



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

NATAN DA SILVA GUEDES

SELF-SUPERVISED LEARNING APLICADO À PREDIÇÃO DE SÉRIES TEMPORAIS
EM CENÁRIOS CLIMÁTICOS

Sorocaba
2025

NATAN DA SILVA GUEDES

SELF-SUPERVISED LEARNING APLICADO À PREDIÇÃO DE SÉRIES TEMPORAIS
EM CENÁRIOS CLIMÁTICOS

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

G924s Guedes, Natan da Silva
Self-supervised learning aplicado à predição de séries temporais em cenários climáticos / Natan da Silva Guedes. -- Sorocaba, 2025
86 p. : il., tabs., mapas

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. Aprendizado do computador. 2. Aprendizagem profunda (Aprendizado do computador). 3. Climatologia. 4. Modelos não lineares (Estatística). 5. Análise de regressão. I. Título.

NATAN DA SILVA GUEDES

TÍTULO DO TRABALHO: SELF-SUPERVISED LEARNING APLICADO À
PREDIÇÃO DE SÉRIES TEMPORAIS EM CENÁRIOS CLIMÁTICOS

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: 08/04/2025

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

Dra. Liliane Moreira Nery
UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

Aos meus pais Wanderlei e Noemia, que mesmo não estando mais neste mundo continuam sendo a base pela qual caminho e me inspiro todo dia a ser uma pessoa melhor. Ao meu irmão Kaue, motivo pelo qual tento ser um pouco melhor todo dia para que possa servir de inspiração e ponto de apoio quando precisar. Aos meus sogros Júnior e Nair, com certeza foram figuras muito importantes durante esses 6 anos que estamos convivendo e tenho vocês como pais e um porto seguro. E claro, a pessoa que mais me apoiou durante todo esse ciclo, minha namorada Natasha. Se estou concluindo esse curso e tendo a honra de ser um engenheiro, com certeza isso é por todo seu apoio nos momentos bons e ruins.

AGRADECIMENTOS

Este trabalho só foi possível graças ao apoio de muitas pessoas queridas.

Gostaria de agradecer, primeiramente, aos meus pais, Wanderlei e Noemia, por todo o suporte dado até o início da minha graduação e pelos ensinamentos que levarei para a vida.

À minha namorada, Natasha, por estar sempre me incentivando e me dando forças para continuar essa longa jornada.

Aos meus sogros, Júnior e Nair, que tenho como pais e que me dão suporte a todo momento.

Ao meu irmão, Kaue, por toda a parceria em todos os momentos da vida, pela amizade e pelos ensinamentos que compartilhamos diariamente sem perceber.

Aos colegas de classe das turmas XVII, XVIII e XIX do curso de Engenharia de Controle e Automação, em especial ao grupo Mecaboís, pela amizade, troca de conhecimento e apoio ao longo do curso.

À Universidade Estadual Paulista (UNESP) e ao seu corpo docente, pelo aprendizado proporcionado, e, em especial, à assistente social Janaína, da Unesp Sorocaba, que foi fundamental para a minha permanência na faculdade nos momentos mais difíceis da minha vida.

Ao meu orientador, Prof. Dr. Leopoldo André Dutra Lusquino Filho, pela sua orientação, pelas valiosas dicas e exemplos, bem como pela parceria durante as aulas e o desenvolvimento deste trabalho.

A todos que, de alguma forma, contribuíram para a realização deste trabalho, meus mais sinceros agradecimentos.

“Tudo o que temos de decidir é o que fazer com o tempo que nos é dado.”
(O Senhor [...], 2001).

RESUMO

O presente estudo investigou a aplicação de *Self-Supervised Learning* (SSL) com *Masked Autoencoders* para previsão de temperatura a 2 metros da superfície (T2M) a partir de dados climáticos multivariados. A abordagem foi proposta para superar limitações de métodos supervisionados tradicionais, que demandam grandes volumes de dados rotulados – frequentemente escassos em aplicações meteorológicas. O *Masked Autoencoder* foi empregado para reconstruir representações latentes dos dados brutos, permitindo que modelos aprendessem padrões climáticos complexos sem dependência exclusiva de rótulos. Foram comparados três modelos principais: *Random Forest* (RF), Redes Neurais Densas (DNN) e *Long Short-Term Memory* (LSTM), em duas condições (dados originais e reconstruídos pelo *Masked Autoencoder*). Os resultados destacaram que: a DNN + SSL obteve o melhor desempenho geral (MAE: 0,5621, R²: 97,6%), superando suas contrapartes com dados originais; o RF + SSL mostrou alta estabilidade (DP MAE: $\pm 0,0006$), sendo ideal para cenários que priorizam consistência; a LSTM, embora superior com dados originais (MAE: 0,8961), teve desempenho prejudicado pelo *Masked Autoencoder*, sugerindo perda de informação temporal na reconstrução. Concluiu-se que o SSL com *Masked Autoencoder* é uma estratégia promissora para problemas climáticos, especialmente quando combinado com redes densas, oferecendo precisão e eficiência computacional. As limitações observadas com a LSTM indicam a necessidade de adaptações para preservar dependências temporais. A metodologia pode ser estendida a outros domínios com dados não rotulados (ex.: energia renovável, poluição atmosférica), e trabalhos futuros devem explorar *autoencoders* temporais e arquiteturas híbridas para otimizar ainda mais os resultados.

Palavras-chave: aprendizado de máquina, inteligência artificial, Self-Supervised Learning, previsão climática, séries temporais.

ABSTRACT

This study investigated the application of Self-Supervised Learning (SSL) with Masked Autoencoders for predicting temperature at 2 meters from the surface (T2M) from multivariate climate data. The approach was proposed to overcome the limitations of traditional supervised methods, which require large volumes of labeled data - often scarce in meteorological applications. Masked Autoencoder was used to reconstruct latent representations of the raw data, allowing models to learn complex weather patterns without relying exclusively on labels. Three main models were compared: Random Forest (RF), Deep Neural Networks (DNN), and Long Short-Term Memory (LSTM), under two conditions: using original data and data reconstructed by Masked Autoencoders. The results highlighted that the DNN + SSL achieved the best overall performance (MAE: 0.5621, R^2 : 97.6%), significantly outperforming its counterparts trained on original data. The RF + SSL demonstrated high stability (MAE standard deviation: ± 0.0006), making it ideal for scenarios prioritizing consistency. In contrast, the LSTM, which performed better with original data (MAE: 0.8961), suffered performance degradation when using Masked Autoencoders, suggesting a loss of temporal information during data reconstruction. Thus, it was concluded that SSL with Masked Autoencoders is a promising strategy for climate-related problems, especially when combined with deep neural networks, offering accuracy, noise robustness, and computational efficiency. However, the limitations observed in the LSTM suggest the need for methodological adaptations to better preserve temporal dependencies in the data. The proposed methodology can be extended to other domains with unlabeled data, such as renewable energy and air pollution. Future studies should explore temporal autoencoders and hybrid architectures to further optimize the obtained results.

Keywords: machine learning; artificial intelligence; self-supervised learning; climate forecasting; time series.

LISTA DE ILUSTRAÇÕES

Figura 1 - Estrutura de uma <i>LSTM</i>	18
Figura 2 - Arquitetura (esquerda) esquema de treinamento (direita) do <i>MAE</i>	28
Figura 3 - Área de abrangência dos dados.	31
Figura 4 - Decomposição da série temporal	32
Figura 5 - Temperatura média por hora do dia.	34
Figura 6 - Comportamento da temperatura em relação ao índice de radiação solar.	34
Figura 7 - Comportamento da temperatura em relação ao índice de umidade relativa do ar. ...	35
Figura 8 - Temperatura média ano a ano.	36
Figura 9 - Distribuição da temperatura por estação do ano.	36
Figura 10 - Comparação dos dados reconstruídos pelos reais em relação a temperatura a 2 metros.	37
Figura 11 - Comparação da temperatura real pela reconstruída.	38
Figura 12 - Distribuição dos erros de reconstrução (T2M).	39
Figura 13 - Evolução da temperatura média ano a ano original e reconstruídos.	39
Figura 14 - Evolução da função de perda durante o treinamento.	42
Figura 15 - Dispersão dos dados originais e reconstruídos	42
Figura 16 - Distribuição dos dados do modelo.	43
Figura 17 - Importância das features.	44
Figura 18 - Matriz de correlação das features problema com o target.	44
Figura 19 - Feature Importance após a remoção de TS.	45
Figura 20 - Matriz de correlação das features problema com o target.	46
Figura 21 - Feature Importance após as correções.	46
Figura 22 - Comparação entre valores registrados e previstos de temperatura usando Random Forest Regressor sem SSL.	49
Figura 23 - Comparação entre valores registrados e previstos de temperatura usando Random Forest Regressor com SSL.	51
Figura 24 - Comparação de desempenho do Random Forest Regressor com e sem SSL.	54
Figura 25 - Comparação entre valores registrados e previstos de temperatura usando <i>DNN</i> sem SSL.	58
Figura 26 - Comparação entre valores registrados e previstos de temperatura usando <i>DNN</i> com SSL.	61
Figura 27 - Comparação de desempenho do <i>DNN</i> com e sem SSL.	64
Figura 28 - Comparação entre valores registrados e previstos de temperatura usando <i>LSTM</i> sem SSL.	68
Figura 29 - Comparação entre valores registrados e previstos de temperatura usando <i>LSTM</i> com SSL.	71
Figura 30 - Comparação de desempenho do <i>LSTM</i> com e sem SSL.	74

LISTA DE TABELAS

Tabela 1 - Temperatura original, reconstruída e erro.	38
Tabela 2 - Resultados do modelo Random Forest Regressor sem SSL.....	50
Tabela 3 - Média e desvio padrão do modelo Random Forest Regressor sem SSL.....	50
Tabela 4 - Resultados do modelo Random Forest Regressor com SSL.	52
Tabela 5 - Média e desvio padrão do modelo Random Forest Regressor com SSL.	52
Tabela 6 - Comparação entre os resultados obtidos e a literatura (RF).....	57
Tabela 7 - Resultados do modelo <i>DNN</i> sem SSL.....	59
Tabela 8 - Média e desvio padrão do modelo <i>DNN</i> sem SSL.	60
Tabela 9 - Resultados do modelo <i>DNN</i> com SSL.	62
Tabela 10 - Média e desvio padrão do modelo <i>DNN</i> com SSL.....	62
Tabela 11 - Comparação entre os resultados obtidos e a literatura (DNN).	67
Tabela 12 - Resultados do modelo <i>LSTM</i> sem SSL.	69
Tabela 13 - Média e desvio padrão do modelo <i>LSTM</i> sem SSL.	69
Tabela 14 - Resultados do modelo <i>LSTM</i> com SSL.....	72
Tabela 15 - Média e desvio padrão do modelo <i>LSTM</i> com SSL.	72
Tabela 16 - Comparação dos modelos com e sem SSL.....	76

SUMÁRIO

1	INTRODUÇÃO	13
2	TRABALHOS RELACIONADOS	16
2.1	Deep Learning.....	16
2.1.1	Introdução ao Deep Learning	16
2.1.2	Treinamento do modelo	19
2.1.3	Desempenho do modelo.....	21
2.2	Funções de custo e métricas utilizadas	23
2.2.1	Mean Absolute Error (MAE)	24
2.2.2	Mean Square Error (MSE)	24
2.2.3	Root Mean Square Error	24
2.2.4	Coeficiente de Determinação (R^2):	24
2.3	Self-Supervised Learning	25
2.3.1	Motivação	25
2.3.2	Estratégias recentes.....	25
3	ABORDAGEM E PROPOSTA.....	27
4	EXPERIMENTOS.....	30
4.1	Setup dos experimentos.....	30
4.2	Dataset original.....	30
4.3	Dataset reconstruído	37
4.4	Hiperparâmetros	40
4.1.1	Random Forest Regressor (RFR).....	40
4.1.2	Dense Neural Networks	40
4.1.3	Long Short-Term Memory (<i>LSTM</i>)	40
4.1.4	Masked Autoencoder	41
4.5	Estrutura do teste	43
4.6	Features conflitantes	43
5	RESULTADO DOS EXPERIMENTOS	48
5.1	Random Forest Regressor	48
5.1.1	Treinamento com dados originais.....	48
5.1.2	Treinamento com dados reconstruídos	51
5.1.3	Comparação entre os modelos sem e com SSL	53
5.1.4	Comparação com Oloyede <i>et al.</i> (2023)	55
5.2	Dense Neural Networks (<i>DNN</i>)	58
5.2.1	Treinamento com dados originais.....	58
5.2.2	Treinamento com dados reconstruídos	61

5.2.3	Comparação entre os modelos sem e com SSL	63
5.1.5	Comparação com Oloyede <i>et al.</i> (2023)	65
5.3	LSTM.....	68
5.3.1	Treinamento com dados originais.....	68
5.3.2	Treinamento com dados reconstruídos	70
5.3.3	Comparação entre os modelos sem e com SSL	73
5.4	Comparativo entre todos os modelos.....	76
6	CONCLUSÃO	78
	REFERÊNCIAS	80
	APÊNDICE A - Principais bibliotecas utilizadas.....	84
	APÊNDICE B - Dados meteorológicos do dataset original.	85

1 INTRODUÇÃO

“As mudanças climáticas são transformações [...] nos padrões de temperatura e clima. Essas mudanças podem ser naturais, como por meio de variações no ciclo solar, influência da atividade vulcânica ou efeitos atmosféricos, como o “El Niño” e “La Niña”. Mas, desde 1800, as atividades humanas têm sido o principal impulsionador das mudanças climáticas, principalmente devido à queima de combustíveis fósseis como carvão, petróleo e gás” (Organização das Nações Unidas, 2024).

Devido a exacerbada busca pela obtenção dos recursos do planeta como petróleo e outros combustíveis fósseis, o consumo cada vez maior de água, o consumismo e principalmente a demanda energética que aumenta exponencialmente, torna a humanidade cada vez mais vulnerável a eventos climáticos extremos e imprevisíveis. Desta forma, torna-se cada vez mais importante o estudo das mudanças climáticas e seus fenômenos para compreender e mitigar os impactos destes no planeta e na sociedade.

Segundo o trabalho *Weather and Climate Extreme Events in a Changing Climate*,

É um fato estabelecido que as emissões de gases de efeito estufa induzidas pelo homem induzidas pelo homem levaram a um aumento na frequência e/ou intensidade de alguns extremos meteorológicos e climáticos desde a era pré-industrial, especialmente no caso de extremos de temperatura. Evidências de mudanças observadas nos extremos e sua atribuição à influência humana (incluindo emissões de gases de efeito estufa e aerossóis e mudanças no uso da terra) se fortaleceram desde o AR5, em especial para extremos de precipitação extrema, secas, ciclones tropicais e extremos compostos (incluindo eventos secos/quentes e clima de fogo). Alguns eventos extremos recentes teriam sido *extremamente improváveis* de ocorrer sem a influência humana sistema climático. (Seneviratne *et al.*, 2021, p. 1517, tradução nossa)

Portanto, se faz necessário a criação e utilização de tecnologias capazes de analisar, prever e mitigar o impacto das mudanças climática, como o aprendizado de máquina. Segundo o autor Sichman (2021, p. 38), aprendizado de máquina “[...] trata-se [de] um ramo da ciência/engenharia da computação, e, portanto, visa desenvolver sistemas computacionais que solucionam problemas. Para tal, utiliza um número diverso de técnicas e modelos, dependendo dos problemas abordados.”

Logo, o uso da inteligência artificial (IA) no combate às mudanças climáticas e a mitigação de seus efeitos diretos e indiretos e torna essencial, visto a aplicabilidade das centenas

de algoritmos que estão abrangidos nesta área da computação, como o estudo de caso de Dominguez *et al.* (2022) e as citações de Nery, Souza e Barros (2024). Neste contexto, para compreender e buscar padrões e previsões cada vez mais completas, o *Machine Learning* (ou aprendizado de máquina, em português) ganha cada vez mais espaço nos meios acadêmicos e profissionais, por uma enorme gama de aplicações e capacidade de prever eventos complexos.

Complementando o raciocínio, conforme descrito por Zhou e Liu (2016),

Enquanto os seres humanos aprendem com a experiência, os computadores podem fazer o mesmo? A resposta é "sim", e o que precisamos é de aprendizado de máquina. O aprendizado de máquina é a técnica que melhora o desempenho do sistema ao aprender com a experiência por meio de métodos computacionais. Nos sistemas computacionais, a experiência existe na forma de dados, e a principal tarefa do aprendizado de máquina é desenvolver algoritmos de aprendizado que constroem modelos a partir dos dados. Ao alimentar o algoritmo de aprendizado com dados de experiência, obtemos um modelo capaz de fazer previsões (por exemplo, determinar se uma melancia está madura) com base em novas observações (por exemplo, uma melancia ainda não cortada). Se considerarmos a ciência da computação como o estudo de algoritmos, então o aprendizado de máquina é o estudo de algoritmos de aprendizado.” (Zhou; Liu, 2016, p.3, tradução nossa).

O aprendizado de máquina, portanto, pode ajudar na predição de eventos climáticos quando olhados sob a óptica de sistemas computacionais. Isto, todavia, só é possível devido a capacidade de algoritmos de aprendizado de máquina terem bom desempenho em situações de comportamento não-linear, como em um sistema complexo como o clima. Segundo Jones e Ricketts (2021),

Os sistemas complexos têm comportamentos emergentes que não podem ser previstos a partir da soma de suas partes (Nicolis e Nicolis, 2012; Broad, 2014; O'Connor e Wong, 2019), portanto, são assimétricos. Os sistemas grandes e complexos contêm uma estrutura hierárquica de vários níveis com diferentes conjuntos de escalas. estrutura hierárquica de vários níveis com diferentes conjuntos de regras dependentes de escala (por exemplo, diferentes níveis tróficos em um ecossistema). Eles não podem ser compreendidos por meio de métodos tradicionais (Anderson, 1972), em conformidade com o que pode ser chamado de hipótese emergentista (Clayton; Davies, 2011; O'Connor, 2020, tradução nossa).

De acordo com Mitchell (2009), um sistema complexo pode ser definido como

[...] um sistema no qual grandes redes de componentes sem controle central e regras simples de operação dão origem a um comportamento coletivo complexo,

processamento sofisticado de informações e adaptação por meio de aprendizado ou evolução.” Além disso, a autora destaca que um sistema complexo também pode ser entendido como “um sistema que apresenta comportamentos emergentes e auto-organizados não triviais” (Mitchell, 2009, p. 13, tradução nossa).

Uma vez que abordagens preditivas tradicionais, envolvendo simulações ou modelos autorregressivos, não são suficientemente eficientes para capturar padrões temporais com alta não-linearidade emergentes de sistemas complexos, os modelos baseados em *Machine Learning* têm se mostrado uma abordagem promissora para cenários climáticos extremos. Essa perspectiva é reforçada por estudos recentes que destacam as dificuldades na previsão de eventos extremos — como ondas de calor, secas e tempestades — devido à sua natureza caótica, não linear e à limitação dos modelos tradicionais. Nesse contexto, técnicas de inteligência artificial vêm demonstrando potencial para aprimorar tais previsões e revelar relações entre esses eventos e fatores climáticos em diferentes escalas (Materia *et al.*, 2023, p.2, tradução nossa). E entre eles, destacam-se os modelos obtido a partir de técnicas de aprendizado auto supervisionado, exploradas neste trabalho.

No capítulo 2, são apresentados os trabalhos relacionados que inspiraram este estudo, e alguns dos conceitos abordados estão explicados em detalhes nas fontes originais. No capítulo 3, é apresentada a abordagem proposta por este trabalho para prever, com a maior exatidão possível, o clima de determinada região, utilizando os conceitos introduzidos no capítulo 2. No capítulo 4, são descritas as condições, hiperparâmetros, configurações e o setup dos experimentos. No capítulo 5, são discutidos os resultados obtidos ao longo deste estudo, os problemas encontrados e possíveis soluções. Por fim, no capítulo 6, são apresentadas as conclusões deste estudo, possíveis contribuições futuras e pendências não resolvidas neste trabalho.

2 TRABALHOS RELACIONADOS

2.1 Deep Learning

2.1.1 Introdução ao Deep Learning

A Inteligência Artificial (IA) é um campo da ciência da computação destinado a criar sistemas computacionais capazes de realizar tarefas que, anteriormente, necessitavam de inteligência humana. Segundo Russel e Norvig (2016, p. 1), "Inteligência artificial é o estudo de agentes inteligentes: qualquer dispositivo que percebe seu ambiente e toma ações que maximizam suas chances de sucesso em algum objetivo".

Esses sistemas utilizam algoritmos que podem reconhecer padrões, fazer previsões de dados, tomar decisões e, como o foco do trabalho aqui desenvolvido, resolver problemas e aprender a partir de dados. No contexto do *machine learning* (aprendizado de máquina), uma subárea da IA, "é o desenvolvimento de algoritmos que permitem aos sistemas aprender e melhorar com a experiência, sem serem explicitamente programados para cada tarefa" (Russel; Norvig, 2016, p. 936). Esse processo de aprendizagem é essencial para que os sistemas se adaptem de maneira eficaz a novos dados e condições.

Dentre os inúmeros paradigmas dentro do *Machine Learning* pode-se notar: o aprendizado supervisionado - em que os dados foram devidamente rotulados e o objetivo do modelo é realizar a predição em comparação com os valores registrados; o aprendizado não supervisionado - em que os dados não foram rotulados e o objetivo do modelo é realizar agrupamentos de dados de acordo com suas semelhanças e realizar a rotulagem e o aprendizado semi-supervisionado - situação em que os dados não estão completamente rotulados e o objetivo do modelo é, dada uma quantidade de dados rotulados, aprender os padrões intrínsecos aos dados e aplicar a metodologia aos dados não rotulados - que será explicado em mais detalhes neste trabalho.

Dentro da subárea do *Machine Learning* existem algoritmos de aprendizado mais robustos a partir do uso de redes neurais de diversas camadas, chamado de Aprendizado Profundo (*Deep Learning*). Segundo LeCun, Bengio e Hinton (2015, pp. 436-444) "os algoritmos de *deep learning* são inspirados na estrutura e função do cérebro e têm como objetivo aprender representações dos dados em múltiplos níveis de abstração, tipicamente através de uma rede de camadas de transformações não lineares.". Logo, para problemas cada vez mais complexos, em que as relações entre as variáveis se tornam cada vez mais difíceis de

interpretar, o *Deep Learning* pode ser uma ferramenta poderosa para a análise de dados modernos de diversas naturezas, desde processamento de imagens a interpretação de texto, nuances da fala ou a predição de eventos futuros a partir de uma série de dados temporais.

Durante o desenvolvimento de modelos de *machine learning*, existem dois conceitos que devem ser observados e evitados: o *overfitting* e o *underfitting*. *Overfitting* e *underfitting* são dois problemas comuns em modelos de *machine learning* que afetam a capacidade de generalização dos algoritmos. O *overfitting* ocorre quando um modelo se ajusta excessivamente aos dados de treinamento, capturando até mesmo o ruído, o que resulta em um desempenho fraco em novos dados (Goodfellow; Bengio; Courville, 2016). Já o *underfitting* acontece quando o modelo é excessivamente simples para capturar os padrões subjacentes nos dados, levando a um desempenho insatisfatório tanto no treinamento quanto na validação (Bishop, 2006).

Devido à enorme quantidade de tarefas para as quais o *Deep Learning* pode ser uma boa solução, diversas arquiteturas foram desenvolvidas ao longo dos anos para operar em diferentes cenários e aplicações. De acordo com LeCun, Bottou, Bengio e Haffner (1998, p. 6) “as redes convolucionais combinam três ideias arquitetônicas para garantir algum grau de deslocamento, escala e invariância de distorção campos receptivos locais, pesos compartilhados ou replicação de pesos e subamostragem espacial ou temporal”. As redes neurais convolucionais (CNNs) são capazes de aprender automaticamente hierarquias espaciais de características visuais, desde bordas e texturas simples nas camadas iniciais até padrões e objetos complexos nas camadas mais profundas, sem a necessidade de intervenção humana explícita (DEEPGRAM, 2025, tradução nossa).

Outra importante arquitetura de redes neurais são as redes neurais recorrentes (RNN). De acordo com Lipton, Berkowitz e Elkan (2015, p. 1),

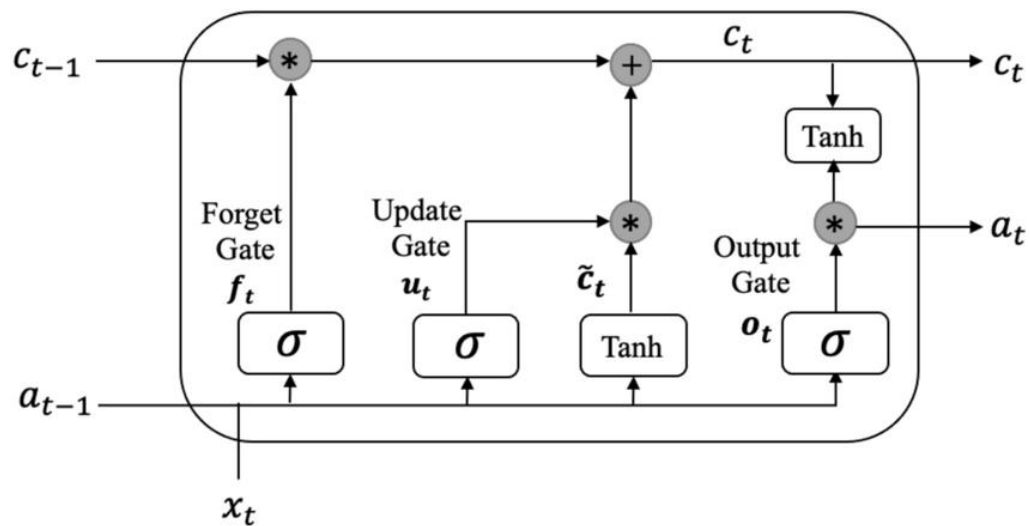
As redes neurais recorrentes (RNNs) são modelos conexionistas que capturam a dinâmica das sequências por meio de ciclos na rede de nós. Diferentemente das redes neurais feedforward padrão, as redes recorrentes retêm um estado que pode representar informações de uma janela de contexto arbitrariamente longa. (Lipton; Berkowitz; Elkan, 2015, p. 1, tradução nossa)

As RNNs têm destaque em atividades que envolvam o tratamento de séries temporais ou texto, pois possuem capacidade de armazenar informação, uma espécie de “memória”, capaz de armazenar dependências temporais intrínsecas aos dados, sendo superior em vários aspectos a outras redes que trabalham com dados sequenciais.

Logo, a evolução natural das *RNN* são as redes *Long Short-term Memory (LSTM)*. Segundo Hochreiter e Schmidhuber (1997, p. 1735), “as unidades de memória de longo prazo (*LSTM*) foram projetadas para combater o problema de gradiente decrescente em redes recorrentes, permitindo que a rede armazene informações por longos períodos de tempo”.

A principal diferença entre as *LSTM* e *RNN* é a presença dos chamados *gates*, que regulam o fluxo das informações dentro da rede, possibilitando a perpetuação de informações relevantes dentro da rede à longo prazo, tendo um desempenho melhor em sequências extensas de dados. A Figura 1 apresenta a estrutura de uma *LSTM*.

Figura 1 - Estrutura de uma *LSTM*



Fonte: Nguyen *et al.* (2022).

Entretanto, existem outras arquiteturas de enorme destaque nos dias atuais que possibilitam a criação de *chatbots* cada vez mais sofisticados e capazes de criar representações completamente únicas baseada nos dados de entrada já existentes, que são os *Transformers*. Segundo Vaswani, *et al.* (2017, p. 6000),

o modelo Transformer é baseado em um novo mecanismo de atenção, que chamamos de auto-atenção, que calcula representações da sequência de entrada considerando as relações entre todos os elementos da sequência, simultaneamente, e não sequencialmente, como nas tradicionais RNNs.

Tal arquitetura vem sendo aplicada em modelos extremamente robustos, como o Bidirectional Encoder Representations from Transformers – BERT (Devlin *et al.*, 2018) e o Generative Pre-trained Transformer – GPT (Brown *et al.*, 2020), modelos otimizados para processamento de linguagem natural (Natural Language Processing – NLP). O BERT é

otimizado para entender o contexto bidirecional de frases, enquanto o GPT é voltado para a geração de texto.

2.1.2 Treinamento do modelo

O desenvolvimento de modelos de *machine learning* representa uma fase essencial em sistemas que utilizam dados. Essa prática envolve capacitar um modelo a reconhecer padrões e realizar previsões corretas com base em conjuntos de dados exemplares. Para alcançar esse objetivo, podem ser usadas várias arquiteturas, de acordo com o problema em questão e a natureza dos dados disponíveis. Esse procedimento abrange arquiteturas como Redes Neurais Artificiais, Redes Neurais Recorrentes, *LSTMs*, Transformers e Autoencoders, que são abordagens comuns em tarefas de classificação, previsão e geração de dados.

Entretanto, para que seja possível aplicar os dados por meio de tais arquiteturas se faz necessário um pré-processamento dos dados. Esta etapa consiste em remover ou corrigir valores ausentes, duplicados ou inconsistentes de forma a preservar a natureza original dos dados; realizar a normalização ou padronização dos dados de forma a ajustar a escala dos dados para uma predição mais eficiente e a divisão dos dados em conjuntos de treino, validação e teste. O conjunto de treino é usado para treinar o modelo, o conjunto de validação para ajustar hiperparâmetros, e o conjunto de teste para avaliar a performance final do modelo.

Os hiperparâmetros são definidos antes do início do processo de treinamento e regem tanto a estrutura quanto o comportamento dos modelos de aprendizado de máquina. Diferentemente dos parâmetros do modelo — aprendidos durante o treinamento —, os hiperparâmetros desempenham um papel fundamental na determinação do desempenho do modelo (Ilemobayo *et al.*, 2024, p. 390, tradução nossa). O ajuste desses hiperparâmetros, conhecido como *fine-tuning*, constitui uma etapa essencial no desenvolvimento do modelo, conforme descrito por Ilemobayo *et al.* (2024),

O ajuste de hiperparâmetros é o processo de encontrar o conjunto ideal de hiperparâmetros que produz o melhor desempenho para um modelo de aprendizado de máquina. Esse processo é fundamental porque os hiperparâmetros controlam o processo de aprendizagem e a estrutura do modelo, como a taxa de aprendizagem, o número de neurônios em uma rede neural ou o tamanho do kernel em uma máquina de vetor de suporte, afetando diretamente seu desempenho. Ao contrário dos parâmetros do modelo, que são aprendidos com os dados, os hiperparâmetros são definidos antes do treinamento e exigem uma seleção cuidadosa. O ajuste dos

hiperparâmetros pode melhorar o desempenho e a generalização do modelo. (Ilemobayo *et al.*, 2024, p. 391, tradução nossa).

Após concluir a etapa de pré-processamento, os dados são introduzidos no modelo para a fase de treinamento, cujo objetivo é aprimorar os pesos do modelo, a fim de minimizar a discrepância entre as previsões geradas e os valores registrados.

Os principais hiperparâmetros em arquiteturas de modelos de *machine learning* são descritos a seguir:

- Forward Pass: os dados de entrada transitam pela rede, camada por camada, até atingirem a saída, resultando nas previsões.
- Feedforward: refere-se à arquitetura de redes neurais onde as conexões entre os neurônios não formam ciclos. A informação flui apenas em uma direção — da entrada para a saída.
- Cálculo da Perda (Loss): utiliza-se uma função de perda para quantificar a diferença entre a saída do modelo e o valor real (rótulo). Entre as opções de função de perda, o erro quadrático médio (*MSE*) é preferido para tarefas de regressão, enquanto a entropia cruzada é comumente aplicada em problemas de classificação.
- Backpropagation: refere-se ao método de modificação dos pesos do modelo. Através do algoritmo de retropropagação, o erro é "enviado de volta" pela rede, o que possibilita o ajustamento dos pesos para reduzir a perda.
- Otimização: um método de otimização (como Gradiente Descendente ou Adam) é empregado para alterar os pesos durante o treinamento, com a finalidade de minimizar a função de perda.

Esse processo é repetido por várias épocas (iterações), até que o modelo aprenda as representações necessárias para realizar previsões precisas. Durante o treinamento, existem vários hiperparâmetros que precisam ser ajustados para melhorar a performance do modelo:

- Taxa de aprendizado: controla o tamanho do passo durante a atualização dos pesos.
- Número de camadas e neurônios: define a profundidade e a capacidade do modelo.
- Tamanho do batch: define quantos exemplos são usados por vez para calcular as atualizações de gradiente.
- Regularização: técnicas como dropout ou L2 regularization podem ser usadas para evitar *overfitting*.

Após o treinamento, o modelo é avaliado em dados de validação (que não foram usados durante o treinamento) para ajustar os hiperparâmetros e evitar *overfitting*. Métricas comuns para avaliação incluem:

- *MAE* (Mean Absolute Error) – Erro absoluto médio: Mede a média das diferenças absolutas entre os valores previstos e os valores registrados.
- *MSE* (Mean Squared Error) – Erro quadrático médio: Mede a média dos quadrados das diferenças entre os valores previstos e os reais.
- Coeficiente de Determinação (R^2): Mede a proporção da variabilidade dos dados que é explicada pelo modelo. Um R^2 próximo de 1 indica um modelo que explica bem os dados.

Após o ajuste dos hiperparâmetros e a validação do modelo, a última etapa é avaliar o modelo em um conjunto de dados de teste, que não foi utilizado em nenhuma fase anterior do treinamento. Isso garante que o modelo tenha uma boa generalização e não tenha apenas memorizado os dados de treino.

2.1.3 Desempenho do modelo

Modelos baseados em *Deep Learning* destacam-se por sua elevada capacidade de generalização e pela aptidão para lidar com problemas não lineares complexos. Essa versatilidade tem favorecido sua ampla aplicação em domínios como previsão climática e de temperatura (Sharma *et al.*, 2024), análise de séries temporais financeiras (Ali *et al.*, 2023), modelagem e controle de sistemas dinâmicos (Raissi; Perdikaris; Karniadakis, 2019), reconhecimento de padrões visuais (Kumar; Sharma, 2021) e processamento de linguagem natural (Vaswani *et al.*, 2017).

Os algoritmos que utilizam *Deep Learning* são especialmente beneficiados com grandes quantidades de dados e recursos computacionais robustos.

A apresenta uma comparação geral entre modelos *Deep Learning* e outros principais modelos de *Machine Learning*, considerando diferentes aspectos

Quadro 1 – Comparação entre diferentes modelos

(continua)

Aspecto	Deep Learning	Random Forest	SVM	Naive Bayes	Regressão Logística	KNN
Capacidade de modelagem	Excelente para padrões não lineares complexos.	Eficaz para capturar relações não lineares, mas pode superestimar em dados ruidosos.	Muito bom para problemas não lineares quando usando kernels.	Simples, mas assume independência entre variáveis.	Bom para relações lineares, mas limitado em padrões mais complexos.	Funciona bem em problemas não lineares simples, mas tem alto custo computacional.
Requisitos de Dados	Necessita de muitos dados para bom desempenho.	Funciona bem com conjuntos menores, mas pode perder desempenho em grandes volumes.	Pode funcionar com conjuntos menores, mas o tempo de treino aumenta com os dados.	Funciona bem com poucos dados, desde que a suposição de independência seja razoável.	Funciona bem com conjuntos pequenos à médios.	Depende do número de vizinhos (K), pode ser afetado por ruído e outliers.
Complexidade Computacional	Muito alta, devido à necessidade de múltiplos parâmetros e treinamento iterativo.	Moderada a alta, dependendo do número de árvores.	Moderada a alta, dependendo do kernel utilizado.	Baixa, muito eficiente para grandes conjuntos de dados.	Baixa, treinamento rápido e eficiente.	Alta, pois depende do cálculo de distâncias para cada nova previsão.
Interpretabilidade	Difícil de interpretar ("caixa preta").	Melhor que <i>Deep Learning</i> , mas difícil para muitas árvores.	Razoável, dependendo do kernel.	Muito alta, fácil de interpretar.	Muito alta, fácil de explicar.	Difícil quando há muitos dados ou variáveis.
Robustez a Outliers	Pode ser afetado por outliers, mas pode	Bastante robusto a outliers.	Sensível a outliers, especialmente com	Muito sensível a outliers.	Sensível a outliers.	Muito sensível a outliers.

Quadro 1 – Comparação entre diferentes modelos

(conclusão)

	aprender a ignorá-los.		kernels lineares.			
Aplicações Principais	Imagens, processamento de linguagem natural, previsão de séries temporais complexas.	Problemas estruturados, como detecção de fraudes e modelagem preditiva.	Classificação de texto, bioinformática, reconhecimento facial.	Classificação de e-mails (spam vs. não spam), diagnósticos médicos.	Problemas de classificação binária como aprovação de crédito.	Reconhecimento de padrões simples e sistemas de recomendação.

Fonte: Adaptado de Goodfellow, Bengio e Courville (2016); Hastie, Tibshirani e Friedman (2009); Géron (2019); Russell e Norvig (2020); Murphy (2012).

Como pode ser observado no Quadro 1, cada modelo apresenta vantagens e desvantagens uns sobre os outros para determinado problema. Entretanto, os modelos baseados em *Deep Learning* apresentam características que os destacam dos outros em termos de performance, capacidade de generalização e aplicações em problemas do mundo real. Todavia, há a necessidade de realizar um levantamento dos prós e contras de cada um antes de escolher aquele que deve ser aplicado à determinado problema, visto que mesmo obtendo melhores resultados é necessário um volume muito grande de dados, um grande poder computacional (o que muitas vezes é um fator determinante para a escolha) para que os modelos baseados em *Deep Learning* performem melhor que os outros, além da realização de um *fine tuning* adequado para evitar problemas de *overfitting*.

Essa análise vai de encontro ao estudo de Bay e Yearick (2024), que destaca que os modelos de *deep learning* (DL) se sobressaem aos modelos tradicionais de *machine learning* (ML) na generalização e extrapolação para dados não vistos, sendo mais eficazes em prever cenários futuros. No entanto, DL exige mais dados, recursos computacionais e tempo de treinamento, enquanto os modelos tradicionais são mais eficientes em contextos com menos dados ou quando a interpretação é essencial. A escolha entre DL e ML deve ser feita com base nas necessidades e limitações específicas de cada tarefa.

2.2 Funções de custo e métricas utilizadas

De acordo com os conceitos abordados por Goodfellow, Bengio e Courville (2016), a função de custo é um componente fundamental que quantifica o erro entre as previsões do

modelo e os valores registrados. A otimização desta função permite ajustar os parâmetros do modelo para minimizar os erros e melhorar a precisão das previsões.

2.2.1 Mean Absolute Error (MAE)

A principal função de custo utilizada na avaliação de modelos de regressão é o erro médio absoluto (Mean Absolute Error – *MAE*), cujo objetivo é medir as diferenças absolutas entre os valores registrados (y_i) e os previstos ($\hat{y}_i$).

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

2.2.2 Mean Square Error (MSE)

Outra importante métrica na avaliação de modelos é o erro médio quadrado (Mean Square Error – *MSE*), que penaliza erros maiores mais fortemente do que o *MAE*, pois os erros são elevados ao quadrado, tornando-o sensível a outliers, sendo útil quando se deseja dar mais peso a previsões com grandes desvios.

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

2.2.3 Root Mean Square Error

A raiz do erro médio quadrado (Root Mean Square Error – *RMSE*), é outra métrica para determinar o desempenho do modelo, baseando-se na raiz quadrada do *MSE*.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3)$$

2.2.4 Coeficiente de Determinação (R^2):

Métrica que mede a proporção da variabilidade dos dados que é explicada pelo modelo. Quanto mais próximo de 1 o coeficiente de determinação é, maior o indicativo que o modelo explica bem os dados. Esse coeficiente é dado por:

$$R^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (4)$$

2.3 Self-Supervised Learning

2.3.1 Motivação

O aprendizado auto supervisionado (*Self-Supervised Learning*) é um dos paradigmas do aprendizado de máquina que têm como objetivo proporcionar uma aprendizagem profunda e poderosa de características sem a necessidade de grandes conjuntos de dados anotados, aliviando assim o estrangulamento da anotação - um dos principais obstáculos à aplicação prática da aprendizagem profunda atualmente. Estas técnicas têm avançado rapidamente nos últimos anos, com a sua eficácia a aproximar-se e, por vezes, a ultrapassar as alternativas de pré-treino totalmente supervisionadas numa variedade de modalidades de dados, incluindo imagem, vídeo, som, texto e gráficos (Ericsson *et al.*, 2021, p. 42).

A vantagem de um modelo auto supervisionado como este trabalho propõe é a capacidade de criar rótulos intrínsecos aos dados e ser capaz de se adaptar a mudanças e criação de novos padrões ao longo do tempo. Isto é importante pois o clima é um sistema complexo e em constante evolução, desta forma o SSL é capaz de compreender estas mudanças e não se tornar obsoleto.

2.3.2 Estratégias recentes

Nos últimos anos, diversas estratégias foram desenvolvidas para entender e prever o comportamento do sistema climático terrestre. Scher (2018) em seu trabalho desenvolveu através de *General Circulation Models* (GCN) - uma aplicação de CNN, um modelo capaz de prever diversos pontos à frente o sistema climático, inclusive a partir de entradas randômicas de parâmetros e mantém uma estabilidade surpreendente mesmo para previsões de até 1000 anos no futuro.

Já Gong, *et al.* (2022) desenvolveram um modelo capaz de realizar predição de temperatura em tempo real através da utilização de vídeos representando o comportamento climático de determinada reunião.

O *Self-Supervised Learning* (SSL), como destacado por Veenu Rani *et al.* (2023), surgiu como uma das técnicas mais promissoras por não exigir anotações manuais. O princípio fundamental dessa abordagem é ocultar parte dos dados e utilizar as informações visíveis para prever as partes ausentes. Diferentemente do aprendizado não supervisionado, o SSL não requer rótulos explícitos, mas também dispensa a rotulagem manual dos dados.

De acordo com as contribuições de An *et al.* (2024), os autores propõem o pós-processamento de precipitação com base na aprendizagem por *transfer learning* do modelo treinado com dados do sistema atmosférico por meio da realização do pré-treinamento no domínio de reconstrução e da integração dos parâmetros no domínio de segmentação e empregar a rotulagem estocástica de suavização em um único momento para superar a aprendizagem tendencioso de dados desequilibrados.

Oloyede *et al.* (2023) em seu trabalho utilizou dois datasets diferentes para a tarefa de regressão: um disponibilizado através da plataforma POWER, da NASA e o dataset local, o NiMet (*Nigerian Meteorological Agency*). O objetivo dos modelos foi utilizar os dados do dataset da NASA para prever os valores de temperatura máximos e mínimos do dataset do NiMet através de 5 algoritmos de predição: *decision trees regression*, XGBoost, MLP com LBFGS (limited-memory Broyden-Fletcher-Goldfarb-Shanno) optimizer, MLP com SGD (stochastic gradient) optimizer e MLP com Adam optimizer.

O trabalho de Jie Cheng, Xiaodong Mei e Ming Liu (2023) traz uma abordagem inovadora de *Self-Supervised Learning* no campo da direção autônoma através da aplicação de *Masked Autoencoder* para prever rotas e possíveis trajetos para os veículos através de uma exposição inicial de dados mascarados, ficando a cargo do modelo reconstruí-los para prever as melhores rotas.

3 ABORDAGEM E PROPOSTA

Como abordado nos capítulos anteriores deste trabalho, o *Self-Supervised Learning* (SSL) apresenta características que o diferencia de outros paradigmas do aprendizado de máquina quando posto em situações do mundo real em que há um grande volume de dados, mas nem sempre os rótulos atribuídos a eles são o suficiente para previsões precisas como no contexto do clima e temperatura.

A previsão de dados climáticos é um desafio complexo que exige modelos capazes de lidar com grandes volumes de dados multivariados, padrões não lineares e incertezas inerentes ao sistema atmosférico. Nesse contexto, de acordo com o estudo apresentado por Ericsson, Gouk e Hospedales (2021), o *Self-Supervised Learning* (SSL) surge como uma abordagem promissora, oferecendo vantagens significativas sobre técnicas tradicionais de *machine learning* supervisionado e não supervisionado.

Os dados meteorológicos frequentemente consistem em grandes volumes de informações temporais e espaciais, mas rotular esses dados manualmente pode ser inviável. O SSL permite que os modelos aprendam representações úteis a partir dos próprios dados, sem a necessidade de rótulos explícitos, o que reduz o esforço de anotação e aproveita ao máximo a informação disponível.

Enquanto modelos supervisionados aprendem mapeando diretamente entradas para saídas com base em rótulos anotados, o SSL desenvolve representações estruturadas dos dados por meio da resolução de tarefas pretextuais, como a previsão de partes ocultas em sequências temporais ou a reconstrução de variáveis mascaradas. Essa abordagem permite que o modelo capture padrões latentes de alta complexidade, promovendo uma melhor generalização em tarefas posteriores, especialmente em cenários com escassez de dados rotulados (Le-Khac *et al.*, 2020; Liu *et al.*, 2021; Zbontar *et al.*, 2021).

Diferentemente de abordagens supervisionadas que podem sofrer com a falta de dados rotulados, o *Self-Supervised Learning* pode aprender de forma mais eficaz mesmo em cenários com poucos exemplos anotados. Além disso, ao aprender representações resilientes, os modelos treinados com SSL tendem a ser mais robustos a ruídos e variações nos dados climáticos, que são comuns devido a falhas em sensores ou mudanças ambientais inesperadas. Abordagens supervisionadas geralmente exigem grandes quantidades de dados rotulados para alcançar bons desempenhos, enquanto o SSL pode aprender representações gerais que são posteriormente

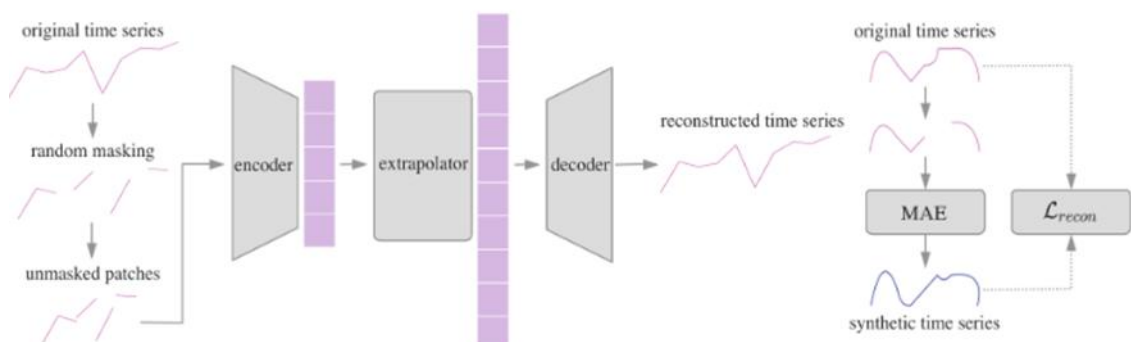
refinadas com poucos exemplos supervisionados. Isso é especialmente útil em contextos climáticos onde séries temporais completas podem não estar disponíveis para todas as variáveis de interesse.

Durante a execução dos testes realizados neste trabalho, a abordagem adotada foi a aplicação de *Masked Autoencoders*. O *Masked Autoencoders* é um tipo de *autoencoder* projetado para reconstruir um conjunto de dados a partir de entradas parcialmente mascaradas. Nesse método, uma fração dos dados (determinada por um fator de mascaramento) é ocultada, e o modelo aprende a reconstruir as informações ausentes com base nos padrões presentes nos dados remanescentes. Esse processo envolve o uso de *encoders* para aprender representações latentes e *decoders* para reconstruir os valores originais (He *et al.*, 2021).

A aplicação do *Masked Autoencoders* na previsão de dados climáticos permite que o modelo aprenda dependências temporais e espaciais de forma mais eficiente, mesmo quando há falhas ou lacunas nos registros meteorológicos. Além disso, essa abordagem pode ser combinada com arquiteturas avançadas, como redes neurais recorrentes (*RNNs*), *Long Short-Term Memory (LSTM)* e *Transformers*, para melhorar ainda mais a captura de dependências temporais nos dados. Isso resulta em previsões mais precisas e interpretáveis em comparação com métodos puramente supervisionados, que podem ter dificuldades em aprender relações climáticas complexas (He *et al.*, 2021).

A Figura 2 apresenta graficamente uma representação do pipeline e a aplicação do SSL.

Figura 2 - Arquitetura (esquerda) esquema de treinamento (direita) do *Masked Autoencoder*.



Fonte: Adaptado de Zha, Wong, Liu, Zhang, e Chen (2022, p. 3)

Dessa forma, a abordagem explorada neste trabalho está alinhada com os experimentos conduzidos por Jie Cheng, Xiaodong Mei e Ming Liu (2023) bem como com os estudos de

Veenu Rani *et al.* (2023), que demonstram que, em tarefas de pretexto, o *SSL* se destaca em relação a outras metodologias ao operar em sistemas nos quais uma fração do conjunto de dados original é ocultada. Nessa configuração, o modelo aprende a prever os valores mascarados com base nas representações que ele próprio constrói a partir dos dados disponíveis, capturando padrões latentes e aprimorando sua capacidade de generalização.

Portanto, este trabalho propõe a reconstrução dos dados por meio do *Masked Autoencoder*, extraíndo as representações dos dados que podem ser utilizados como entrada para diferentes modelos preditivos, cujos desempenhos foram avaliados na tarefa de previsão da temperatura. A análise comparativa desses modelos permitiu verificar os impactos da aprendizagem auto supervisionada na modelagem climática e na precisão das previsões.

4 EXPERIMENTOS

4.1 Setup dos experimentos

Para a elaboração deste trabalho, foram utilizados dois hardwares diferentes para a construção dos algoritmos. Para testes mais simples, foi utilizado um laptop ACER ASPIRE 5 A515 51G, com 12 GB de RAM DD4, uma placa gráfica (GPU) MX130 e um processador Intel i5 de 8ª geração.

Já para os testes dos modelos apresentados neste trabalho, foi utilizado um hardware mais potente devido ao tempo de compilação excessivo do hardware anterior. Esse hardware é composto por um processador Ryzen 7 5800X3D de 8 núcleos e 16 unidades lógicas, 32 GB de memória RAM DDR4, uma placa gráfica MSI Radeon RX 6750XT de 12 GB VRAM GDDR6.

Durante o desenvolvimento do algoritmo, diversas bibliotecas foram utilizadas para atingir o desempenho esperado para este trabalho, sendo algumas das principais apresentadas no Apêndice A. Para a execução deste trabalho foi utilizado o framework *TensorFlow/Keras*. Ambos os frameworks foram desenvolvidos pelo Google para operações com tensores, operações matemáticas complexas com CPU e GPU e desenvolvimento de redes neurais complexas multicamadas.

Todos os códigos desenvolvidos neste projeto estão disponíveis no repositório público do GitHub do autor¹.

4.2 Dataset original

O conjunto de dados utilizados neste trabalho foi construído do uso da API da plataforma POWER² (Prediction of Worldwide Energy Resources) da NASA (National Aeronautics and Space Administration), cuja disponibilização dos dados é pública. Os dados coletados na plataforma podem ser coletados de duas formas: pontualmente ou em área. Para a análise dos modelos propostos por este trabalho, foi escolhido a segunda forma de obtenção de dados. Logo, a região compreendida pelos Estados de São Paulo, Rio de Janeiro, Minas Gerais, Mato Grosso do Sul e Paraná foi escolhida devido a ser uma região de geografia diversa e compreender uma população estimada em mais de 44 milhões de pessoas, cerca de

¹ Códigos do projeto desenvolvido. Disponível em: <https://github.com/NSG0409/pfc>. Acesso em: 20 out. 2024.

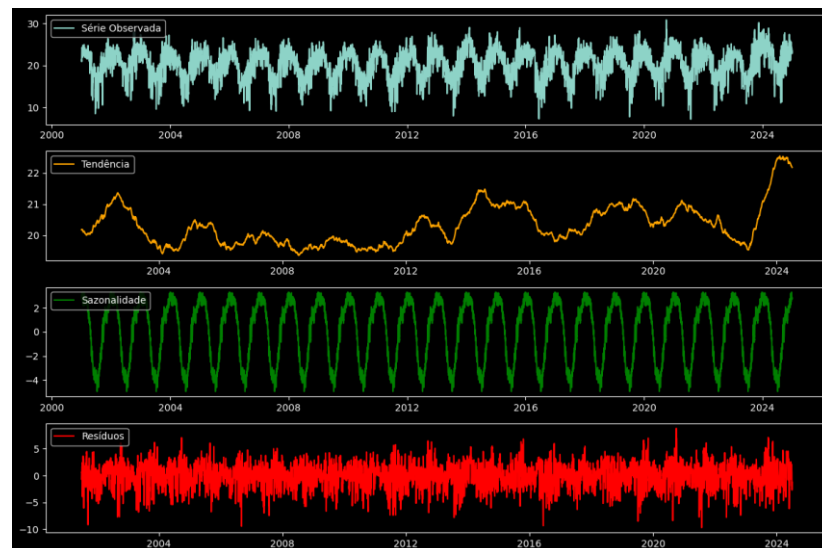
² Dados e informações da plataforma POWER. Disponível em: <https://power.larc.nasa.gov/>. Acesso em: 18 nov. 2024.

- WS50M: velocidade do vento a 50 metros;
- ALLSKY_SFC_UV_INDEX: índice de UV ao nível da superfície;
- TOA_SW_DWN: Radiação solar descendente no topo da atmosfera (W/m^2);
- CLRSKY_SFC_SW_DWN: Radiação de onda curta descendente na superfície em céu limpo (W/m^2);
- RH2M: Umidade relativa do ar a 2 metros (%).

Os dados obtidos compreendem o ano de 2001 a 2024, de acordo com a disponibilidade dos dados na plataforma POWER, cujo limite de requisições de dados eram (na época ao qual os dados foram extraídos via API) de 22 anos. Períodos anteriores de dados apresentaram alguns problemas na sua coleta, conforme explicitamente informado na plataforma, portanto foram descartados.

A Figura 4 apresenta a decomposição da série temporal dos dados da temperatura à 2 metros, de 2001 a 2024, utilizados em seus principais componentes: tendência, sazonalidade e resíduos. O método de decomposição da série temporal utilizado foi o aditivo e a decomposição é baseada em médias móveis de acordo com a biblioteca utilizada, Statsmodels⁴.

Figura 4 - Decomposição da série temporal



Fonte: Elaborado pelo autor (2025).

⁴ Parâmetros e definições da biblioteca Statsmodels para decomposição de séries temporais disponível em: https://www.statsmodels.org/stable/generated/statsmodels.tsa.seasonal.seasonal_decompose.html#statsmodels.tsa.seasonal.seasonal_decompose-parameters. Acesso em: 18 abr. 2025.

A curva em azul claro apresenta os dados originais da série temporal ao longo dos anos, com sazonalidades anuais bem definidas e tendência de longo prazo. Pela análise das curvas é possível notar que os padrões consistentes condizem com as estações do ano, com períodos de maior aquecimento da superfície (coincidindo com fim e início de um ano) e períodos de menor aquecimento (coincidindo com o período do meio de um ano).

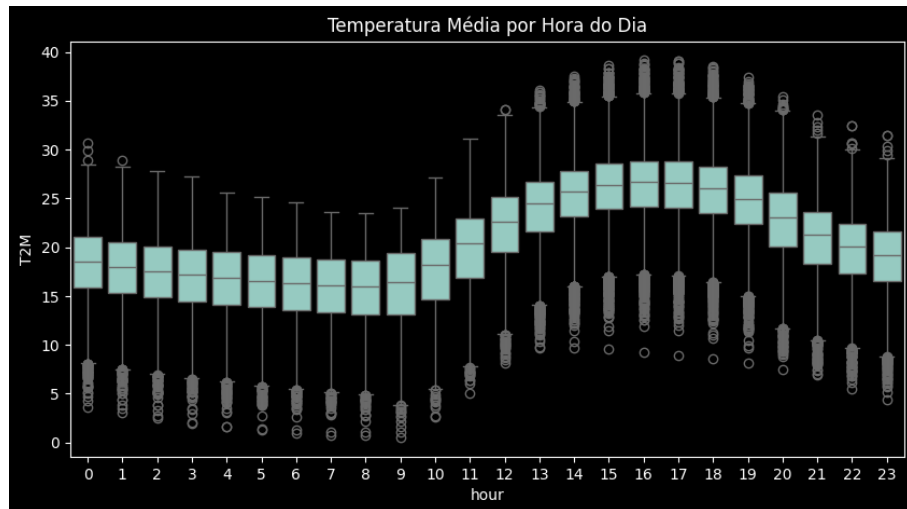
O gráfico laranja apresenta a tendência dos dados de temperatura ao longo dos anos (2001 a 2024) descartando a sazonalidade e os ruídos da informação. A partir da análise visual dos dados, é possível notar que entre os anos de 2004 à 2012 a tendência da temperatura se manteve estável, entretanto, a partir de 2012 nota-se uma tendência de aquecimento, com um pico notável em 2024.

A curva em verde, por sua vez, apresenta em detalhes a sazonalidade dos dados ao longo dos anos. É possível notar o padrão repetitivo que ocorre em intervalos regulares de tempo. A amplitude da variação sazonal pode estar relacionada a mudanças nos padrões atmosféricos, como maior incidência de radiação solar no verão e menor no inverno.

Por fim, o gráfico em vermelho apresenta os resíduos da série temporal, removendo a tendência e a sazonalidade a fim de apresentar as variações inesperadas da temperatura. Os picos e vales representam momentos extremos, como ondas de calor ou frio. Nota-se que há maior incidência de eventos extremos conforme o passar dos anos, com maiores incidência a partir de 2016.

Analisando em mais detalhes o comportamento da temperatura da série histórica coletada através de imagens de satélite por meio de médias de temperatura da área selecionada, é possível notar a distribuição da temperatura ao longo das 24 horas como pode ser observado na Figura 5.

Figura 5 - Temperatura média por hora do dia.

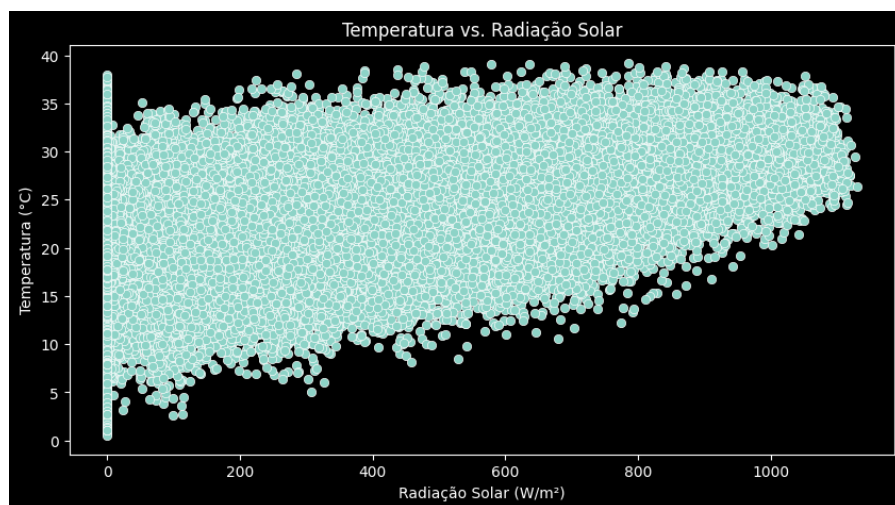


Fonte: Elaborado pelo autor (2025).

O comportamento da temperatura média mostrou o comportamento do ambiente conforme o passar do tempo, começando com uma temperatura média em torno de 20 graus Celsius e tendo um leve decréscimo ao longo das horas da madrugada. Em torno de 8 horas foi possível observar a menor temperatura registrada em média ao longo do dia. Ao longo das horas seguintes a temperatura se eleva até o pico de temperatura registrado entre 15h 17h, que decresce novamente e tem seu ciclo repetido.

A Figura 6 apresenta o comportamento de T2M em relação ao índice de radiação solar média da região de análise.

Figura 6 - Comportamento da temperatura em relação ao índice de radiação solar.

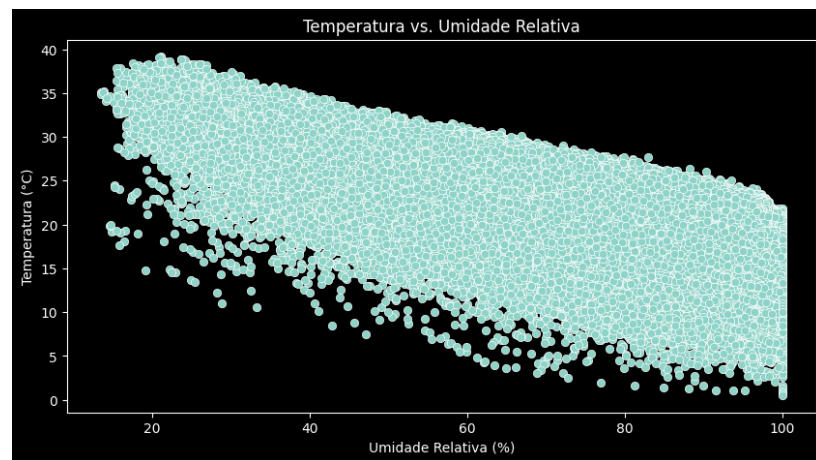


Fonte: Elaborado pelo autor, 2025.

No eixo horizontal apresenta o índice de radiação solar em watts por metro quadrado (W/m^2). O eixo vertical apresenta a temperatura em graus Célsius. Neste gráfico é possível observar a relação proporcional entre a temperatura e o índice de radiação solar, quanto maior o índice maior a temperatura média e vice versa.

A Figura 7 apresenta o comportamento da temperatura média a 2 metros em relação a umidade relativa do ar média da região de análise.

Figura 7 - Comportamento da temperatura em relação ao índice de umidade relativa do ar.

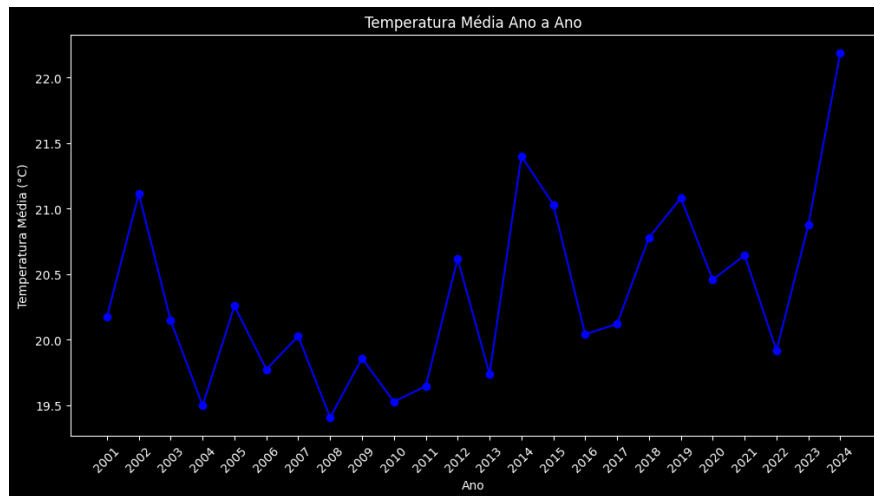


Fonte: Elaborado pelo autor (2025).

No eixo horizontal apresenta o índice de umidade relativa do ar (%). O eixo vertical apresenta a temperatura em graus Célsius. Neste gráfico é possível observar a relação inversamente proporcional entre a temperatura e o índice de umidade relativa do ar, quanto menor o índice maior a temperatura média e vice versa.

A Figura 8 apresenta o valor de temperatura média a 2 metros de 2001 a 2024.

Figura 8 - Temperatura média ano a ano.

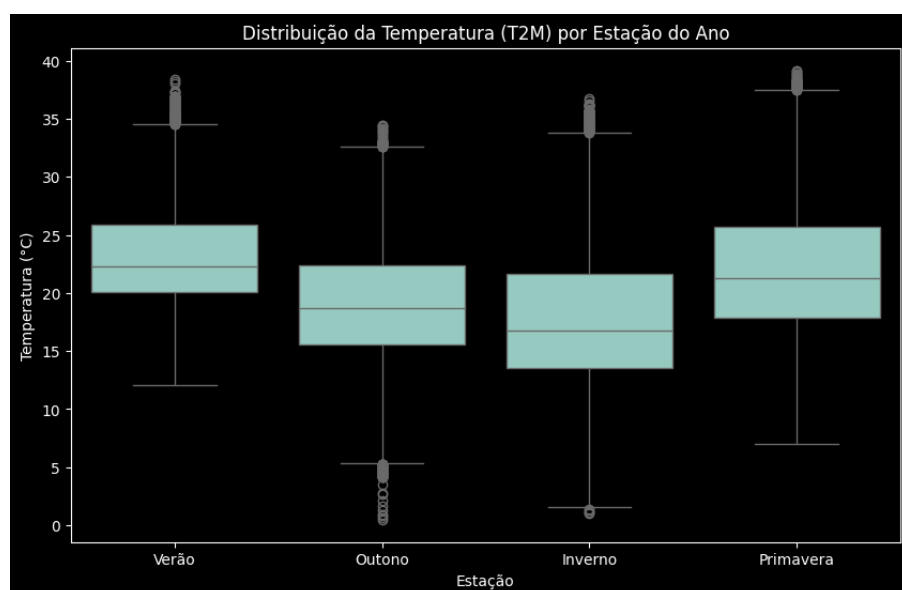


Fonte: Elaborado pelo autor (2025).

O eixo horizontal apresenta o tempo em anos. O eixo vertical apresenta a temperatura em graus Célsius. Neste gráfico é possível observar o aumento da temperatura ao longo dos anos, com picos notáveis em 2014 e 2024.

A temperatura, como uma grandeza sazonal, possui valores diferentes para determinadas estações do ano. A Figura 9 apresenta a distribuição da temperatura média ao longo das quatro estações do ano: verão, outono, inverno e primavera.

Figura 9 - Distribuição da temperatura por estação do ano.



Fonte: Elaborado pelo autor (2025).

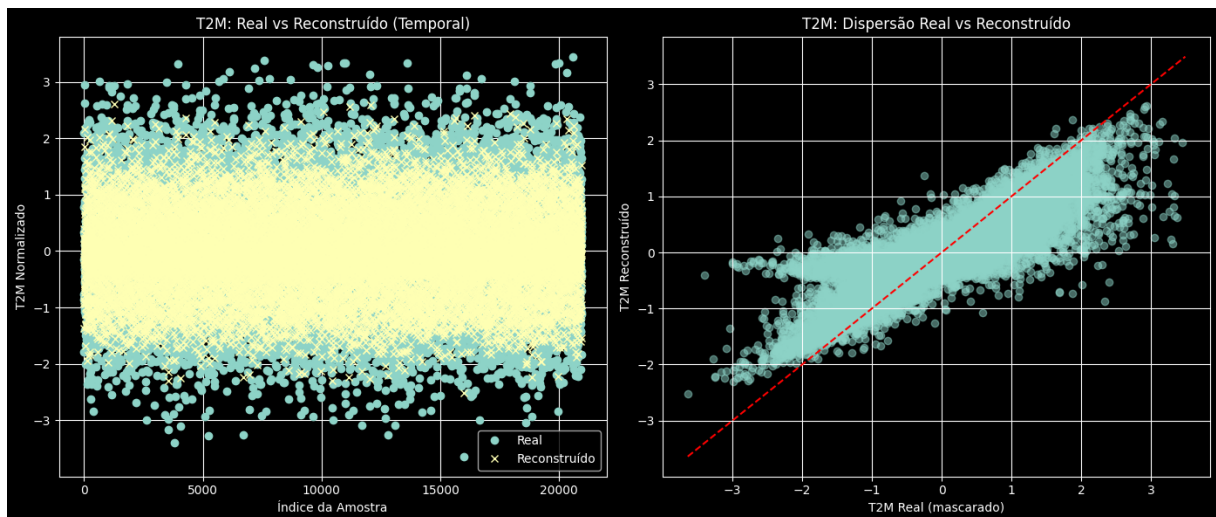
4.3 Dataset reconstruído

Como citado anteriormente neste trabalho, o clima é um sistema complexo ao qual a combinação de suas propriedades traz à tona comportamentos que não podem ser previstos pela simples análise de seus componentes de forma separada. As mudanças climáticas, por sua vez, adicionam uma complexidade ainda maior para o sistema, tornando características antes conhecidas em fenômenos novos e imprevisíveis.

Por esse motivo, o uso de *Self-Supervised Learning* se faz tão importante na análise do clima, como é proposto por esse trabalho, já que sua capacidade de capturar correlações intrínseca aos dados antigos e novos o torna uma ferramenta tão poderosa para ajudar a compreender e prevenir as adversidades cada vez maiores que as mudanças climáticas trazem para a humanidade.

Assim, o conjunto de dados utilizados também foi submetido a um modelo responsável pela reconstrução dos dados, o *Masked Autoencoder*. Os dados foram reconstruídos e seus valores foram comparados com os dados originais. A Figura 10 a) apresenta a comparação temporal entre os dados reais e reconstruídos e b) a dispersão dos valores registrados pelos reconstruídos.

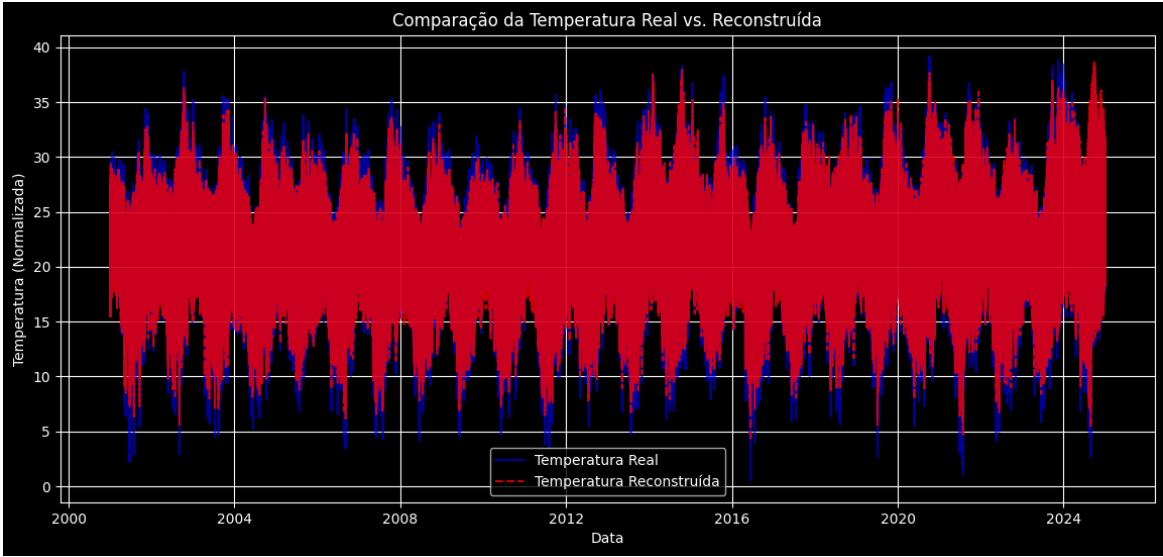
Figura 10 - Comparação dos dados reconstruídos pelos reais em relação a temperatura a 2 metros.



Fonte: Elaborado pelo autor (2025).

A Figura 11 abaixo apresenta um gráfico temporal comparando a temperatura média a 2 metros ao longo dos anos.

Figura 11 - Comparação da temperatura real pela reconstruída.



Fonte: Elaborado pelo autor (2025).

A Tabela 1 apresenta uma amostra de 10 registros dos dados originais, dados reconstruídos e erro de temperatura na base histórica.

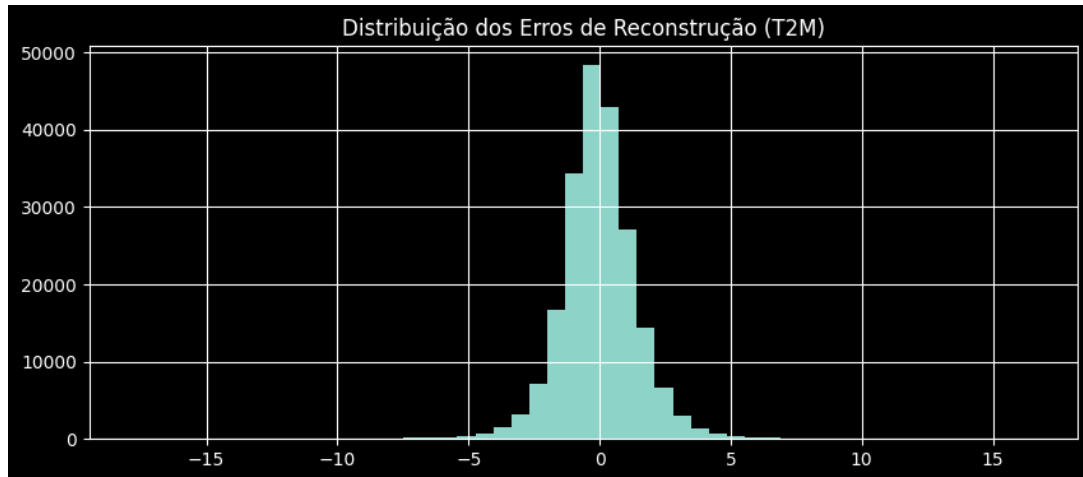
Tabela 1 - Temperatura original, reconstruída e erro.

Temperatura Original	Temperatura Reconstruída	Erro entre valores registrados e reconstruídos
18.93	19.303720	-0.373720
18.54	19.020370	-0.480370
18.21	19.312595	-1.102595
17.93	18.679619	-0.749619
17.70	17.540056	0.159944
17.46	18.624704	-1.164704
17.27	18.130436	-0.860436
17.16	17.827562	-0.667562
17.18	16.613478	0.566522
18.34	18.456207	-0.116207

Fonte: Elaborado pelo autor (2025).

A Figura 12 apresenta o gráfico de distribuição dos erros de reconstrução da temperatura a 2 metros (T2M).

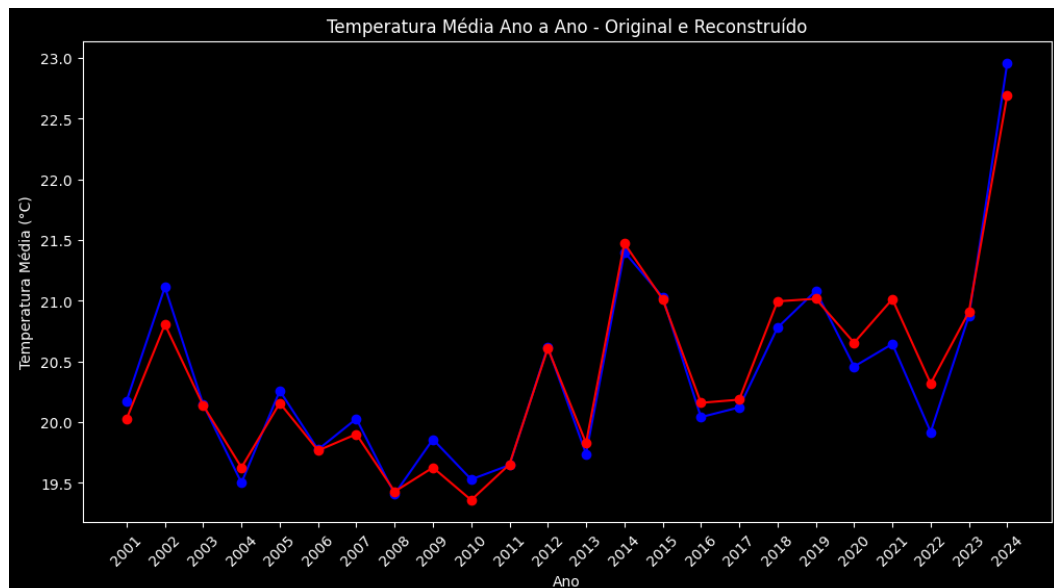
Figura 12 - Distribuição dos erros de reconstrução (T2M).



Fonte: Elaborado pelo autor (2025).

Por fim, a Figura 13 apresenta a evolução da temperatura média ano a ano dos dados originais e dados reconstruídos.

Figura 13 - Evolução da temperatura média ano a ano original e reconstruídos.



Fonte: Elaborado pelo autor (2025).

4.4 Hiperparâmetros

Os algoritmos de *Deep Learning* possuem hiperparâmetros que afetam o desempenho e sua capacidade de generalização para as tarefas propostas. A seguir foram apresentados os hiperparâmetros utilizados em diferentes arquiteturas de redes neurais e modelos de aprendizado.

4.1.1 Random Forest Regressor (RFR)

O modelo *Random Forest Regressor* utilizado possui as seguintes características:

- Número de árvores na floresta: 100 regressores;
- Critério de divisão: Dados de 2001 a 2022 como treino, dados de 2023 a 2024 como teste. Este modelo foi escolhido devido à sua robustez e capacidade de reduzir *overfitting* ao combinar múltiplas árvores de decisão, conforme descrito por Breiman (2001, p.29).

4.1.2 Dense Neural Networks

A topologia da rede neural artificial densas adotada foi estruturada da seguinte forma:

- Camada de entrada: 64 neurônios;
- Segunda camada oculta: 32 neurônios com função de ativação ReLU;
- Camada de saída: 1 neurônio com função de ativação linear;
- Função de erro: *MSE*;
- Épocas de treinamento: 100 épocas;
- Early Stopping: paciência de 10 épocas.

Essa estrutura visa capturar relações complexas entre as variáveis de entrada e fornecer previsões com alto grau de precisão, segundo LeCun, Bengio e Hinton (2015).

4.1.3 Long Short-Term Memory (*LSTM*)

A rede *LSTM* foi configurada da seguinte maneira:

- Camada *LSTM* 128 unidades;
- Camada *LSTM* com 64 unidades;

- Camada densa com 32 neurônios e ativação ReLU;
- Camada densa de saída com ativação ReLU;
- Função de perda: Mean Squared Error (*MSE*);
- Otimizador: Adam;
- Tamanho do batch: 256;
- Número de épocas: 30;
- Early Stopping: paciência de 10 épocas.

A *LSTM* foi escolhida devido à sua capacidade de capturar dependências temporais nos dados, sendo amplamente utilizada para modelagem de séries temporais, segundo Hochreiter e Schmidhuber (1997).

4.1.4 Masked Autoencoder

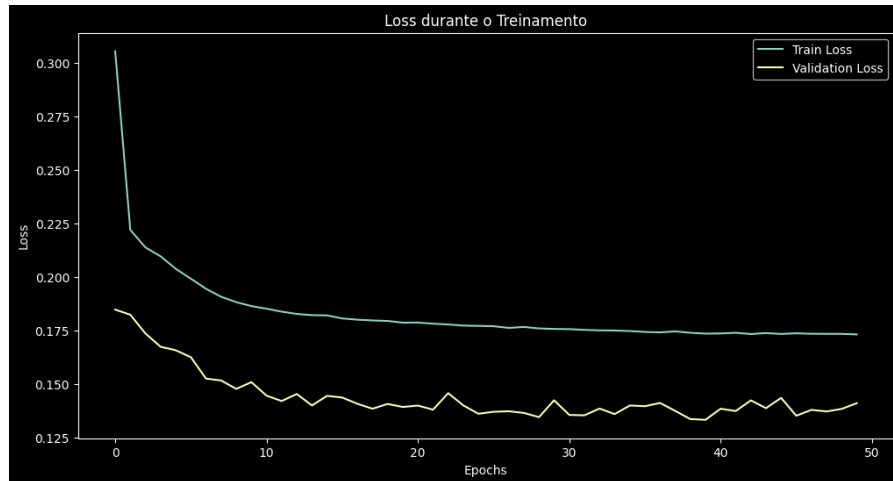
O *masked autoencoder* foi implementado para aprendizado não supervisionado, visando melhorar a representação dos dados e facilitar o treinamento de modelos subsequentes. Sua estrutura é descrita a seguir:

- Encoder: Primeira camada densa com 128 neurônios e ativação ReLU, Regularizer L2 = 0.01. Dropout: 0.2.
- Encoder: Segunda camada densa com 64 neurônios e ativação ReLU.
- Decoder: Primeira camada densa com 128 neurônios e ativação ReLU, Regularizer L2 = 0.01. Dropout: 0.2.
- Decoder: Camada de saída com 20 neurônios. Função de ativação linearç
- Otimizador: Adam;
- Taxa de aprendizado: 0.001.

O *dropout* utilizado foi de 0.2 em todas as camadas do *encoder* para evitar *overfitting*. A taxa de aprendizado da função de otimização Adam foi de 0.001 e função de custo *MSE*. A taxa de mascaramento dos dados foi de 30%, o número de épocas usado no treinamento foi 50 épocas e o *batch-size* de 64.

A Figura 14 apresenta a curva da função de perda durante o treinamento na reconstrução dos dados originais do *dataset* após ser submetido ao *Masked Autoencoder*.

Figura 14 - Evolução da função de perda durante o treinamento.

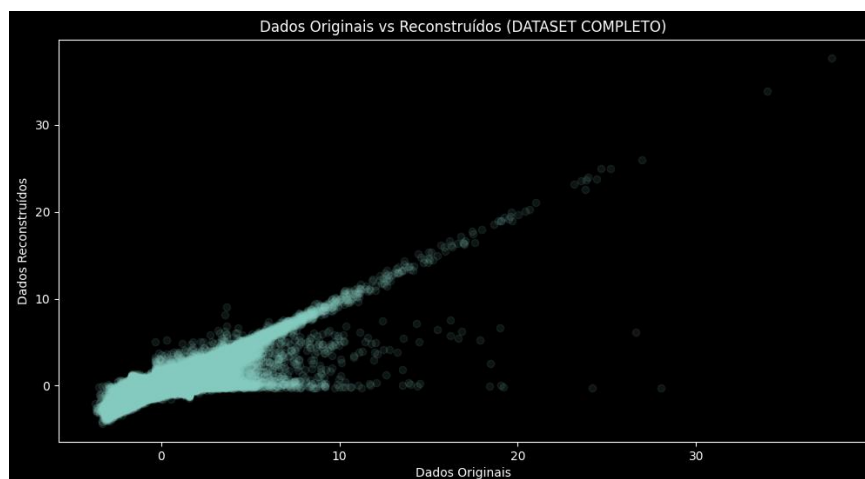


Fonte: Elaborado pelo autor, 2025.

É possível notar que o modelo convergiu em aproximadamente 20 épocas de treinamento, com valores de perda no treinamento maiores que valores de perda na validação, o que demonstra que o modelo não está sofrendo *overfitting*.

A Figura 15 apresenta o gráfico de dispersão dos dados originais e reconstruídos.

Figura 15 - Dispersão dos dados originais e reconstruídos



Fonte: Elaborado pelo autor, 2025.

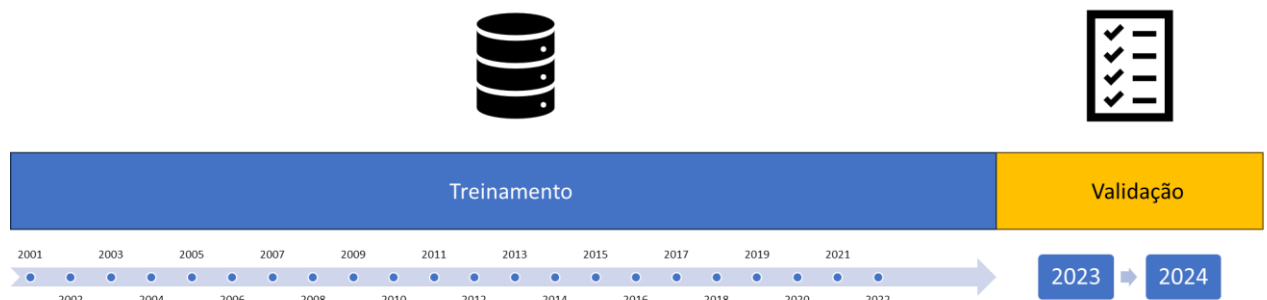
4.5 Estrutura do teste

Para a execução deste trabalho, inicialmente todos os dados foram tratados de forma a serem possíveis de serem processados pelos modelos. Como diferentes grandezas foram medidas em diferentes escalas (por exemplo: temperatura e velocidade do vento) os dados inicialmente foram normalizados por meio de transformações lineares, de forma a assegurar que as escalas não impedissem o desempenho dos modelos. Para garantir a qualidade da predição dos pontos futuros, foram utilizados 24 pontos passados para prever 1 ponto futuro.

Outra característica do pré-processamento dos dados realizada foi a criação de *features* capazes de correlacionar os dados já obtidos para ajudar na predição de eventos próximos, o chamado *lag* temporal. Desta forma, para auxiliar o modelo, as temperaturas de 48h e 72h anteriores foram tratadas como preditores. A média móvel e o desvio padrão das últimas 24h de temperatura também foram registrados.

Desta forma, com todo o procedimento de pré-processamento, o conjunto de dados foi enfim dividido em conjunto de treino (dados de 2001-2022) e conjunto de teste (dados do ano de 2023 e 2024), como pode ser visto na Figura 16.

Figura 16 - Distribuição dos dados do modelo.

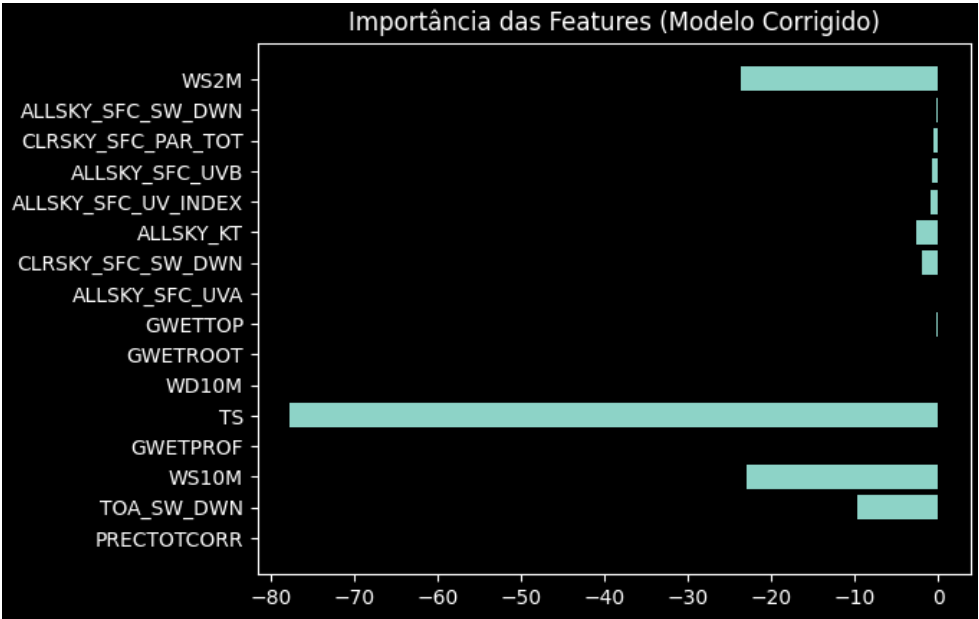


Fonte: Elaborado pelo autor (2025).

4.6 Features conflitantes

Durante a análise das features de ambos os datasets, notou-se a presença de ruídos que impediam o bom funcionamento dos modelos em suas fases iniciais. Após uma análise mais detalhada das features mais importantes positiva e negativamente, obteve-se o seguinte resultado. O Apêndice B traz o significado das siglas utilizadas na figura a seguir.

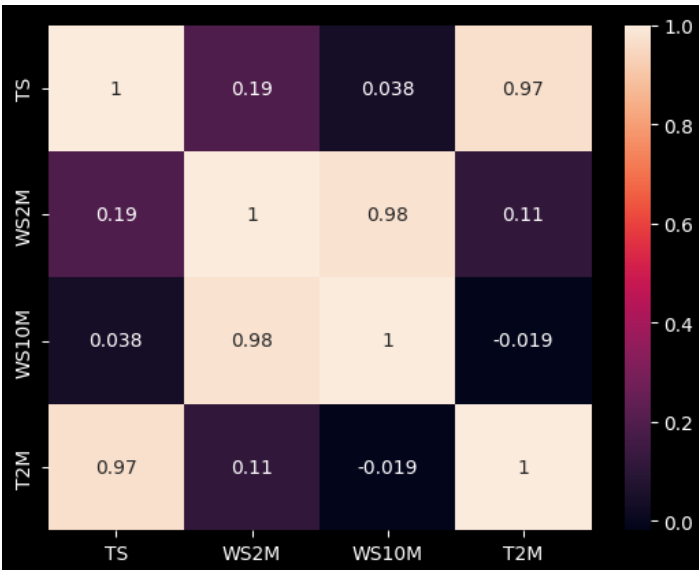
Figura 17 - Importância das features.



Fonte: Elaborado pelo autor (2025).

Pela análise da Figura 17, observou-se que existiam muitas features que causavam ruído aos modelos e reduziam o seu desempenho, em especial as features: TS, WS2M (velocidade do vento a dois metros) e WS10M (velocidade do vento a 10 metros). O valor tão expressivo para TS pode ser explicado pela alta correlação entre a classe (TS) e o *target* (T2M), conforme pode ser observado na Figura 18, que apresenta a matriz de correlação destas features com o target.

Figura 18 - Matriz de correlação das features problema com o target.

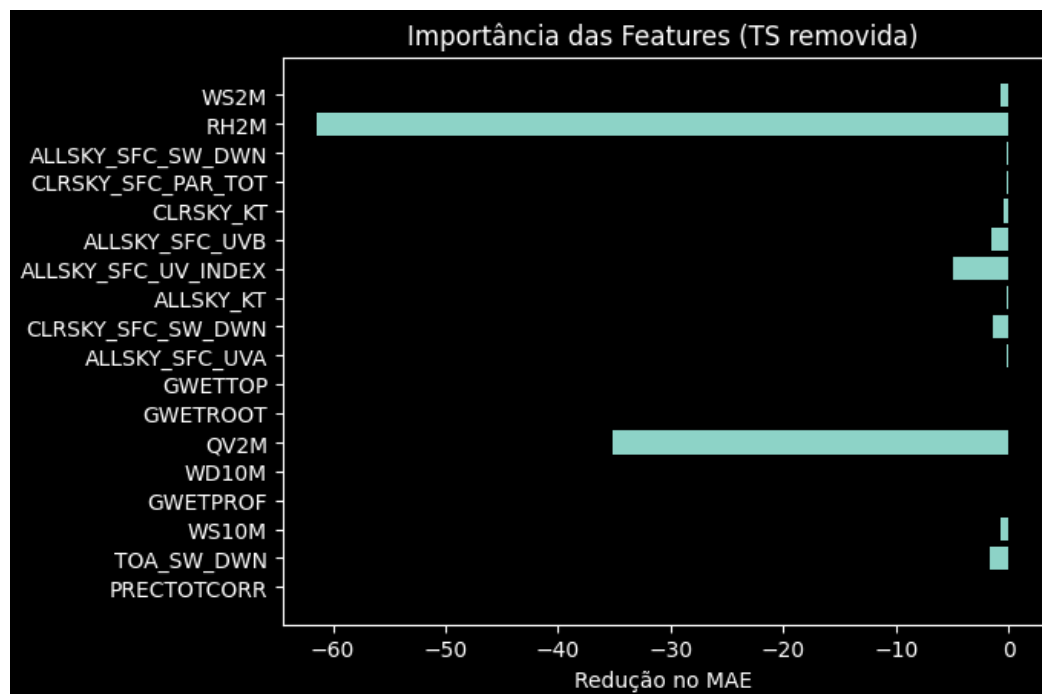


Fonte: Elaborado pelo autor (2025).

Verificou-se que a correlação entre as variáveis TS e T2M é de 0,97, valor considerado extremamente elevado, o que compromete a capacidade de generalização e previsão do modelo. Tal correlação acentuada ocasiona um desbalanceamento na rede em favor da variável TS, reduzindo a influência das demais variáveis presentes no conjunto de dados. Diante disso, optou-se pela remoção da variável TS, com o objetivo de reavaliar as importâncias das variáveis nos modelos.

A Figura 19 apresenta a *feature importance* do modelo de LSTM após a remoção de TS.

Figura 19 - Feature Importance após a remoção de TS.

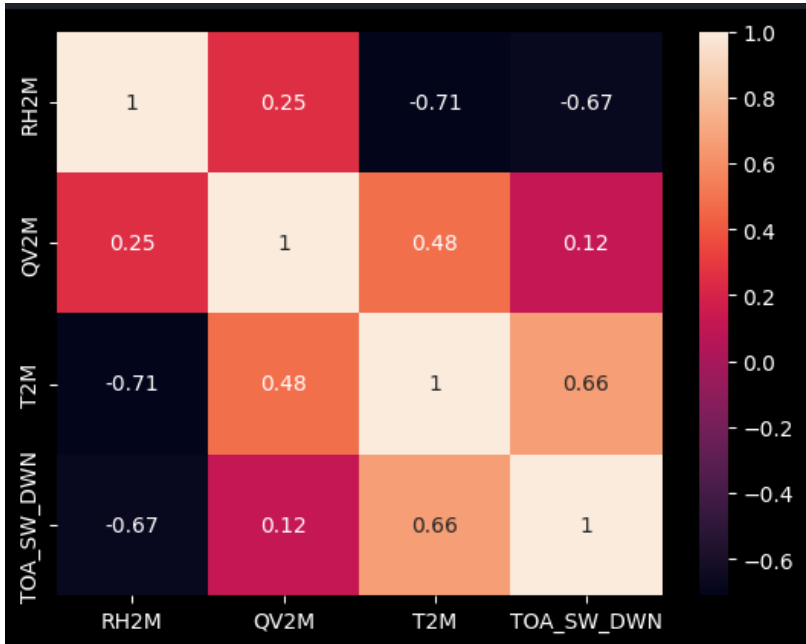


Fonte: Elaborado pelo autor (2025).

Após a remoção de TS, a feature RH2M (umidade a 2 metros) apresentou um comportamento muito semelhante a TS, apesar dessa influência negativa ser contraintuitiva quando se pensa em sistemas climáticos.

A Figura 20 apresenta a matriz de correlação destas features com o target.

Figura 20 - Matriz de correlação das features problema com o target.

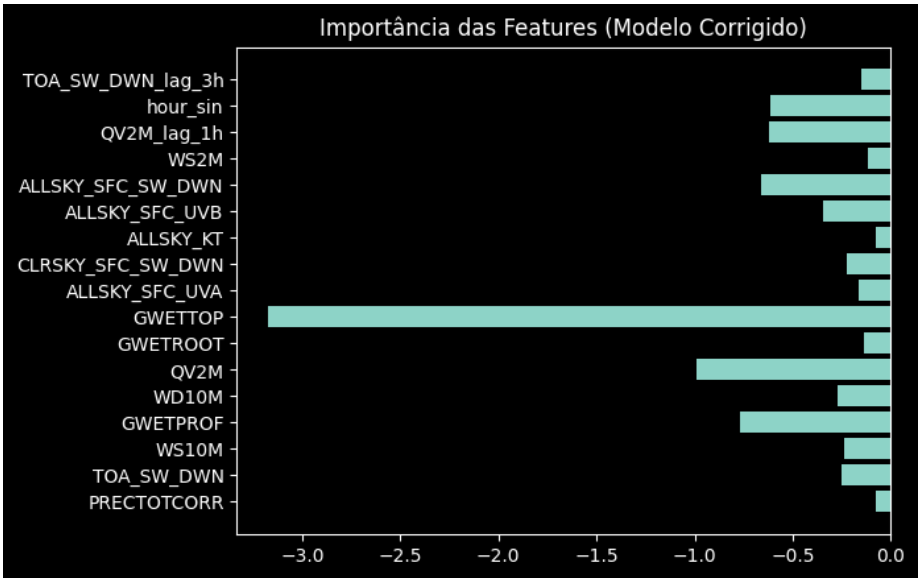


Fonte: Elaborado pelo autor (2025).

Novamente, notou-se uma importância extremamente negativa para RH2M valores de correlação considerados ruins entre QV2M, TOA_W_DWM e T2M. Desta forma, optou-se pela remoção destas features dos datasets para o treinamento dos modelos.

A Figura 21 apresenta o novo *feature importance* do modelo após as correções.

Figura 21 - Feature Importance após as correções.



Fonte: Elaborado pelo autor (2025).

Por fim, nota-se que mesmo após a remoção das features problemáticas do dataset, ainda há features ruidosas nos modelos. Entretanto, apesar de possivelmente melhorar o desempenho dos modelos caso sejam removidas, esta atitude descaracteriza significativamente o dataset original e, portanto, foram mantidas nos testes futuros.

5 RESULTADO DOS EXPERIMENTOS

Os experimentos realizados contemplaram diferentes arquiteturas, incluindo modelos baseados em árvores de decisão (*Random Forest*), redes neurais profundas (*DNN*) e redes neurais recorrentes (*LSTM*), bem como uma proposta inédita que combina *Masked Autoencoder* com aprendizado auto-supervisionado (*SSL*) e redes *LSTM*. Dada a originalidade do método proposto, não foram identificados na literatura modelos que apresentem arquitetura e configuração experimental idênticas.

Dessa forma, optou-se por realizar comparações indiretas com abordagens semelhantes, disponíveis em trabalhos relacionados da literatura, como o de Oloyede *et al.* (2023), os quais compartilham o mesmo objetivo preditivo e o uso de técnicas de aprendizado de máquina, entretanto possuem suas diferenças, como: tamanho do dataset, área de estudo, frequência dos dados e quantidade de classes e *target*. A análise comparativa foi conduzida com base em métricas quantitativas amplamente utilizadas, como *MAE* (erro médio absoluto), *RMSE* (erro quadrático médio) e R^2 (coeficiente de determinação). Tais indicadores permitiram avaliar o desempenho relativo dos modelos, demonstrando a eficácia da abordagem proposta frente às estratégias já consolidadas na literatura.

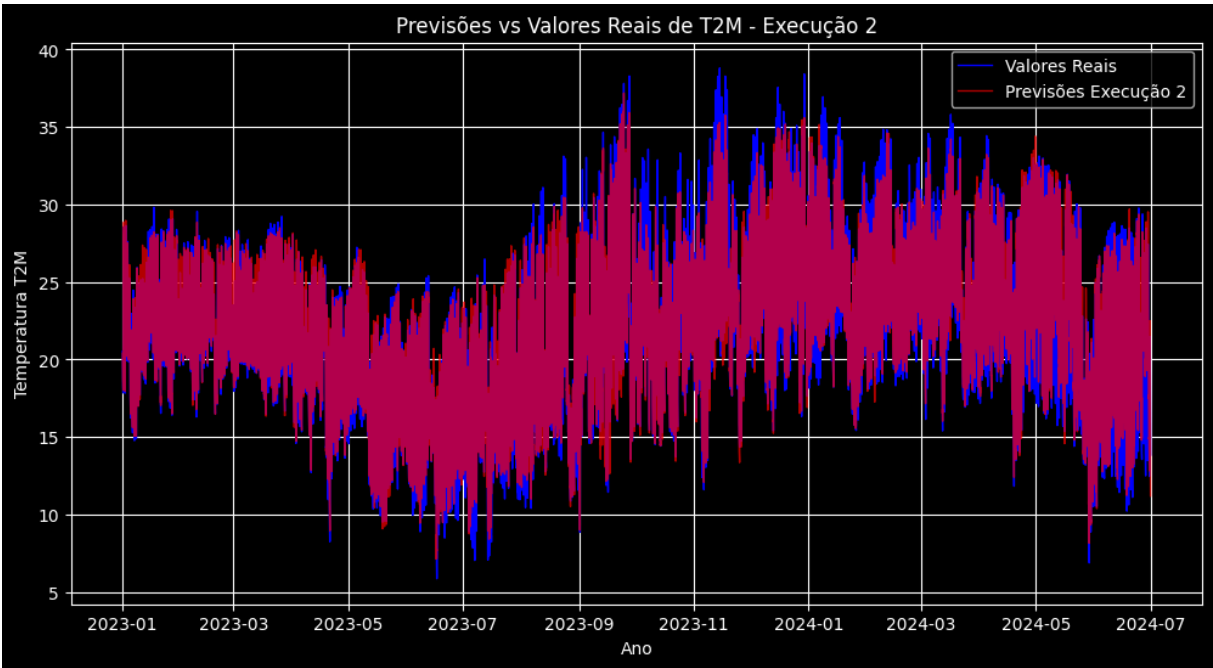
5.1 Random Forest Regressor

5.1.1 Treinamento com dados originais

Os modelos foram treinados com a função de perda *MSE* ou erro médio quadrático e avaliados com a métrica *MAE*, que são quantificadores expressivos para análise de séries temporais complexas, como dados climáticos.

A Figura 22, portanto, apresenta o desempenho (de forma gráfica) do *Random Forest Regressor* em comparação com os dados reais medidos para os anos de 2023 a 07/2024.

Figura 22 - Comparação entre valores registrados e previstos de temperatura usando Random Forest Regressor sem SSL.



Fonte: Elaborado pelo autor (2025).

A Figura 22 é composta pela sobreposição dos valores previstos (em vermelho) pelos valores registrados (em azul). A escala impressa em temperatura pelo tempo em dias ao longo do ano.

Foi possível observar que para o primeiro experimento, o *Random Forest Regressor* performou bem em comparação com os valores registrados, entretanto, para determinado período do ano notou-se que, apesar de prever com certa exatidão momentos de maior ou menor aquecimento da temperatura, não foi possível prever sua intensidade ao longo do tempo. Nota-se também que em regiões da série temporal onde há pouca variabilidade dos dados a sua precisão foi positiva. O modelo pareceu ser capaz de captar bem a faixa geral de valores da temperatura (média e tendências sazonais).

Para garantir a assertividade do modelo *Random Forest Regressor* e a replicabilidade dos resultados obtidos o modelo em questão foi testado cinco vezes e coletado os valores médios e desvio padrão dos resultados obtidos.

A Tabela 2 abaixo apresenta os valores de *MAE* e *MSE* ao longo dos testes e a Tabela 3 apresenta os resultados médios do modelo.

Tabela 2 - Resultados do modelo Random Forest Regressor sem SSL.

Execução	MAE	MSE	R ²
1	1,2181	3,0182	~90%
2	1,2175	3,0190	~90%
3	1,2212	3,0512	~90%
4	1,2225	3,0457	~90%
5	1,2197	3,0273	~90%

Fonte: Elaborado pelo autor (2025).

Tabela 3 - Média e desvio padrão do modelo Random Forest Regressor sem SSL.

Média do Modelo	Desvio Padrão do Modelo
Média MAE	1,2198 ± 0,0019
Média MSE	3,0323 ± 0,0137

Fonte: Elaborado pelo autor (2025).

Pela análise dos resultados dos testes observados na Tabela 3 com o *Random Forest Regressor*, foi possível observar valores baixos de *MAE*, indicando que o modelo erra entre 1,2175 °C e 1,225 °C, um valor relativamente baixo para modelos de previsão de temperatura, mas com espaço para melhorias. Olhando para valores médios de *MAE*, o modelo erra em média 1,2198 °C e o desvio padrão de ± 0,0019, mostrando que o modelo é estável e não sofre muitas variações entre splits de treino e teste. O *MSE* médio (de acordo com a Tabela 3) foi de 3,0323 e desvio padrão de ± 0,0137. Como o *MSE* penaliza erros grandes mais severamente, isso indica que o modelo raramente comete erros muito altos (ex.: > 5°C). Pela análise dos coeficientes de

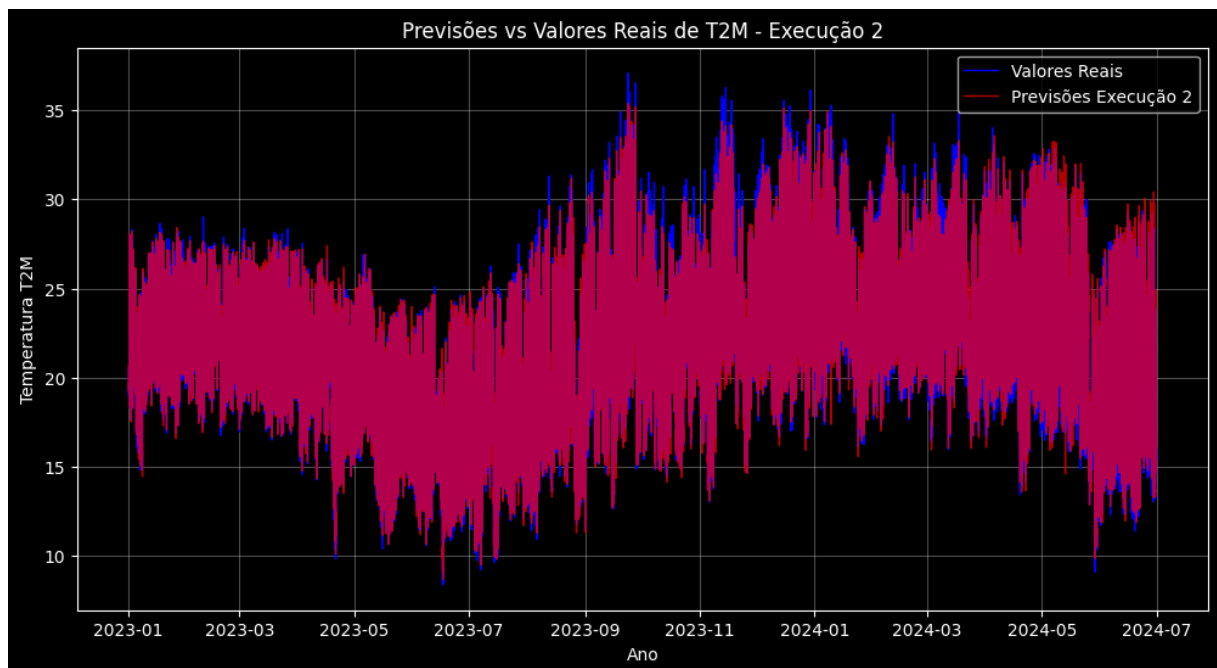
determinação (R^2), valores de > 0.9 indicam que o modelo explica $\sim 90.6\%$ da variância dos dados e que o modelo captura bem as relações entre as features e o target.

Logo, nota-se pela análise dos dados que o modelo de *Random Forest* teve um desempenho bom, apesar de haver margem para a melhora das previsões com um *fine-tuning* de hiperparâmetros.

5.1.2 Treinamento com dados reconstruídos

A Figura 23 apresenta o mesmo modelo treinado com dados reconstruídos pelo *Masked Autoencoder* e submetidos ao modelo de *Random Forest*.

Figura 23 - Comparação entre valores registrados e previstos de temperatura usando Random Forest Regressor com SSL.



Fonte: Elaborado pelo autor (2025).

O gráfico acima apresenta os valores registrados de temperatura a 2 metros no solo pelos valores previstos pelo modelo *DNN*. A curva em azul apresenta os dados reais para o período e a curva em vermelho os dados previstos pelo modelo.

Como pode ser observado, o modelo que utiliza técnicas de *Self-Supervised Learning* na composição do dataset obteve um desempenho visualmente superior ao modelo treinado com dados originais. Entretanto foi possível observar que devido a reconstrução dos dados em

relação aos originais, a escala de temperatura foi deslocada em +5 graus no limite inferior e -5 graus no limite inferior. Nota-se que os ruídos presentes no dataset reconstruído foram reduzidos.

As Tabela 4 e Tabela 5 a seguir apresentam os resultados individuais e médios do Regressor sob nova situação.

Tabela 4 - Resultados do modelo Random Forest Regressor com SSL.

Execução	MAE	MSE	R^2
1	0,5961	0,8020	~96%
2	0,5946	0,8037	~96%
3	0,5959	0,8022	~96%
4	0,5956	0,8029	~96%
5	0,5948	0,7992	~96%

Fonte: Elaborado pelo autor (2025).

Tabela 5 - Média e desvio padrão do modelo Random Forest Regressor com SSL.

Média do Modelo	Desvio Padrão do Modelo
Média MAE	0,5954 ± 0,0006
Média MSE	0,8020 ± 0,0015

Fonte: Elaborado pelo autor (2025).

Pela análise dos resultados dos testes em Tabela 4 e com o *Random Forest Regressor*, foi possível observar valores ainda mais baixos de *MAE*, variando entre 0,5946 °C e 0,5961 °C, o erro médio absoluto foi de 0,5954 ± 0,0006. Ou seja, o modelo com os dados reconstruídos erra em média 0,5954 °C e o baixo desvio padrão indica que as execuções foram consistentes

entre os testes. Pela análise dos valores médios de *MSE*, nota-se que os erros grandes foram reduzidos pela aplicação do *Masked Autoencoder*. Pela análise dos coeficientes de determinação (R^2) em torno de $\sim 96\%$, indicando que o modelo explica 96% da variância nos dados. Também se nota a estabilidade entre as execuções, com resultados bem próximos entre si, o que sugere que o modelo está devidamente ajustado e não é sensível a variações nos processos de treinamento

Quando comparando o desempenho dos modelos treinados com os dados originais e com os dados reconstruídos com *SSL* nota-se que erro médio absoluto caiu de **$\sim 1,2198$ para $\sim 0,5954$** , uma redução de cerca de 51%, indicando que os dados reconstruídos pelo *Masked Autoencoder* permitiram que a *Random Forest* fizesse previsões muito mais precisas. Quando analisando o *MSE*, nota-se que o valor caiu de **$\sim 3,0323$ para $\sim 0,8020$** , uma melhora ainda mais expressiva (74%), sugerindo que os erros maiores foram reduzidos e que o modelo tem menos previsões com grandes desvios. Nota-se uma alta consistência entre execuções.

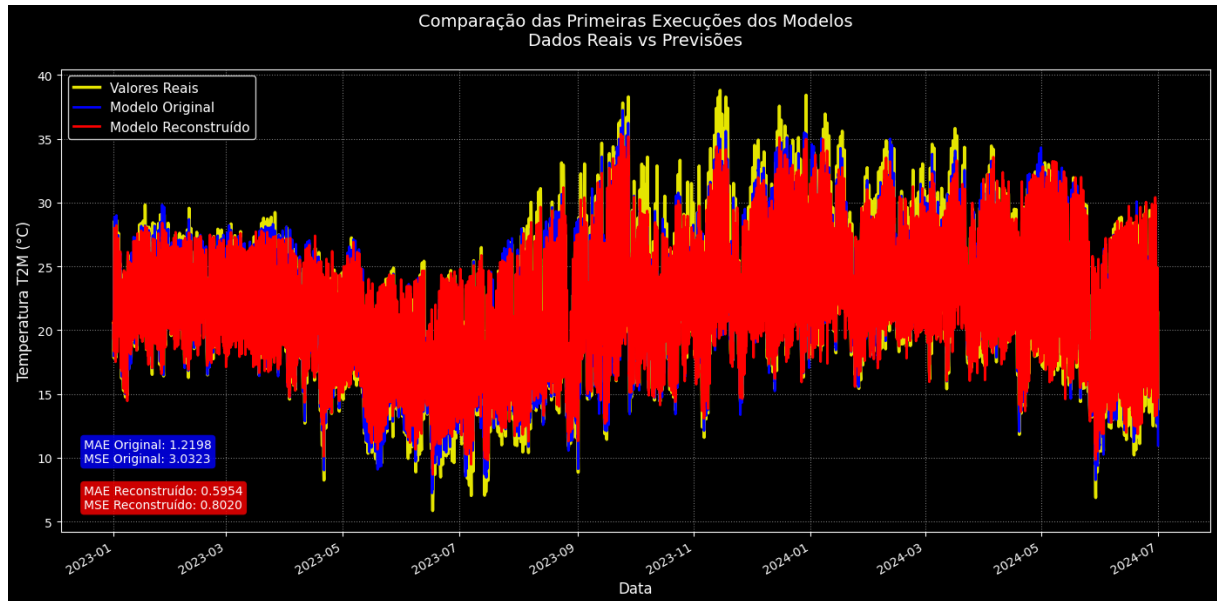
Essa redução expressiva dos valores de *MAE* pode indicar que o *Masked Autoencoder* pode ter reduzido ruídos nos dados meteorológicos originais, deixando apenas os padrões mais significativos e a extração de representações mais compactas e úteis das variáveis climáticas, facilitando a previsão do *Random Forest Regressor*.

O *Masked Autoencoder* parece ter sido muito eficiente na remoção de ruídos ou na captura de padrões ocultos dos dados, resultando em entradas mais limpas para a *Random Forest*. O *Masked Autoencoder* reconstruiu os dados de maneira que suavizou flutuações aleatórias ou medições ruidosas. O modelo conseguiu aprender representações mais informativas, preservando características importantes para a previsão.

5.1.3 Comparação entre os modelos sem e com *SSL*

A Figura 24 apresenta um comparativo entre o desempenho do *Random Forest Regressor* com e sem técnicas de *Self-Supervised Learning*.

Figura 24 - Comparação de desempenho do Random Forest Regressor com e sem SSL.



Fonte: Elaborado pelo autor (2025).

A curva em amarelo representa os dados de temperaturas originais para os anos de 2023 e 2024. A curva em azul representa os dados treinados pelo modelo de *Random Forest* utilizando o dataset original e a curva em vermelho representa os dados treinados pelo modelo de *Random Forest* utilizando o dataset reconstruído pelo *Masked Autoencoder*.

Uma análise quantitativa revela diferenças marcantes no desempenho dos dois modelos. O *Random Forest* convencional, trabalhando com os dados brutos, apresentou um *MAE* de 1,2198 e um *MSE* de 3,0323. Em contraste, a versão que utilizou os dados processados pelo *Masked Autoencoder* alcançou métricas significativamente melhores: *MAE* de 0,5954 (redução de 51%) e *MSE* de 0,8020 (redução de 74%). Essa melhoria expressiva nas métricas de erro demonstra claramente o impacto positivo do pré-processamento com *autoencoder* na qualidade das previsões.

A comparação visual das curvas no gráfico corrobora esses resultados numéricos. A curva vermelha (modelo com *SSL*) mostra uma aderência notavelmente maior aos valores registrados (amarelo), mantendo uma trajetória suave e coerente ao longo de todo o período analisado.

A superioridade do modelo com *SSL* torna-se ainda mais evidente ao examinar seu comportamento frente a valores extremos. A redução de 74% no *MSE* indica uma capacidade

muito superior de lidar com situações atípicas, que normalmente representam um desafio para modelos preditivos. Enquanto a curva azul exibe diversos picos discrepantes, a vermelha mantém uma trajetória estável e confiável, mesmo em condições meteorológicas mais extremas.

Esses resultados sugerem fortemente que o *Masked Autoencoder* atuou de maneira eficaz na purificação dos dados climáticos. O processo de reconstrução parece ter filtrado ruídos e inconsistências presentes nos dados originais, preservando, porém os padrões meteorológicos essenciais. Essa "limpeza" dos dados permitiu ao algoritmo *Random Forest* extrair relações mais significativas entre as variáveis, resultando em previsões mais precisas e confiáveis.

A maior estabilidade temporal demonstrada pelo modelo com *SSL* é particularmente valiosa para aplicações práticas em meteorologia. A redução das oscilações nas previsões aumenta a confiabilidade do modelo para fins de planejamento e tomada de decisão, especialmente em contextos sensíveis às variações climáticas, como agricultura ou gestão de recursos energéticos.

Em síntese, a comparação entre as duas abordagens revela que a incorporação do *Masked Autoencoder* no pipeline de processamento elevou significativamente a qualidade das previsões do *Random Forest*. Os resultados não apenas validam a eficácia desta técnica de aprendizado auto supervisionado para tarefas de previsão climática, mas também destacam a importância de etapas avançadas de pré-processamento no fluxo de trabalho de *machine learning* aplicado a dados meteorológicos.

5.1.4 Comparação com Oloyede *et al.* (2023)

Apesar da semelhança entre os problemas trabalhados, não foram encontrados na literatura trabalhos utilizando as mesmas técnicas e área de estudo iguais aos desenvolvidos neste trabalho. Entretanto, o trabalho de Oloyede *et al.* (2023) possui semelhanças que podem ser comparadas, dada suas características individuais, aos resultados obtidos pelo presente trabalho. A Tabela 6 abaixo mostra a composição dos datasets comparados, as principais diferentes entre os dados e os modelos.

Tabela 6 - Comparação entre os datasets.

Característica	Trabalho desenvolvido	Trabalho de Oloyede <i>et al.</i> (2023)
Período	01/01/2001 a 01/08/2024	01/01/2010 a 17/08/2017
Frequência	Horária	Diária
Número de Registros	~200.000	~3000
Área de estudo	Partes da região centro-oeste, sudeste e sul do Brasil	Victor Attah International Airport, Nigéria
Modelo	Random Forest Regressor	Decision Tree Regression e XGBoost Regression

Fonte: Elaborado pelo autor (2025) com informações de Oloyede *et al.* (2023).

Ao comparar a literatura com o trabalho desenvolvido, nota-se algumas divergências entre as propostas. O trabalho de Oloyede *et al.* (2023) possui um dataset com frequência temporal diferente, o que interfere na quantidade de registros que são atribuídos aos modelos. Outra característica importante que difere os dois trabalhos é a área de estudo: Partes da região centro-oeste, sudeste e sul do Brasil e a região do Aeroporto Internacional Victor Attah, na Nigéria, que possuem características climáticas diferentes entre si.

Ao analisar os resultados apresentados por Oloyede *et al.* (2023), observa-se uma semelhança nas abordagens adotadas em ambos os trabalhos, especialmente no que se refere ao uso de modelos baseados em árvores de decisão. No presente estudo, o modelo Random Forest Regressor obteve desempenho expressivo, com MAE médio de 0,5954, MSE médio de 0,8020 e coeficiente de determinação (R^2) de 0,96, indicando elevada precisão nas previsões e forte capacidade explicativa do modelo sobre a variabilidade dos dados.

Em comparação com os resultados reportados por Oloyede *et al.* (2023), observa-se que o modelo de Árvore de Decisão Regressora aplicado à previsão de temperatura máxima obteve MAE de 1,471 °C e R^2 de 0,531, enquanto para a temperatura mínima, os resultados foram ainda menos expressivos (MAE de 0,805 °C, R^2 de 0,253). Da mesma forma, o modelo

XGBoost apresentou desempenho semelhante ou inferior, com MAE de 1,444 °C e R² de 0,551 para a temperatura máxima, e MAE de 0,816 °C e R² de 0,211 para a temperatura mínima.

A Tabela 7 abaixo apresenta uma comparação entre os resultados dos modelos utilizados neste trabalho e os resultados obtidos por Oloyede *et al.* (2023). Os valores de *RMSE* do trabalho desenvolvido foram calculados em cima dos valores de *MSE* apresentados anteriormente, para que as informações sejam apresentadas em uma mesma grandeza.

Tabela 7 - Comparação entre os resultados obtidos e a literatura (RF).

Modelo	MAE	RMSE	R ²
RF Regressor (p/ temp. média)	1,2198	1,7415	~90%
RF Regressor + SSL (p/ temp. média)	0,5954	0,8955	~96%
Decision Tree (p/ temp. máx) ¹	1,471	1,863	53,1%
Decision Tree (p/ temp. min) ¹	0,805	1,150	25,3%
XGBoost (p/ temp. máx) ¹	1,444	1,822	55,1%
XGBoost (p/ temp. min) ¹	0,816	1,182	21,1%

Fonte: Elaborado pelo autor (2025) com informações de Oloyede *et al.* (2023).¹.

Analisando os dados apresentados, nota-se que, considerando a heterogeneidade dos dados utilizados, o modelo *Random Forest* + *SSL* desenvolvido neste trabalho é mais eficaz na captura de padrões complexos e não lineares da temperatura média, resultando em menores erros absolutos e quadráticos e um R² significativamente superior. Todavia, as condições de teste dos modelos são completamente diferentes, visto que a quantidade de dados, frequência, configuração de redes neurais artificiais simples e *DNN* é diferente.

Portanto, apesar de visualmente os esforços apresentados neste trabalho apresentarem um resultado superior, é necessário futuramente comparar as metodologias utilizando os mesmos dados e condições ambientais para avaliar o possível ganho dos modelos utilizando SSL.

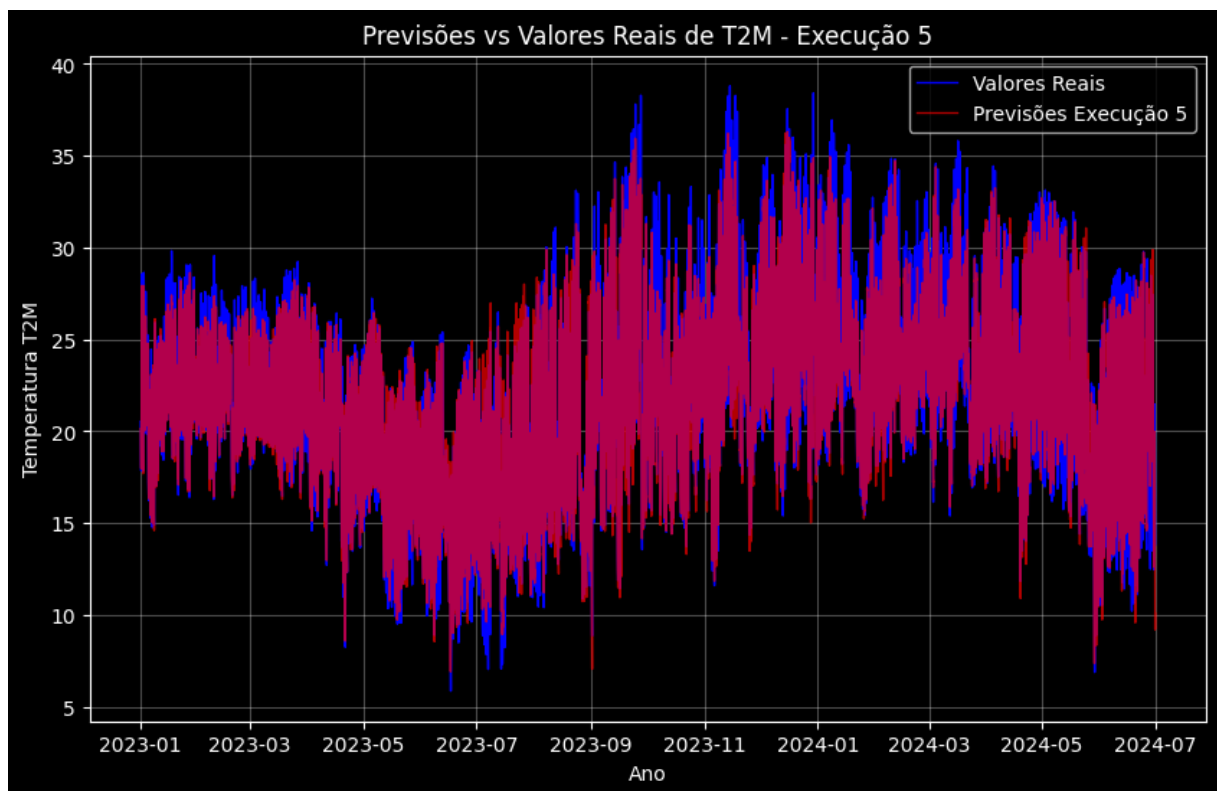
5.2 Dense Neural Networks (DNN)

5.2.1 Treinamento com dados originais

Em seguida, os dados tratados anteriormente passaram pelo modelo de redes neurais artificiais para a predição da temperatura. Foi utilizado o *TensorFlow/Keras* como framework de trabalho. A rede neural foi desenhada contendo 64 neurônios na entrada, a segunda camada com 32 e camada de saída com 1 neurônio por se tratar de uma tarefa de regressão.

Para garantir a assertividade do modelo *DNN* e a replicabilidade dos resultados obtidos o modelo em questão foi testado cinco vezes e coletado os valores médios e desvio padrão dos resultados obtidos. A Figura 25 apresenta os valores de temperatura (T2M) reais em comparação com os valores previstos pelo modelo.

Figura 25 - Comparação entre valores registrados e previstos de temperatura usando *DNN* sem SSL.



Fonte: Elaborado pelo autor (2025).

O gráfico representado pela Figura 25 apresenta os valores registrados de temperatura a 2 metros no solo pelos valores previstos pelo modelo *DNN*. A curva em azul apresenta os dados reais para o período e a curva em vermelho os dados previstos pelo modelo.

Pela análise visual dos resultados plotados graficamente, nota-se que o modelo *DNN* conseguiu prever bem a temperatura para o período analisado e em certos momentos foi capaz de capturar a amplitude da temperatura além da tendência da série temporal, entretanto, nota-se que visualmente o desempenho foi inferior ao *Random Forest com SSL*.

A Tabela 8 abaixo apresenta os valores de *MAE* e *MSE* ao longo dos testes e a Tabela 9 apresenta os resultados médios do modelo.

Tabela 8 - Resultados do modelo *DNN* sem SSL.

Execução	MAE	MSE	R ²
1	1,1683	2,7180	91,60%
2	1,1360	2,5253	92,12%
3	1,1532	2,6646	91,76%
4	1,1788	2,7362	91,54%
5	1,1100	2,4571	92,40%

Fonte: Elaborado pelo autor (2025).

Tabela 9 - Média e desvio padrão do modelo *DNN* sem SSL.

	Média do Modelo	Desvio Padrão do Modelo
Média MAE	1,1492	$\pm 0,0244$
Média MSE	2,6202	$\pm 0,1101$

Fonte: Elaborado pelo autor (2025).

Os resultados do modelo *DNN* treinado com dados originais mostram um desempenho significativamente superior aos modelos *RF* tanto em *MAE* quanto *MSE*. O *MAE* caiu 56% (de 0,156 para 0,0688) e o *MSE* caiu 86% (de 0,066 para 0,0089). Isso sugere que, assim como a teoria dita, o modelo de *Deep Learning* tem maior capacidade em capturar padrões não-lineares dos dados em comparação ao *Random Forest*.

Os resultados do modelo *DNN* treinado com dados originais mostram um desempenho superior ao *RF*, porém com uma diferença menos drástica do que a observada em outros cenários. O *MAE* caiu ~5.8% (de 1,2198 para 1,1492) e o *MSE* caiu ~13.6% (de 3,0323 para 2,6202), enquanto o R^2 aumentou de ~90% para ~92%. Isso confirma que, como esperado, redes neurais têm maior capacidade de capturar relações não-lineares complexas nos dados em comparação com métodos baseados em árvores.

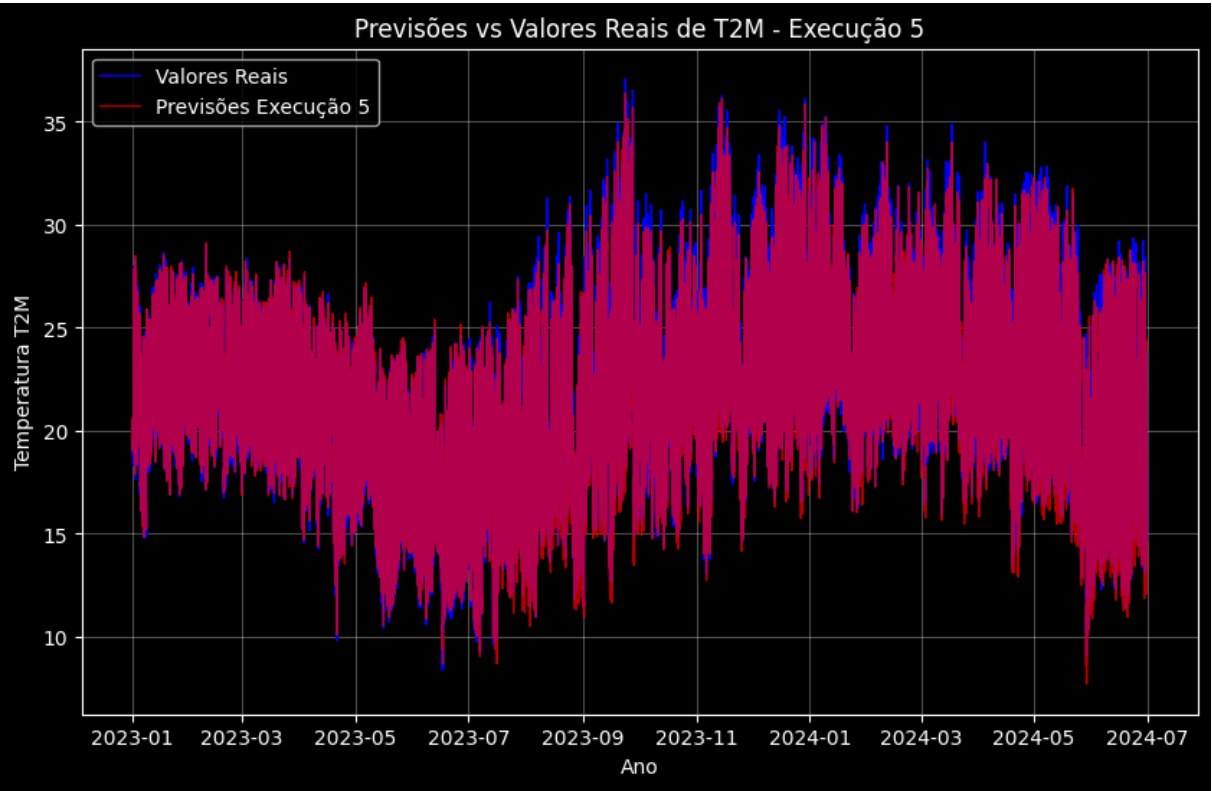
Entretanto, a variação entre execuções (*MAE*: $\pm 0,0244$, *MSE*: $\pm 0,1101$) foi maior que a do *RF* (*MAE*: $\pm 0,0019$, *MSE*: $\pm 0,0137$), indicando uma certa instabilidade, possivelmente devido a: Inicializações aleatórias dos pesos da rede e sensibilidade a divisões treino-teste ou batches de treinamento.

Apesar disso, a *DNN* superou o *RF* nos dados originais em todas as métricas, reforçando que modelos profundos podem extrair padrões mais sofisticados. Contudo, o melhor desempenho absoluto ainda pertence ao *RF* + SSL pelo *Autoencoder* (*MAE*: 0,5954, *MSE*: 0,8020, R^2 : ~96%), sugerindo que o pré-processamento avançado dos dados pode ser ainda mais crítico que a escolha do modelo.

5.2.2 Treinamento com dados reconstruídos

A Figura 26 apresenta o desempenho do modelo de *Deep Learning* quando submetido aos dados climáticos reconstruídos pelo *Masked Autoencoder*.

Figura 26 - Comparação entre valores registrados e previstos de temperatura usando *DNN* com SSL.



Fonte: Elaborado pelo autor (2025).

O gráfico acima apresenta os valores registrados de T2M pelos valores previstos pelo modelo *DNN*. A curva em azul apresenta os dados reais para o período e a curva em vermelho os dados previstos pelo modelo.

A Tabela 10 abaixo apresenta os valores de *MAE* e *MSE* ao longo dos testes e a Tabela 11 apresenta os resultados médios do modelo.

Tabela 10 - Resultados do modelo *DNN* com SSL.

Execução	<i>MAE</i>	<i>MSE</i>	<i>R</i> ²
1	0,5656	0,5675	97,6%
2	0,5614	0,5673	97,6%
3	0,5591	0,5675	97,6%
4	0,5703	0,5748	97,6%
5	0,5544	0,5487	97,7%

Fonte: Elaborado pelo autor (2025).

Tabela 11 - Média e desvio padrão do modelo *DNN* com SSL.

Média do Modelo		Desvio Padrão do Modelo
Média <i>MAE</i>	0,5621	± 0,0055
Média <i>MSE</i>	0,5652	± 0,0087

Fonte: Elaborado pelo autor (2025).

Como primeira análise, os resultados demonstram que a *DNN* supera consistentemente o *RF* em todas as métricas, tanto com dados originais quanto reconstruídos. Essa vantagem pode ser atribuída à maior capacidade da rede neural de capturar relações não-lineares complexas e padrões sutis nos dados, enquanto o *RF*, embora robusto, tem limitações inerentes a modelos baseados em árvores para lidar com variações abruptas ou interdependências sofisticadas entre features climáticas.

Quando aplicada aos dados reconstruídos pelo *SSL*, a *DNN* atingiu um desempenho excepcional: **o *MAE* caiu de 1,149 para 0,562**, uma redução de 51% no erro absoluto médio. **O *MSE* por sua vez foi reduzido de 2,620 para 0,565**, uma redução de 78% nos erros grandes (outliers). E o coeficiente de determinação (R^2) aumentou de ~92% para ~97.6%, ou seja, possibilitando uma explicação quase total da variabilidade dos dados.

Além disso, a estabilidade do modelo melhorou significativamente, o desvio padrão do *MAE* caiu de $\pm 0,0244$ (dados originais) para $\pm 0,0055$ (dados reconstruídos), indicando que a *DNN* se tornou mais consistente entre execuções.

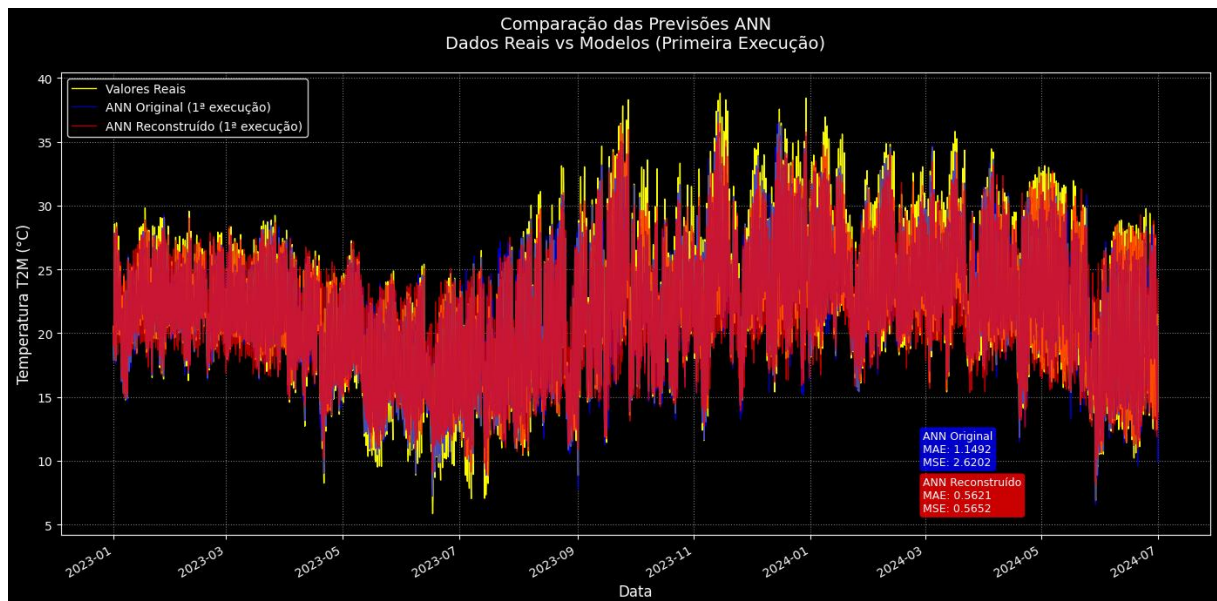
A comparação direta entre os modelos com dados reconstruídos revela que a *DNN* aproveitou melhor a reconstrução do *Masked Autoencoder*: *MAE* da *DNN* (0,562) vs. *RF* (0,595): 5.5% menor e *MSE* da *DNN* (0,565) vs. *RF* (0.802): 29.5% menor.

Isso reforça que arquiteturas de *Deep Learning* são mais eficientes para extrair informações de representações latentes de alta qualidade, como as geradas pelo *Autoencoder*. O uso do *Masked Autoencoder* não apenas elevou a precisão de ambos os modelos, mas também destacou a superioridade da *DNN* em explorar padrões complexos. A combinação *DNN* + *SSL* se consolida como a melhor abordagem para previsão de temperatura (T2M) quando em comparação aos modelos de *Random Forest Regressor*.

5.2.3 Comparação entre os modelos sem e com *SSL*

A Figura 27 apresenta um comparativo entre o desempenho do *Dense Neural Networks* com e sem técnicas de *Self-Supervised Learning*.

Figura 27 - Comparação de desempenho do *DNN* com e sem SSL.



Fonte: Elaborado pelo autor (2025).

O gráfico apresenta a comparação entre as previsões de temperatura geradas por redes *DNN* utilizando dois conjuntos de dados distintos: os originais e os reconstruídos pelo *Masked Autoencoder (SSL)*. A curva em amarelo representa os valores registrados de temperatura (T2M) entre 2023 e 2024, servindo como referência para avaliar a precisão dos modelos.

Ao analisar as previsões, fica evidente que a *DNN* treinada com os dados reconstruídos (curva vermelha) apresenta um desempenho significativamente superior àquela treinada com os dados originais (curva azul). Essa melhoria é quantificada pelas métricas de erro: enquanto o modelo com dados originais registrou um *MAE* de 1,4192 e um *MSE* de 2,6202, a versão com dados reconstruídos alcançou um *MAE* de apenas 0,5621 e um *MSE* de 0,5652. Essa redução de aproximadamente 60% no *MAE* e 78% no *MSE* demonstra não apenas uma maior precisão média, mas também uma capacidade muito melhor de lidar com valores extremos, que são penalizados mais fortemente pelo *MSE*.

Visualmente, a superioridade do modelo com dados reconstruídos é clara. A curva vermelha acompanha de perto a linha amarela dos valores registrados, mantendo-se estável mesmo em períodos de variação mais abrupta da temperatura. Em contraste, a curva azul exibe flutuações mais pronunciadas e frequentemente se desvia da trajetória real, especialmente durante eventos climáticos mais extremos. Esse comportamento reforça que o *MAE* foi eficaz em filtrar ruídos

e capturar os padrões essenciais dos dados, proporcionando à rede neural um conjunto de features mais limpo e informativo.

Além da precisão, outro aspecto notável é a consistência do modelo com dados reconstruídos. O desvio padrão do *MAE* caiu de $\pm 0,0244$ para $\pm 0,0055$, indicando que as previsões se tornaram mais estáveis entre diferentes execuções do modelo. Essa redução na variabilidade é crucial para aplicações práticas, onde a confiabilidade do sistema é tão importante quanto sua acurácia.

Em resumo, os resultados destacam o impacto transformador do *Masked Autoencoder* no pipeline de previsão. Ao reestruturar os dados originais, o *MAE* permitiu que a *DNN* atingisse níveis de performance muito mais altos, com erros significativamente menores e uma aderência impressionante aos valores registrados. Essa combinação de técnicas não apenas valida a eficácia dos *autoencoders* para pré-processamento de dados climáticos, mas também estabelece um novo patamar de qualidade para modelos de previsão de temperatura.

5.1.5 Comparação com Oloyede *et al.* (2023)

Os autores Oloyede *et al.* (2023) utilizaram, em seu trabalho, diversas arquiteturas de redes neurais com duas camadas ocultas para prever os valores máximos e mínimos da temperatura, abordagem semelhante à desenvolvida no presente estudo.

Assim como descrito anteriormente, as informações contidas em ambos os trabalhos divergem em volume de dados, área de estudo, metodologia e arquitetura de modelos, sendo assim, a Tabela 12 apresenta as configurações utilizadas em cada estudo.

Tabela 12 - Comparação entre os datasets

Característica	Trabalho desenvolvido	Trabalho de Oloyede et al. (2023)
Período	01/01/2001 a 01/08/2024	01/01/2010 a 17/08/2017
Frequência	Horária	Diária
Número de Registros	~200.000	~3000
Área de estudo	Partes da região centro-oeste, sudeste e sul do Brasil	Victor Attah International Airport, Nigéria
Modelo	DNN	MLP with LBFGS optimizer, MLP with SGD optimizer e MLP with Adam optimizer
Função de ativação	ReLu	ReLu

Fonte: Fonte: Elaborado pelo autor (2025) com informações de Oloyede *et al.* (2023).

A Tabela 13 apresenta um resumo dos resultados obtidos utilizando redes DNN sem e com SSL e os resultados obtidos por Oloyede *et al.* (2023). Os valores de *RMSE* do trabalho desenvolvido foram calculados em cima dos valores de *MSE* apresentados anteriormente, para que as informações sejam apresentadas em uma mesma grandeza.

Tabela 13 - Comparação entre os resultados obtidos e a literatura (DNN).

Modelo	MAE	RMSE	R²
DNN (p/ temp. média)	1,1492	1,6187	91,8%
DNN + SSL (p/ temp. média)	0,5621	0,7517	97%
MLP with LBFGS optimizer (p/ temp. máx.) ¹	1,412	1,778	57,3%
MLP with LBFGS optimizer (p/ temp. min.) ¹	0,788	1,127	28,2%
MLP with SGD optimizer (p/ temp. máx.) ¹	1,424	1,786	56,9%
MLP with SGD optimizer (p/ temp. min.) ¹	0,764	1,126	28,4%
MLP with Adam optimizer (p/ temp. máx.) ¹	1,428	1,797	56,4%
MLP with Adam optimizer (p/ temp. min.) ¹	0,795	1,128	28,2%

Fonte: Elaborado pelo autor (2025) em conjunto com os dados de Oloyede et al. (2023) ¹.

Analisando os dados apresentados, nota-se que, considerando a heterogeneidade dos dados utilizados, o modelo *DNN+ SSL* desenvolvido neste trabalho é mais eficaz na captura de padrões complexos e não lineares da temperatura média, resultando em menores erros absolutos e quadráticos e um R² significativamente superior. Todavia, as condições de teste dos modelos são completamente diferentes, visto que a quantidade de dados, frequência, configuração de redes neurais artificiais simples e *DNN* é diferente.

Portanto, apesar de visualmente os esforços apresentados neste trabalho apresentarem um resultado superior, é necessário futuramente comparar as metodologias utilizando os mesmos dados e condições ambientais para avaliar o possível ganho dos modelos utilizando *SSL*.

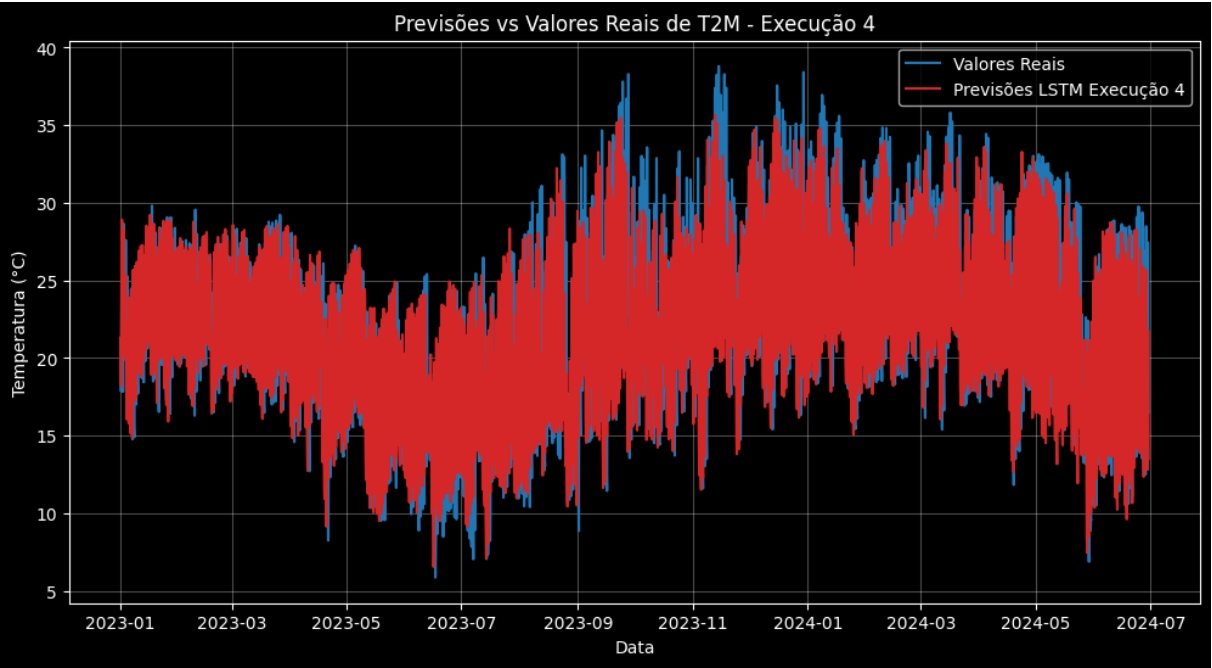
5.3 LSTM

5.3.1 Treinamento com dados originais

Para trabalhar com séries temporais longas e complexas, a *LSTM* é uma abordagem robusta para capturar os padrões intrínsecos e muitas vezes imperceptíveis entre os dados. Utilizando dados de 2001 a 2022 para treinamento e dados de 2023 e 2024 para validação.

A Figura 28 apresenta a comparação entre os valores previstos pelo modelo e os valores registrados na base histórica.

Figura 28 - Comparação entre valores registrados e previstos de temperatura usando *LSTM* sem SSL.



Fonte: Elaborado pelo autor (2025).

O gráfico acima apresenta os valores registrados de temperatura a 2 metros no solo pelos valores previstos pelo modelo *LSTM*. A curva em azul apresenta os dados reais para o período e a curva em vermelho os dados previstos pelo modelo.

A Tabela 14 abaixo apresenta os valores de *MAE* e *MSE* ao longo dos testes e a Tabela 15 os resultados médios do modelo.

Tabela 14 - Resultados do modelo *LSTM* sem SSL.

Execução	MAE	MSE	R ²
1	0,9056	1,6549	94,8%
2	0,8814	1,5559	95,1%
3	0,9113	1,6748	94,8%
4	0,8778	1,5196	95,3%
5	0.9046	1.6359	94,9%

Fonte: Elaborado pelo autor (2025).

Tabela 15 - Média e desvio padrão do modelo *LSTM* sem SSL.

	Média do Modelo	Desvio Padrão do Modelo
Média MAE	0,8961	± 0,0137
Média MSE	1,6082	± 0,0599

Fonte: Elaborado pelo autor (2025).

Pela análise dos resultados dos testes na Tabela 14 e Tabela 15 com o *LSTM*, foi possível observar resultados da *LSTM* em comparação com o *Random Forest Regressor* e o modelo *DNN*. Os resultados da *LSTM* (Long Short-Term Memory) aplicada aos dados originais demonstram um desempenho notável na tarefa de previsão de temperatura (T2M). O modelo apresentou uma média de *MAE* de 0,89611 (± 0,0137) e *MSE* de 1,6082 (± 0,0599) ao longo de

cinco execuções independentes, com um coeficiente de determinação (R^2) consistentemente acima de 94.8%.

Comparando com os modelos anteriores, **a LSTM com dados originais superou significativamente o Random Forest convencional** (MAE : 1,2198, MSE : 3,0323), mostrando uma redução de aproximadamente 26.5% no erro absoluto médio e 47% no erro quadrático médio. Esse avanço pode ser atribuído à capacidade inerente das *LSTMs* de capturar padrões temporais em séries temporais, uma característica particularmente valiosa em dados meteorológicos que possuem dependência temporal marcante. **Quando comparada à DNN convencional com os mesmos dados originais, a LSTM mostrou uma vantagem considerável.** Enquanto a *DNN* obteve MAE de 1,1492 e MSE de 2,6202, a *LSTM* reduziu esses erros em aproximadamente 22% e 38,6%, respectivamente. Essa diferença evidencia a importância de considerar a dimensão temporal nos dados meteorológicos - algo que as redes densas tradicionais não conseguem capturar com a mesma eficiência.

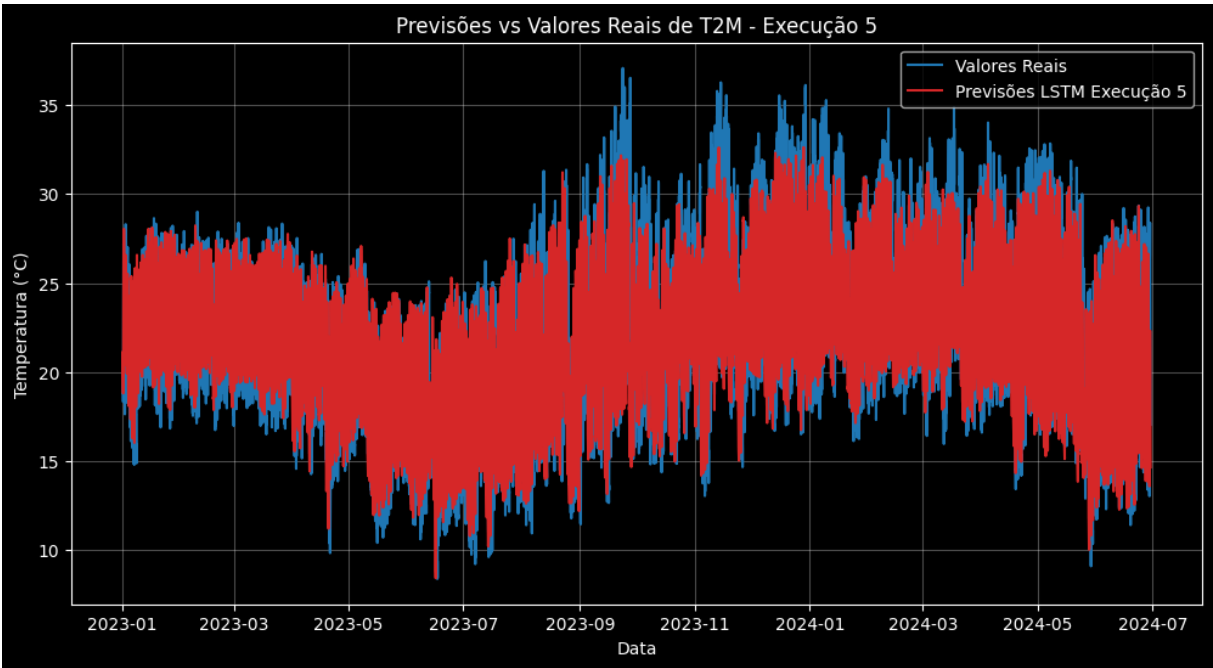
A análise da variabilidade entre execuções revela que a *LSTM* manteve uma consistência razoável, com desvio padrão de $\pm 0,0137$ para o MAE e $\pm 0,0599$ para o MSE . Embora ligeiramente superior à estabilidade observada no *Random Forest* com dados reconstruídos, essa variação permanece dentro de limites aceitáveis para um modelo de deep learning.

O desempenho notavelmente superior da *LSTM* dá-se a arquitetura da rede que leva em consideração. Enquanto a *DNN* trata os dados como independentes, a *LSTM* modela explicitamente as relações temporais entre observações, crucial para variáveis climáticas como temperatura. O mecanismo de "gates" da *LSTM* permite reter informações relevantes por períodos mais longos, identificando padrões sazonais e tendências que a *DNN* não consegue isolar.

5.3.2 Treinamento com dados reconstruídos

Por fim foram combinadas as arquiteturas de *LSTM* e o *Masked Autoencoder*, conforme sugerido por Cheng, Mei e Liu (2023), com o objetivo de aprimorar a capacidade preditiva do sistema por meio do aprendizado de representações extraídas pelo modelo de SSL. O sistema foi treinado com dados históricos compreendidos entre os anos de 2001 a 2022, sendo validado com os dados dos anos de 2023 e 2024.

Figura 29 - Comparação entre valores registrados e previstos de temperatura usando *LSTM* com *SSL*.



Fonte: Elaborado pelo autor (2025).

O gráfico acima apresenta os valores registrados de temperatura a 2 metros no solo pelos valores previstos pelo modelo *LSTM com SSL*. A curva em azul apresenta os dados reais para o período e a curva em vermelho os dados previstos pelo modelo.

A Tabela 16 abaixo apresenta os valores de MAE e MSE ao longo dos testes e a Tabela 17 apresenta os resultados médios do modelo.

Tabela 16 - Resultados do modelo *LSTM* com SSL.

Execução	MAE	MSE	R²
1	1,0416	2,0048	91,8
2	1,0525	2,0298	91,7
3	1,0486	2,0156	91,7
4	1,0585	2,0632	91,5
5	1,0484	2,0435	91,6

Fonte: Elaborado pelo autor (2025).

Tabela 17 - Média e desvio padrão do modelo *LSTM* com SSL.

	Média do Modelo	Desvio Padrão do Modelo
Média MAE	1,0499	± 0,0055
Média MSE	2,0314	± 0,0206

Fonte: Elaborado pelo autor (2025).

Os resultados da *LSTM* aplicada aos dados reconstruídos pelo *SSL* apresentam um **desempenho surpreendentemente inferior ao esperado**. O modelo alcançou um *MAE* médio de 1,0499 ($\pm 0,0055$) e *MSE* de 2,0314 ($\pm 0,0206$), com R^2 em torno de 91,6% - resultados significativamente piores que os obtidos pela mesma arquitetura com dados originais (*MAE*: 0,8961, *MSE*: 1,6082).

A *LSTM* com dados reconstruídos apresentou um *MAE* 17% maior e *MSE* 26% maior que sua versão com dados originais, entretanto o modelo apresentou maior estabilidade em

relação à *LSTM* com dados originais. A performance do modelo foi muito inferior à *DNN* com os mesmos dados reconstruídos (*MAE*: 0,5621 vs 1,0499).

A queda de desempenho observada contrasta significativamente com o comportamento registrado em outras arquiteturas submetidas ao mesmo processo de pré-processamento. A *DNN* treinada com dados reconstruídos apresentou resultados expressivos, com *MAE* de 0,5621 e *MSE* de 0,5652, representando melhorias de 51% e 78%, respectivamente, em relação aos resultados obtidos com os dados originais. De maneira semelhante, o modelo *Random Forest* também apresentou ganhos consideráveis, com redução de 51% no *MAE* e 74% no *MSE*. Em contrapartida, a arquitetura *LSTM* foi a única que não demonstrou ganhos de desempenho com o processo de reconstrução dos dados.

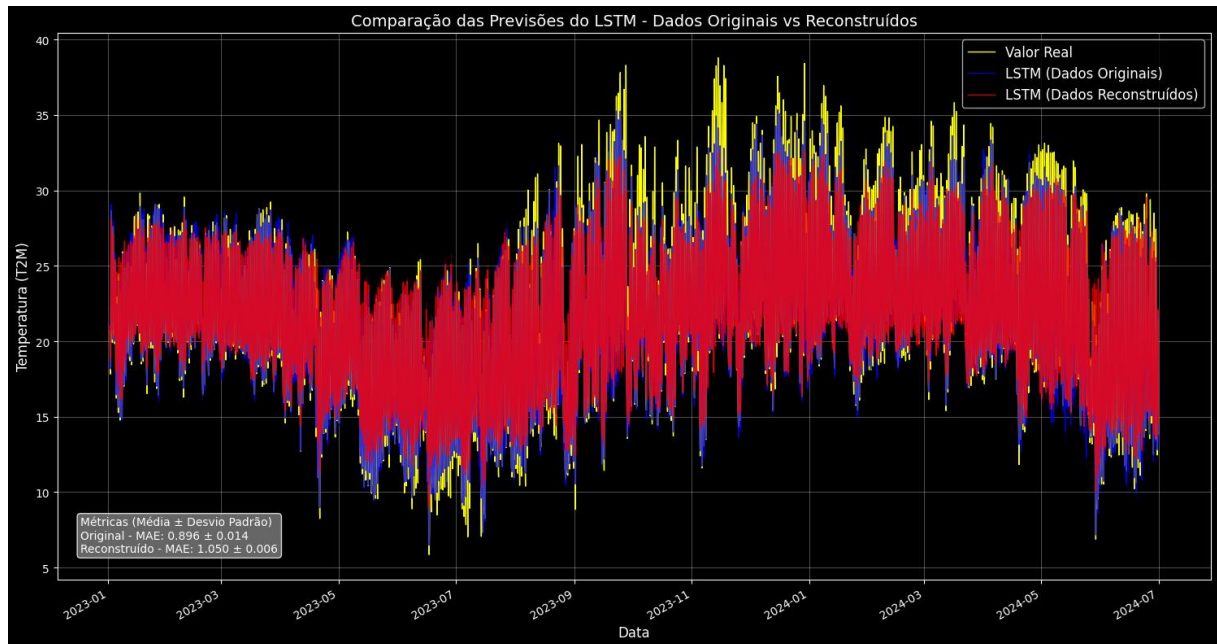
A comparação direta coloca a *LSTM + SSL* em desvantagem clara contra os modelos anteriores. Em relação à *DNN + SSL*, o *MAE* 87% maior e *MSE* 3,6 vezes maior. Quando comparando em relação ao *RF + MAE*, o *MAE* 76% maior e *MSE* 2,5 vezes maior. E por fim mesmo comparando com sua própria versão com dados originais, a performance foi inferior.

Esses resultados sugerem que **o processo de reconstrução pode estar eliminando ou distorcendo justamente as características temporais que a *LSTM* explora tão bem nos dados originais**. As *DNNs* parecem conseguir extrair melhor valor das representações latentes criadas pelo *autoencoder* e o benefício do *Masked Autoencoder* varia significativamente dependendo da arquitetura do modelo final.

5.3.3 Comparação entre os modelos sem e com SSL

A Figura 30 apresenta um comparativo entre o desempenho da *LSTM* com e sem técnicas de *Self-Supervised Learning*.

Figura 30 - Comparação de desempenho do *LSTM* com e sem SSL.



Fonte: Elaborado pelo autor (2025).

O gráfico comparativo entre as previsões do modelo *LSTM* treinado com dados originais e aquele treinado com dados reconstruídos pelo *Masked Autoencoder* revela diferenças significativas no desempenho dos dois modelos. Enquanto o *LSTM* baseado nos dados originais (curva azul) apresenta uma trajetória mais próxima dos valores registrados (curva amarela), o modelo que utilizou dados reconstruídos (curva vermelha) demonstra um desvio mais acentuado, especialmente em períodos de maior variabilidade térmica.

A *LSTM* com dados originais (azul) consegue capturar com maior precisão tanto as tendências de longo prazo quanto muitas das flutuações de curto prazo presentes nos dados reais. Em contraste, a versão com dados reconstruídos (vermelha) apresenta uma trajetória mais suavizada, perdendo parte da riqueza temporal dos dados climáticos. Essa suavização excessiva é particularmente visível em períodos de mudanças abruptas de temperatura, onde a curva vermelha frequentemente subestima ou superestima os picos térmicos.

Paradoxalmente, embora menos precisa, a *LSTM* com dados reconstruídos demonstra maior estabilidade entre execuções, com desvio padrão significativamente menor ($\pm 0,006$ contra $\pm 0,014$). Essa aparente contradição sugere que o processo de reconstrução pode estar atuando como um filtro excessivamente agressivo, removendo não apenas ruído, mas também

informações meteorológicas relevantes. O resultado é uma representação excessivamente homogênea dos dados, que embora mais consistente, perde em precisão preditiva.

O *Masked Autoencoder*, ao focar na reconstrução ponto a ponto, pode ter comprometido as relações temporais entre observações consecutivas - justamente o tipo de padrão que as *LSTMs* são especializadas em explorar. Dados meteorológicos possuem dependências temporais complexas (como ciclos diários e sazonais) que podem ter sido enfraquecidas no processo de reconstrução. O *autoencoder* foi otimizado para reconstruir features individuais, não necessariamente para preservar as dinâmicas temporais que a *LSTM* utiliza para fazer previsões. Essa diferença nos objetivos de aprendizado pode explicar por que redes densas (*DNN*) se beneficiaram do *MAE*, enquanto a *LSTM* não. A suavização observada nos dados reconstruídos pode ter removido componentes legítimos da variabilidade climática, limitando a capacidade da *LSTM* de prever eventos extremos ou variações rápidas de temperatura.

Para aplicações onde a precisão absoluta é prioritária, a *LSTM* com dados originais permanece como a melhor opção entre as arquiteturas testadas. A maior consistência da *LSTM* + *SSL*, embora com erro médio maior, pode ser interessante em cenários onde a estabilidade das previsões é mais importante que a precisão pontual. Os resultados sugerem que, para modelos temporais, técnicas de pré-processamento específicas para séries temporais (como *autoencoders* convolucionais ou variacionais) podem ser mais adequadas que o *MAE* tradicional.

5.4 Comparativo entre todos os modelos

A Tabela 18 a seguir traz um levantamento dos desempenhos médios e desvio padrão de todos os modelos testados com valores registrados e reconstruídos usando *Self-Supervised Learning*.

Tabela 18 - Comparação dos modelos com e sem SSL.

Modelos	MAE médio	MSE médio	R ² (%)
Random Forest Regressor	1,2198 ± 0,0019	3,0323 ± 0,0137	~ 90
Random Forest Regressor (SSL)	0,5954 ± 0,0006	0,8020 ± 0,0015	~ 96
<i>DNN</i>	1,1492 ± 0,0244	2,6202 ± 0,1101	~ 92
<i>DNN</i> (SSL)	0,5621 ± 0,0055	0,5652 ± 0,0087	~ 97,6
<i>LSTM</i>	0,8961 ± 0,0137	1,6082 ± 0,0137	~ 95
<i>LSTM</i> (SSL)	1,0499 ± 0,0055	2,0314 ± 0,0206	~ 91,6

Fonte: Elaborado pelo autor (2025).

A avaliação dos diferentes modelos de *machine learning* para previsão de temperatura (T2M) apresentados na Tabela 18 apresentou desempenhos distintos conforme a arquitetura utilizada e o tipo de dados (originais ou processados pelo *Masked Autoencoder*). Entre todos os modelos testados, **a *DNN* combinada com dados reconstruídos pelo *Masked Autoencoder* emergiu como a abordagem mais eficaz**, alcançando um *MAE* de apenas 0,5621°C, *MSE* de 0,5652°C² e um impressionante R² de 97,6%. Esses resultados demonstram não apenas uma alta precisão nas previsões médias, mas também uma excelente capacidade de lidar com valores extremos, além de explicar quase toda a variabilidade presente nos dados climáticos.

Quando comparado o desempenho das diferentes arquiteturas com dados originais, a ***LSTM* se mostrou superior às redes *DNN* e ao *Random Forest***, registrando *MAE* de 0,8961°C contra 1,1492°C da *DNN* e 1,2198°C do *Random Forest*. Essa vantagem da *LSTM* evidencia sua capacidade inata de capturar padrões temporais em séries climáticas, algo essencial para previsões meteorológicas precisas. No entanto, o cenário mudou radicalmente

quando aplicamos o pré-processamento com *Masked Autoencoder*: enquanto a *DNN* e o *Random Forest* apresentaram melhorias expressivas (com reduções de 51% e 74% no *MSE*, respectivamente), a *LSTM* teve seu desempenho prejudicado, com aumento de 17% no *MAE* e 26% no *MSE*.

Esta diferença de comportamento sugere que o processo de reconstrução do *Masked Autoencoder*, embora benéfico para modelos que operam em dados independentes (como *DNN* e *Random Forest*), pode estar removendo ou alterando justamente as relações temporais que são cruciais para o funcionamento das *LSTMs*. A análise gráfica apresentada na Figura 29 confirma essa hipótese, mostrando que a *LSTM* com dados reconstruídos produz uma trajetória excessivamente suavizada, perdendo a capacidade de acompanhar variações abruptas de temperatura que sua versão com dados originais captura com maior eficiência.

Em termos de estabilidade, o *Random Forest + SSL* apresentou a menor variação entre execuções (desvio padrão do *MAE* de apenas $\pm 0,0006$), seguido pela *LSTM com SSL* ($\pm 0,0055$) e pela *DNN com SSL* ($\pm 0,0055$). Essa consistência, aliada a seu baixo custo computacional, faz do *Random Forest + SSL* uma opção atraente para aplicações onde a estabilidade das previsões é prioritária, mesmo que sua precisão absoluta seja ligeiramente inferior à da *DNN*.

Para situações práticas, a escolha do modelo ideal depende dos requisitos específicos de cada aplicação: a combinação *DNN + SSL* é a melhor opção quando se busca a máxima precisão absoluta; a *LSTM* com dados originais deve ser preferida quando não é possível pré-processar os dados ou quando a natureza temporal da previsão é crítica; já o *Random Forest + SSL* oferece o melhor equilíbrio entre desempenho e eficiência computacional para sistemas que demandam previsões estáveis e rápidas.

6 CONCLUSÃO

O presente trabalho abordou o desafio da previsão de temperatura (T2M) a partir de dados climáticos multivariados, um problema notoriamente complexo devido à natureza não linear e temporal das variáveis meteorológicas. Para superar as limitações dos métodos supervisionados tradicionais, que dependem fortemente de grandes volumes de dados rotulados, foi adotada uma abordagem baseada em *Self-Supervised Learning (SSL)* com *Masked Autoencoders*. Essa técnica permitiu extrair representações latentes dos dados sem a necessidade de rótulos explícitos, aproveitando ao máximo a informação disponível e contornando o custo elevado de anotação manual. O *Masked Autoencoder* foi escolhido por sua capacidade de reconstruir dados parcialmente mascarados, aprendendo padrões temporais e espaciais mesmo na presença de falhas ou ruídos nos registros climáticos.

Os resultados demonstraram que o pré-processamento com *Masked Autoencoder* melhorou significativamente o desempenho de modelos tradicionais. A *DNN* com dados reconstruídos alcançou os melhores resultados (*MAE*: 0,5621, *MSE*: 0,5652, R^2 : 97,6%), superando tanto a *LSTM* com dados originais (*MAE*: 0,8961) quanto o *Random Forest* convencional (*MAE*: 1,2198). No entanto, a *LSTM* não se beneficiou do *Masked Autoencoder*, apresentando piora no desempenho (*MAE*: 1,0499), o que sugere que a reconstrução pode ter prejudicado as relações temporais essenciais para seu funcionamento. Por outro lado, o *Random Forest + SSL* mostrou-se uma alternativa eficiente, com alta estabilidade (DP do *MAE*: $\pm 0,0006$) e custo computacional reduzido, indicando que o SSL pode ser útil mesmo em modelos menos complexos.

A abordagem proposta apresenta vantagens claras: redução da dependência de dados rotulados, robustez a ruídos e capacidade de capturar padrões climáticos complexos. Contudo, também enfrenta algumas limitações, como a possível perda de informações temporais em modelos sequenciais (como a *LSTM*) e a necessidade de um ajuste cuidadoso dos parâmetros de mascaramento. Apesar dessas restrições, a abordagem pode ser utilizada em outros contextos com grandes quantidades de dados não rotulados, como na previsão da energia eólica, no monitoramento de poluição ou na análise de séries temporais na área da saúde.

Futuros trabalhos podem explorar o uso de *Masked Autoencoders* especializados em dados temporais (como *Transformers/TFT*) para melhorar o desempenho de *LSTMs*, além de investigar técnicas como *GRU* e *Bidirecional LSTM*. /GRU. Em suma, este estudo demonstra

que o *SSL*, especialmente com o *MAE*, é uma abordagem útil para lidar com questões climáticas, proporcionando um equilíbrio entre precisão, eficiência e aplicabilidade prática.

REFERÊNCIAS

- ALI, T.; YIN, Y.; ZHANG, Y.; HUANG, T. **Deep learning-based models for forecasting financial time series: a comprehensive survey**. Machine Learning and Knowledge Extraction, Basel, v. 5, n. 4, p. 723–741, 2023.
- AN, S.; LEE, J.; JANG, J.; NA, I.; PARK, W.; YOU, S. **Self-supervised pre-training for precipitation post-processor**. In: Development of a Next-Generation Operational, [S.l.: s.n.], fev. 2024.
- ARTAXO, P.; RIZZO, L. V.; MACHADO, L. A. T. **Inteligência artificial e mudanças climáticas**. Revista USP, São Paulo, n. 141, p. 29–40, 2024.
- BAY, Yong Yi; YEARICK, Kathleen A. **Machine learning vs deep learning: the generalization problem**. arXiv e-prints, [S.l.], mar. 2024. Disponível em: <https://arxiv.org/abs/2403.01628>. Acesso em: 6 maio 2025.
- BISHOP, C. M. **Pattern recognition and machine learning**. New York: Springer Science+Business Media, 2006.
- BREIMAN, L. **Random forests**. Machine Learning, [S.l.], v. 45, p. 5–32, 2001.
- BROWN, T. B.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A.; AGARWAL, S.; HERBERT-VOSS, A.; KRUEGER, G.; HENIGHAN, T.; CHILD, R.; RAMESH, A.; ZIEGLER, D. M.; WU, J.; WINTER, C.; HESSE, C.; CHEN, M.; SIGLER, E.; LITWIN, M.; GRAY, S.; CHESS, B.; CLARK, J.; BERGAMO, A.; RAO, S.; GABRIELI, T.; KING, M.; NARASIMHAN, K.; KERNION, J.; KRUEGER, G.; KAPLAN, J.; SUTSKEVER, I.; AMODEI, D. **Language models are few-shot learners**. arXiv e-prints, [S.l.], maio 2020. Disponível em: <https://arxiv.org/abs/2005.14165>. Acesso em: 6 maio 2025.
- CHENG, Jie; MEI, Xiaodong; LIU, Ming. **Forecast-MAE: self-supervised pre-training for motion forecasting with masked autoencoder**. arXiv e-prints, [S.l.], ago. 2023. Disponível em: <https://arxiv.org/abs/2308.04604>. Acesso em: 6 maio 2025.
- DEEPGRAM. **Convolutional neural networks**. [S.l.], 2025. Disponível em: <https://deepgram.com/ai-glossary/convolutional-neural-networks>. Acesso em: 12 fev 2025.
- DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. **BERT: pre-training of deep bidirectional transformers for language understanding**. arXiv e-prints, [S.l.], out. 2018. Disponível em: <https://arxiv.org/abs/1810.04805>. Acesso em: 12 fev 2025.
- DOMINGUEZ, D.; JUAN DEL VILLAR, L. D.; PANTOJA, O.; GONZÁLEZ-RODRIGUEZ, M. **Forecasting Amazon rain-forest deforestation using a hybrid machine learning model**. Sustainability, Basel, v. 14, n. 2, p. 691, 2022.
- ERICSSON, L.; GOUK, H.; HOSPEDALES, T. M. **How well do self-supervised models transfer?** arXiv e-prints, [S.l.], mar. 2021. p. 1–8. Disponível em: <https://arxiv.org/abs/2103.03282>. Acesso em: 15 fev 2025.

- ERICSSON, L.; GOUK, H.; LOY, C. C.; HOSPEDALES, T. M. **Self-supervised representation learning: introduction, advances and challenges**. arXiv e-prints, [S.l.], 2021. Disponível em: <https://arxiv.org/abs/2111.12899>. Acesso em: 15 fev 2025.
- GONG, Bing; LI, Chenxu; JIA, Wenwen; LI, Zhaoxue; HU, Xiyang; CHEN, Long; LIU, Hongyu; ZHANG, Tong; WANG, Yuan; YANG, Bo; TANG, Xuebo; ZHANG, Xian; WANG, Jinyu. **Temperature forecasting by deep learning methods**. Geoscientific Model Development, [S.l.], 2022, p. 1–26.
- GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. Cambridge: MIT Press, 2016.
- GÉRON, Aurélien. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow**. 2. ed. Sebastopol: O'Reilly Media, 2019.
- HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. **The elements of statistical learning: data mining, inference, and prediction**. New York: Springer, 2009.
- HE, Kaiming; CHEN, Xinlei; XIE, Saining; LI, Yanghao; DOLLÁR, Piotr; GIRSHICK, Ross. **Masked autoencoders are scalable vision learners**. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), [S.l.], nov. 2021. p. 16000–16009.
- HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. **Long short-term memory**. Neural Computation, [S.l.], v. 9, p. 1735–1780, 1997.
- IEMOBAYO, Justus A.; OLATUNJI, Olaoluwa O.; ONI, Olamide O.; AKINBAYO, Folasade. **Hyperparameter tuning in machine learning: a comprehensive review**. Journal of Engineering Research and Reports, [S.l.], v. 26, n. 6, p. 388–395, jun. 2024.
- JONES, Roger N.; RICKETTS, James H. **Climate as a complex, self-regulating system**. Earth System Dynamics, [S.l.], v. 12, p. 1–47, jul. 2021.
- KUMAR, R.; KUMAR, M.; SHARMA, A. **Deep learning approaches for image classification: a comprehensive survey**. SN Computer Science, [S.l.], v. 2, p. 1–15, 2021.
- LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. **Deep learning**. Nature, [S.l.], v. 521, p. 436–444, 2015.
- LECUN, Yann; BOTTOU, Léon; BENGIO, Yoshua; HAFFNER, Patrick. **Gradient based learning applied to document recognition**. IEEE Xplore, [S.l.], nov. 1998. p. 6.
- LE-KHAC, Nhien-An; HEALY, Geoff; SWEENEY, Leif. **A survey on self-supervised learning: auxiliary pretext tasks and methodologies**. Journal of Big Data, [S.l.], v. 7, p. 1–30, 2020.
- LI, Haotian; SIEBES, Arno; MEHRKANOON, Siamak. **Self-supervised spatial-temporal learner for precipitation nowcasting**. arXiv e-prints, [S.l.], dez. 2023. Disponível em: <https://arxiv.org/abs/2302.12322>. Acesso em: 6 maio 2025.
- LIM, Bryan; ZOHREN, Stefan. **Time-series forecasting with deep learning: a survey**. Philosophical Transactions of the Royal Society A, [S.l.], p. 1–14, 2021.

LIPTON, Zachary C.; BERKOWITZ, John; ELKAN, Charles. **A critical review of recurrent neural networks for sequence learning**. arXiv e-prints, [S.l.], maio 2015. Disponível em: <https://arxiv.org/abs/1506.02318>. Acesso em: 6 maio 2025.

LIU, Bin et al. **Self-supervised learning: generative or contrastive**. IEEE Transactions on Knowledge and Data Engineering, [S.l.], v. 33, p. 1–12, 2021.

MATERIA, Stefano; AL., et al. **Artificial intelligence for prediction of climate extremes: state of the art, challenges and future perspectives**. arXiv preprint, [S.l.], 2023.

MITCHELL, Melanie. **Complexity: A guide tour**. Oxford: Oxford University Press, 2009.

MURPHY, Kevin P. **Machine learning: a probabilistic perspective**. Cambridge: MIT Press, 2012.

OLOYEDE, Adamson; OZUOMBA, Simeon; ASUQUO, Philip; OLATOMIWA, Lanre; LONGE, Omowunmi Mary. **Data driven techniques for temperature data prediction: big data analytics approach**. Environmental Monitoring and Assessment, [S.l.], 2023.

ORGANIZAÇÃO DAS NAÇÕES UNIDAS. Nações Unidas no Brasil. **O que são as mudanças climáticas?**. Brasília, 2024. Disponível em: <https://brasil.un.org/pt-br/175180-o-que-s%C3%A3o-mudan%C3%A7as-clim%C3%A1ticas>. Acesso em: 6 maio 2025.

RAISSI, M.; PERDIKARIS, P.; KARNIADAKIS, G. E. **Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations**. Journal of Computational Physics, [S.l.], v. 378, p. 686–707, 2019.

RUSSELL, Stuart J.; NORVIG, Peter. **Artificial intelligence: a modern approach**. 4. ed. Harlow: Pearson, 2016.

SCHER, S. **Toward data-driven weather and climate forecasting: approximating a simple general circulation model with deep learning**. Geophysical Research Letters, [S.l.], 2018.

SENEVIRATNE, S. I. et al. **Weather and climate extreme events in a changing climate**. Sixth Assessment Report, vol. 6. Cambridge: Cambridge University Press, 2021.

SHARMA, S. et al. **A hybrid deep learning model for accurate weather forecasting using ResNet50 and LSTM**. Scientific Reports, [S.l.], v. 14, 2024.

O SENHOR dos anéis: a sociedade do anel. Direção: Peter Jackson. Produção: Barrie M. Osborne, Peter Jackson, Fran Walsh e Tim Sanders. Estados Unidos: New Line Cinema, 2001. Filme, 178 min.

SICHMAN, Jaime Simão. **Inteligência artificial e sociedade: avanços e riscos**. Estudos Avançados, [S.l.], v. 35, n. 3, p. 37–49, 12 dez. 2021.

SINGH, Karandeep; AL., et al. **Self supervised vision for climate downscaling**. arXiv e-prints, [S.l.], jan. 2024. Disponível em: <https://arxiv.org/abs/2401.10089>. Acesso em: 6 maio 2025.

VASWANI, Ashish; AL., et al. **Attention is all you need.** Advances in Neural Information Processing Systems, [S.l.], v. 30, p. 5998–6008, 2017. Disponível em: <https://arxiv.org/abs/1706.03762>. Acesso em: 6 maio 2025.

VEENU RANI, Syed Tufael Nabi, Munish Kumar, Ajay Mittal, e Krishan Kumar. **Self supervised learning: A succinct review.** Archives of Computational Methods in Engineering, [S.l.], 5 jan. 2023. Disponível em: <https://link.springer.com/article/10.1007/s11831-023-09743-4>. Acesso em: 6 maio 2025.

ZBONTAR, Jure et al. Barlow twins: **self-supervised learning via redundancy reduction.** Proceedings of the 38th International Conference on Machine Learning (ICML), [S.l.], p. 12310–12320, 2021. Disponível em: <https://arxiv.org/abs/2103.03230>. Acesso em: 6 maio 2025.

ZHA, Mengyue; WONG, SiuTim; LIU, Mengqi; ZHANG, Tong; CHEN, Kani. **Time series generation with masked autoencoder.** [S.l.], maio 2022. p. 3. Disponível em: <https://arxiv.org/abs/2205.06009>. Acesso em: 6 maio 2025.

ZHOU, Zhi-Hua; LIU, Shaowu. **Machine learning.** 1. ed. Tradução: Shaowu Liu. Nanjing, Jiangsu: Tsinghua University Press, 2016.

APÊNDICE A - Principais bibliotecas utilizadas.

Biblioteca	Descrição
Numpy	Biblioteca para computação numérica eficiente. Oferece suporte a arrays multidimensionais e funções matemáticas de alto desempenho para operações em vetores e matrizes.
Pandas	Focada em análise e manipulação de dados, fornece estruturas como DataFrame e Series, com funcionalidades para limpeza, filtragem, agregação e transformação de dados tabulares.
Scikit Learn	Biblioteca para <i>machine learning</i> em Python. Oferece algoritmos de classificação, regressão, clustering e ferramentas para validação de modelos e engenharia de características.
Statsmodels	Voltada para análise estatística, fornece ferramentas para modelagem estatística (como regressão linear e ARIMA), testes estatísticos e análise de séries temporais.
TensorFlow	Framework para <i>machine learning</i> e deep learning desenvolvido pelo Google. Suporta redes neurais, aprendizado por reforço e operações matemáticas em GPUs e CPUs.
Matplotlib	Biblioteca para visualização de dados. Permite criar gráficos estáticos, interativos e personalizados, como histogramas, gráficos de linha e barras.
Seaborn	Biblioteca para visualização de dados semelhante ao Matplotlib. Permite criar gráficos mais personalizáveis.

APÊNDICE B - Dados meteorológicos do dataset original.

Feature	Descrição	Máximo	Mínimo
YEAR	Ano da medição.	2024	2001
MO	Mês da medição.	12	1
DY	Dia da medição.	31	1
HR	Hora da medição.	23	0
PRECTOTCORR	Precipitação total corrigida (mm).	369.14	0.0
TOA_SW_DWN	Radiação solar de onda curta descendente no topo da atmosfera (W/m^2).	1403.28	-999.0
WS10M	Velocidade do vento a 10 metros de altura (m/s).	12.03	0.02
GWETPROF	Umidade do solo integrada ao longo do perfil.	0.92	0.52
TS	Temperatura da superfície do solo.	44.88	-1.29
WD10M	Direção do vento a 10 metros de altura (graus).	359.9	0.0
QV2M	umidade específica a 2 metros.	20.87	2.11
GWETROOT	Umidade do solo na zona radicular.	0.91	0.5
GWETTOP	Umidade do solo na camada superior.	1.0	0.34
ALLSKY_SFC_UVA	Irradiação UVA na superfície para todas as condições de céu (W/m^2).	70.1	-999.0
CLRSKY_SFC_SW_DWN	Radiação solar de onda curta descendente na superfície para condições de céu limpo (W/m^2).	1129.8	-999.0
ALLSKY_KT	Índice de transmissão atmosférica para todas as condições de céu.	0.83	-999.0
ALLSKY_SFC_UV_INDEX	Índice UV na superfície para todas as condições de céu.	19.52	-999.0
ALLSKY_SFC_UVB	Irradiação UVB na superfície para todas as condições de céu (W/m^2).	3.13	-999.0
CLRSKY_KT	Índice de transmissão atmosférica para condições de céu limpo.	0.83	-999.0

CLRSKY_SFC_PAR_TOT	Radiação fotos sinteticamente ativa para condições de céu limpo (W/m²).	502.23	-999.0
T2M	temperatura a 2 metros, sendo a grandeza alvo da predição	39.16	0.49
ALLSKY_SFC_SW_DWN	Radiação solar de onda curta descendente na superfície para todas as condições de céu (W/m²).	1129.6	-999.0
RH2M	Umidade relativa a 2 metros de altura (%).	100.0	13.54
WS2M	velocidade do vento a 2 metros (m/s).	8.02	0.01