios quando comparados a

medições pontuais realizadas por estações meteorológicas em solo. Apesar dessa limitação, o projeto foi escolhido devido à sua vasta cobertura global, à consistência temporal da série histórica e à facilidade de acesso, características essenciais para o treinamento de modelos de aprendizado de máquina (NASA, 2025).

A plataforma de acesso aos dados do projeto POWER da NASA apresenta uma interface web intuitiva para a consulta e extração de dados, conforme ilustrado na Figura 7. A interface permite que o usuário selecione coordenadas geográficas diretamente em um mapa interativo do globo terrestre e especifique os parâmetros de interesse para a requisição. Além da interface visual, o projeto disponibiliza uma API REST (Representational State Transfer), que possibilita a automação de requisições personalizadas via código, facilitando a extração de dados em larga escala para quaisquer coordenadas geográficas.

Figura 7- Interface NASA POWER LARC

Fonte: Autoria Própria

No trabalho, foi utilizada a API disponibilizada no para fazer a solicitação das séries temporais em .CSV (Comma Separated Values). Utilizando o módulo *request* do *python* foi desenvolvido um *script* que solicita o arquivo com uma série temporal com um determinado número de dias com as seguintes variáveis:

Tabela 1- Lista de parâmetros utilizados para treinamento

Parâmetro	Descrição (Inglês)	Descrição (Português)
QV2M	Specific Humidity at 2 Meters	Umidade Específica a 2 Metros
RH2M	Relative Humidity at 2 Meters	Umidade Relativa a 2 Metros
PRECTOTCORR	Precipitation Corrected	Precipitação Corrigida
PS	Surface Pressure	Pressão na Superfície
WS2M	Wind Speed at 2 Meters	Velocidade do Vento a 2 Metros
WS2M_MAX	Maximum Wind Speed at 2 Meters	Velocidade Máxima do Vento a 2 Metros
WS2M_MIN	Minimum Wind Speed at 2 Meters	Velocidade Mínima do Vento a 2 Metros
WS2M_RANGE	Wind Speed Range at 2 Meters	Variação da Velocidade do Vento a 2 Metros
WD2M	Wind Direction at 2 Meters	Direção do Vento a 2 Metros
WS10M	Wind Speed at 10 Meters	Velocidade do Vento a 10 Metros
WS10M_MAX	Maximum Wind Speed at 10 Meters	Velocidade Máxima do Vento a 10 Metros
WS10M_MIN	Minimum Wind Speed at 10 Meters	Velocidade Mínima do Vento a 10 Metros
WS10M_RANGE	Wind Speed Range at 10 Meters	Variação da Velocidade do Vento a 10 Metros
WD10M	Wind Direction at 10 Meters	Direção do Vento a 10 Metros
T2M_RANGE	Temperature Range at 2 Meters	Variação da Temperatura a 2 Metros
T2M_MAX	Maximum Temperature at 2 Meters	Temperatura Máxima a 2 Metros
T2M_MIN	Minimum Temperature at 2 Meters	Temperatura Mínima a 2 Metros
T2M	Temperature at 2 Meters	Temperatura a 2 Metros
T2MDEW	Dew Point Temperature at 2 Meters	Temperatura do Ponto de Orvalho a 2 Metros

Fonte: Autoria própria

Buscou-se garantir a cobertura de diferentes dimensões físicas do sistema climático, incluindo variáveis relacionadas à temperatura, umidade, radiação solar e vento, a fim de fornecer ao modelo um conjunto de informações abrangente e diversificado. A escolha final foi consolidada pela disponibilidade de séries históricas longas e consistentes para estes parâmetros na plataforma NASA POWER.

A obtenção desses dados foi essencial para o desenvolvimento deste estudo, fornecendo informações detalhadas sobre as diversas variáveis meteorológicas relevantes para a previsão de temperatura. Com os arquivos exportados via API, foi possível estruturar a base de dados utilizada na modelagem preditiva, garantindo um histórico confiável e consistente para a aplicação de técnicas de aprendizado de máquina e estatísticas.

3.5 - Topologia dos modelos

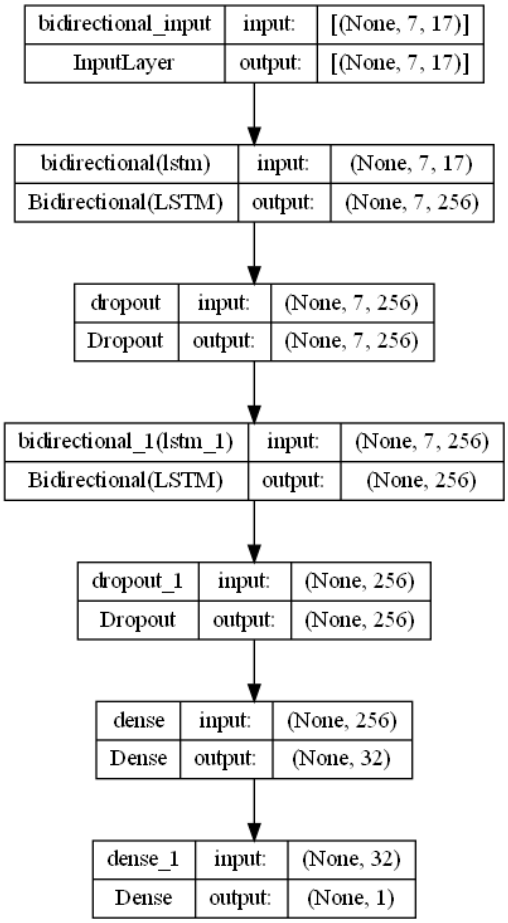
Neste trabalho, foram utilizados quatro modelos de redes neurais, LSTM, RNN, CNN e GRU. Como abordado no Capítulo 2, esses diferentes modelos têm alta capacidade preditiva com diferentes particularidades para cada um deles. Como modelo de referência, foi utilizado o SARIMA, uma abordagem estatística para a previsão de séries temporais multivariadas.

Os modelos baseados em redes neurais (LSTM, RNN, CNN e GRU) foram implementados utilizando a biblioteca *TensorFlow/Keras*. Diversas camadas são utilizadas, cada uma com o seu propósito dentro do mecanismo preditivo, dentre elas estão descritas as camadas recorrentes, camadas bidirecionais, camadas convolucionais, camadas densas e camadas de regularização.

As camadas recorrentes foram utilizadas na LSTM, na RNN e na GRU, para capturar padrões temporais nas séries. A camada LSTM foi escolhida por conta de sua capacidade de aprender e armazenar informações de longo prazo, superando os problemas de desvanecimento do gradiente, como comentado no Capítulo 2. Essa camada é composta por células de memória que controlam seletivamente a retenção e o esquecimento de informações ao longo tempo.

Para aprimorar a captação de padrões complexos na LSTM, foi implementada a camada LSTM bidirecional, cuja topologia é detalhada na Figura 8. Diferente da camada LSTM convencional, ela permite que a rede interprete a série temporal em ambas as direções. Melhorando a capacidade do modelo em compreender relações temporais mais amplas, capturando padrões que podem não ser perceptíveis ao considerar apenas uma direção do tempo.

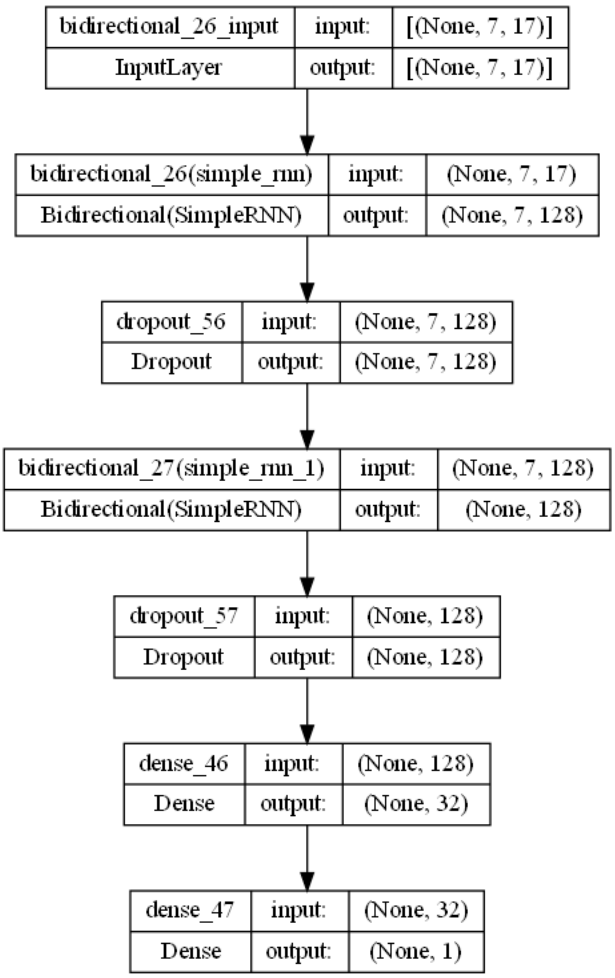
Figura 8- Topologia do modelo da LSTM utilizada para treinamento



Fonte: Autoria própria

Em alternativa ao primeiro modelo, foi testado o modelo baseado em RNN, detalhado na Figura 9, que utiliza uma abordagem mais simples e menos custosa computacionalmente, adequada para identificar dependências temporais curtas.

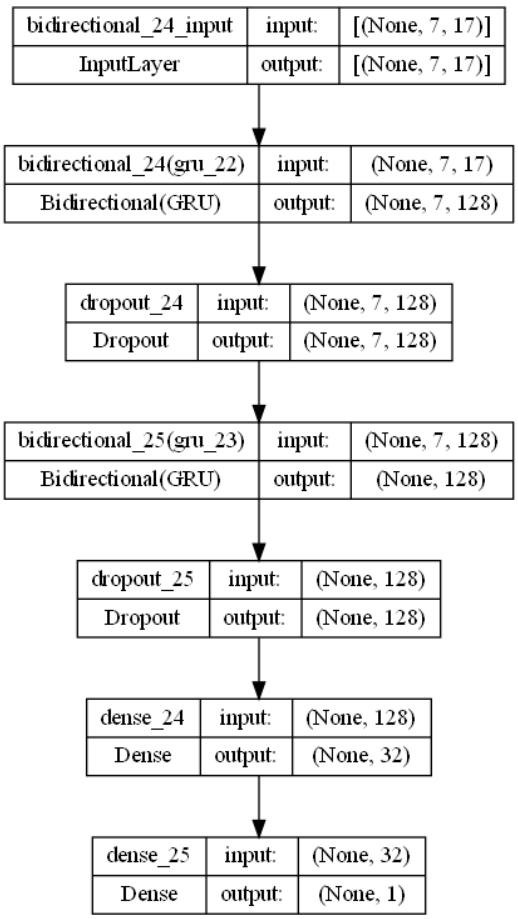
Figura 9- Topologia do modelo da RNN utilizada para treinamento



Fonte: Autoria própria

Em contraponto às duas abordagens de redes recorrentes, foi utilizada a GRU (Figura 10), uma rede semelhante à topologia da LSTM, porém com células de estruturas diferentes, mais simples, que promete ser mais eficiente computacionalmente.

Figura 10- Topologia do modelo da GRU utilizada para treinamento



Fonte: Autoria própria

Além das redes recorrentes, foi testada uma abordagem baseada em redes neurais convolucionais (CNN) para a extração de padrões nos dados temporais, conforme ilustrado na Figura 11. A camada Conv1D utilizada aplica filtros convolucionais, permitindo a detecção de variações sutis nas séries temporais. Juntamente das camadas convolucionais, são aplicadas camadas de *pooling*, responsáveis por diminuir a dimensão dos dados, mantendo as características mais relevantes e melhorando a eficiência do modelo.

Figura 11- Topologia modelo CNN utilizado para treinamento

Fonte: Autoria própria

Após as camadas densas e recorrentes, foram adicionadas densas (*fully connected layers)* para processar as informações extraídas e gerar a previsão final. Nessas camadas, cada neurônio é completamente conectado aos neurônios da camada anterior, permitindo a combinação de diferentes padrões aprendidos. A ativação ReLU (Rectified Linear Unit) foi utilizada para introduzir não linearidade e ampliar a capacidade de aprendizado. A última camada possui somente um neurônio e ativação linear, e é responsável por realizar a previsão.

Durante o treinamento de modelos de aprendizado profundo, é comum observar o fenômeno do *overfitting*, que ocorre quando o modelo aprende excessivamente os detalhes presentes nos dados de treinamento, comprometendo sua capacidade de generalização para dados novos. Em outras palavras, o modelo se ajusta tão bem ao conjunto de treinamento que perde desempenho em dados nunca vistos.

Para mitigar esse problema, aplicam-se técnicas de regularização, cujo objetivo é restringir a complexidade do modelo, favorecendo uma aprendizagem mais generalizável. Uma técnica bastante utilizada é o *Dropout*, introduzido por Srivastava et al. (2014), que consiste em desativar aleatoriamente um conjunto de neurônios durante o treinamento. Isso força a rede a não depender exclusivamente de determinados caminhos ou combinações de neurônios.

Outra técnica importante é a *Batch Normalization*, proposta por Ioffe e Szegedy (2015), que normaliza as ativações das camadas intermediárias em cada *minibatch*. Além de ajudar na estabilização e aceleração do treinamento, essa técnica também atua como uma forma de regularização, reduzindo a sensibilidade do modelo a pequenas variações nos dados de entrada.

Os modelos foram treinados utilizando a função de perda *Mean Squared Error* (MSE), utilizado em problemas de regressão, pois penaliza erros de forma significativa.

Como otimizador, foi escolhido o Adam (*Adaptive Moment Estimation*), que combina métodos de descida de gradiente com momento e ajuste adaptativo da taxa de aprendizado, garantindo um treinamento eficiente.

3.5 - Transfer Learning

Como explicado no Capítulo 2, o TL foi aplicado para aproveitar o conhecimento aprendido por um dos modelos previamente treinados, e treinar novamente de modo a acelerar o aprendizado em um ambiente diferente e melhorar a generalização do modelo.

No experimento, foi selecionado o melhor modelo entre os testados (LSTM, RNN, CNN e GRU). Para aplicar TL, as primeiras camadas da rede foram congeladas utilizando a própria biblioteca do Keras, impedindo que os seus pesos fossem atualizados, enquanto as camadas finais permaneceram treináveis. Assim, o modelo mantém a base de treinamento já aprendida e ajusta apenas os últimos estágios da rede para refinar as previsões.

O treinamento foi realizado utilizando a função de perda Mean Squared Error (MSE) e o otimizador Adam, seguindo a mesma configuração inicial dos outros modelos. Essa abordagem possibilitou a adaptação do modelo a novos dados sem a necessidade de um treinamento do zero, reduzindo o tempo computacional e melhorando a capacidade preditiva.

3.7 - Forecasting

Para garantir um aprendizado adequado dos padrões temporais, a previsão foi feita utilizando uma abordagem de janelas deslizantes (sliding window), onde os últimos 7 dias de dados foram utilizados para prever a temperatura do dia seguinte.

Para os modelos de redes neurais (LSTM, GRU, RNN e CNN), a entrada do modelo consiste em um vetor de características multivariadas ao longo de um período fixo de tempo, denominado janela temporal. Esse vetor é composto por fatores meteorológicos relevantes, incluindo umidade, precipitação, temperatura do ponto de orvalho, pressão, velocidade e direção do vento, entre outros.

A previsão foi feita em horizonte de um passo à frente (one-step ahead forecast), onde o modelo retorna a temperatura esperada para o próximo instante de tempo. Os valores previstos foram comparados com os valores reais, utilizando métricas como Erro Quadrático Médio (MSE), Erro Médio Absoluto (MAE) e Coeficiente de Determinação (R^2). Além disso, foi

utilizada a Distância Dinâmica de Tempo (DTW) para avaliar similaridades entre as sequências temporais previstas e observadas.

Para avaliar o desempenho dos modelos, os dados foram divididos em conjuntos de treinamento e validação, com a proporção de 60% e 40% respectivamente, garantindo que os modelos fossem testados em dados nunca antes vistos. A divisão foi feita de forma temporal, sem embaralhamento, para preservar a estrutura sequencial dos dados.

Por fim, para validar a robustez dos modelos, foram realizados múltiplos treinamentos, e os resultados foram analisados estatisticamente para garantir a estabilidade das previsões.

4 - Resultados e Discussões

Essa seção apresenta e discute todos os resultados dos experimentos e desenvolvimento do trabalho, mostrando todos os resultados dos treinamentos realizados, comparando os resultados e discutindo a respeito.

São apresentados os resultados de treinamento das redes LSTM, RNN, GRU e CNN, comparando os resultados obtidos e escolhendo a que melhor performa para a aplicação do Transfer Learning. Por fim, são comparados os resultados de algumas aplicações de Transfer Learning contra um modelo estatístico SARIMA.

4.1 - Conjuntos de Dados

A seleção das variáveis foi baseada na relevância para a previsão da temperatura a 2 metros. O conjunto de características preditivas inclui variáveis relacionadas à umidade, precipitação, pressão, temperatura, vento e suas respectivas variações. A variável-alvo é a temperatura a 2 metros.

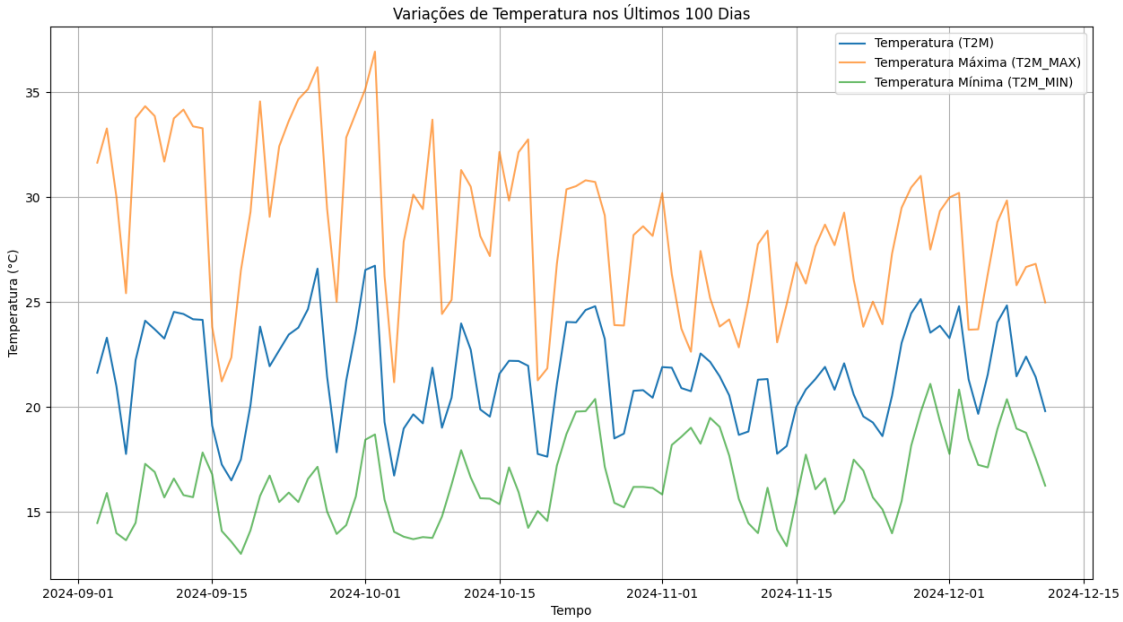
O dataset foi extraído do site NASA POWER LARC e abrange um período de 24 anos (2000 - 2024) nas cidades de São Paulo - SP/Brasil e Calgary - AB/Canadá. Antes da aplicação nos modelos, os dados passaram por um tratamento que incluiu uma etapa de limpeza essencial.

Nesse processo, foi aplicado um threshold (limiar técnico) para remover outliers — valores que fogem ao padrão e são considerados erros de leitura dos sensores ou "lixos" gerados no processamento. Embora exista a possibilidade de um valor extremo ser um evento real, a remoção de dados flagrantemente anômalos é fundamental para não comprometer o resultado preditivo do modelo.

Essa etapa garante que o modelo seja treinado com mais confiança, aprendendo com um conjunto de dados mais fiel à realidade e, consequentemente, gerando previsões mais precisas.

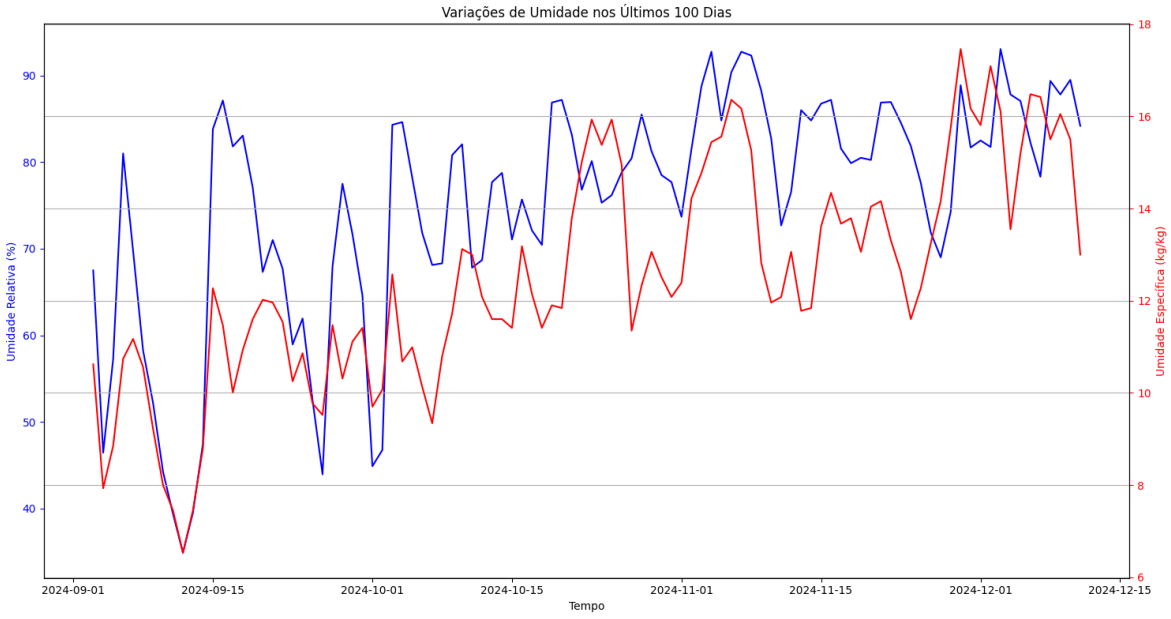
As figuras 12 a 21 ilustram os últimos 100 dias dos dados considerados no nosso *dataset*:

Figura 12 - Temperatura Máxima, Mínima e Média (ºC) a 2 metros de altitude em São Paulo



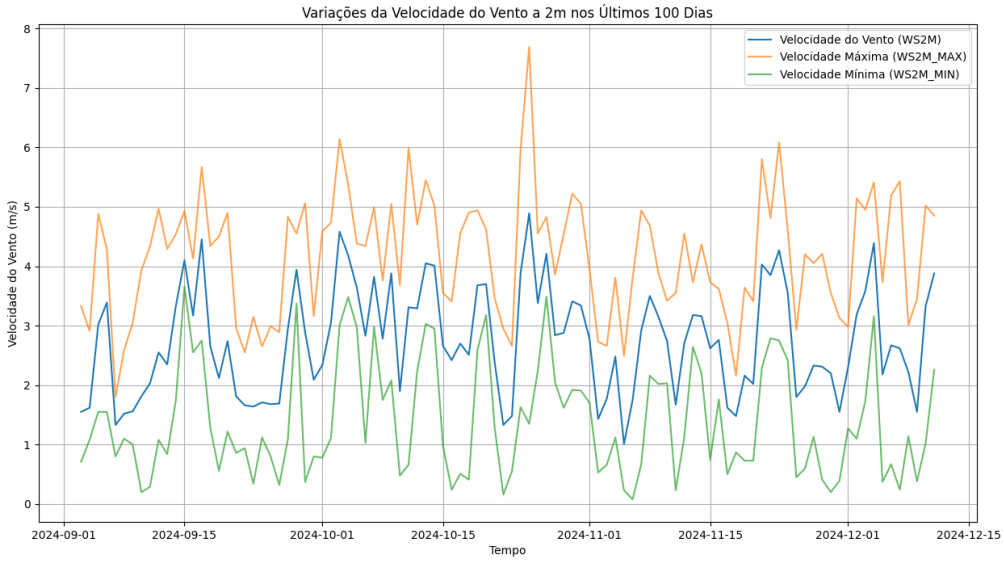
Fonte: Autoria própria

Figura 13 - Umidade relativa do ar (%) e umidade específica do ar (kg /kg) a 2 metros de altitude em São Paulo



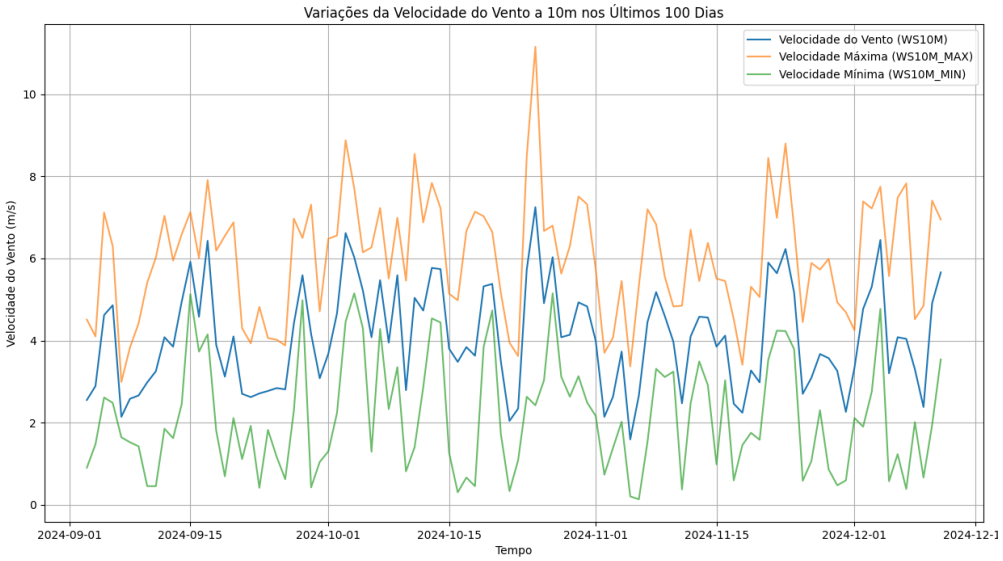
Fonte: Autoria própria

Figura 14 - Velocidades máxima, mínima e média do vento (m/s) a 2 metros de altitude em São Paulo



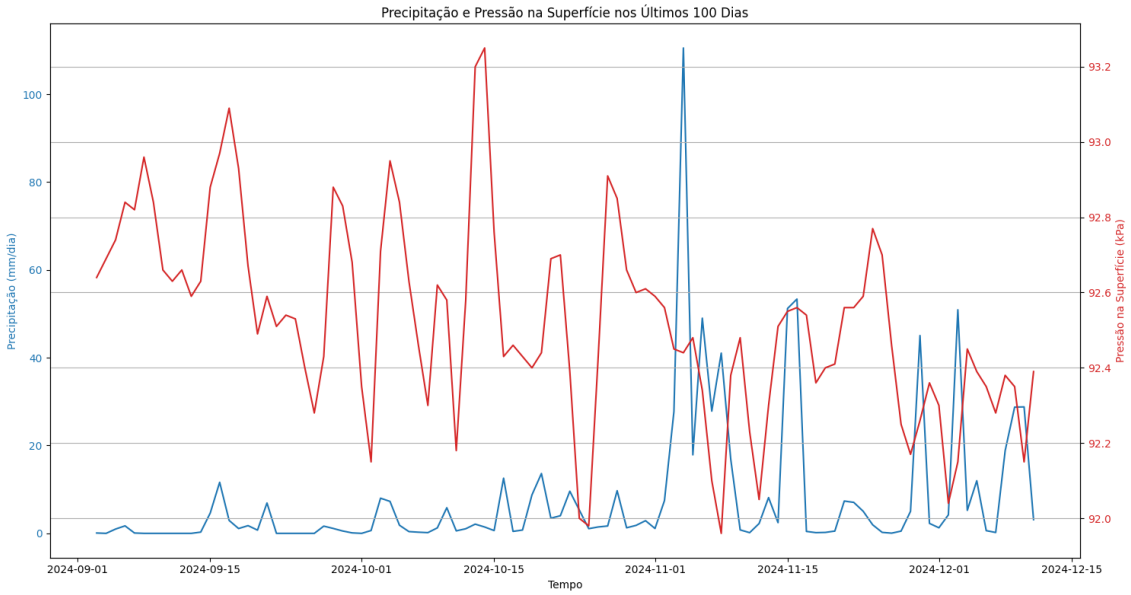
Fonte: Autoria própria

Figura 15 - Velocidades máxima, mínima e média do vento (m/s) a 10 metros de altitude em São Paulo



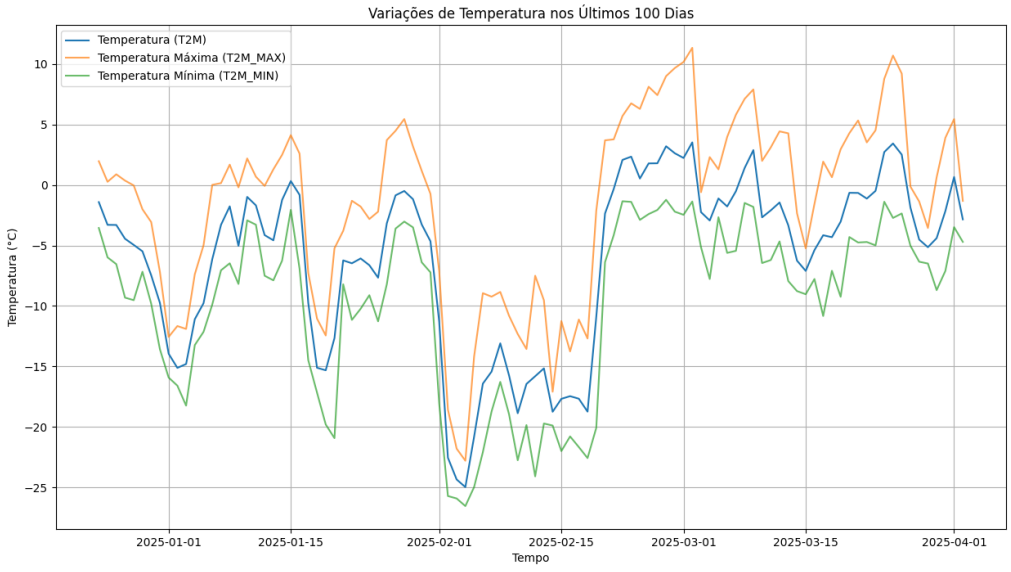
Fonte: Autoria própria

Figura 16 - Precipitação (mm) e pressão atmosférica (kPa) em São Paulo



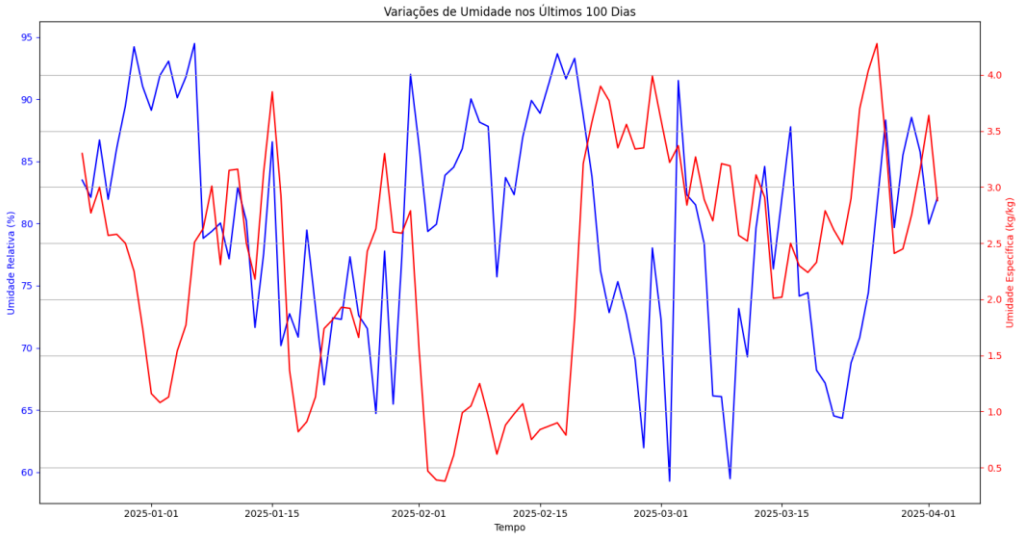
Fonte: Autoria própria

Figura 17 - Temperatura Máxima, Mínima e Média (ºC) a 2 metros de altitude em Calgary



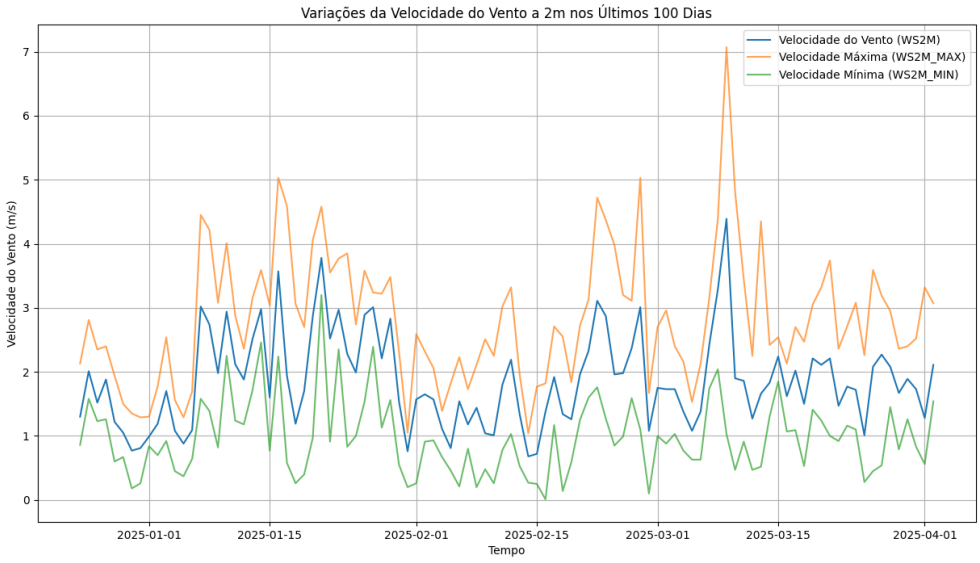
Fonte: Autoria própria

Figura 18 - Umidade relativa do ar (%) e umidade específica do ar (kg /kg a 2 metros de altitude em Calgary



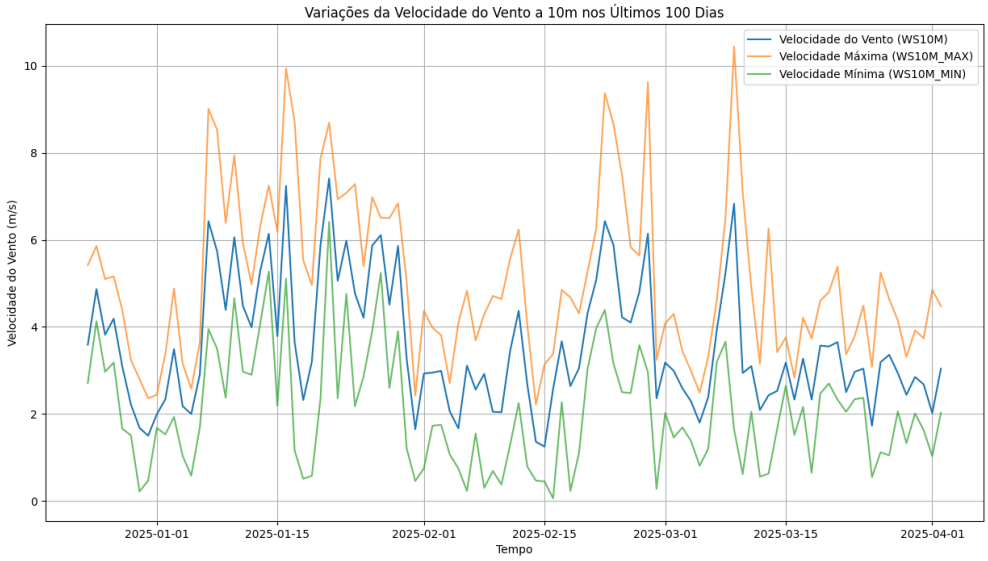
Fonte: Autoria própria

Figura 19 - Velocidades máxima, mínima e média do vento (m/s) em 2 metros de altitude em Calgary



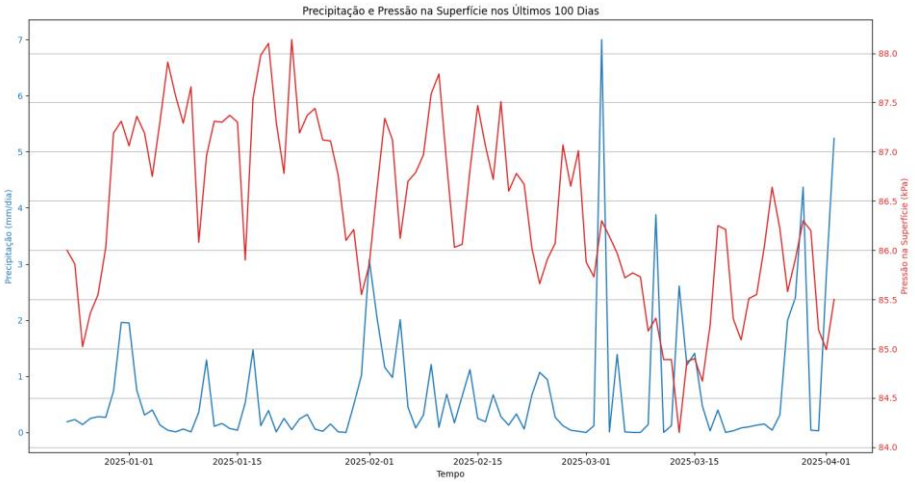
Fonte: Autoria própria

Figura 20 - Velocidades máxima, mínima e média do vento (m/s) a 10 metros de altitude em Calgary



Fonte: Autoria própria

Figura 21 – Precipitação (mm) e pressão atmosférica (kPa) em Calgary



Fonte: Autoria própria

Após o tratamento dos dados brutos, os dados foram normalizados utilizando a técnica de *Min-Max Scaling*, que reescalou os valores para o intervalo entre 0 e 1. Esse processo foi aplicado em todas as variáveis preditores (X) e à variável alvo (y), de forma a garantir que diferenças de escala não impactam os modelos.

Para o desenvolvimento dos modelos, o dataset foi segmentado em 60% para treinamento e 40% para validação. Essa proporção foi definida buscando um equilíbrio entre

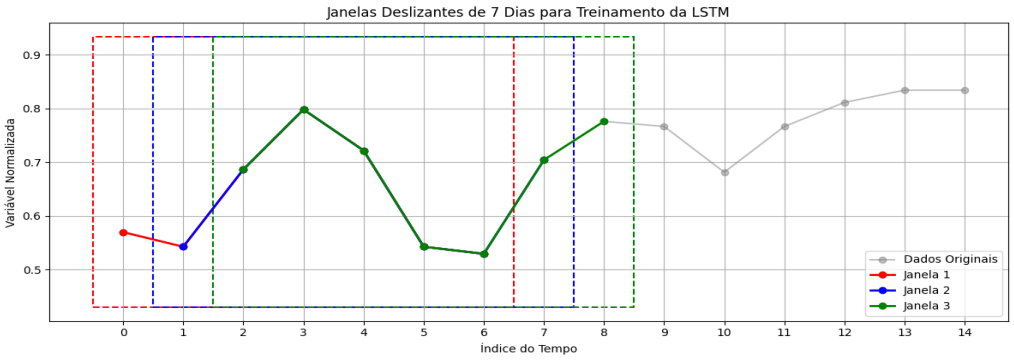
fornecer ao modelo um volume de dados suficiente para o aprendizado dos padrões e, ao mesmo tempo, permitir uma avaliação de desempenho abrangente sobre um conjunto de dados futuros. A fim de preservar a dependência temporal, a divisão foi realizada mantendo a ordem cronológica dos dados.

É importante ressaltar uma limitação inerente à abordagem de usar os 60% iniciais dos dados para treino e os 40% finais para validação. A divisão cronológica de uma série temporal longa (2000-2024) pode introduzir desafios relacionados a mudanças nos padrões climáticos ao longo dos anos.

Os impactos ambientais recentes podem fazer com que os dados de validação tenham características estatísticas diferentes dos dados de treino, um fenômeno conhecido como concept drift ou não estacionariedade. Para atenuar esse efeito em trabalhos futuros, poderiam ser exploradas técnicas como o retreinamento periódico do modelo com dados mais recentes ou a inclusão de variáveis que ajudem o modelo a capturar tendências de longo prazo

Para treinar os modelos foram criadas sequências temporais utilizando uma janela de 7 dias, com a intenção de fazer os modelos serem capazes de prever a temperatura dentro dessa janela. As sequências foram geradas a partir dos dados já normalizados.

Figura 22 - Exemplo de como são criadas as sequências temporais



Fonte: Autoria própria

4.2 - Treinamento dos Modelos

Para o treinamento dos modelos de redes neurais (LSTM, RNN, GRU e CNN), foram utilizados os dados históricos climáticos da cidade de São Paulo - SP previamente tratados e normalizados. O processo de treinamento foi conduzido utilizando as bibliotecas comentadas no capítulo 3 (*TensorFlow/Keras*), e os hiper parâmetros foram ajustados empiricamente para otimizar o desempenho dos modelos como já representados no capítulo anterior.

A fim de garantir a robustez estatística e mitigar os efeitos da inicialização aleatória de pesos, cada arquitetura de modelo foi treinada 10 vezes de forma independente. Dessa forma,

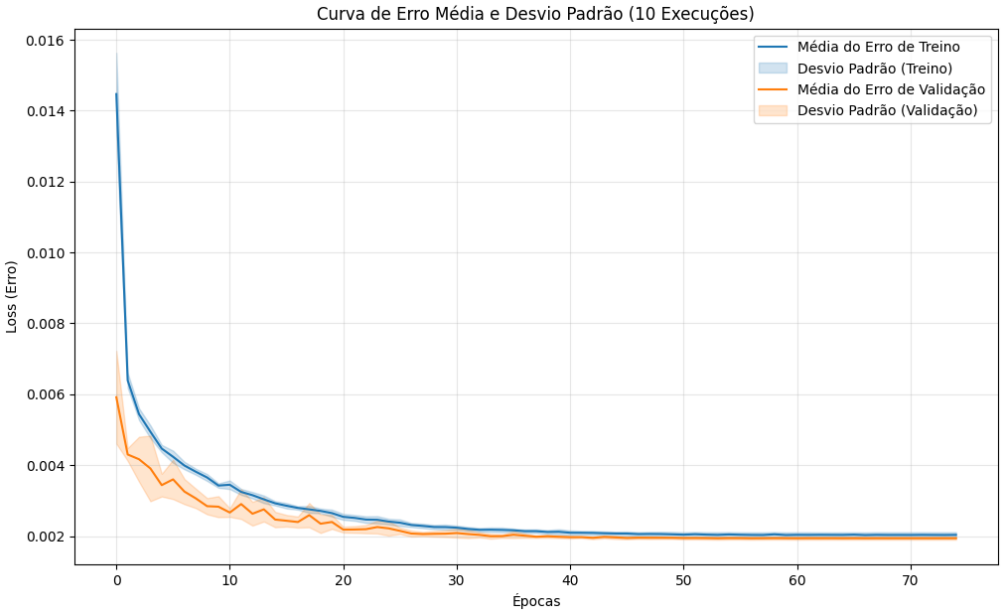
o desempenho final considerado representa uma média dos resultados, oferecendo uma medida mais confiável da capacidade de generalização do modelo. Os treinamentos foram configurados para um máximo de 100 épocas, valor que funciona como um limite superior para o processo, sendo o número efetivo de épocas controlado pelo critério de parada antecipada.

Para evitar o *overfitting*, foi empregado o critério de parada antecipada (*early stopping*) com uma "paciência" de 10 épocas. Essa configuração interrompe o treinamento caso a métrica de validação não apresente melhora por 10 épocas consecutivas. A escolha por esta paciência é uma prática comum que busca um balanço entre evitar uma interrupção prematura, devido a flutuações no desempenho, e parar o treinamento de forma eficiente quando o modelo atinge sua melhor performance. Cada modelo foi treinado 10 vezes utilizando a função de perda MSE. Além disso, foi empregado o otimizador Adam devido a sua eficiência de adaptação à taxa de aprendizado. Os treinamentos individuais são realizados de maneira iterativa, com um número máximo de épocas, e ao validar o desempenho, podendo parar antes o treinamento caso não avance.

O critério de parada antecipada (*early stopping*) foi utilizado para evitar o *overfitting*, interrompendo o treinamento quando a métrica de validação não melhorava após 10 épocas consecutivas.

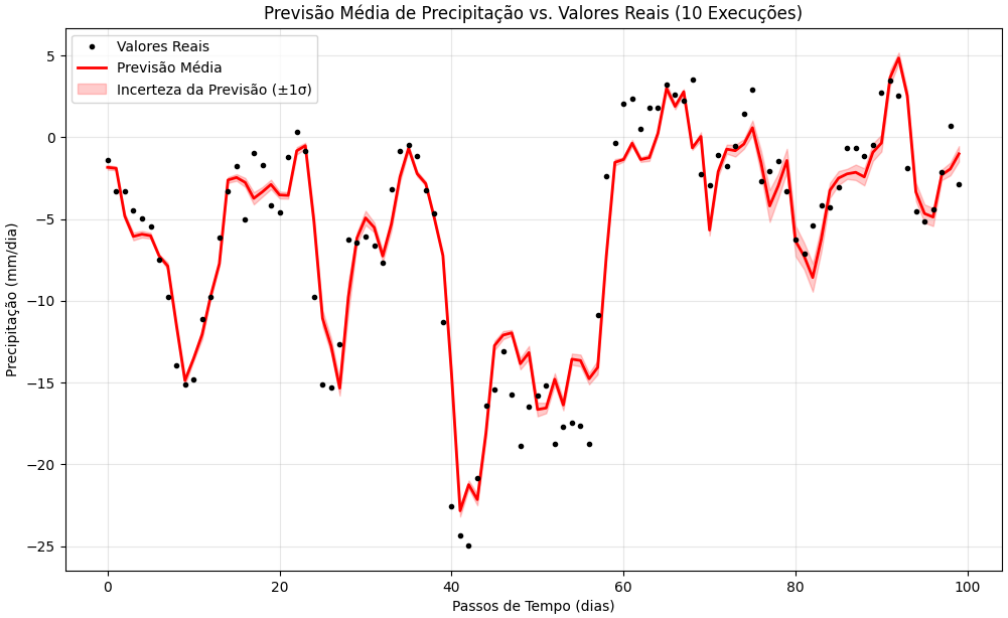
As curvas de erro (Treinamento vs Validação) encontradas para os treinamentos realizados foram as seguintes:

Figura 23- Curvas de erro Treinamento x Validação da LSTM



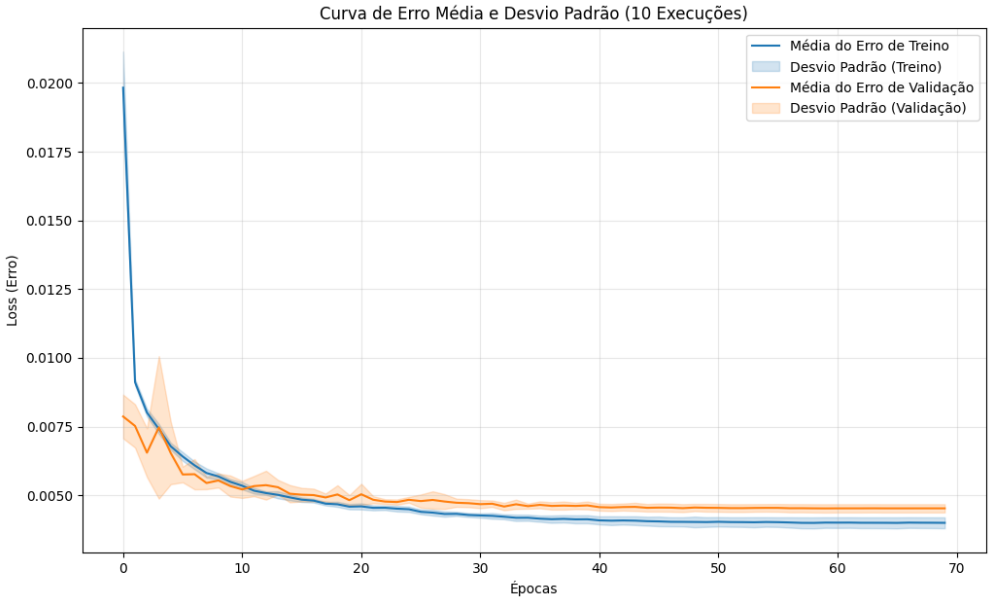
Fonte: Autoria própria

Figura 24- Comparação das previsões da LSTM contra os dados reais para os 100 primeiros dias



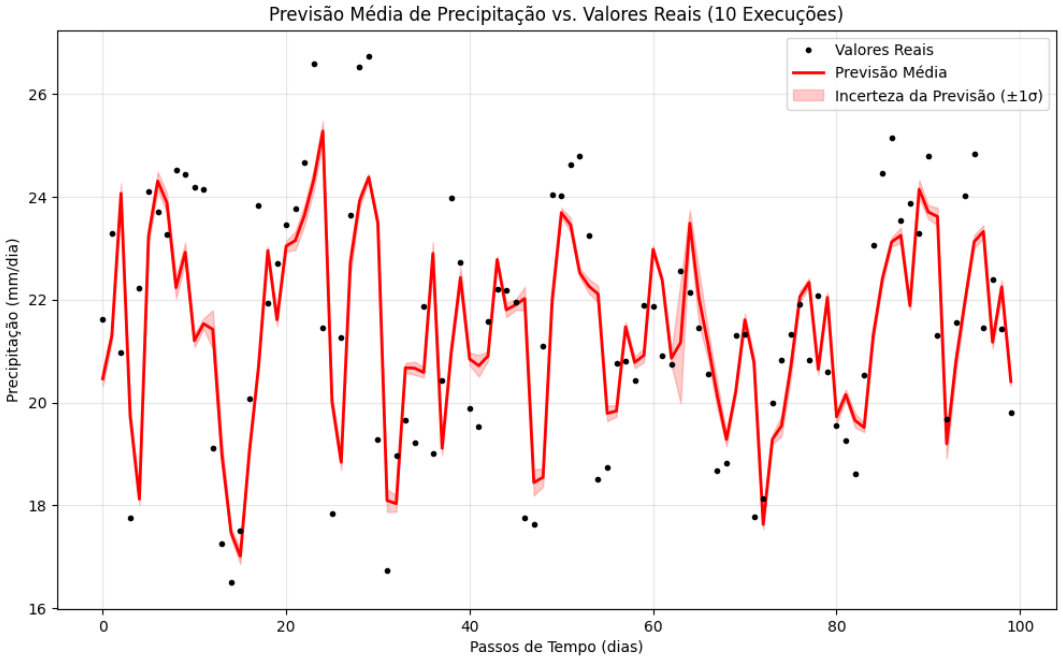
Fonte: Autoria própria

Figura 25- Curvas de erro Treinamento x Validação da GRU



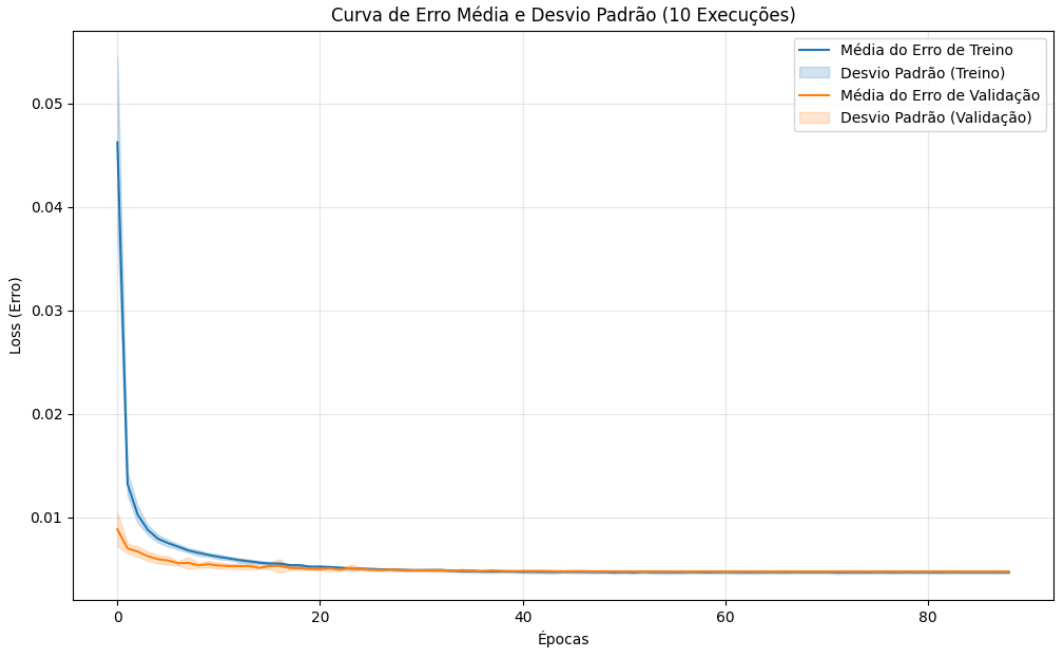
Fonte: Autoria própria

Figura 26- Comparação das previsões da GRU contra os dados reais para os 100 primeiros dias



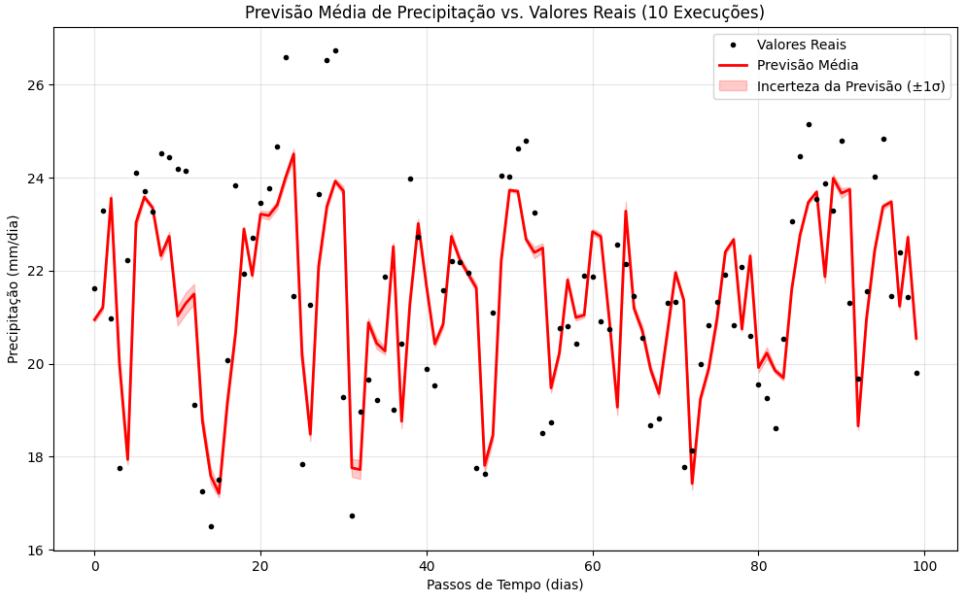
Fonte: Autoria própria

Figura 27- Curvas de erro Treinamento x Validação da RNN



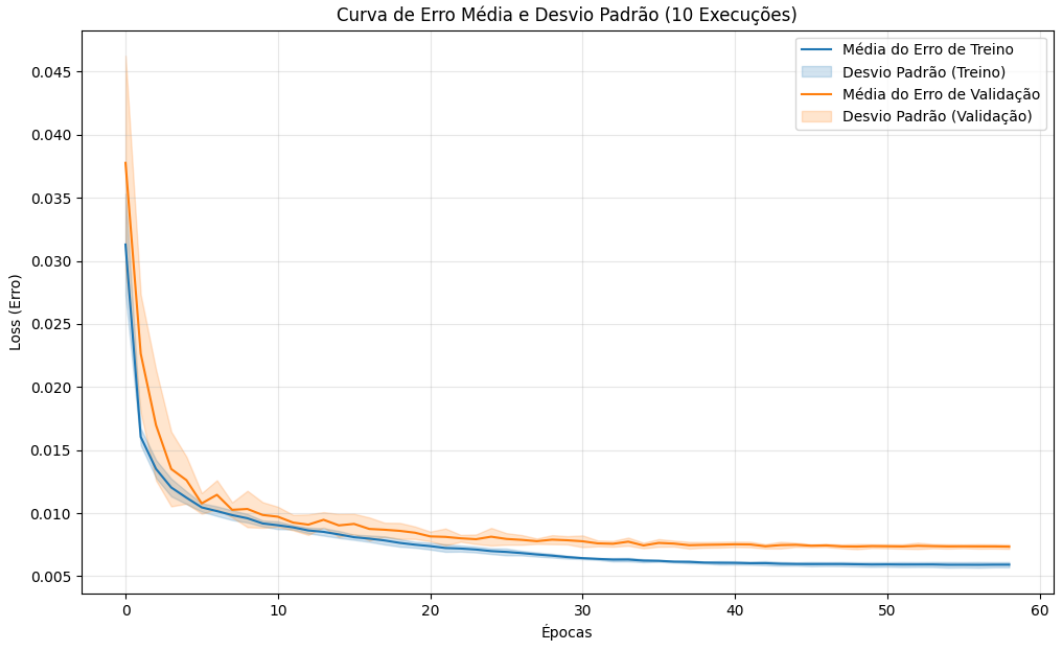
Fonte: Autoria própria

Figura 28 - Comparação das previsões da RNN contra os dados reais para os 100 primeiros dias



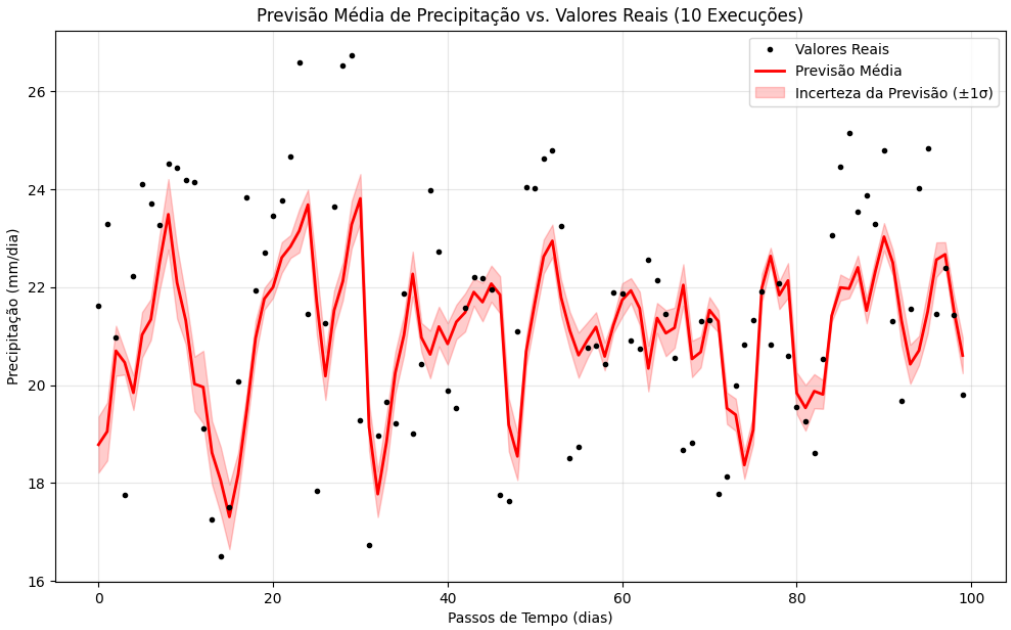
Fonte: Autoria própria

Figura 29- Curvas de erro Treinamento x Validação da CNN



Fonte: Autoria própria

Figura 30 - Comparação das previsões da CNN contra os dados reais para os 100 primeiros dias



Fonte: Autoria própria

As curvas de erro apresentadas nas Figuras 25, 27, 29 e 30 mostram que os modelos convergem de maneira adequada durante o treinamento. Como é esperado, foi observada uma queda acentuada do erro nas primeiras épocas, seguida de uma estabilização, indicando que os modelos foram capazes de aprender os padrões relevantes dos dados.

Nesse contexto, o critério de parada antecipada mostrou-se essencial para o controle do overfitting. A necessidade desta técnica justifica-se porque, após certo ponto, o erro no conjunto de validação começa a aumentar, mesmo que o erro de treinamento continue a diminuir. Esse fenômeno indica que o modelo parou de aprender padrões generalizáveis e começou a memorizar ruídos específicos dos dados de treino. Ao interromper o processo no momento de menor erro de validação, a parada antecipada garante que o modelo final seja aquele com a melhor capacidade de generalização para dados não vistos.

Ao analisar as curvas de cada modelo treinado, percebe-se que a GRU e a LSTM apresentam uma maior capacidade de generalização, por conta de sua proximidade entre as curvas de treinamento e de validação. A curva observada na CNN apesar de ainda estar próxima, apresenta maior divergência em relação aos outros modelos apresentados, o que pode explicar seu desempenho inferior em relação às redes recorrentes.

Nos gráficos das previsões em comparação com os valores do *dataset* é possível observar que todos os modelos foram capazes de capturar adequadamente as tendências dos dados de temperatura. Porém, são observadas nos indicadores inferiores no grau de aderência

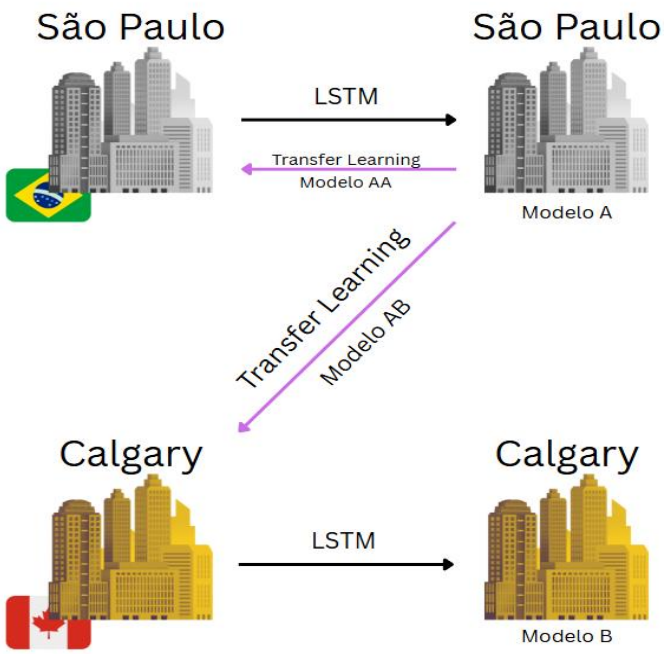
às curvas reais: enquanto a LSTM e a GRU mantêm previsões mais próximas aos valores observados. Já a RNN, apesar de capturar o padrão geral, mostra maior variabilidade entre as execuções, o que sugere menor robustez em comparação aos outros modelos recorrentes.

4.3 - Avaliação Quantitativa dos Modelos

Por fim, foram obtidos e analisados os resultados com a aplicação dos modelos de aprendizado profundo e do método de TL para previsão de temperatura em diferentes domínios geográficos. As métricas avaliadas foram o MAE, a MSE, o R^2, a métrica DTW, além do tempo médio de execução de cada modelo. As análises foram divididas em três partes principais: comparação entre modelos base, TL no mesmo domínio e TL entre domínios distintos.

As comparações serão feitas da forma como é apresentado no esquemático na figura 31:

Figura 31- Esquemático das comparações que serão feitas na análise



Fonte: Autoria própria

A estratégia de análise deste trabalho foi desenhada para avaliar a eficácia da técnica de TL na previsão de séries temporais de temperatura, conforme ilustrado no esquema da Figura 31.

A metodologia se inicia com o estabelecimento de dois modelos de referência (baseline) treinados de forma convencional: o Modelo A, utilizando exclusivamente dados de São Paulo, e o Modelo B, utilizando apenas dados de Calgary. O desempenho destes modelos serve como

um ponto de partida para medir os ganhos obtidos com os modelos que empregam a transferência de conhecimento.

O experimento principal, denominado Modelo AB, consiste em pré-treinar um modelo com dados de São Paulo (domínio fonte) e, posteriormente, transferir e ajustar esse conhecimento para a tarefa de previsão em Calgary (domínio alvo).

Adicionalmente, um modelo de controle, o Modelo AA, foi desenvolvido aplicando o mesmo processo de transferência dentro do próprio domínio de São Paulo, visando validar a metodologia.

Portanto, a análise comparativa entre o Modelo B (treinado do zero em Calgary) e o Modelo AB (com conhecimento de São Paulo) permitirá quantificar o impacto e os benefícios da aplicação de transfer learning entre localidades com características climáticas distintas.

Os resultados médios de desempenho estão descritos na Tabela 2. Entre os modelos avaliados, a LSTM obteve os melhores desempenhos em termos de MAE (1,038 °C), MSE (1,945 (°C)²) e R^2 (0,825), superando as demais abordagens. O modelo RNN apresentou desempenho próximo, porém inferior, enquanto a CNN, apesar de apresentar o menor tempo médio de execução (48,56 s), teve o pior desempenho preditivo, com um MAE de 1,355 °C e um R^2 de apenas 0,722.

Além dos modelos base de aprendizado profundo, foi avaliado o modelo estatístico SARIMA como baseline. Seus resultados foram consideravelmente inferiores aos das redes neurais, especialmente em termos de R^2 (0,7605), indicando menor capacidade explicativa da variância dos dados. A DTW do SARIMA (14,7036 °C) foi substancialmente mais alta, sugerindo que sua previsão tem menos similaridade temporal com os dados reais em comparação com os modelos neurais.

Tabela 2 - Comparação de resultados de treinamento

	GRU	**CNN**	**LSTM**	**RNN**	**SARIMA**
DTW (ºC)	2,32979	3,18767	**2,25810**	2,30648	14,7036
MAE (ºC)	1,05179	1,35482	**1,03817**	1,06526	1,3823
MSE (ºC²)	1,98155	3,08546	**1,94506**	2,06716	2,5877
R2	0,82173	0,72243	**0,82492**	0,81393	0,7605
Tempo médio (s)	122,714794	**48,56331**	130,665905	213,72730	-

Fonte: Autoria própria

Esse resultado reforça a eficácia da arquitetura LSTM na modelagem de séries temporais meteorológicas, possivelmente devido à sua capacidade de capturar dependências de longo prazo nos dados. A métrica DTW também foi menor para a LSTM (2,258°C), indicando uma boa correspondência temporal entre a série prevista e a real. O tempo de execução da LSTM, embora mais elevado que o da CNN, foi inferior ao da RNN, o que sugere um bom equilíbrio entre custo computacional e precisão.

Na sequência, foi aplicada a técnica de Transfer Learning dentro do mesmo domínio, ou seja, a rede LSTM pré-treinada com dados de São Paulo foi refinada com os mesmos dados da cidade de São Paulo. O objetivo foi verificar se a transferência de conhecimento poderia preservar ou melhorar o desempenho, mesmo sem alterar o domínio geográfico. Os resultados obtidos foram: MAE de 1,0907 °C, RMSE de 1,4743 °C, R^2 de 0,8267 e DTW de 1,1404°C. Em comparação com o Modelo A (MAE de 1,0381 °C e R^2 de 0,8249), observa-se que o desempenho foi bastante similar, com aumento do MAE em 5,06% e leve melhora no R^2 de 0,22%. A métrica DTW, no entanto, apresentou melhora significativa, caindo de 2,258°C para 1,140°C, o que sugere que o modelo transferido teve melhor alinhamento temporal com a série real.

Apesar de não serem melhoras significativas, tais resultados indicam que o uso de TL em um mesmo domínio pode ser vantajoso do ponto de vista computacional, permitindo o reaproveitamento de modelos já treinados sem comprometer a performance. Em cenários onde o tempo de treinamento ou a disponibilidade de dados seja limitada, essa abordagem pode ser especialmente útil.

A terceira análise consistiu em testar a eficácia do TL em um domínio distinto. Para isso, foi aplicada a rede LSTM treinada em São Paulo diretamente sobre os dados de Calgary, uma cidade com padrões climáticos significativamente diferentes. Após o fine-tuning no novo domínio, o modelo atingiu um MAE de 1,9348 °C, RMSE de 2,4958 °C, R^2 de 0,9443 e DTW de 0,6087 °C.

Em comparação com o modelo LSTM treinado exclusivamente em Calgary (sem transferência), que obteve um MAE de 1,9833 °C, RMSE de 2,6300 °C, R^2 de 0,9381 e DTW de 0,6659 °C, observa-se que o modelo transferido obteve melhor desempenho em todas as métricas avaliadas. A melhoria no R^2 (de 0,9381 para 0,9443) indica que o modelo transferido foi capaz de explicar uma fração ainda maior da variância dos dados de Calgary. Além disso, a redução nas métricas de erro e na DTW evidencia que a rede foi capaz de generalizar o conhecimento aprendido em São Paulo para um ambiente climático diferente.

Esse resultado demonstra a robustez e generalização do conhecimento aprendido pela LSTM, sendo possível transferir e adaptar o modelo para novas regiões com bom desempenho, mesmo quando há diferenças entre os domínios. Isso é particularmente relevante em contextos onde não há dados históricos suficientes para treinar um modelo do zero, tornando o Transfer Learning uma alternativa promissora.

Os experimentos demonstraram que a rede LSTM, além de apresentar o melhor desempenho entre os modelos testados, mostrou-se adequada para aplicação de Transfer Learning em diferentes cenários. No mesmo domínio, o modelo transferido manteve a performance do modelo treinado do zero, enquanto no domínio diferente foi capaz de superar o modelo local, demonstrando sua capacidade de adaptação. A redução significativa na métrica DTW em ambos os casos reforça a qualidade das previsões em termos de similaridade temporal com os dados reais.

Em síntese, os resultados confirmam que o uso de Transfer Learning em redes LSTM é viável e vantajoso, especialmente em contextos com escassez de dados ou necessidade de generalização entre diferentes regiões climáticas. Essa abordagem representa uma contribuição relevante para o desenvolvimento de sistemas preditivos mais eficientes e transferíveis no campo da previsão meteorológica com aprendizado de máquina.

A tabela com os resultados obtidos nas últimas comparações segue a seguir:

Tabela 3- Comparação de resultados de Transfer Learning x Modelos base

	SP -> SP	**SP -> Calgary**	**SP_{base}**	**$Calgary_{base}$**
DTW(°C)	1,14042	**0,60876**	2,25810	2,30648
MAE (ºC)	**1,09071**	1,93483	1,03817	1,98334
MSE (ºC²)	**1,47436**	2,49582	1,94506	2,63009
R2	0,82678	**0,9443**	0,82492	0,93816
Tempo médio (s)	**16,45 s**	20,97 s	130,66 s	134,88 s

Fonte: Autoria própria

A aplicação de Transfer Learning à LSTM resultou em melhorias significativas de desempenho. No cenário de SP → SP, observou-se uma redução de 49,5% no DTW, 24,2% no MSE e uma expressiva economia de 87,4% no tempo de execução, embora com leve aumento no MAE. Já no caso de generalização para outro domínio climático (SP → Calgary), os ganhos foram ainda mais notáveis: redução de 73,6% no DTW, 5,1% no MSE, 84,4% no tempo de

execução e um aumento de 0,7% no R^2, em relação ao modelo base treinado diretamente com dados de Calgary. Outro ponto de destaque do Transfer Learning, é a capacidade de refinar um modelo treinado em uma base de dados grande em pouco tempo, sendo necessário menos recurso para ampliar a capacidade de generalização e predição de um modelo complexo de machine learning.

Esses resultados reforçam o potencial do Transfer Learning como ferramenta promissora para previsão meteorológica em ambientes com dados limitados, possibilitando a reutilização de conhecimento prévio de forma eficiente e precisa.

5 - Conclusão

Este trabalho teve como objetivo principal investigar e comparar o desempenho de diferentes modelos de deep learning na previsão de séries temporais climáticas, com destaque para a aplicação de técnicas de TL como forma de otimizar recursos computacionais e melhorar a capacidade de generalização em contextos com escassez de dados.

Inicialmente, foram explorados os fundamentos das séries temporais e suas particularidades, além de abordagens modernas de aprendizado profundo, como as redes LSTM, GRU, RNN e CNN. A partir disso, foi possível evidenciar a superioridade das arquiteturas baseadas em LSTM na modelagem de padrões temporais complexos, especialmente devido à sua habilidade de capturar dependências de longo prazo. A GRU também apresentou resultados consistentes, com menos tempo de treinamento, configurando-se como uma alternativa promissora.

O experimento com TL mostrou-se bastante eficiente. A reutilização de um modelo previamente treinado permitiu não apenas uma redução de 84,4% no tempo de treinamento (de 134,88 s para 20,97 s), como também manteve — e em alguns casos até superou — o desempenho preditivo em domínios distintos. Isso foi observado especialmente na transferência de conhecimento do cenário climático de São Paulo para Calgary, onde o modelo obteve melhora em todas as métricas avaliadas: o DTW foi reduzido em 73,6% (de 2.30648 °C para 0,60876 °C), o MSE caiu 5,1% (de 2,63009 °C² para 2,49582 °C²), o MAE reduziu em 2,4% (de 1,98334 °C para 1,93483 °C), e o coeficiente de determinação R^2 aumentou em 0,7% (de 0,93816 para 0,9443), mesmo diante de diferentes padrões climáticos. Esses resultados reforçam a capacidade do modelo em generalizar para novos contextos, mesmo com características meteorológicas distintas.

Como continuidade deste estudo, futuras pesquisas podem explorar o uso de transformers aplicados a séries temporais, que têm ganhado destaque por sua capacidade de capturar relações de longo alcance com eficiência. Outra direção promissora seria a aplicação de técnicas de *data augmentation* para enriquecer conjuntos de dados escassos e aumentar a robustez dos modelos. Por fim, a integração de arquiteturas híbridas combinando modelos estatísticos e redes neurais, ou ainda o uso de ensembles, pode representar um avanço em termos de precisão e generalização dos modelos preditivos. Essas técnicas de ensemble funcionam combinando as previsões de múltiplos modelos individuais para gerar um resultado final mais robusto. A ideia central é que os erros de um modelo tendem a ser compensados pelos acertos dos outros, criando uma previsão agregada mais precisa e confiável.

Em síntese, os resultados confirmam que o uso de *TL* em redes LSTM é viável e vantajoso, especialmente em contextos com escassez de dados ou necessidade de generalização entre diferentes regiões climáticas. Essa abordagem representa uma contribuição relevante para o desenvolvimento de sistemas preditivos mais eficientes e transferíveis no campo da previsão meteorológica com aprendizado de máquina.

Referências

BOX, G. E. P.; JENKINS, G. M.; REINSEL, G. C. **Time series analysis: forecasting and control**. 4. ed. Oxford: Wiley, 2008. Disponível em: https://doi.org/10.1002/9781118619193. Acesso em: 1 maio 2025.

BOROVYKH, A.; BOHTE, S.; OOSTERLEE, C. W. **Conditional time series forecasting with convolutional neural networks.** arXiv preprint, arXiv:1703.04691, 2017. Disponível em: https://arxiv.org/pdf/1703.04691. Acesso em: 22 dez. 2024.

CHATFIELD, C. **The analysis of time series: an introduction.** 6. ed. Boca Raton: CRC Press, 2016. Disponível em: https://www.google.com.br/books/edition/The_Analysis_of_Time_Series/K-KEDwAAQBAJ. Acesso em: 20 jan. 2025.

EVERITT, B. S.; SKRONDAL, A. **The Cambridge Dictionary of Statistics**. 4. ed. Cambridge: Cambridge University Press, 2010. Acesso em: 08 jan. 2025.

GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. **Learning to forget: continual prediction with LSTM. Neural Computation,** v. 12, n. 10, p. 2451–2471, 2000. Acesso em: 1 maio 2025.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. Cambridge: MIT Press, 2016. Disponível em: https://www.deeplearningbook.org/. Acesso em: 23 jan. 2025.

GRAVES, A. **Generating sequences with recurrent neural networks.** arXiv preprint, arXiv:1308.0850, 2013. Disponível em: https://arxiv.org/pdf/1308.0850. Acesso em: 10 mar. 2025.

HAMILTON, J. D. **Time series analysis.** Princeton: Princeton University Press, 1994. Disponível em: https://api.pageplace.de/preview/DT0400.9780691218632_A40156688/preview-9780691218632_A40156688.pdf. Acesso em: 12 mar. 2025.

HEWAMALAGE, Hansika; BERGMEIR, Christoph; BANDARA, Kasun. **Recurrent neural networks for time series forecasting: Current status and future directions**. International Journal of Forecasting, v. 37, n. 1, p. 388–427, 2021. DOI: https://doi.org/10.1016/j.ijforecast.2020.06.008. Acesso em: 14 fev. 2025.

HOCHREITER, S.; SCHMIDHUBER, J. **Long short-term memory**. Neural Computation, v. 9, n. 8, p. 1735–1780, 1997. Acesso em: 11 jan. 2025.

HOFER, M. et al. **Deep learning architectures for time series forecasting.** Institute of Bioinformatics, Johannes Kepler University, 2017. Disponível em: https://www.bioinf.jku.at/publications/older/2604.pdf. Acesso em: 15 jan. 2025.

HYNDMAN, R. J.; ATHANASOPOULOS, G. **Forecasting: principles and practice**. 3. ed. Melbourne, Australia: OTexts, 2021. Disponível em: https://otexts.com/fpp3/. Acesso em: 15 dez. 2024.

IOFFE, S.; SZEGEDY, C. **Batch normalization: Accelerating deep network training by reducing internal covariate shift**. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING – ICML, 32., 2015, Lille. Proceedings… Lille: PMLR, 2015. Disponível em: https://proceedings.mlr.press/v37/ioffe15.html. Acesso em: 2 maio 2025.

LATORRE, Maria do Rosário Dias de Oliveira. **Análise de séries temporais em epidemiologia: uma introdução sobre os aspectos metodológicos.** Revista Brasileira de Epidemiologia, São Paulo, v. 20, n. 3, p. 1–12, set. 2017. Disponível em: https://www.scielo.br/j/rbepid/a/KM9MndgpCGSnjSNDddSydCG/. Acesso em: 20 jan. 2025.

LATHUILIÈRE, Stéphane; MESEJO, Pablo; ALAMEDA-PINEDA, Xavier; HORAUD, Radu. **A comprehensive analysis of deep regression. IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 42, n. 9, p. 2065–2081, 2020. DOI: 10.1109/TPAMI.2019.2910523. Acesso em: 1 maio 2025.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. **Gradient-based learning applied to document recognition. Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998.

LI, Zewen; YANG, Wenjie; PENG, Shouheng; LIU, Fan; ZHOU, Jun. **A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Transactions on Neural Networks and Learning Systems**, v. 33, n. 12, p. 6999–7019, dez. 2022. DOI: 10.1109/TNNLS.2021.3084827. Acesso em: 14 fev. 2025.

MORETTIN, Pedro A.; TOLOI, Clélia M. C. **Análise de séries temporais**. 3. ed. São Paulo: Blucher, 2018. Acesso em: 15 jan. 2025.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA). **Prediction Of Worldwide Energy Resources (POWER).** Washington, D.C.: NASA, 2025. Disponível em: https://power.larc.nasa.gov/. Acesso em: 12 jun. 2025.

OLAH, C. **Understanding LSTM Networks**. Colah's blog, 2015. Disponível em: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Acesso em: 10 fev. 2025.

ORUCHE, Roland; EGEDE, Lisa; BAKER, Tracy; O'DONNCHA, Fearghal. **Transfer learning to improve streamflow forecasts in data sparse regions**. arXiv preprint, 2021. DOI: 10.48550/arXiv.2112.03088. Acesso em: 1 maio 2025.

SCHERRER, L. C. et al. **Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting.** arXiv preprint, arXiv:1803.08450, 2020. Disponível em: https://arxiv.org/pdf/1803.08450. Acesso em: 14 fev. 2025.

SHUMWAY, R. H.; STOFFER, D. S. **Time series analysis and its applications: with R examples.** 4. ed. Springer, 2017. Disponível em: https://eprints.ukh.ac.id/id/eprint/232/1/2016_Book_IntroductionToTimeSeriesAndFor.pdf. Acesso em: 14 fev. 2025.

SMYL, S. **A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting**. International Journal of Forecasting, v. 36, n. 1, p. 75–85, 2020. Acesso em: 14 fev. 2025.

SRIVASTAVA, N. et al. Dropout: **A simple way to prevent neural networks from overfitting.** Journal of Machine Learning Research, [S.l.], v. 15, n. 56, p. 1929–1958, 2014. Disponível em: https://jmlr.org/papers/v15/srivastava14a.html. Acesso em: 3 maio 2025.