UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE FÍSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA AMBIENTAL

Utilização de Redes Neurais Informadas Pela Física no Preenchimento de Falhas em Séries de Temperatura do Ar

Anisio Alfredo da Silva Junior

Orientador: **Prof. Dr. Raphael de Souza Rosa Gomes**

Cuiabá - MT
Dezembro/2023

UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE FÍSICA
PROGRAMA DE PÓS GRADUAÇÃO EM FÍSICA AMBIENTAL

Utilização de Redes Neurais Informadas Pela Física no Preenchimento de Falhas em Séries de Temperatura do Ar

Anisio Alfredo da Silva Junior

Tese apresentada ao Programa de Pós-Graduação em Física Ambiental da Universidade Federal de Mato Grosso, como parte dos requisitos para obtenção do título de Doutor em Física Ambiental.

Orientador: **Prof. Dr. Raphael de Souza Rosa Gomes**

Cuiabá, MT

Dezembro/2023

Dados Internacionais de Catalogação na Fonte.

D111u da Silva Junior, Anisio Alfredo.
Utilização de redes neurais informadas pela física no preenchimento de falhas em séries de temperatura do ar [recurso eletrônico] / Anisio Alfredo da Silva Junior. -- Dados eletrônicos (1 arquivo : 58 f., il. color., pdf). -- 2023.

Orientador: Raphael de Souza Rosa Gomes.
Tese (doutorado) - Universidade Federal de Mato Grosso, Instituto de Física, Programa de Pós-Graduação em Física Ambiental, Cuiabá, 2023.

Modo de acesso: World Wide Web: https://ri.ufmt.br.
Inclui bibliografia.

1. Redes Neurais Informadas por Física, Aprendizado de Máquina, Equações Diferenciais Parciais, inteligência artificial, tratamento de dados.. I. de Souza Rosa Gomes, Raphael, *orientador*. II. Título.

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

MINISTÉRIO DA EDUCAÇÃO

UNIVERSIDADE FEDERAL DE MATO GROSSO

PRÓ-REITORIA DE ENSINO DE PÓS-GRADUAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM [NOME DO PPG]

FOLHA DE APROVAÇÃO

TÍTULO: Utilização de Redes Informadas Pela Física no Preenchimento de Falhas em Séries de Temperatura do Ar

AUTOR: DOUTORANDO ANÍSIO ALFREDO DA SILVA JUNIOR

Tese defendida e aprovada em 13 de novembro de 2023.

COMPOSIÇÃO DA BANCA EXAMINADORA

1. **Prof. Dr. Raphael de Souza Rosa Gomes** (Presidente da Banca / Orientador)

INSTITUIÇÃO: Universidade Federal de Mato Grosso

2. **Profa. Dra. Daniela de Oliveira Maionchi** (Membro Interno)

INSTITUIÇÃO: Universidade Federal de Mato Grosso

3. **Prof. Dr. Carlo Ralph De Musis** (Membro Interno)

INSTITUIÇÃO: POLITEC - MT

4. **Prof. Dr. Josiel Maimoni Figueiredo** (Membro Externo)

INSTITUIÇÃO: Universidade Federal de Mato Grosso

5. **Prof. Dr. Jonathan Willian Zangeski Novais** (Membro Externo)

INSTITUIÇÃO: Universidade de Cuiabá

Cuiabá, 13/11/2023.

Documento assinado eletronicamente por **MARCELO SACARDI BIUDES**, **Coordenador(a) de Programas de Pós-Graduação em Física Ambiental - IF/UFMT**, em 17/11/2023, às 12:28, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **RAPHAEL DE SOUZA ROSA GOMES**, **Docente da Universidade Federal de Mato Grosso**, em 17/11/2023, às 12:37, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **JOSIEL MAIMONE DE FIGUEIREDO**, **Docente da Universidade Federal de Mato Grosso**, em 17/11/2023, às 12:42, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **DANIELA DE OLIVEIRA MAIONCHI**, **Docente da Universidade Federal de Mato Grosso**, em 18/11/2023, às 15:38, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **CARLO RALPH DE MUSIS**, **Usuário Externo**, em 18/11/2023, às 17:45, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Jonathan Willian Zangeski Novais**, **Usuário Externo**, em 20/11/2023, às 14:42, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site http://sei.ufmt.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **6388223** e o código CRC **D9219CCD**.

Referência: Processo nº 23108.084474/2023-06 SEI nº 6388223

"Aquele que prevê calamidades sofrea-as duas vezes"
-Beilby Porteus

Agradecimentos

- A Deus por tudo.
- A minha mãe, eterna heroína.
- Aos meus irmãos Leonardo e Cláudia, companheiros para toda vida.
- A minha amada esposa Leilane, por incentivar a concluir esse árduo trabalho.
- A Lucielly, Paloma e Marlus, pela amizade persistente.
- Ao professor Raphael de Souza Rosa Gomes pela paciência e compreensão durante a orientação.
- Ao professor José de Souza Nogueira (In memoriam), sua esposa professora Marta Cristina de Jesus Albuquerque Nogueira, e aos demais professores e corpo docente que fazem do PPGFA um ambiente acolhedor e vivo.
- A todos que contribuíram de alguma forma para a realização deste trabalho.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

ADAM *Adaptive Moment Estimation* (Estimativa de Momento Adaptativo) 23

CART *Classification And Regression Tree* (Árvores de Classificação e Regressão) 2

EMA Estação Meteorológica Automática 27

EN *Elastic Net* (Rede Elástica) 2

INMET Instituto Nacional de Meteorologia 27

KNN *k-nearest neighbors* (k-vizinhos mais próximos) 2

LASSO *Least Absolute Shrinkage and Selection Operator* (Operador de Redução e Seleção Mínima Absoluta) 2

LR *Linear Regression* (Regressão Linear) 2

ML *Machine Learning* (Aprendizado de Máquina) 2

MLP *Multilayer Perceptron* (Perceptron Multicamada) 2

MSE *Mean Squared Error* (Erro Quadrático Médio) 23

PDE *Partial Differential Equation* (Equações Diferenciais Parciais) 24

PINN *Physics-Informed Neural Networks* (Redes Neurais Informadas pela Física) 2

ReLU *Rectified Linear Unit* (Unidade Linear Retificada) 19

RESUMO

SILVA JUNIOR, A. A. **Utilização de Redes Neurais Informadas Pela Física no Preenchimento de Falhas em Séries de Temperatura do Ar**. Cuiabá, 2023, 60f. Tese (Doutorado em Física Ambiental) - Instituto de Física, Universidade Federal de Mato Grosso.

Neste estudo, um modelo baseado em *Physics-Informed Neural Networks* (PINN) foi desenvolvido e implementado com o objetivo de preencher falhas em dados micrometeorológicos e comparado com outros sete modelos. A função de custo do modelo foi derivada a partir de uma versão empírica da equação da difusão molecular. Os resultados foram comparados com sete modelos tradicionais de aprendizado de máquina, incluindo Regressão Linear, Regressão LASSO, Rede Elástica (EN), KNN, Árvores de Classificação e Regressão (CART), *Support Vector Regression* (SVR) e *Multi-Layer Perceptron* (MLP).Os resultados mostraram que os modelos baseados em redes neurais, incluindo o PINN, superaram a maioria dos outros modelos, com um erro médio quadrático (RMSE) médio de 0,08 °C em comparação com os valores médios de RMSE de 0,12 °C para os outros modelos. O PINN também exigiu menos épocas de treinamento em comparação com o MLP, com uma diferença máxima de apenas 62 épocas em cinco das oito regiões analisadas. No entanto, os modelos LASSO e EN apresentaram os maiores valores de RMSE, com uma média de aproximadamente 0,37 °C durante todas as fases e em todas as regiões analisadas. Além disso, foi observada a resiliência dos modelos em relação à proporção dos dados de treinamento, demonstrando sua capacidade de se adaptar a diferentes proporções de dados, com todos os modelos atendendo à quantidade mínima necessária para um aprendizado eficaz dos dados utilizados. Em suma, este estudo destacou a eficácia dos modelos baseados em redes neurais, incluindo o PINN, na previsão de temperaturas em dados micrometeorológicos, proporcionando uma solução precisa para o preenchimento de lacunas. Esses resultados contribuem para o avanço das Ciências Ambientais e oferecem percepções para estudos futuros na área de preenchimento de dados faltantes, melhorando a integridade das análises climáticas e ambientais.

Palavras-chave: Redes Neurais Informadas por Física, Aprendizado de Máquina, Equações Diferenciais Parciais, inteligência artificial, tratamento de dados.

ABSTRACT

SILVA JUNIOR, A. A. **Utilization of Physics-Informed Neural Networks in Filling Gaps in Air Temperature Time Series**. Cuiabá, 2023, 60f. Tese (Doutorado em Física Ambiental) - Instituto de Física, Universidade Federal de Mato Grosso.

In this study, a model based on Physics-Informed Neural Networks (PINN) was developed and implemented with the aim of filling gaps in micrometeorological data and compared with seven other models. The model's cost function was derived from an empirical version of the molecular diffusion equation. Results were compared with seven traditional machine learning models, including Linear Regression, LASSO Regression, Elastic Net (EN), K-Nearest Neighbors (KNN), Classification and Regression Trees (CART), Support Vector Regression (SVR), and Multi-Layer Perceptron (MLP). The results indicated that neural network-based models, including PINN, outperformed most other models, with an average Root Mean Square Error (RMSE) of 0.08 °C compared to the average RMSE values of 0.12 °C for the other models. PINN also required fewer training epochs compared to MLP, with a maximum difference of only 62 epochs in five of the eight analyzed regions. However, LASSO and EN models exhibited the highest RMSE values, averaging approximately 0.37 °C across all phases and regions. Additionally, the models demonstrated resilience to varying proportions of training data, showcasing their ability to adapt to different data ratios, with all models meeting the minimum requirement for effective learning from the utilized data. In summary, this study highlighted the effectiveness of neural network-based models, including PINN, in predicting temperatures in micrometeorological data, providing a precise solution for gap filling. These results contribute to the advancement of Environmental Sciences and offer insights for future studies in the field of missing data imputation, enhancing the integrity of climatic and environmental analyses.

Keywords: *Physics-Informed Neural Networks, Machine Learning, Partial Differential Equations, Artificial Intelligence, Data Processing.*

INTRODUÇÃO

1.1 PROBLEMÁTICA

O estudo da temperatura do ar é uma área de pesquisa de grande importância, uma vez que está intrinsecamente ligada às mudanças climáticas e ao aquecimento global, fenômenos que têm repercussões significativas nas vidas humanas e no meio ambiente (THOBER et al., 2018). O aumento da demanda por informações meteorológicas precisas e de curto prazo tem impulsionado o desenvolvimento de novas abordagens e tecnologias para melhorar a precisão das previsões meteorológicas (WEN et al., 2020).

Nos últimos anos, uma dessas abordagens tem se destacado: o uso de métodos de Aprendizado de Máquina (ML). Esses métodos, que englobam uma série de algoritmos e técnicas computacionais, têm demonstrado sua eficácia em uma variedade de aplicações, desde análise de dados até previsões complexas, e muitas vezes um baixo custo em comparação com os métodos tradicionais (XU et al., 2021).

Porém para alcançar melhores precisões meteorológicas se faz necessária uma monitoria climática eficaz exigindo a coleta e armazenamento de dados micrometeorológicos, essenciais para o estudo das mudanças nas temperaturas, precipitação, umidade e outros fatores climáticos. Entretanto, a disponibilidade desses dados muitas vezes é afetada por falhas e lacunas, prejudicando a precisão das previsões climáticas levando a análises distorcidas e a tomadas de decisões inadequadas, afetando a eficiência das políticas públicas e das medidas de proteção ambiental.

Existem muitos estudos sobre a utilização de ML no preenchimento de dados. A exemplo: Bonfante et al. (2013) e KATİPOĞLU e Reşat (2021) utilizaram redes neurais artificiais para preencher as lacunas nos dados de temperatura e obtiveram resultados bem-sucedidos. Coulibaly e Evora (2007) aplicaram Perceptron Multicamada (MLP do inglês *Multilayer Perceptron*) para preenchimento de dados de precipitação e temperaturas extremas. Kajewska-Szkudlarek e Stańczyk

(2018) investigaram a eficácia do método Regressão de Vetor de Suporte (SVR - do inglês *Support Vector Regression*) entre outros para completar os dados diários ausentes de temperatura e umidade. Tosunoğlu et al. (2020) investigaram a eficiência do k-vizinhos mais próximos (KNN - do inglês *k-nearest neighbors*) entre outros algoritmos para prever o fluxo hídrico mensal.

Na conjuntura de preenchimento de falhas em séries de dados da temperatura do ar, a utilização de ML mostrou-se particularmente relevante e promissora. Muitos estudos têm explorado o uso desses algoritmos para prever não apenas a temperatura do ar, mas também outros parâmetros meteorológicos, como ponto de orvalho (MOHAMMADI et al., 2015), precipitação (LATIF et al., 2023) e estudo populacional de espécies (TCHAKONTE et al., 2023). No entanto, a literatura revela uma lacuna notável: a falta de estudos que realizem uma comparação abrangente e crítica da aplicação de diversos métodos de ML no mesmo contexto.

Um questionamento crucial relacionado aos métodos de ML diz respeito à sua capacidade de generalização. Isso envolve a indagação sobre se o método efetivamente aprende com os dados de treinamento e consegue fazer previsões precisas para situações que não estão contidas nos dados utilizados para treiná-lo. Em outras palavras, é uma preocupação fundamental sobre se o ML é capaz de extrapolar e aplicar o conhecimento adquirido para novas e diferentes circunstâncias.

Com isso, espera-se fornecer uma solução mais precisa e confiável para o preenchimento de falhas em dados micrometeorológicos, contribuindo para o avanço da área de Ciências Ambientais e permitindo uma melhor compreensão dos fenômenos climáticos em escala local. Além disso, essa pesquisa pode servir como base para futuros estudos e aprimoramentos na área de preenchimento de dados faltantes, proporcionando uma maior confiabilidade e integridade às análises climáticas e ambientais.

1.2 JUSTIFICATIVA

As Redes Neurais Informadas pela Física, conhecidas como PINN (do inglês *Physics-Informed Neural Networks*), representam uma abordagem inovadora e promissora na interseção entre a física e o ML. Essa técnica, desenvolvida por Raissi et al. (2019), combina o conhecimento científico preexistente sobre um fenômeno com a flexibilidade e capacidade de aprendizado das redes neurais artificiais. Dessa forma, são capazes de incorporar princípios físicos fundamentais em seus modelos, tornando-as particularmente adequadas para problemas de modelagem

em áreas como ciências ambientais, engenharia, física, entre outras (LAUBSCHER, 2021). O resultado é uma ferramenta poderosa que permite a previsão e otimização de sistemas complexos, muitas vezes com alta precisão, ao mesmo tempo em que mantém uma conexão sólida com as leis físicas subjacentes. Isso as torna uma escolha atraente para lidar com uma ampla gama de problemas do mundo real, nos quais a física e o aprendizado de máquina podem se unir para fornecer *insights* valiosos e soluções práticas.

Embora no início, essa área vem ganhando bastante enfoque nos últimos anos. Karpatne et al. (2017) usaram uma rede neural totalmente conectada para prever o perfil de temperatura e profundidade dos lagos, dado um conjunto de fatores de entrada, incluindo velocidade do vento, temperatura do ar e a época do ano. Para tal feito eles definiram uma função de perda adicional incorporando uma relação empírica conhecida para converter os perfis de temperatura previstos em perfis de densidade. Com isso descobriram que ao utilizar esse termo de perda junto com uma perda supervisionada padrão melhorou a precisão das previsões da rede e as tornaram mais consistentes fisicamente em comparação com a mesma rede treinada sem ele. Zhang et al. (2018) também utilizaram uma rede neural totalmente conectada em combinação com uma função de perda informada pela física e obtiveram resultados promissores para a dinâmica molecular.

O objetivo deste trabalho é desenvolver e implementar um modelo utilizando os conceitos de PINN. Para isso foi desenvolvida uma função de custo informada pela física baseada em uma versão empírica da equação do calor.

Os resultados obtidos foram avaliados quanto à acurácia e comparados com outros sete modelos tradicionais de aprendizado de máquina utilizados na literatura: Regressão Linear (LR), Regressão LASSO (do inglês *Least Absolute Shrinkage and Selection Operator*), Rede Elástica (EN - do inglês *Elastic Net*), KNN, Árvores de Classificação e Regressão (CART - do inglês *Classification And Regression Tree*), SVR e MLP.

Para atingir o objetivo geral deste trabalho, foram definidos os seguintes objetivos específicos:

- Propor uma função de perda informada pela física para aplicar no preenchimento de falhas em séries temporais da temperatura do ar;
- Comparação com outros 7 métodos de ML para a mesma tarefa;
- Treinamento e validação dos modelos criados;
- Análise do desempenho dos modelos criados para 8 localidades;

- Análise da relação do fator climático na performance dos métodos empregados;
- Análise da relação entre a quantidade de dados utilizados no treinamento e a precisão dos modelos criados;
- Analise do tempo de processamento dos modelos.

FUNDAMENTAÇÃO TEÓRICA

2.1 SÉRIES TEMPORAIS MICROMETEOROLÓGICAS

Na pesquisa de fenômenos meteorológicos, é preciso o aferimento de variáveis microclimáticas. Para tal, utilizam-se instrumentos, com sensores diversificados, possibilitando a leitura e análise de uma ou mais variáveis de estudo, como a temperatura do ar, velocidade do vento, radiação, entre outras. As leituras são armazenadas em sistemas de aquisição de dados.

O agrupamento sequencial desses dados corresponde a uma Série Temporal (ST), que pode ser definida como um conjunto de observações coletadas ao longo do tempo, em uma determinada ordem (MORETTIN; TOLOI, 2006, p. 1). Para o criador do modelo auto-regressivo, Yule (YULE, 1927), uma ST deveria ser vista simplesmente como a realização de um processo estocástico. Cada uma das observações pode ser considerada como pontos em um gráfico de duas dimensões, onde o eixo das ordenadas determina as medições dos dados e o eixo das abscissas delimita em que momento discreto do tempo tais medições foram aferidas.

Séries temporais se diferenciam em tendência e sazonalidade. A tendência capta elementos de longo prazo relacionados com a série em si, conforme aponta Chatfield (CHATFIELD, 1996). Já a sazonalidade capta seus padrões regulares ao longo do tempo. A característica elementar de uma ST é o ciclo: uma completa descrição do fenômeno observado, que deve conter as frequências de todos os seus ciclos dominantes (OLIVEIRA; FAVERO, 2002).

Diversos propósitos levam à análise de séries temporais, por exemplo: a análise exploratória (HOAGLIN et al., 2011), análise de entropia (STOSIC et al., 2016), extração de características (CAVALCANTE et al., 2016), e segmentação (KEOGH et al., 2004). Além disso, as análises podem visar a previsão do valor de determinada variável com base nos valores de outras variáveis, medidas no mesmo instante (WNDERLEY, 2012).

No processo de leitura e gravação, falhas podem ocorrer, originárias do próprio funcionamento do equipamento utilizado, devido a mau uso ou defeitos

de fábrica, intempéries, falta de manutenção ou até mesmo erros na manipulação dos sensores. Essas falhas caracterizam a ausência de dados, formando lacunas na ST. Dessa forma, podem levar a afirmações tendenciosas, pois uma série com falhas pode se distanciar do real comportamento do fenômeno estudado. Nesse caso, o erro absoluto e o desvio-padrão tornam-se maiores à medida que aumenta a diferença entre as leituras (SIROIS, 1990).

Para que toda a ST não seja descartada, efetua-se um método de preenchimento de falhas. Teegavarapu e Chu (TEEGAVARAPU; CHANDRAMOULI, 2005) enfatizam que o uso de tratamentos de dados meteorológicos é frequente, pois várias estações apresentam falhas em seus bancos de dados. De acordo com Chibana e Vieira (CHIBANA et al., 2005), vários métodos podem ser utilizados no preenchimento de falhas de dados meteorológicos, incluindo a utilização de médias dos dados observados ou dados sintéticos obtidos de geradores de dados. Portanto, falhas em uma ST micrometeorológica é um problema iterado, que pode prejudicar a pesquisa.

2.2 PREENCHIMENTO DE FALHAS EM SÉRIES TEMPORAIS MICROMETEOROLÓGICAS

O preenchimento de falhas em dados micrometeorológicos refere-se ao processo de estimar valores ausentes em séries temporais de dados meteorológicos. Essas falhas nas séries temporais ocorrem quando não há registros ou observações disponíveis para um determinado intervalo de tempo em uma estação meteorológica. Isso pode ocorrer devido a várias razões, tais como problemas técnicos na coleta de dados, falhas nos sensores, problemas de transmissão, erros humanos ou condições climáticas extremas que impossibilitam a coleta de informações (BRUBACHER et al., 2020).

Diversos métodos estatísticos são amplamente utilizados na manipulação de dados relacionados às variáveis climáticas. No contexto de dados da temperatura do ar, por exemplo, técnicas como LR e KNN têm sido aplicadas com sucesso. Além disso, é comum incorporar dados coletados em estações próximas àquelas afetadas pela lacuna de informações (SABINO; SOUZA, 2022). Em alternativa, frequentemente recorre-se a abordagens que podem incluir a exclusão completa de períodos de dados afetados pela falta de informação ou a aplicação de métodos simplificados, como a repetição ou cálculo da média dos valores mais próximos no tempo ou no espaço conforme detalha o estudo de Júnior et al. (2019).

Para lidar com essas falhas, é necessário realizar o preenchimento, ou seja, estimar os valores ausentes para manter a continuidade e a integridade das séries

temporais. Esse processo é essencial para a análise climática, previsões meteorológicas e outros estudos relacionados ao clima e ao tempo. Portanto, o preenchimento de falhas em dados micrometeorológicos desempenha um papel crucial na obtenção de conjuntos de dados completos e confiáveis (BONFANTE et al., 2013).

A complexidade desse processo reside na necessidade de estimar com precisão os valores ausentes com base no comportamento histórico dos dados, levando em consideração a variabilidade e sazonalidade das condições meteorológicas. A escolha de métodos estatísticos ou algoritmos de ML para essa tarefa depende das características dos dados, da quantidade de dados históricos disponíveis e do contexto específico do problema. Além disso, o preenchimento de falhas também pode ser afetado por características locais, como a geografia do local da estação meteorológica, que pode influenciar os padrões climáticos observados.

Portanto, o preenchimento de falhas em dados micrometeorológicos é uma etapa crítica na análise de séries temporais climáticas e meteorológicas, e a escolha da abordagem adequada é fundamental para obter resultados confiáveis e precisos.

2.3 SELEÇÃO DE VARIÁVEIS E SEPARAÇÃO EM DADOS DE TREINO E TESTE

O objetivo principal de um modelo de aprendizado de máquina é ter um bom desempenho em exemplos não vistos. Uma maneira de avaliar o quão bem o algoritmo aprendeu uma tarefa é avaliar seu desempenho em exemplos que nunca encontrou antes, ou seja, dados que não foram usados durante o treinamento. Se um algoritmo de aprendizado de máquina fosse avaliado usando o mesmo conjunto de dados em que foi treinado, provavelmente alcançaria um bom desempenho. No entanto, suas previsões sobre novos dados seriam inconsistentes porque os dados de treinamento podem não ser representativos, fazendo com que o modelo se tornasse excessivamente específico para determinados dados. Esse fenômeno é conhecido como *overfitting* que ocorre quando o modelo perde sua capacidade de generalização, pois fica muito focado no comportamento específico dos dados de treinamento e falha em capturar os padrões gerais (DIETTERICH, 1995). Por outro lado, *underfitting* também pode ocorrer, onde o modelo falha em capturar o comportamento geral do problema e fornece previsões excessivamente simplificadas (MJALLI et al., 2007).

Em outras palavras, o *overfitting* ocorre quando um modelo não consegue generalizar adequadamente de dados observados para dados que não foram vistos anteriormente. Nesse cenário, o modelo se comporta bem ao lidar com o con-

junto de treinamento, mas seu desempenho é insatisfatório no conjunto de testes. Isso ocorre porque o modelo enfrenta dificuldades ao processar informações do conjunto de teste, que podem ser distintas das informações do conjunto de treinamento (YING, 2019). Outro problema bastante comum é o *underfitting* que significa que o treinamento do modelo foi insuficiente e a precisão do aprendizado é baixa (ZHANG et al., 2019). Sendo assim o melhor cenário é quando não ocorre a presença desses comportamentos. A Figura 1 ilustra os três cenários.

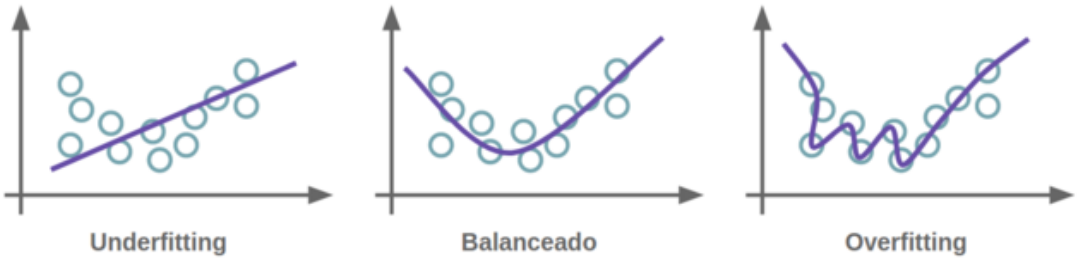


Figura 1: Comparação dos Cenários de *Overfitting*, *Underfitting* e Ajuste Ótimo. Fonte: Wikimedia Commons[1]

A avaliação fornece uma estimativa de quão bem o algoritmo pode funcionar em cenários práticos. Uma boa prática para avaliação é usar reamostragem, que envolve dividir o conjunto de dados em subconjuntos de treinamento e teste. O subconjunto de treinamento é usado para treinar o modelo, enquanto o subconjunto de teste é usado para avaliar seu desempenho, conforme descrito por Brownlee (2016). A Figura 9 abaixo ilustra esse processo.

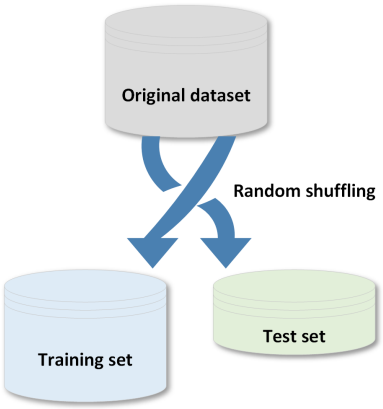


Figura 2: Particionamento dos dados em dados de treino e teste.

Fonte: (BONACCORSO, 2017, p. 44–45)

Na reamostragem dos dados em treinamento e validação, cada ponto de dados removido do subconjunto de treinamento para o conjunto de teste é perdido

[1]Disponível em: <http://commons.wikimedia.org/wiki/File:Underfitting_e_overfitting.png>

para o conjunto de treinamento, o que pode resultar na perda de informações valiosas para o treinamento. Uma maneira de minimizar esse problema é utilizar a validação cruzada (BENGIO; GRANDVALET, 2004). Nesse método, os dados são divididos em K segmentos de tamanho igual, e em cada iteração, um segmento é usado como conjunto de teste, enquanto os segmentos $K-1$ restantes são usados para treinamento. Esse processo é repetido K vezes, alternando circularmente o segmento de teste em cada iteração, e a acurácia do modelo é repetida em cada iteração. A média das acurácias seguidas nas K iterações é então seguida. Esse tipo de validação é conhecido como validação K-fold (Figura 3).

Figura 3: k-fold, onde $k = 4$.

Fonte: do autor.

Outro fator importante a ser considerado é a seleção das variáveis utilizadas no treinamento do modelo de aprendizado de máquina, pois isso tem um impacto significativo no desempenho alcançado pelo modelo. Variáveis irrelevantes ou parcialmente relevantes podem ter um efeito negativo no desempenho do modelo. De acordo com Hawkins (2004), é importante seguir o princípio da parcimônia, que sugere que, se um modelo de regressão com apenas dois preditores é suficiente para explicar uma variável de interesse, não devemos usar mais do que esses dois preditores.

A seleção de variável é um processo em que as variáveis são automaticamente separadas dos dados, levando em consideração aqueles que mais intervieram para a variável de previsão ou saída. Ter variáveis irrelevantes pode diminuir a precisão de muitos modelos, especialmente de algoritmos lineares, como a regressão linear.

A aplicação da seleção de variáveis antes da modelagem traz três benefícios: redução do *overfitting*, melhoria da precisão e redução do tempo de treinamento. Para o processo de seleção pode ser utilizado por exemplo o coeficiente de determinação (R^2) ou o de correlação de Pearson, que mede o grau da correlação linear entre duas variáveis quantitativas, sendo assim, é um índice adimensional

com valores situados ente -1,0 e 1.0, que reflete a intensidade de uma relação linear entre dois conjuntos de dados (STEEL et al., 1997).

Para avaliar a eficácia de um modelo, é essencial que o algoritmo seja capaz de diferenciar dados significativos daqueles que representam ruídos no processo de aprendizado, conforme discutido em (PARIS et al., 2003).

2.4 MODELOS DE APRENDIZADO DE MÁQUINA

O aprendizado de máquina é um subcampo da Inteligência Artificial (IA), definido por Robert (2014) como um conjunto de métodos capazes de detectar automaticamente padrões em dados e utilizá-los para realizar previsões de dados futuros ou tomar decisões em situações de incerteza.

As técnicas de aprendizado de máquina utilizam modelos matemáticos especializados para representar o comportamento dos sistemas e aplicar o conhecimento adquirido. Essas abordagens têm a capacidade de aprimorar o desempenho dos modelos à medida que recebem mais dados, permitindo um aprendizado mais profundo e direcionado para tarefas específicas (KHAN et al., 2019).

Inicialmente, as técnicas de aprendizado de máquina foram aplicadas em domínios desafiadores nos quais a programação manual era inviável, como exemplificado no artigo seminal de Samuel (1959), que destaca o uso do aprendizado de máquina no contexto do xadrez devido à complexidade intrínseca desse jogo, demandando a análise de um grande volume de jogadas. Esse marco histórico demonstrou a capacidade do aprendizado de máquina em abordar problemas complexos e influenciou significativamente o desenvolvimento do campo, aplicando-se em uma ampla variedade de setores.

Quando se depara com uma grande quantidade de dados que precisa ser analisada, pode ser desafiador para a percepção humana identificar padrões, tornando os algoritmos de aprendizado de máquina ferramentas valiosas para identificar regras e exceções, além de substituir programas baseados em regras.

A vantagem da utilização de modelos de aprendizado de máquina em relação aos modelos puramente estatísticos é que eles são orientados a dados, ou seja, não é necessário compreender previamente o comportamento do conjunto de dados que será utilizado para o treinamento do algoritmo. Nesse contexto, não é preciso ter conhecimento prévio sobre a linearidade ou não do modelo, pois o próprio algoritmo realiza o ajuste automaticamente (MICHALSKI et al., 2013). Isso torna o processo de aprendizado mais flexível e adaptável, permitindo que o algoritmo seja aplicado em diferentes tipos de dados sem a necessidade de alterações significativas em sua estrutura.

Uma definição formal pode ser encontrada em Mitchell (1997): "Um programa de computador aprende com a experiência E em relação a uma classe de tarefas T, e medida de desempenho P, se seu desempenho em tarefas T, medido por P, melhora com a experiência E". A exemplo, um programa de aprendizado de máquina que preenche falhas nos dados. Preencher falhas é a tarefa T, estimar o valor do dado faltante é a experiência E e a precisão da estimativa é mensurada pela medida de desempenho P. A tarefa T não aprende como preencher falhas com precisão, mas sim a ação de estimar valores. Normalmente, tarefas de aprendizado de máquina definem qual técnica deve ser usada para processar exemplos, que são experiências pertencentes ao domínio da tarefa. Dessa forma as técnicas de aprendizado de máquina podem aprender usando características medidas quantitativamente de um exemplo.

Muitos tipos de problemas podem ser resolvidos com aprendizado de máquina. Tarefas comuns de aprendizado de máquina incluem:

- tarefas de clusterização (Figura 4A): em que o modelo de aprendizado de máquina tenta agrupar exemplos em grupos por suas semelhanças de recursos.

- tarefas de regressão (Figura 4B): em que se espera que o modelo de aprendizado de máquina gere um valor real para algum exemplo, tentando ajustar uma curva aos dados.

- tarefas de classificação (Figura 4C): em que se espera que o modelo de aprendizagem por máquina retorne qual categoria algum exemplo pertence, em outras palavras, encontrar uma fronteira de decisão.

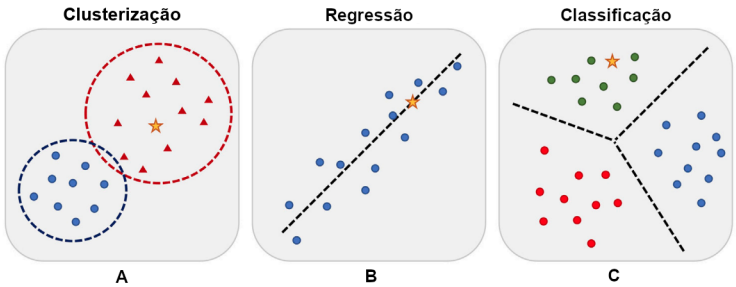


Figura 4: Tipos de aprendizado de máquina: (A) agrupamento de dados por clusterização; (B) ajuste de curva aos dados utilizando regressão e (C) classificação de dados

Fonte: Modificado de Peng et al. (2021)

A avaliação do desempenho P é fundamental para determinar o quão bem uma técnica de aprendizado de máquina realiza uma tarefa T, ou seja, P deve ter uma métrica de avaliação bem definida e diretamente relacionada à tarefa.

Normalmente, a performance de um algoritmo de aprendizado de máquina mede quão bem ele se comporta em novas experiências, a fim de avaliar a generalização do modelo. Uma melhor capacidade de generalização significa que o modelo ainda pode realizar sua tarefa com bom desempenho, mesmo na presença de variações em novos exemplos não vistos anteriormente.

Conforme mencionado na experiência E, que pode ser definida como um conjunto de exemplos, o aprendizado do modelo é realizado a partir desses dados. Quanto mais experiência estiver disponível para o aprendizado, mais o modelo aprenderá e obterá melhor desempenho. As abordagens de ML podem ser geralmente divididas em três tipos: aprendizagem supervisionada, não supervisionada e por reforço, adaptadas para fins de investigação distintos. Algoritmos de aprendizagem supervisionada investigam relações entre variáveis preditivas e resultados de conjuntos de dados de treinamento rotulados e aplicam a regra aprendida para estabelecer um modelo para classificar novos dados (RUSSELL; NORVIG, 2016). Classificação e regressão são duas abordagens principais na aprendizagem supervisionada, onde o modelo de classificação visa prever o resultado da categoria (por exemplo, classificação de espécie de flores (SWAIN et al., 2012)) e o modelo de regressão visa prever um resultado contínuo, como a previsão de preços de imóveis.

Por outro lado, algoritmos de aprendizado não supervisionado são aplicados para descobrir padrões ocultos em dados de treinamento sem rótulos. Abordagens de agrupamento dentro da aprendizagem não supervisionada, incluindo agrupamento hierárquico, agrupamento K-means e modelos de mistura gaussiana, são as técnicas mais populares para reunir dados em classificações anteriormente ambíguas.

Finalmente, a aprendizagem por reforço é programada para se autocorrigir sequencialmente a partir do feedback ambiental (positivo ou negativo) e, portanto, melhorar a função geral do modelo sem ter dados rotulados (KAELBLING et al., 1996). Isso é frequentemente exemplificado em cenários nos quais agentes de inteligência artificial são treinados para tomar decisões, como no aprendizado de máquinas em jogos, em que o agente aprende a otimizar sua estratégia com base nas recompensas ou penalidades recebidas após cada ação.

Com base nos dados e no domínio do problema, disponíveis neste estudo, foi adotado o aprendizado supervisionado. Nessa categoria, o algoritmo tenta ajustar o modelo a partir de dados em que a saída correta já é conhecida, pressupondo uma relação entre a entrada e a saída. Cada exemplo é um par consistindo de um conjunto de entradas e um valor de resposta correspondente. Assim, a

aprendizagem supervisionada envolve observar vários exemplos (X, y) e ajustar o modelo para prever y a partir de X.

A seguir, serão abordados os principais métodos empregados na condução deste estudo.

2.4.1 Regressão Linear (LR)

A regressão linear é um método estatístico utilizado para modelar a relação entre uma variável dependente contínua e um ou mais preditores independentes, descrita pela Equação 2.1.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon \tag{2.1}$$

onde: - y representa a variável dependente contínua que desejamos prever. - $\beta_0, \beta_1, \ldots, \beta_n$ são os coeficientes, também conhecidos como "bias" ou pesos associados aos preditores independentes $x_1, x_2, \ldots, x_n$. - o termo ε na equação representa o erro residual, que é a diferença entre os valores reais da variável dependente (y) e os valores previstos pelo modelo de regressão linear. Em outras palavras, ele captura a parte da variabilidade de y que não pode ser explicada pelos preditores independentes $(x_1, x_2, \ldots, x_n)$ e é atribuída a fatores não considerados no modelo ou à variabilidade aleatória nos dados. Essa parte "não explicada" ou "residual" é fundamental para entender a qualidade do ajuste do modelo aos dados (CONNELLY, 2020). Um erro residual pequeno indica que o modelo se ajusta bem aos dados, enquanto um erro residual grande sugere que o modelo não consegue explicar completamente a variabilidade de y. A análise do erro residual desempenha um papel crucial na avaliação e na validação de modelos de regressão linear.

2.4.2 Regressão Linear Múltipla

A Regressão Linear Múltipla (RLM) é uma técnica estatística que envolve o uso de duas ou mais variáveis independentes para prever uma única variável dependente y. É uma extensão da análise de regressão linear simples, onde o objetivo é estabelecer uma equação que possa ser utilizada para fazer previsões da variável dependente com base nos valores das variáveis independentes ou explanatórias ou covariáveis ou regressoras x. A principal vantagem da RLM em relação à regressão linear simples é que a inclusão de variáveis independentes adicionais melhora a capacidade de previsão do modelo. Isso ocorre porque as variáveis independentes adicionais podem capturar mais informações e explicar melhor a variação na variável dependente (LEVINE et al., 2005), (MONTGOMERY et al.,

2021).

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_k x_{ik} + \varepsilon_i, i = 1, ..., n \tag{2.2}$$

onde:

n é o tamanho da amostra;

y_i representa a observação da variável dependente para a i-ésima posição na amostra, onde i varia de 1 a n, representando cada ponto de dados individual.

β_0 o termo independente da variável;

β_k a inclinação de y em relação à variável x_k;

ε_i = é um componente de erro aleatório em y, para a observação i.

Assume-se que esses erros são independentes e seguem distribuição normal com média zero e variância desconhecida (σ^2).

A denominação múltipla vem do fato do modelo envolver mais de um coeficiente de regressão. O modelo é linear aos parâmetros $\beta = (\beta_0, \beta_1, \beta_2, ..., \beta_k)$, por isso a descrição de linearidade.

2.4.3 Regressão LASSO

Lasso (Equação 2.3) é um tipo de regressão linear usada para seleção e regularização de recursos (FRIEDMAN et al., 2010). Adicionar um termo de penalidade à função de custo do modelo é uma técnica usada para evitar o sobreajuste (*overfitting*). Isso incentiva o modelo a usar menos variáveis ou recursos na equação de regressão final. O termo de penalidade é baseado nos valores absolutos dos coeficientes na equação,significando que alguns podem ser zerados se forem considerados menos importantes.

$$(Y - X\beta)^T(Y - X\beta) + \lambda|\beta|_1 \tag{2.3}$$

- onde $|\beta|_1 = \sum_{j=1}^{p} |\beta|_j$.

A soma $|\beta|_1 = \sum_{j=1}^{p} |\beta|_j$ representa a soma dos valores absolutos dos coeficientes β no contexto:

- p é o número total de coeficientes no modelo de regressão.

- Cada $|\beta|_j$ é o valor absoluto de um coeficiente específico β_j no modelo.

- A notação $|\beta|_1$ denota a soma de todos esses valores absolutos dos coeficientes, variando de $j = 1$ a $j = p$.

A penalidade $|\beta|_1$ é conhecida como "norma L1" ou "norma do valor abso-

luto". A sua aplicação no LASSO tem o efeito de incentivar o modelo a reduzir alguns coeficientes a zero. Isso resulta na seleção automática de variáveis, mantendo apenas aquelas consideradas mais relevantes para a previsão da variável dependente. Portanto, o LASSO atua como um método de seleção de variáveis, o que é útil para controlar a complexidade do modelo, evitar o sobreajuste e manter apenas as variáveis mais informativas.

O maior benefício do LASSO é a capacidade do modelo de criar matrizes esparsas. Uma desvantagem é que, como a penalidade L_1 contém valores absolutos, é muito difícil resolvê-la analiticamente (CHATTERJEE et al., 2011).

2.4.4 Rede Elástica (EN)

A rede elástica (Equação 2.4) é um modelo estatístico utilizado na regressão linear para lidar com problemas de multicolinearidade, em que existem altas correlações entre as variáveis independentes. É uma combinação entre a regressão Ridge e a regressão LASSO, incorporando termos de regularização para controlar a complexidade do modelo e evitar o *overfitting* (RAOUHI et al., 2022).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \lambda_1 \sum_{i=1}^{n} |\beta_i| + \lambda_2 \sum_{i=1}^{n} \beta_i^2 \tag{2.4}$$

onde:

- y representa a variável dependente que estamos tentando prever.

- $\beta_0, \beta_1, \ldots, \beta_n$ são os coeficientes associados às variáveis independentes $x_1, x_2, \ldots, x_n$.

- $x_1, x_2, \ldots, x_n$ são as variáveis independentes.

- λ_1 e λ_2 são os parâmetros de regularização que controlam a penalização aplicada aos coeficientes.

O primeiro termo $\lambda_1 \sum_{i=1}^{n} |\beta_i|$ corresponde à regularização LASSO, que introduz uma penalidade proporcional à soma dos valores absolutos dos coeficientes. O segundo termo $\lambda_2 \sum_{i=1}^{n} \beta_i^2$ corresponde à regularização Ridge, que introduz uma penalidade proporcional à soma dos quadrados dos coeficientes.

A rede elástica combina essas duas penalidades para encontrar o equilíbrio entre a seleção de variáveis e a estabilização dos coeficientes. Os parâmetros λ_1 e λ_2 controlam o grau de regularização e podem ser ajustados para melhor se adequarem aos dados.

A utilização da rede elástica na regressão linear ajuda a lidar com pro-

blemas de multicolinearidade, reduzindo a complexidade do modelo, ao mesmo tempo em que mantém a interpretabilidade e a estabilidade dos coeficientes estimados.

2.4.5 K-vizinhos Mais Próximos (KNN)

O KNN é um algoritmo de aprendizado de máquina usado para problemas de regressão e classificação. Embora seja mais comumente usado para classificação, é possível aplicá-lo à regressão linear multivariada usando uma abordagem modificada.

A ideia básica é prever o valor de uma variável dependente com base nos valores das variáveis independentes das k observações mais próximas no espaço de atributos (FIX; HODGES, 1989). A equação da 2.5 multivariada usando o modelo KNN pode ser representada da seguinte forma:

$$y = \frac{1}{k} \sum_{i=1}^{k} y_i \tag{2.5}$$

onde:

- y é a variável dependente que estamos tentando prever.
- y_i é o valor da variável dependente da i-ésima observação mais próxima.
- k é o número de observações mais próximas que serão consideradas para a previsão.

O modelo KNN para regressão linear multivariada simplesmente estima o valor da variável dependente y como a média dos valores das variáveis dependentes (y_i) das k observações mais próximas (PATRICK; III, 1970).

Nesse trabalho foi adotada a distância euclidiana, mostrada na Equação 2.6, como técnica padrão. Essa distância é calculada entre as observações no espaço de atributos, que são representadas por vetores de valores das variáveis independentes.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.6}$$

2.4.6 Árvores de Classificação e Regressão (CART)

Quando utilizado para regressão linear multivariada, o CART busca criar uma árvore binária com base nas entradas $x_1, x_2, \ldots, x_n$ para prever uma resposta. Em cada nó da árvore, é realizado um teste em uma das entradas, representado

como x_i. Conforme o resultado do teste, o ramo esquerdo ou direito da árvore é selecionado. Eventualmente, chega-se a um nó folha, onde uma previsão é gerada. Essa previsão consiste na agregação ou cálculo da média de todos os pontos de dados de treinamento que alcançam aquele nó folha. O processo de criação de um modelo é repetido para cada uma das variáveis independentes. No caso de cada variável individual, utiliza-se o erro quadrático médio para determinar a melhor divisão possível. O número máximo de recursos considerados em cada divisão é definido como o total de recursos disponíveis (MORI; TAKAHASHI, 2012).

Uma vez construída a árvore, a previsão é feita seguindo os caminhos da árvore a partir do nó raiz até chegar a um nó folha. O valor previsto para uma observação é o valor contido no nó folha alcançado (MEGETO et al., 2014).

2.4.7 Regressão de Vetor de Suporte (SVR)

O SVR é um modelo de aprendizado de máquina utilizado para realizar tarefas de regressão. Ao contrário dos modelos de regressão linear tradicionais, o SVR é baseado em *Support Vector Machines* (SVM) e busca encontrar uma função que se ajuste aos dados de treinamento, tentando minimizar a distância entre os pontos e a função estimada, ao mesmo tempo em que limita o erro total (AWAD et al., 2015). Sua formulação matemática pode ser expressa pela Equação 2.7.

Dado um conjunto de treinamento $(\mathbf{x}_i, y_i)$, onde $\mathbf{x}_i$ é o vetor de características (entrada) do i-ésimo exemplo e y_i é o valor real (saída), a tarefa é encontrar a função $f(\mathbf{x})$ que estima os valores alvo y.

$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\zeta_i + \zeta_i^*) \tag{2.7}$$

sujeito a:
$y_i - \langle w, \phi(\mathbf{x}_i)\rangle - b \leq \varepsilon + \zeta_i$
$\langle w, \phi(\mathbf{x}_i)\rangle + b - y_i \leq \varepsilon + \zeta_i^*$
$\zeta_i, \zeta_i^* \geq 0, i = 1, \ldots, N$

onde: - w é o vetor de pesos que multiplica os vetores de características $\phi(\mathbf{x}_i)$; - $\phi(\mathbf{x}_i)$ é uma função que mapeia as características de entrada para um espaço de maior dimensão (kernel trick); - b é o termo de viés (bias); - ζ_i e ζ_i^* são variáveis de folga, que representam a quantidade de erro tolerado para cada exemplo; - C é um hiperparâmetro que controla o trade-off entre o erro de treinamento e a complexidade do modelo; - ε é uma margem de tolerância que controla a largura da faixa em torno da função estimada em que os pontos de

treinamento podem residir.

O objetivo é minimizar a função de custo, que consiste em duas partes: o termo $\frac{1}{2}||w||^2$, que busca minimizar a norma dos pesos e, consequentemente, a complexidade do modelo; e o termo $\sum_{i=1}^{N}(\zeta_i + \zeta_i^*)$, que penaliza valores fora da faixa de tolerância.

O SVR é um poderoso modelo para tarefas de regressão, permitindo lidar com dados não lineares usando diferentes funções de kernel (ϕ). Isso possibilita a captura de relações complexas entre as características e os valores alvo, tornando o SVR uma opção versátil para diversas aplicações (SMOLA; SCHÖLKOPF, 2004).

2.4.8 Redes Neurais Artificiais

Redes Neurais Artificiais (RNA) são modelos computacionais inspirados no cérebro humano, que buscam imitar a função dos neurônios biológicos e suas conexões sinápticas. Essas redes são capazes de processar dados, identificar padrões e memorizar informações, assim como o cérebro humano faz. O modelo inicial de um neurônio artificial, conhecido como perceptron, foi proposto por Rosenblatt (1958) e aprimorado posteriormente por Widrow et al. (1960). O perceptron é um exemplo fundamental de uma RNA de única camada, capaz de realizar classificação linear simples. A sua operação envolve a soma ponderada das entradas, seguida de uma função de ativação para produzir uma saída. A Figura 5 ilustra uma rede neural com um único neurônio artificial, exemplificando suas entradas x_1, x_2 ... x_m, com seus correspondentes pesos w_{k1}, w_{k2}... w_{km}, soma ponderada das entradas e a função de ativação.

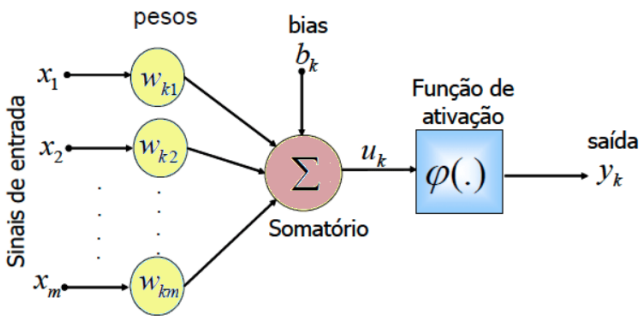


Figura 5: Uma rede neural com um único neurônio. Nessa configuração, a saída do neurônio é a saída geral da rede neural.

Fonte: Adaptado de Haykin (2009)

O somatório $\sum_{i=1}^{n} w_i x_i$, onde w_i é o peso de cada sinapse i, que representa a ponderação relativa da mesma, enquanto x_i é o valor do sinal de entrada. A unidade dispara se a soma ponderada das entradas atingir ou exceder o limite definido pela a função de ativação.

De acordo com Loesch e Sari (1996), as sinapses nas redes neurais artificiais podem ser descritas como atualizações dos pesos das conexões entre os neurônios. Esses pesos representam a importância relativa das conexões na transmissão de informações pela rede. Através do ajuste desses pesos, as redes neurais podem aprender e adaptar-se a diferentes tarefas e conjuntos de dados. Principe et al. (2000) define as redes neurais como máquinas de aprendizado distribuídas, adaptativas e geralmente não-lineares, compostas por várias unidades de processamento independentes. Essas unidades de processamento, também conhecidas como neurônios artificiais, recebem entradas ponderadas pelos pesos correspondentes e produzem uma saída através de uma função de ativação.

A função de ativação, também conhecida como função de transferência, desempenha um papel fundamental no comportamento de uma rede neural. Essa função é predefinida, parametrizada e tem um impacto direto na saída de um neurônio, limitando seus valores a um intervalo específico. Existem várias funções matemáticas que podem ser usadas como funções de ativação, com a função ReLU sendo um exemplo amplamente adotado, muitas vezes empiricamente, em vez de por meio de um processo de seleção adequado baseado em dados (BANERJEE et al., 2019). A função ReLU é não linear e possui um comportamento distinto, retornando zero para valores negativos e crescendo de forma linear para valores positivos, como ilustrado na Figura 6 (GULLI; PAL, 2017). Em síntese trata-se uma função linear amplamente utilizada em redes neurais artificiais (AGARAP, 2018).Essa função retorna o valor da entrada se for positivo ou zero, caso contrário, retorna zero. Em termos simples, a função ReLU ativa um neurônio se a entrada for positiva, e o desativa se a entrada for negativa.

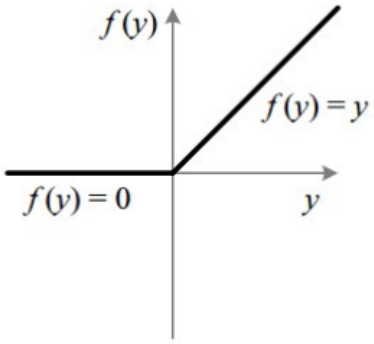


Figura 6: Comportamento da função ReLU.

Fonte: modificada de Gulli e Pal (2017)

Para um modelo com um único neurônio, a saída é um único valor. No entanto, nas redes neurais multicamadas, a saída de um neurônio se torna a entrada para outros neurônios. De acordo com Cybenko (1989), uma rede neural com mais de uma camada é capaz de aproximar qualquer função matemática. Essa propriedade torna as redes neurais multicamadas uma ferramenta poderosa para modelagem e previsão em diversas áreas.

Uma arquitetura comum de rede neural é o MLP (do inglês *Multilayer Perceptron*). Nessa arquitetura, os neurônios são organizados em camadas, e a saída de cada neurônio em uma camada é utilizada como entrada para os neurônios da camada seguinte. Essa estrutura em camadas permite que a rede neural aprenda representações hierárquicas e complexas dos dados. A Figura 7 ilustra um exemplo de uma MLP (HAYKIN, 2009).

A arquitetura MP é amplamente utilizada em problemas de classificação, regressão e outras tarefas de aprendizado de máquina. Ela oferece flexibilidade e capacidade de lidar com dados de alta dimensionalidade. Além disso, a MLP pode ser treinada usando algoritmos de aprendizado supervisionado, como o algoritmo de retro-propagação de erro. Essas características tornam a arquitetura uma escolha popular e eficaz para muitas aplicações de redes neurais. No entanto, é importante ressaltar que existem outras arquiteturas de redes neurais, cada uma com suas próprias características e funcionalidades específicas. (ENGELBRECHT, 2007),(BARROW D. K., 2016).

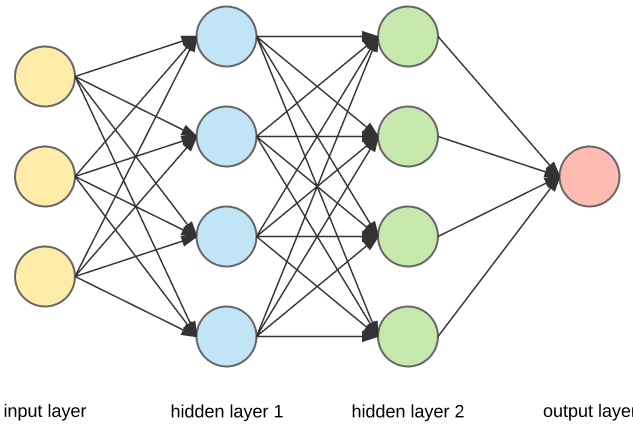


Figura 7: Rede neural multicamada. Fonte: Dertat, 2017 [2]

As camadas intermediárias de uma rede neural, comumente chamadas de camadas ocultas, desempenham um papel crucial no processamento de informações. No entanto, essas camadas são denominadas "ocultas" porque não há interação direta com o mundo exterior; elas servem como estágios intermediários no cálculo de saídas a partir das entradas. A arquitetura de uma rede neural é caracterizada por ser completamente conectada, o que significa que cada neurônio em uma camada está conectado a todos os neurônios da camada seguinte. A propagação de dados através dessas camadas intermediárias segue o princípio de *feed-forward*, que envolve a transformação progressiva dos dados à medida que passam pelas diferentes camadas. Nesse processo, cada neurônio aplica uma função de ativação aos seus inputs ponderados. *Feed-forward* significa que os dados fluem em uma única direção, da camada de entrada para as camadas ocultas e, finalmente, para a camada de saída, sem que haja ciclos ou retroalimentação direta, o que é uma característica fundamental na operação de redes neurais (HAYKIN, 2009).

A formulação matemática geral do MLP para regressão pode ser expressa da seguinte forma:

Dado um conjunto de treinamento $(\mathbf{x}_i, y_i)$, onde $\mathbf{x}_i$ é o vetor de características (entrada) do i-ésimo exemplo e y_i é o valor real (saída), a tarefa é encontrar a função $f(\mathbf{x})$ que estima os valores alvo y. Seja L o número de camadas no MLP (incluindo a camada de entrada e a camada de saída), N_l o número de neurônios na camada l, $\mathbf{W}_l$ a matriz de pesos da camada l e b_l o vetor de viés da camada l. A propagação para a frente do MLP é feita da seguinte forma:

$$\mathbf{a}^{(0)} = \mathbf{x} \quad \text{(entrada)}$$

[2]Fonte: Dertat, 2017. Disponível em: <https://towardsdatascience.com>

$$\mathbf{z}^{(l)} = \mathbf{W}_l \mathbf{a}^{(l-1)} + \mathbf{b}_l \quad \text{(ativação ponderada)}$$

$$\mathbf{a}^{(l)} = \text{ReLU}(\mathbf{z}^{(l)}) \quad \text{(ativação ReLU)}$$

onde $l = 1, 2, \ldots, L-1$ e $\mathbf{a}^{(l)}$ a ativação na camada l.

A camada de saída L pode ter um único neurônio para problemas de regressão, e a saída é dada por:

$$\hat{y} = \mathbf{w}^{(L)} \mathbf{a}^{(L-1)} + b^{(L)}$$

de forma que $\mathbf{w}^{(L)}$ é o vetor de pesos da camada de saída e $b^{(L)}$ é o viés da camada de saída.

Após definir a arquitetura da rede neural, é necessário escolher um algoritmo de treinamento adequado. O objetivo do treinamento é ajustar os pesos sinápticos da rede de forma a minimizar o erro médio quadrático entre a saída desejada e a saída obtida pelo neurônio de saída. O treinamento pode ser visto como um processo de estimativa dos pesos sinápticos (FACELI et al., 2011).

Um algoritmo comumente utilizado para o treinamento supervisionado de redes neurais multicamadas é o algoritmo de retropropagação, proposto por Rumelhart et al. (1986). Esse algoritmo é baseado no método do gradiente descendente e requer o uso de uma função de ativação contínua, diferenciável e preferencialmente não decrescente.

A principal ideia por trás do algoritmo de retropropagação reside na percepção de que os termos ε (por vezes chamados de "sinais de erro") podem ser calculados de forma recursiva por meio da regra da cadeia, em vez de serem medidos diretamente por meio de injeção de ruído e observação dos resultados. Na camada de saída, a forma de calcular é relativamente direta e é dada por $\epsilon_i = y_i - t_i$, onde ϵ_i é a saída da rede no neurônio de saída e t_i é o valor alvo correspondente. Em todas as outras camadas, ϵ_j é calculado a partir de todos os ϵ_k na camada acima, permitindo que os sinais de erro fluam retroativamente pela rede, começando pelos sinais de erro nas unidades de saída, que são as derivadas da função de erro (LILLICRAP et al., 2020).

A Equação 2.8 representa uma fórmula para a atualização dos pesos sinápticos, levando em consideração a segunda derivada da função de ativação f em relação à variável v_j.

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} = -\eta h_i \epsilon_j \text{ onde } \epsilon_j = e_j f'(v_j) = \left(\sum_k \epsilon_k W_{jk} \right) f'(v_j) \qquad (2.8)$$

Onde: - ΔW_{ij} representa a mudança no peso W_{ij} que está sendo atualizado. - η é a taxa de aprendizado, um hiperparâmetro que controla o tamanho dos ajustes de peso. - h_i é a saída do neurônio i na camada atual. - ϵ_j é o "sinal de erro" do neurônio j na camada atual.

O "sinal de erro" ϵ_j é calculado com base na diferença entre a saída do neurônio e o valor-alvo $(y_j - t_j)$, multiplicado pela derivada da função de ativação $f'(v_j)$. A atualização de peso ΔW_{ij} é proporcional a esse sinal de erro e à saída do neurônio i.

Uma função de custo, muitas vezes chamada de função de perda ou erro, quantifica o grau pelo qual as saídas finais da rede (y_i) se desviam de seus valores-alvo (t_i), fazendo uma média dos erros de todos os exemplos de treinamento e retornando um valor positivo. Uma escolha comum para essa função é o erro quadrático médio (MSE), representada pela Equação 2.9.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.9}$$

-onde: - n representa o número de observações ou pontos de dados no conjunto de dados. - y_i é o valor real (observado) para a i-ésima observação. - $\hat{y}_i$ é o valor previsto pelo modelo para a i-ésima observação.

2.4.9 Parâmetros Externos da Rede

Fora a definição de configurações internas, como a quantidade de neurônios e camadas a serem utilizadas no modelo, é necessário estabelecer alguns parâmetros externos (KANDEL et al., 2000), conhecidos como hiperparâmetros. Esses hiperparâmetros incluem a quantidade de épocas, a função de custo, o otimizador da descida do gradiente, taxa de aprendizado (TA) e parada antecipada. A configuração adequada desses parâmetros pode ter um impacto significativo na capacidade de generalização da rede neural (BENGIO et al., 2007), (GRABOCKA et al., 2014).

A quantidade de épocas refere-se ao número de iterações do algoritmo de treinamento. Geralmente, é definida como o número de ciclos de treinamento completos, representado por um valor inteiro. A escolha adequada desse valor é importante para garantir um treinamento eficiente e evitar o *overfitting*.

O otimizador da descida do gradiente é responsável por otimizar a função de custo durante o treinamento. Um método amplamente utilizado é o algoritmo de Estimativa de Momento Adaptativo (ADAM do inglês *Adaptive Moment Estimation*), proposto por Kingma e Ba (2014). Esse otimizador é eficiente pois

requer apenas gradientes de primeira ordem e consome pouca memória.

TA é um hiperparâmetro crucial que determina o tamanho do passo que o algoritmo de treinamento dá em direção à convergência. A definição adequada desse valor requer conhecimento da tarefa em questão. Valores muito pequenos podem levar a um treinamento lento, enquanto valores muito grandes podem causar divergência no processo de treinamento (KLEINBAUM; KLEIN, 1994).

A parada antecipada é uma técnica utilizada para evitar que o treinamento continue além do ponto de convergência, economizando recursos computacionais e evitando o *overfitting*. Esta estratégia envolve o monitoramento do desempenho do modelo em um conjunto de validação durante o treinamento. Quando não se observa melhoria no desempenho, medida por uma variação Δ, por um período contínuo de X épocas (paciência), o treinamento é interrompido. Isso assegura que o modelo pare de treinar quando já não está mais aprendendo de forma significativa e evita que ele se ajuste demais aos dados de treinamento.

A escolha adequada dos hiperparâmetros é essencial para obter um bom desempenho da rede neural. É comum realizar experimentos e ajustes iterativos para encontrar a combinação ideal de hiperparâmetros para cada tarefa específica (CHEN et al., 2023).

2.5 REDES NEURAIS INFORMADAS POR FÍSICA

PINN é uma técnica desenvolvida por Raissi et al. (2019) que representa uma abordagem inovadora para resolver problemas de Equações Diferenciais Parciais (PDEs) e outras equações físicas complexas. Essa metodologia combina conceitos do aprendizado de máquina e da física para obter estimativas precisas e eficientes de soluções em diferentes contextos científicos e de engenharia (KIDGER et al., 2020).

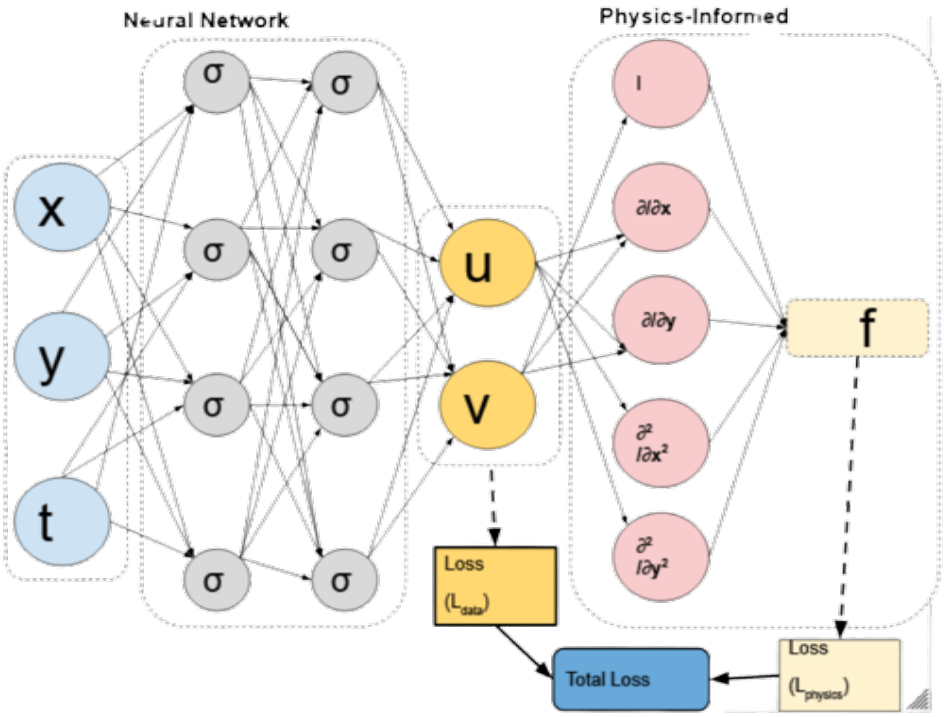


Figura 8: Diagrama esquemático de um modelo PINN. Fonte: Lawal et al. (2022)

Uma das principais características das PINNs é a sua capacidade de incorporar conhecimento prévio da física do problema diretamente na estrutura da rede neural. Esse conhecimento é representado pelas equações governantes que descrevem o comportamento do sistema físico (RAISSI et al., 2019). Ao fazer isso, as PINNs conseguem explorar a estrutura subjacente do problema e fornecer estimativas confiáveis mesmo em cenários com poucos dados observados ou com ruídos nos dados (MISHRA; MOLINARO, 2022).

O treinamento de uma PINN envolve a minimização de uma função de perda, que é composta por vários termos. Esses termos incluem as condições iniciais e de contorno, que são diretamente incorporadas nas camadas iniciais da rede neural, bem como o resíduo da equação governante, que é avaliado em pontos específicos do domínio chamados pontos de colocação (GAO et al., 2023). Dessa forma, a rede neural é ajustada para se adequar às condições físicas do problema e produzir estimativas coerentes em todo o domínio (MISHRA S., 2020).

Uma das vantagens das PINNs é sua versatilidade para lidar com diferentes tipos de equações físicas, incluindo EDPs não lineares, equações integrais, equações fracionárias e até mesmo EDPs estocásticas. Isso as tornam uma poderosa ferramenta para uma ampla gama de aplicações científicas e de engenharia, como modelagem climática, simulação de escoamentos de fluidos, mecânica dos sólidos, dinâmica de partículas e muitos outros campos(LU L., 2019).

Outro ponto interessante é que PINNs também podem ser usadas para resolver problemas inversos. Ou seja, além de estimar as soluções de um sistema físico conhecido, podem ser usadas para identificar parâmetros desconhecidos do modelo a partir de dados observados. Isso tem implicações importantes em problemas de calibração de modelos, identificação de propriedades materiais e otimização de projetos (WANG et al., 2022).

No entanto, apesar de suas vantagens, as PINNs ainda apresentam desafios e questões a serem abordadas. O treinamento dessas redes neurais pode ser computacionalmente custoso, especialmente para problemas com domínios de alta dimensão (WANG et al., 2022). Além disso, a formulação adequada da função de perda e a escolha dos hiperparâmetros são cruciais para o desempenho e a convergência do modelo.

A pesquisa contínua em PINNs está avançando rapidamente, e novas variantes e melhorias estão sendo propostas regularmente. Pesquisadores estão explorando diferentes estratégias de treinamento, arquiteturas de rede, abordagens de discretização e técnicas de otimização para melhorar a eficiência e a precisão dos PINNs (LU L., 2019).

Em resumo, as Redes Neurais Informadas por Física são uma inovadora abordagem que combina aprendizado de máquina com física para resolver problemas de equações diferenciais parciais e outras equações físicas complexas. Essa metodologia tem mostrado resultados promissores em diversas aplicações científicas e de engenharia e continua a ser uma área de pesquisa ativa e empolgante. Com avanços contínuos, espera-se que as PINNs sejam cada vez mais adotadas como uma ferramenta valiosa para a modelagem e simulação de sistemas físicos em uma variedade de áreas.

MATERIAIS E MÉTODOS

Os dados utilizados neste trabalho são provenientes de oito estações meteorológicas automáticas (EMA), pertencentes ao Instituto Nacional de Meteorologia (INMET), disponíveis em sua própria base de dados[1], localizadas em diferentes estados, nas cidades de Campina Verde - MG (EMA A519), Sorriso - MT (EMA A904), Diamante do Norte - PR (EMA A849), Campo Bom - RS (EMA A884), Barcelos - AM (EMA A128), Macapá - AP (EMA A249), Petrolina - PE (EMA A307) e Irêce - BA (EMA A424), coletados no período de 03/12/2013 a 01/06/2023. As localizações podem ser conferidas na Figura 9.

As estações foram selecionadas com o objetivo de assegurar diversidade nas séries temporais. Isso leva em consideração fatores climáticos que desempenham um papel significativo na influência sobre o comportamento das variáveis analisadas neste estudo, característica fundamental para a avaliação do desempenho de cada modelo. Portanto, cada localidade possui seu próprio conjunto de dados, permitindo a análise individual de cada uma delas.

[1]Base de dados INMET: http://www.inmet.gov.br

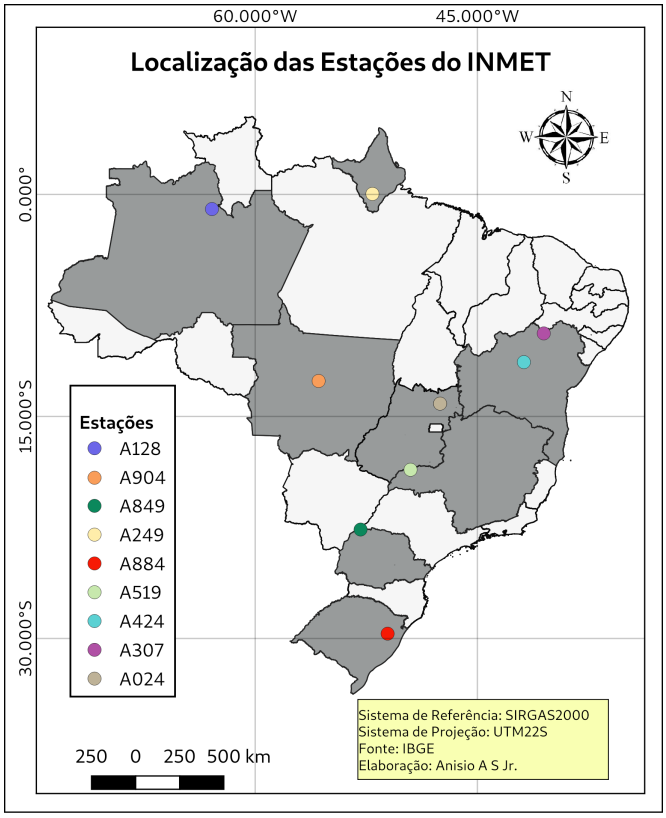


Figura 9: Localização das oito estações meteorológicas, pertencentes ao INMET, utilizadas no estudo.

De acordo com INMET (2011), as leituras foram realizadas a cada cinco segundos, no final da décima segunda leitura, é armazenada a média das mesmas. Nesse estudo foi adotada a média horária. Os dados são compostos pelas as variáveis microclimáticas: radiação solar (kJ/m^2), velocidade do vento (m/s), ponto de orvalho (°C), umidade relativa do ar (%), pressão atmosférica (hPa) e temperatura do ar (°C). A Tabela 1 exibe os valores dos desvios padrão e média de cada variável por localidade juntamente com a classificação climática de Köppen e Geiger (1928).

Tabela 1: Variáveis microclimáticas por localidade com desvio padrão (σ) e média (μ).

Localizações	**Clima Predominante(Köppen)**	**Radiação Solar** (kJ/m^2)		**Velocidade do Vento** (m/s)		**Ponto de Orvalho (°C)**		**Umidade Relativa do ar (%)**		**Pressão atmosférica** (hPa)		**Temperatura do ar (°C)**	
		σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ
Campo Bom – RS	Cfa	1007,763	656,199	0,865	1,274	4,715	15,431	17,030	75,691	5,625	1012,790	6,166	20,416
Sorriso – MT	Aw	1012,098	729,877	1,247	0,952	3,598	19,076	19,885	69,285	2,417	969,247	4,173	26,018
Diamante do Norte – PR	Cfa	1160,585	867,862	1,758	1,927	4,268	17,080	18,526	69,107	3,751	971,295	5,034	23,803
Campina Verde -MG	Aw	1126,969	854,563	1,377	1,797	4,968	15,972	21,083	64,330	3,183	950,645	5,117	24,266
Barcelos -AM	Af	1057,887	742,776	0,951	0,752	1,160	23,186	13,389	84,466	3,548	1007,413	3,156	26,287
Macapá -AP	Am	1140,148	818,429	1,157	2,044	0,917	22,927	13,088	78,755	1,872	1009,780	2,656	27,214
Petrolina -PE	BSh	975,549	722,797	1,330	3,471	2,851	15,471	16,709	51,347	2,869	971,259	3,939	27,336
Irêce -BA	BSh	1211,223	923,154	1,530	2,773	3,018	15,176	19,877	61,261	2,546	928,487	4,391	24,113

3.1 CONFIGURAÇÃO DOS MODELOS

Neste estudo, um modelo de PINN foi desenvolvido e implementado para avaliar sua capacidade de previsão de dados observados. Além disso, seus resultados foram comparados com os de sete outros modelos, a saber, LR, LASSO, EN, KNN, CART, SVR e MLP. Todos os modelos foram implementados no TensorFlow em Python. Os scripts utilizados estão disponíveis para download[2].

Nos modelos EN (Equação 2.4) e LASSO (Equação 2.3), foi utilizado um valor de $\lambda = 1$, realizando 1000 iterações (p) e com uma tolerância de 10^{-3} para o algoritmo de otimização. Essa configuração permite controlar a intensidade da penalização $L1$ aplicada aos coeficientes do modelo, garantindo que alguns coeficientes sejam reduzidos a zero e, assim, realizando a seleção de variáveis de forma automática. O número de iterações e a tolerância são ajustados para garantir que o algoritmo de otimização convirja eficientemente e forneça uma solução satisfatória.

No modelo KNN, foi aplicado o número de vizinhos igual a cinco ($k = 5$) associado à distância euclidiana. Para o modelo CART foi utilizado o MSE (Equação 2.9) como critério para medição da impureza de um nó, com um número mínimo de amostras igual a 2 e número por folha igual a 1.

Para as RNAs: PINN e MLP, foi utilizada a mesma arquitetura de rede *feedfoward* totalmente conectada, com as unidades básicas de computação (neurônios) empilhadas em camadas, compostas por uma camada de entrada, sete camadas ocultas e uma camada de saída (com um neurônio). A quantidade de neurônios na camada de entrada é de igual valor a dimensionalidade da alimentação inicial (neurônios = 6), com cada neurônio ligado a todos neurônios da camada anterior com parâmetros ajustáveis. A Tabela 2 apresenta detalhes da arquitetura.

A função ReLu (Figura 6) foi empregada como função de ativação. O algoritmo ADAM foi selecionado como otimizador da descida do gradiente, com uma taxa de aprendizado configurada em $TA = 10^{-4}$. Além disso, foi implementada uma estratégia de parada antecipada, onde o treinamento é interrompido quando a melhoria no desempenho se torna inferior a $\Delta = 10^{-4}$ por um período contínuo de 30 épocas.

[2]Scripts: https://github.com/anisio002/tese_script.git

Camadas	**Neurônios**	**Parâmetros**
Camada de entrada	6	0
camada oculta 1	320	2240
camada oculta 2	80	25680
camada oculta 3	80	6480
camada oculta 4	80	6480
camada oculta 5	16	1296
camada oculta 6	12	204
camada oculta 7	7	91
Camada de saída	1	8
Total parâmetros		**42,479**
Parâmetros treinados		**42,479**
Parâmetros não treinados		**0**

Tabela 2: Resumo da arquitetura MLP.

No modelo PINN, foi adotado o modelo teórico não-empírico proposto por Paulo et al. (2023). Este modelo considera que a variação da temperatura do ar está intimamente relacionada ao equilíbrio entre a radiação solar líquida (diferença entre a radiação solar incidente e a refletida) e a radiação infravermelha emitida pela superfície e pela atmosfera. O equilíbrio é expresso por meio da Equação 3.1, que descreve como a taxa de variação da temperatura está condicionada a esses fatores, além da densidade do ar, do calor específico do ar seco e de constantes físicas relevantes.

$$\frac{dT}{dt} = \frac{\alpha R - \gamma \sigma T^4}{\rho_a c_a} \tag{3.1}$$

Neste contexto, ρ_a representa a densidade do ar (com o valor de $1.184\,\mathrm{kg/m^3}$), c_a é o calor específico do ar seco ($1012\,\mathrm{J/kg^{\circ}C}$), R $\left(W/m^2\right)$ denota o fluxo líquido de radiação solar (ou seja, a radiação solar absorvida efetivamente pelo ar ou ecossistema), T representa a temperatura do ar em Kelvin (K), σ é a constante de Stefan-Boltzmann ($5.67 \times 10^{-8}\,\mathrm{W\ m^{-2}K^{-4}}$), e α e γ são parâmetros ajustáveis ($2.7 \times 10^{-3}\,\mathrm{m^{-1}}$ e $6.5 \times 10^{-4}\,\mathrm{m^{-1}}$, respectivamente).

É importante notar que as constantes γ e α são tratadas como parâmetros ajustáveis e, no estudo citado, foram definidas com base nas condições específicas do município de Sinop (MT). No presente estudo, os mesmos valores foram empregados, mantendo a coerência com o estudo anterior. No valor de R foi emprega a média de cada localidade.

3.2 PRÉ-PROCESSAMENTO, RE-AMOSTRAGEM E AVALIAÇÃO

O pré-processamento dos dados envolveu inspeção e exclusão de linhas com dados ausentes ou inválidos e ajuste de dimensionalidade dos mesmos (Equação 3.2), uma vez que as variáveis independentes apresentam grandezas incongruentes (GARRETA; MONCECCHI, 2013).

$$z = \frac{X - u}{s} \tag{3.2}$$

onde X representa a amostra de treinamento, u é a média das amostras, e s é o desvio padrão. Essa padronização dos valores não altera a distribuição inicial de X.

No experimento foram empregadas duas técnicas de validação, sendo elas a validação cruzada e a reamostragem dos dados. Na validação cruzada foi empregada a técnica *k-fold* com $k = 10$ conforme ilustrado na Figura 10.

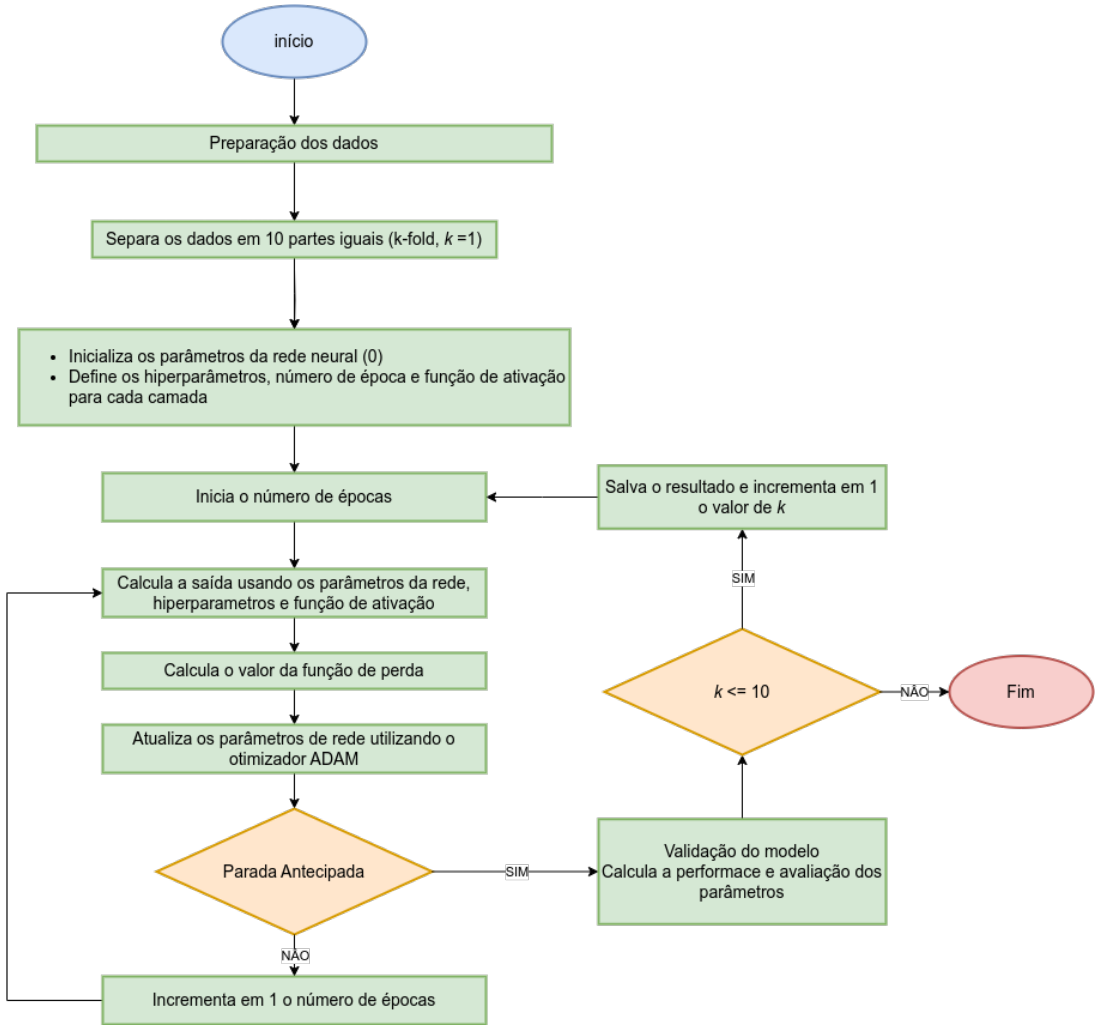


Figura 10: Diagrama de fluxo do modelo de rede neural utilizado para os modelos MLP e PINN com validação cruzada

Na aleatoriedade da divisão dos *folds*, utilizou-se uma semente de aleatoriedade, a fim de garantir que a divisão seja reproduzível, ou seja, resulte sempre na mesma distribuição dos dados quando a mesma semente é utilizada. É importante ressaltar que, nesse contexto, a aleatoriedade é baseada em números pseudo aleatórios (MATSUMOTO; NISHIMURA, 1998).

Na segunda validação empregou-se a re-amostragem em bases para treinamento e validação variando a proporção da distribuição dos dados conforme explicitado na Tabela 3.

Reamostragem	Treinamento (%)	Validação (%)
A	80	20
B	60	40
C	40	60

Tabela 3: Re-amostragem dos dados em bases de treinamento e validação com três diferentes configurações

Para a avaliação dos erros, adotou-se a raiz do erro quadrático médio (RMSE, do inglês *Root Mean Squared Error*) (Equação 3.3), o valor do viés (Equação 3.4) e o coeficiente de linearidade de Pearson (r).

$$\text{RMSE} = -\sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{3.3}$$

$$\text{Viés} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i) \tag{3.4}$$

onde y_i representa o valor real do alvo e $\hat{y}_i$ o valor previsto pelo modelo para a amostra i e n o número total de amostras.

RESULTADOS E DISCUSSÃO

As séries históricas da temperatura do ar utilizadas neste trabalho, divergem em variações sazonais, possuindo diferentes amplitudes e comportamentos (Figura 11).

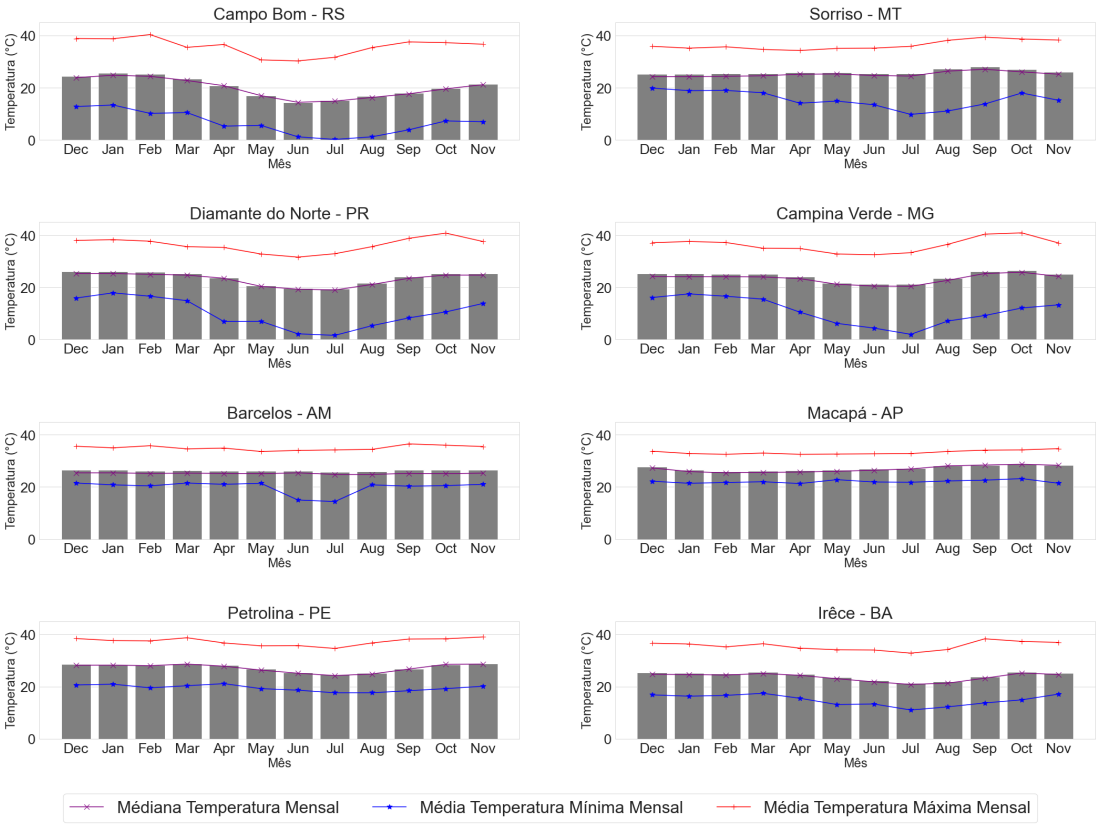


Figura 11: Médias mensais da temperatura do ar nas localidades de estudo considerando toda a base de dados. Calculadas entre o período de novembro/2013 a janeiro/2022.

4.1 VALIDAÇÃO CRUZADA

A Figura 12 apresenta os valores de RMSE, organizados por localização e modelo, obtidos durante a fase de treinamento por meio da técnica de validação cruzada. Para todas as localizações, os modelos baseados em redes neurais (PINN

e MLP) demonstraram a maior precisão em relação aos dados observados, com valor mínimo registrado nos dados de Barcelos (μ: 0,086 °C). Para os demais, o modelo LASSO atingiu o melhor desempenho (μ: 0,68 °C) seguido do modelo EN (μ: 0,69 °C).

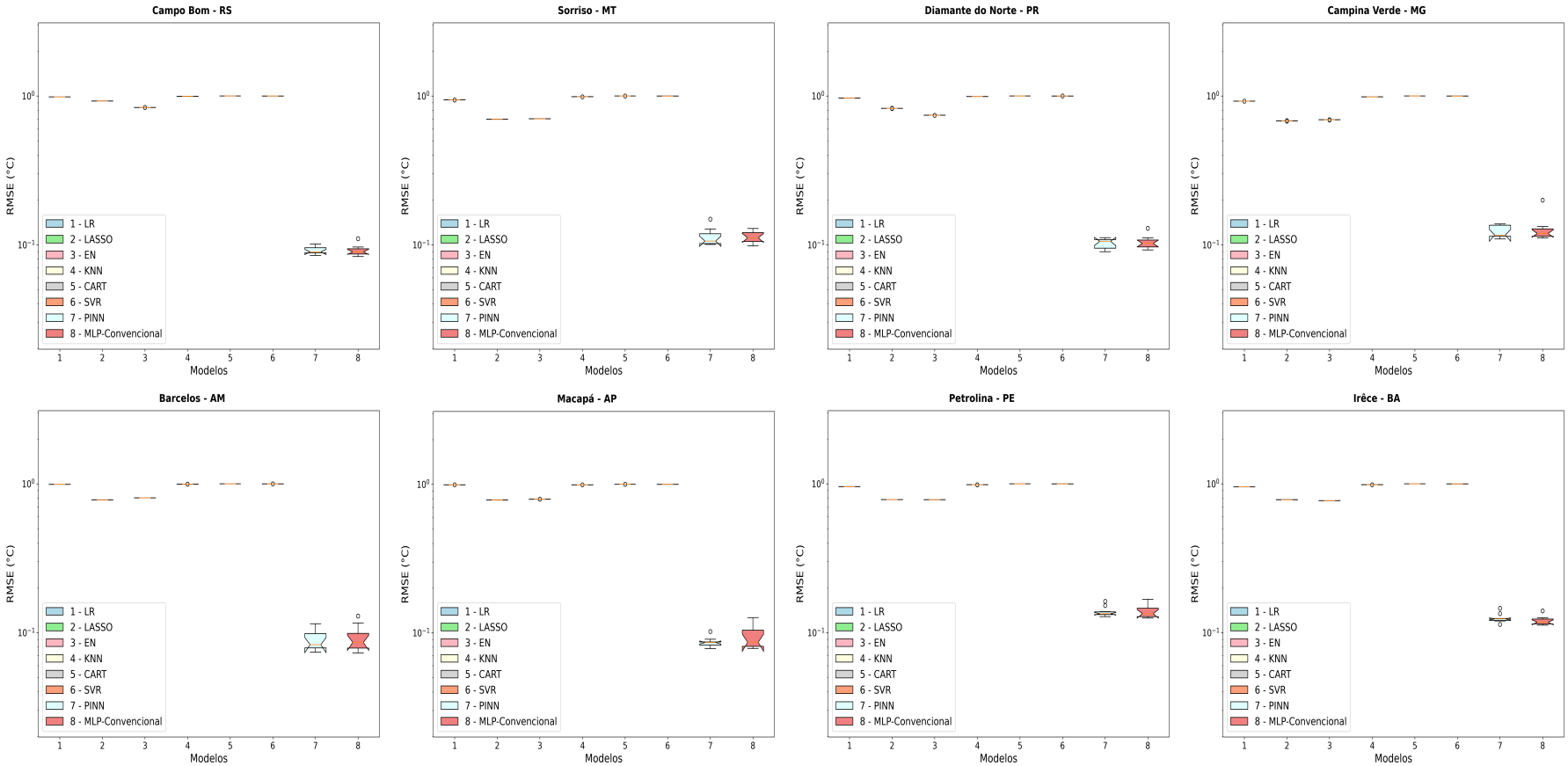


Figura 12: Avaliação de precisão dos modelos para estimativa da temperatura do ar por região na fase de treinamento

Os modelos PINN e MLP resultaram em estimativas muito próximas entre si. Por exemplo, em Campo Bom, Petrolina e Sorriso as diferenças entre os valores de RMSE são ínfimas, sendo que a maior divergência ocorrida foi na série histórica de Macapá -AP, correspondendo ao valor de 0,007 °C (Figura 13).

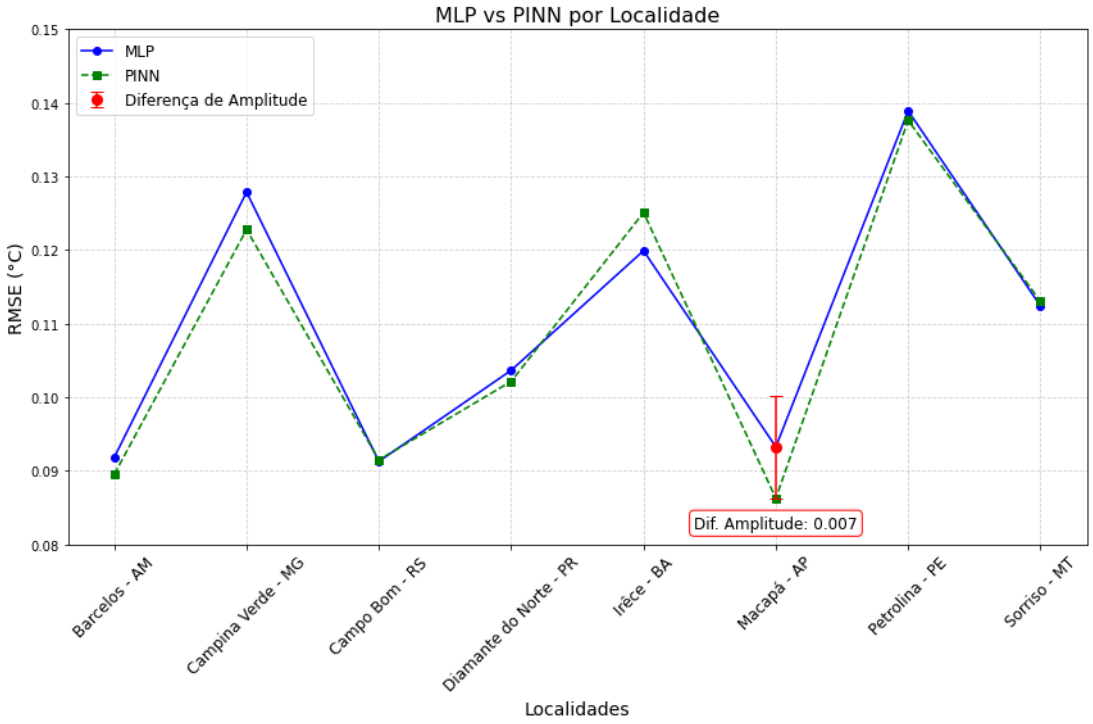


Figura 13: Precisão dos modelos MLP e PINN na etapa de treinamento.
Observação: Médias dos valores de RMSE obtidos na validação cruzada. A barra em vermelho indica a maior amplitude entre os modelos.

Na análise do tempo de execução na fase de treinamento, os modelos baseados em redes neurais apresentou um alto custo computacional ($10^3 ms$), para todas as localidades, quando comparados com os demais (de $10^{-1} a 10^2 ms$).

Com a configuração dos hiperparâmetros, incluindo uma parada antecipada definida em 10^{-4}, em conjunto com um limiar de "paciência" de 30 épocas durante o treinamento, foram observados os resultados de quantidade de épocas necessárias para o treinamento. Ao calcular a média das dez iterações de validação cruzada (Figura 14), constatou-se que o modelo PINN, quando comparado ao modelo MLP, exigiu menos épocas de treinamento em três das oito regiões avaliadas, com uma diferença máxima de apenas 25 épocas ocorrida no treinamento dos dados de Campo Bom (RS).

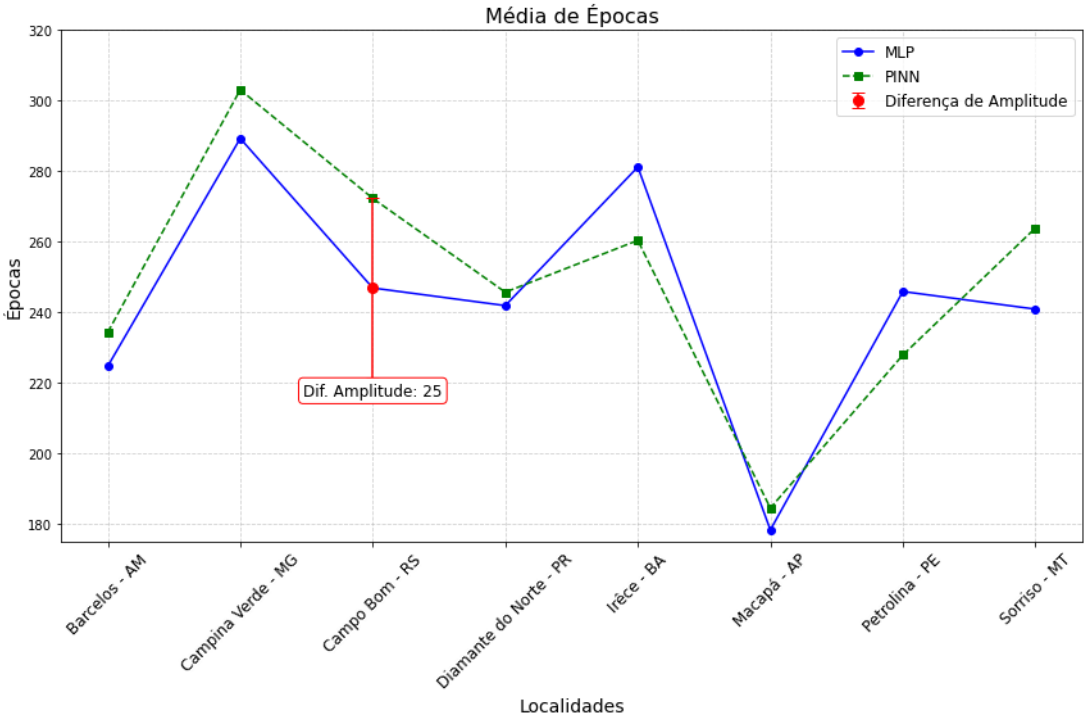


Figura 14: Quantidade de épocas necessárias no treinamento dos modelos MLP e PINN.
Observação: Médias das quantidades de épocas obtidas na validação cruzada. A barra em vermelho indica a maior amplitude entre os modelos.

Na etapa de validação (Figura 15), assim como na fase de treinamento, os modelos baseados em redes neurais se destacaram em relação aos demais. Entretanto, é notável uma exceção nos dados de Macapá, onde o modelo SVR apresentou um desempenho ligeiramente superior em comparação com todos os outros modelos. Essa distinção fica evidente ao analisar as médias obtidas durante a validação cruzada (Figura 16), com um valor de RMSE de 0,092 °C para o modelo SVR, enquanto o modelo MLP registrou 0,094 °C e o modelo PINN obteve 0,098 °C.

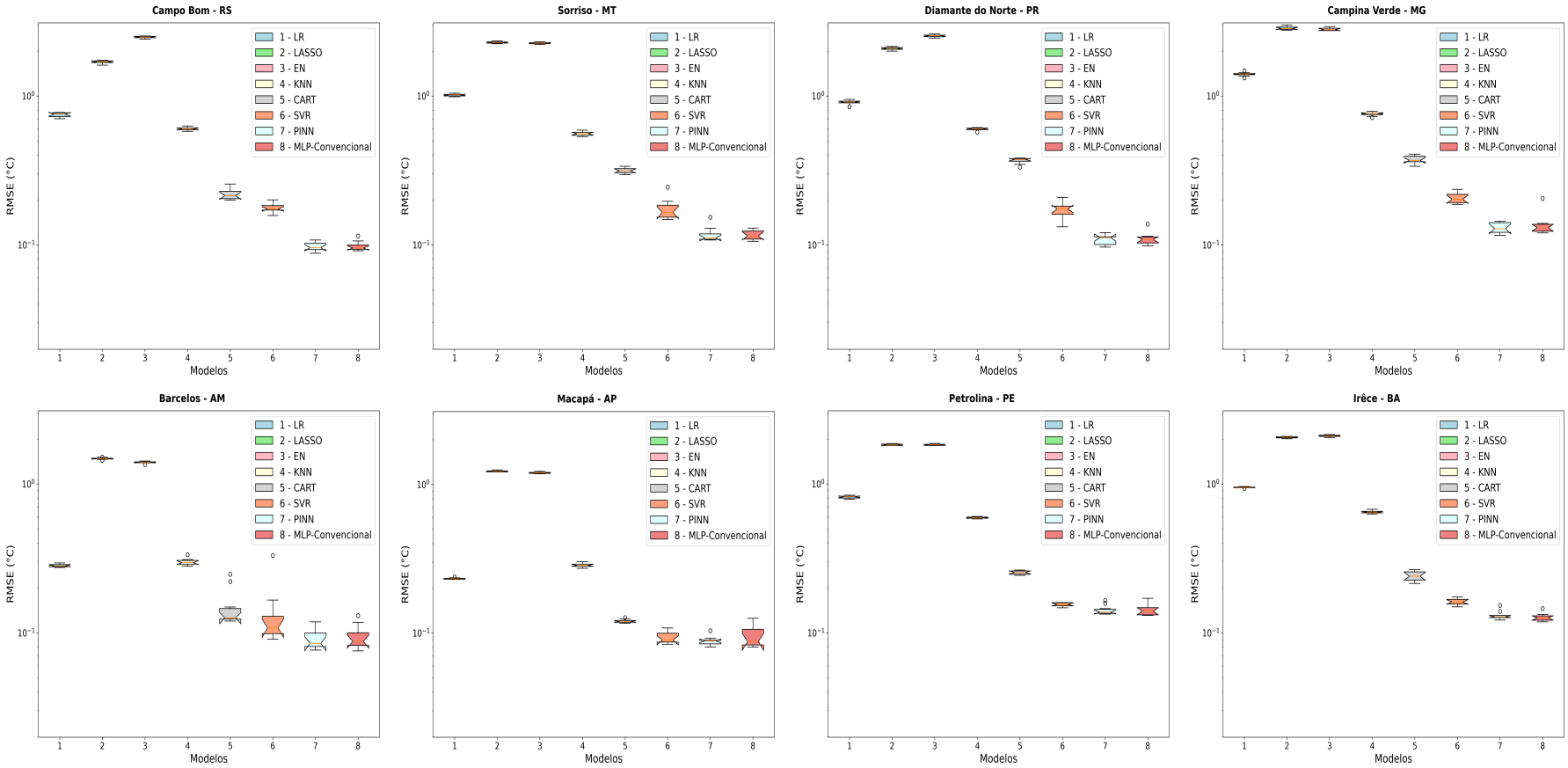


Figura 15: Avaliação de precisão dos modelos para estimativa da temperatura do ar por região na fase de validação

É importante ressaltar que, assim como na fase de treinamento, as discrepâncias entre os modelos MLP e PINN são bastante sutis, com a maior diferença ocorrendo nos dados de Campina Verde (MG), onde a amplitude atinge apenas 0,006 °C.

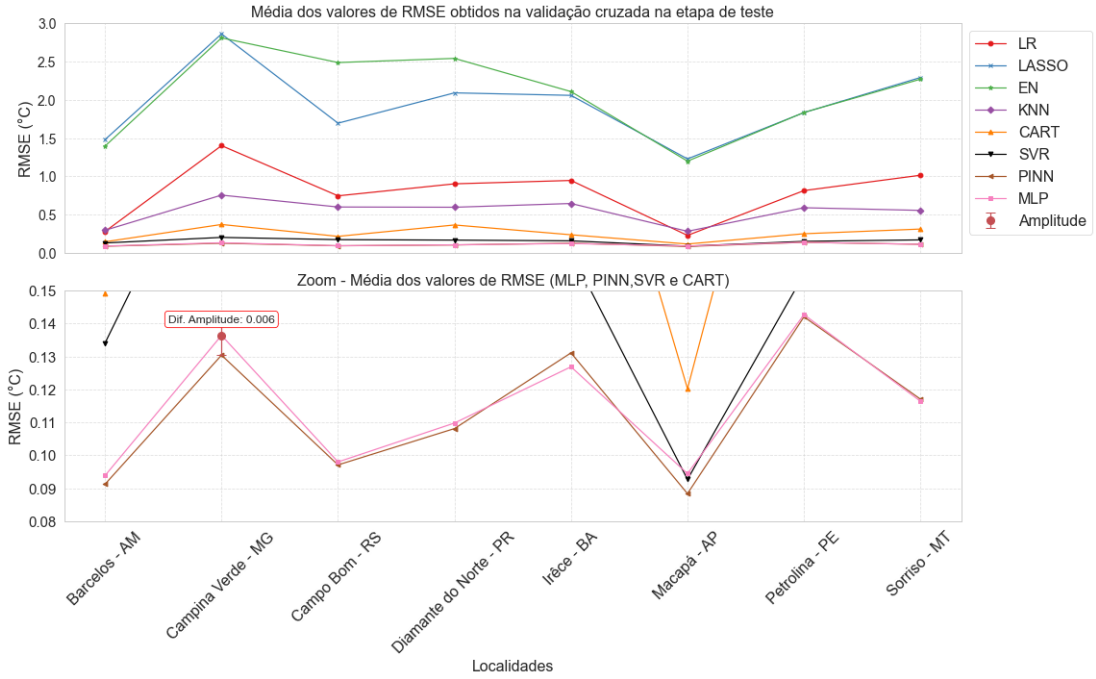


Figura 16: Média dos valores de RMSE para a fase de validação

Observação: Médias dos valores de RMSE obtidos na validação cruzada na etapa de validação. A barra em vermelho indica a maior amplitude entre os modelos MLP e PINN.

4.2 REAMOSTRAGEM DE BASE

Na abordagem de divisão das bases históricas em dados para treinamento e validação, com variação na proporção desses conjuntos (conforme ilustrado na Figura 17), observou-se que os modelos LR, KNN, CART e SVR mantiveram uma linearidade notável em seus resultados, pouco sendo afetados pela variação das proporções.

Por outro lado, os modelos LASSO e EN exibiram um comportamento semelhante, evidenciando picos e declínios em suas previsões, especialmente quando consideradas diferentes localizações.

Ao analisar as médias agrupadas de RMSE por modelo e tamanho da base de validação, notou-se que, para as redes neurais com reamostragem de 20%, o modelo PINN apresentou um valor de RMSE médio (μ: 0,120 °C) ligeiramente inferior em comparação com o modelo MLP (μ: 0,132 °C). Essa tendência também se repetiu na reamostragem de 40%, onde a média do RMSE foi de 0,122 °C

para o modelo PINN e 0,130 °C para o modelo MLP. No entanto, em caso de reamostragem de 60%, ocorreu o oposto, com o modelo PINN obtendo uma média de RMSE de 0,132 °C e o modelo MLP registrando uma média de 0,122 °C.

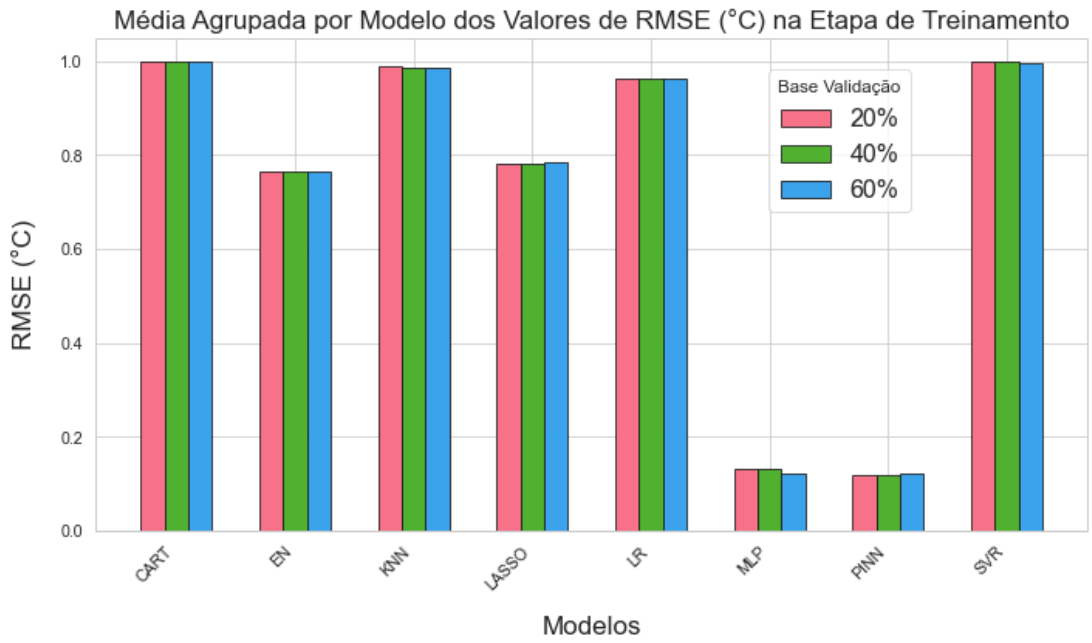


Figura 17: Médias dos valores de RMSE (°C) referentes a etapa de treinamento agrupadas por reamostragem e modelo

Em termos de tempo de treinamento (Figura 18), os modelos LR, LASSO, EN e CART demonstraram um notável desempenho, exigindo menos de 0,2 segundos (s) para serem treinados. O modelo KNN apresentou valores inferiores a 7s, o modelo SVR obteve valores inferiores a 130 segundos e para os modelos PINN e MLP média de 316 ms.

É notável que, em todos os modelos, tenha sido observada uma relação direta entre a quantidade de dados e o tempo de execução. No entanto, exceções notáveis foram encontradas nos modelos EN e LASSO. Embora ambos tenham consistentemente mostrado tempos de execução mais longos para a reamostragem de 20%, os resultados revelaram uma inversão dessa tendência quando comparados aos casos de 40% e 60%. Isso sugere que esses modelos podem ser mais sensíveis à quantidade de dados em cenários específicos, o que pode exigir ajustes cuidadosos dos hiperparâmetros para otimizar o tempo de treinamento.

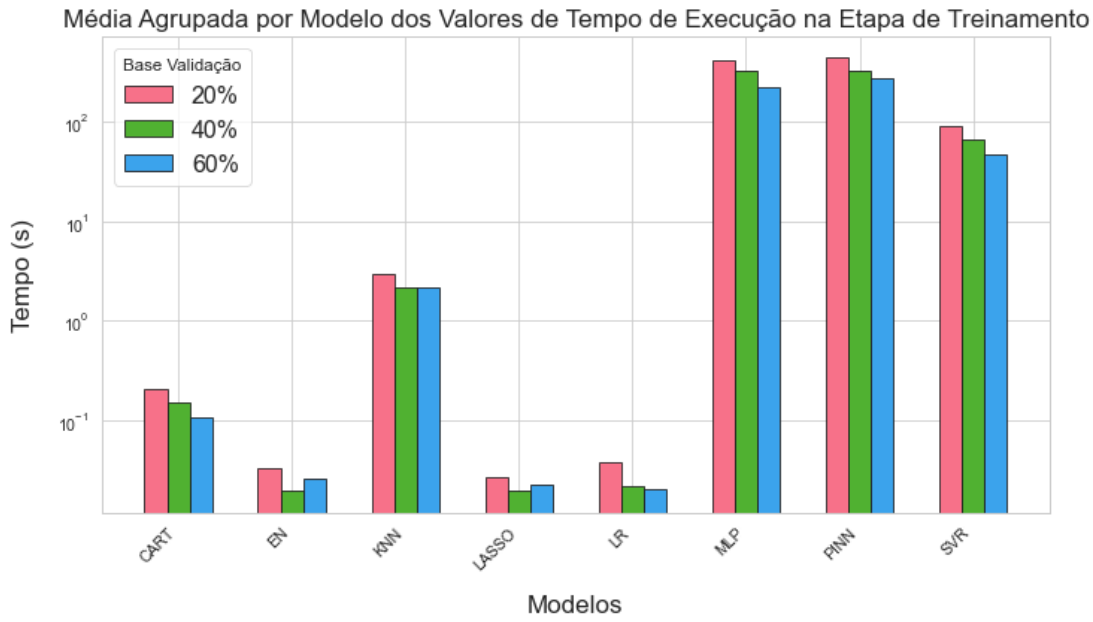
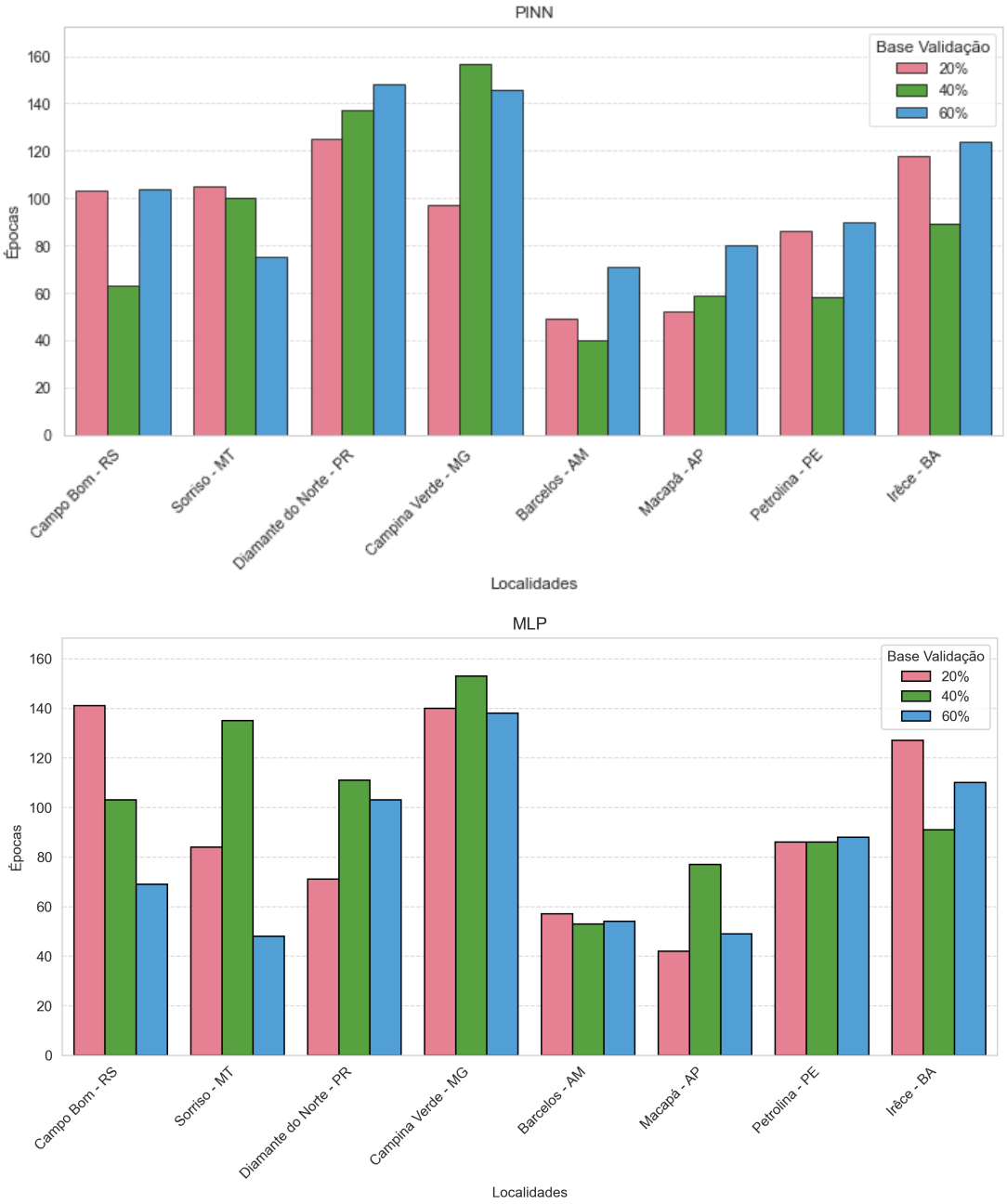


Figura 18: Médias de tempo de treinamento agrupadas por reamostragem e modelo

Não foi possível detectar uma relação direta da quantidade de épocas com o tamanho da base de treinamento (Figura 19), exceto que em grande parte das localidade (5/8) a reamostragem de 40% exigiu maior quantidade de épocas em comparação com a de 60%.

Figura 19: Quantidades de Épocas Necessárias na Etapa de Treinamento



A Figura 20 apresenta gráficos de dispersão para cada modelo aplicado em diversas localizações. Essas análises revelam informações importantes sobre o desempenho dos modelos e sua capacidade de lidar com a complexidade dos dados.

Inicialmente, observa-se que os modelos LASSO e EN demonstraram desempenhos menos favoráveis, com valores de RMSE variando entre 2,48°C e 3,07°C. Além disso, foi notado que esses modelos tendem a subestimar os da-

dos observados. Essas evidências sugerem que a configuração do experimento pode não ter a sensibilidade necessária para abranger todas as complexidades dos dados climáticos.

Por outro lado, o modelo de Regressão Linear (LR) demonstrou um desempenho notável, alcançando valores de RMSE próximos a 0,15°C em três configurações de reamostragem. Isso evidencia a capacidade do modelo de compreender o comportamento da temperatura, mesmo quando a quantidade de dados de treinamento é reduzida.

No caso do modelo KNN, observou-se uma forte e linear relação entre os dados medidos e observados, apesar da presença de alguma dispersão nos pontos. Os valores de RMSE permaneceram baixos (0,47°C), com um viés praticamente nulo (abaixo de 0,001) e um coeficiente linear superior a 0,98.

Um desempenho notável foi alcançado pelo modelo CART, com coeficientes lineares superiores a 0,99, um viés próximo a 10^{-3} e valores de RMSE abaixo de 0,22°C. Esses resultados sugerem uma excelente capacidade do modelo CART em se ajustar aos dados observados.

Por fim, os modelos SVR, PINN e MLP se destacaram, superando os demais. Apresentaram coeficientes lineares superiores a 0,999 e valores de RMSE variando de 0,10°C a 0,12°C. Um aspecto notável dos modelos baseados em redes neurais, MLP e PINN, é a perfeita linearidade entre os pontos, o que sugere um ajuste excepcionalmente próximo aos dados observados.

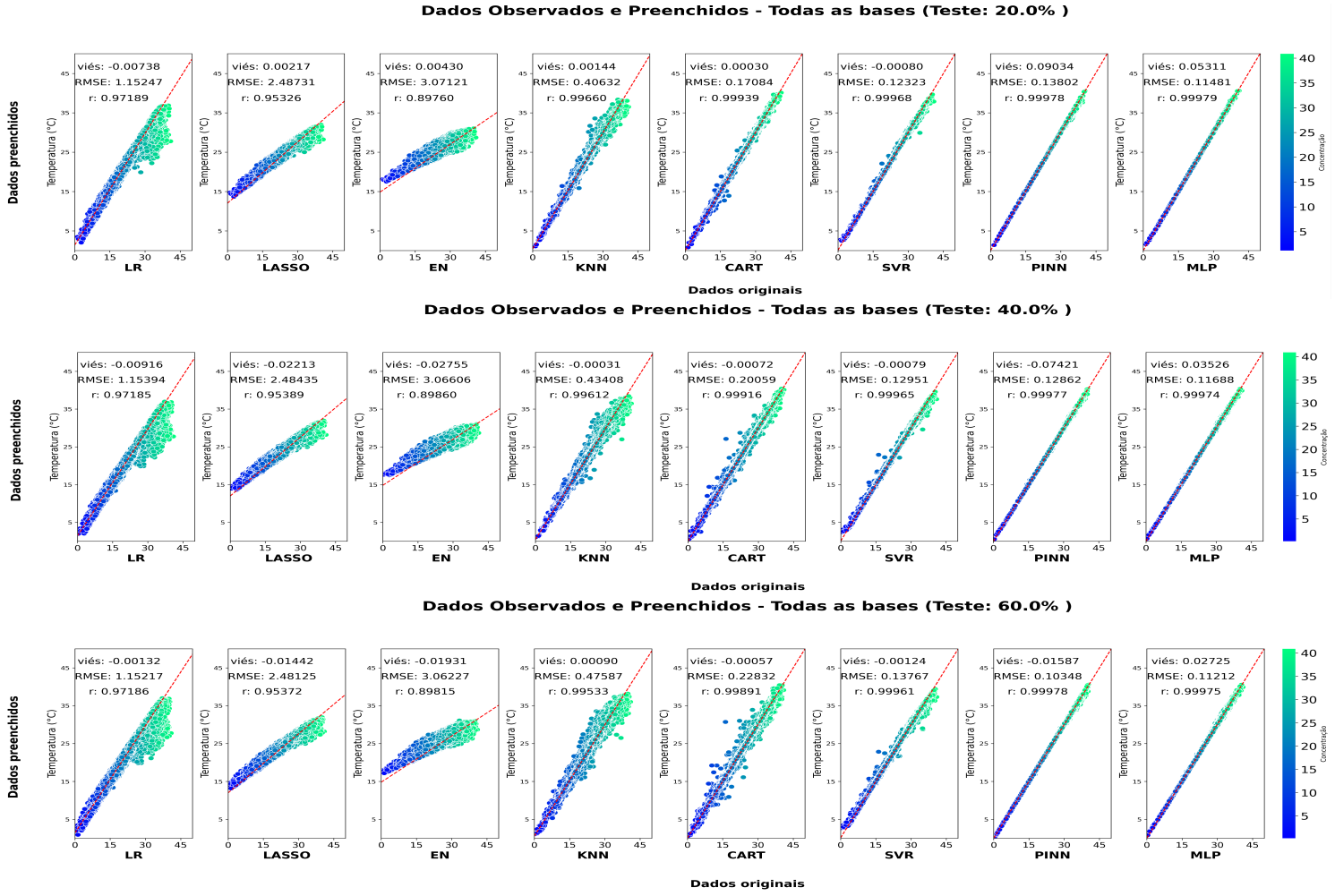


Figura 20: Comparação dos dados estimados com os observados para a base unificada com todas as localizações e configurações de reamostragem.

Os modelos LASSO e EN apresentaram os maiores valores de RMSE, com uma média de aproximadamente 2,37 °C, durante ambas as fases e em todas as regiões analisadas. Esses resultados estão em consonância com uma investigação anterior realizada por Hammami et al. (2012).

Em seu estudo, Coutinho et al. (2018) realizou o preenchimento de séries temporais de temperatura do ar e umidade relativa em quatro regiões do Rio de Janeiro. Ele comparou MLP e RL, e os resultados mostraram que o MLP se destacou, com coeficientes lineares superiores a 0,94 e RMSE inferior a 0,49 °C em três das regiões de estudo. Esses resultados são consistentes com os obtidos no presente estudo em relação aos modelos mencionados. A melhoria do desempenho pode ser atribuída à configuração diferente da arquitetura da rede e seus hiperparâmetros adotados neste estudo.

A pequena diferença entre os resultados do MLP e do PINN sugere que a função de custo utilizada não supera a função MSE, embora se iguale ao MLP, que atingiu um desempenho altamente otimizado, com um RMSE em torno de 0,113 e R = 0,999. É possível comparar os modelos com base em um estudo recente de Ren et al. (2023), que tratou da previsão das composições de gás de síntese a partir da gaseificação de biomassa. Nesse estudo, foi proposto um otimizador de descida do gradiente informado pela física, que resultou em um desempenho superior ao MLP, embora em um contexto diferente.

CONCLUSÕES

Ao avaliar a eficácia do modelo de PINN desenvolvido neste estudo para prever temperaturas, destacam-se algumas observações. Em primeiro lugar, os modelos baseados em redes neurais se destacaram em comparação com outros métodos, com exceção do Support Vector Regression (SVR), que se manteve próximo em desempenho e até superou os modelos neurais em uma das oito localidades.

Além disso, observou-se uma notável semelhança entre os modelos baseados em redes neurais, tanto em termos da quantidade de épocas necessárias para o treinamento quanto no tempo necessário para atingir a convergência. Isso sugere que a função de custo introduzida pela física, embora não tenha superado a função de erro quadrático médio, também não resultou em um desempenho inferior.

No que diz respeito ao estudo da influência da proporção dos dados no treinamento, notou-se que os modelos demonstraram resiliência, com todas as reamostragens atendendo à quantidade mínima necessária para um aprendizado eficaz dos dados utilizados. Isso realça a capacidade dos modelos de se adaptarem a diferentes proporções de dados de treinamento, contribuindo para sua robustez e versatilidade.

REFERÊNCIAS

AGARAP, A. F. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. Citado na página 19.

AWAD, M.; KHANNA, R.; AWAD, M.; KHANNA, R. Support vector regression. *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*, Springer, p. 67–80, 2015. Citado na página 17.

BANERJEE, C.; MUKHERJEE, T.; JR, E. P. An empirical study on generalizations of the relu activation function. In: *Proceedings of the 2019 ACM Southeast Conference*. [S.l.: s.n.], 2019. p. 164–167. Citado na página 19.

BARROW D. K., C. S. F. A comparison of adaboost algorithms for time series forecast combination. *International Journal of Forecasting*, v. 32, p. 1103–1119, 2016. Citado na página 20.

BENGIO, Y.; GRANDVALET, Y. No unbiased estimator of the variance of k-fold cross-validation. *Journal of machine learning research*, v. 5, n. Sep, p. 1089–1105, 2004. Citado na página 9.

BENGIO, Y.; LECUN, Y. et al. Scaling learning algorithms towards ai. *Large-scale kernel machines*, v. 34, n. 5, p. 1–41, 2007. Citado na página 23.

BONACCORSO, G. *Machine Learning Algorithms*. Birmingham – Mumbai: Packt Publishing Ltd, 2017. 331 p. Citado na página 8.

BONFANTE, A. G.; VENTURA, T. M.; OLIVEIRA, A. G. de; MARQUES, H. O.; OLIVEIRA, R. S.; MARTINS, C. A.; FIGUEIREDO, J. M. de. Uma abordagem computacional para preenchimento de falhas em dados micro meteorológicos. *Brazilian Journal of Environmental Sciences (RBCIAMB)*, n. 27, p. 61–70, 2013. Citado 2 vezes nas páginas 1 e 7.

BROWNLEE, J. *Machine Learning Mastery with Python*. Melbourne, Australia: Machine Learning Mastery Pty Ltd, 2016. 256 p. Citado na página 8.

BRUBACHER, J. P.; OLIVEIRA, G. G. d.; GUASSELLI, L. A. Preenchimento de falhas e espacialização de dados pluviométricos: desafios e perspectivas. *Revista Brasileira de Meteorologia*, SciELO Brasil, v. 35, p. 615–629, 2020. Citado na página 6.

CAVALCANTE, R. C.; MINKU, L. L.; OLIVEIRA, A. L. Fedd: Feature extraction for explicit concept drift detection in time series. p. 740–747, 2016. Citado na página 5.

CHATFIELD, C. *The analysis of time series: an introduction*. New York: CRC press, 1996. 352 p. Citado na página 5.

CHATTERJEE, S.; BANERJEE, A.; CHATTERJEE, S.; GANGULY, A. R. Sparse group lasso for regression on land climate variables. In: IEEE. *2011 IEEE 11th International Conference on Data Mining Workshops*. [S.l.], 2011. p. 1–8. Citado na página 15.

CHEN, L.; LI, Y.; MA, S.; JING, Y.; ZHOU, H.; GU, X.; ZHENG, Q.; WANG, F.; ZHAO, Y. Fast fault localization based on deep learning in optical networks. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. [S.l.: s.n.], 2023. v. 12617, p. 126174Z. Citado na página 24.

CHIBANA, E. Y.; FLUMIGNAN, D.; MOTA, R. G. Estimativa de falhas em dados meteorológicos. *Congresso Brasileiro de Agroinformática*, v. 9, 2005. Citado na página 6.

CONNELLY, L. Logistic regression. *Medsurg Nursing*, Anthony J. Jannetti, Inc., v. 29, n. 5, p. 353–354, 2020. Citado na página 13.

COULIBALY, P.; EVORA, N. Comparison of neural network methods for infilling missing daily weather records. *Journal of hydrology*, Elsevier, v. 341, n. 1-2, p. 27–41, 2007. Citado na página 1.

COUTINHO, E. R.; SILVA, R. M. d.; MADEIRA, J. G. F.; COUTINHO, P. R. d. O. d. S.; BOLOY, R. A. M.; DELGADO, A. R. S. Application of artificial neural networks (anns) in the gap filling of meteorological time series. *Revista Brasileira de Meteorologia*, SciELO Brasil, v. 33, p. 317–328, 2018. Citado na página 47.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, Springer, v. 2, n. 4, p. 303–314, 1989. Citado na página 20.

DIETTERICH, T. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, ACM, v. 27, n. 3, p. 326–327, 1995. Citado na página 7.

ENGELBRECHT, A. P. *Computational intelligence: an introduction*. [S.l.]: John Wiley & Sons, 2007. Citado na página 20.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. d. L. F. d. *Inteligência Artificial: Uma abordagem de aprendizado de máquina*. Rio de Janeiro: [s.n.], 2011. 394 p. Citado na página 22.

FIX, E.; HODGES, J. L. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, JSTOR, v. 57, n. 3, p. 238–247, 1989. Citado na página 16.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, NIH Public Access, v. 33, n. 1, p. 1, 2010. Citado na página 14.

GAO, Y.; QIAN, L.; YAO, T.; MO, Z.; ZHANG, J.; ZHANG, R.; LIU, E.; LI, Y. et al. An improved physics-informed neural network algorithm for predicting the phreatic line of seepage. *Advances in Civil Engineering*, Hindawi, v. 2023, 2023. Citado na página 25.

GARRETA, R.; MONCECCHI, G. *Learning scikit-learn: machine learning in python*. Birmingham – Mumbai: Packt Publishing Ltd, 2013. 10–12 p. Citado na página 32.

GRABOCKA, J.; SCHILLING, N.; WISTUBA, M.; SCHMIDT-THIEME, L. Learning time-series shapelets. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2014. p. 392–401. Citado na página 23.

GULLI, A.; PAL, S. *Deep Learning with Keras*. Birmingham – Mumbai: Packt Publishing Ltd, 2017. Citado 2 vezes nas páginas 19 e 20.

HAMMAMI, D.; LEE, T. S.; OUARDA, T. B.; LEE, J. Predictor selection for downscaling gcm data with lasso. *Journal of Geophysical Research: Atmospheres*, Wiley Online Library, v. 117, n. D17, 2012. Citado na página 47.

HAWKINS, D. M. The problem of overfitting. *Journal of chemical information and computer sciences*, ACS Publications, v. 44, n. 1, p. 1–12, 2004. Citado na página 9.

HAYKIN, S. *Neural networks and learning machines, 3/E*. [S.l.]: Pearson Education India, 2009. Citado 3 vezes nas páginas 18, 20 e 21.

HOAGLIN, D. C.; MOSTELLER, F.; TUKEY, J. W. *Exploring data tables, trends, and shapes*. Hoboken, New Jersey: John Wiley & Sons, 2011. v. 101. Citado na página 5.

INMET, N. T. N. S. Rede de estação meteorológicas automáticas do inmet. *Nota Técnica do INMET, Rio de Janeiro*, 2011. Accesso: 2019-01-08. Disponível em: <http://www.inmet.gov.br/portal/css/content/topo_iframe/pdf/Nota_Tecnica-Rede_estacoes_INMET.pdf>. Citado na página 28.

JÚNIOR, A. A. da S.; GOMES, R. d. S. R.; VENTURA, T. M.; RODRIGUES, T. R.; NOGUEIRA, J. de S.; OLIVEIRA, A. G. de; FIGUEIREDO, J. M. de. Visão geral sobre o tratamento de dados meteorológicos no brasil. *Natural Resources*, v. 9, n. 2, p. 59–66, 2019. Citado na página 6.

KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement learning: A survey. *Journal of artificial intelligence research*, v. 4, p. 237–285, 1996. Citado na página 12.

KAJEWSKA-SZKUDLAREK, J.; STAŃCZYK, J. Filling missing meteorological data with computational intelligence methods. In: EDP SCIENCES. *ITM Web of Conferences*. [S.l.], 2018. v. 23, p. 00015. Citado na página 2.

KANDEL, E. R.; SCHWARTZ, J. H.; JESSELL, T. M.; SIEGELBAUM, S.; HUDSPETH, A. J.; MACK, S. et al. *Principles of neural science*. [S.l.]: McGraw-hill New York, 2000. v. 4. Citado na página 23.

KARPATNE, A.; WATKINS, W.; READ, J.; KUMAR, V. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, v. 2, 2017. Citado na página 3.

KATİPOĞLU, O. M.; REŞAT, A. Estimation of missing temperature data by artificial neural network (ann). *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, Dicle University, v. 12, n. 2, p. 431–438, 2021. Citado na página 1.

KEOGH, E.; CHU, S.; HART, D.; PAZZANI, M. Segmenting time series: A survey and novel approach. World Scientific, p. 1–21, 2004. Citado na página 5.

KHAN, M.; JAN, B.; FARMAN, H.; AHMAD, J.; FARMAN, H.; JAN, Z. Deep learning methods and applications. *Deep learning: convergence to big data analytics*, Springer, p. 31–42, 2019. Citado na página 10.

KIDGER, P.; MORRILL, J.; FOSTER, J.; LYONS, T. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, v. 33, p. 6696–6707, 2020. Citado na página 24.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado na página 23.

KLEINBAUM, D.; KLEIN, M. Logistic regression: a self-learning text springer-verlag. *New York, NY*, 1994. Citado na página 24.

KÖPPEN, W.; GEIGER, R. Klimate der erde. gotha: verlag justus perthes. *Wall-map 150cmx200cm*, p. 91–102, 1928. Citado na página 28.

LATIF, S. D.; HAZRIN, N. A. B.; KOO, C. H.; NG, J. L.; CHAPLOT, B.; HUANG, Y. F.; EL-SHAFIE, A.; AHMED, A. N. Assessing rainfall prediction models: Exploring the advantages of machine learning and remote sensing approaches. *Alexandria Engineering Journal*, Elsevier, v. 82, p. 16–25, 2023. Citado na página 2.

LAUBSCHER, R. Simulation of multi-species flow and heat transfer using physics-informed neural networks. *Physics of Fluids*, AIP Publishing, v. 33, n. 8, 2021. Citado na página 3.

LAWAL, Z. K.; YASSIN, H.; LAI, D. T. C.; IDRIS, A. C. Physics-informed neural network (pinn) evolution and beyond: a systematic literature review and bibliometric analysis. *Big Data and Cognitive Computing*, MDPI, v. 6, n. 4, p. 140, 2022. Citado 2 vezes nas páginas I e 25.

LEVINE, D.; STEPHAN, D.; KREHBIEL, T.; BERENSON, M. Estatística–teoria e aplicações usando o microsoft excel em português. 3ª edição. *Rio de Janeiro: LTC, Edição, 819p*, 2005. Citado na página 13.

LILLICRAP, T. P.; SANTORO, A.; MARRIS, L.; AKERMAN, C. J.; HINTON, G. Backpropagation and the brain. *Nature Reviews Neuroscience*, Nature Publishing Group UK London, v. 21, n. 6, p. 335–346, 2020. Citado na página 22.

LOESCH, C.; SARI, S. T. *Redes neurais artificiais: fundamentos e modelos*. Blumenau - Santa Catarina: EDIFURB, 1996. 166 p. Citado na página 19.

LU L., M. X. M. Z. K. G. E. Deepxde: a deep learning library for solving differential equations. 2019. Citado 2 vezes nas páginas 25 e 26.

MATSUMOTO, M.; NISHIMURA, T. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, ACM, v. 8, n. 1, p. 3–30, 1998. Citado na página 33.

MEGETO, G. A.; OLIVEIRA, S. R. d. M.; PONTE, E. M. d.; MEIRA, C. A. Decision tree for classification of soybean rust occurence in commercial crops based on weather variables. *Engenharia Agrícola*, SciELO Brasil, v. 34, p. 590–599, 2014. Citado na página 17.

MICHALSKI, R. S.; CARBONELL, J. G.; MITCHELL, T. M. *Machine learning: An artificial intelligence approach*. Heidelberg, Berlin: Springer Science & Business Media, 2013. 3–13 p. Citado na página 10.

MISHRA, S.; MOLINARO, R. Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for pdes. *IMA Journal of Numerical Analysis*, Oxford University Press, v. 42, n. 2, p. 981–1022, 2022. Citado na página 25.

MISHRA S., M. R. Estimates on the generalization error of physics informed neural networks (pinns) for approximating pdes. 2020. Citado na página 25.

MITCHELL, T. M. Machine learning (mcgraw-hill international editions computer science series). McGraw-Hill, 1997. Citado na página 11.

MJALLI, F. S.; AL-ASHEH, S.; ALFADALA, H. Use of artificial neural network black-box modeling for the prediction of wastewater treatment plants performance. *Journal of Environmental Management*, Elsevier, v. 83, n. 3, p. 329–338, 2007. Citado na página 7.

MOHAMMADI, K.; SHAMSHIRBAND, S.; MOTAMEDI, S.; PETKOVIĆ, D.; HASHIM, R.; GOCIC, M. Extreme learning machine based prediction of daily dew point temperature. *Computers and Electronics in Agriculture*, Elsevier, v. 117, p. 214–225, 2015. Citado na página 2.

MONTGOMERY, D. C.; PECK, E. A.; VINING, G. G. *Introduction to linear regression analysis*. [S.l.]: John Wiley & Sons, 2021. Citado na página 14.

MORETTIN, P. A.; TOLOI, C. *Análise de séries temporais*. Universidade de São Paulo: Edgard Blucher, 2006. 474 p. Citado na página 5.

MORI, H.; TAKAHASHI, A. A data mining method for selecting input variables for forecasting model of global solar radiation. In: IEEE. *PES T&D 2012*. [S.l.], 2012. p. 1–6. Citado na página 17.

OLIVEIRA, M. A.; FAVERO, L. P. L. Uma breve descrição de algumas técnicas para análise de séries temporais: Séries de fourier, wavelets, arima, modelos estruturais para séries de tempos e redes neurais. *VI SEMEAD Ensaio mqi. USP. Anais, São Paulo*, 2002. Citado na página 5.

PARIS, G.; ROBILLIARD, D.; FONLUPT, C. Exploring overfitting in genetic programming. In: SPRINGER. *International Conference on Artificial Evolution (Evolution Artificielle)*. [S.l.], 2003. p. 267–277. Citado na página 10.

PATRICK, E. A.; III, F. P. F. A generalized k-nearest neighbor rule. *Information and control*, Elsevier, v. 16, n. 2, p. 128–152, 1970. Citado na página 16.

PAULO, I. de; FERREIRA, H.; PAULO, S. de; NOGUEIRA, J. de S.; AGUIAR, R.; SÁ, M. Termodinâmica noturna dos ecossistemas amazônicos. *Revista Brasileira de Climatologia*, v. 32, p. 269–291, 2023. Citado na página 31.

PENG, J.; JURY, E. C.; DÖNNES, P.; CIURTIN, C. Machine learning techniques for personalised medicine approaches in immune-mediated chronic inflammatory diseases: applications and challenges. *Frontiers in pharmacology*, Frontiers Media SA, v. 12, p. 720694, 2021. Citado na página 11.

PRINCIPE, J. C.; EULIANO, N. R.; LEFEBVRE, W. C. *Neural and adaptive systems: fundamentals through simulations*. New York: Wiley New York, 2000. v. 672. 137–139 p. Citado na página 19.

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*, Elsevier, v. 378, p. 686–707, 2019. Citado 3 vezes nas páginas 2, 24 e 25.

RAOUHI, E. M.; LACHGAR, M.; KARTIT, A. Comparative study of regression and regularization methods: Application to weather and climate data. In: SPRINGER. *WITS 2020: Proceedings of the 6th International Conference on Wireless Technologies, Embedded, and Intelligent Systems*. [S.l.], 2022. p. 233–240. Citado na página 15.

REN, S.; WU, S.; WENG, Q. Physics-informed machine learning methods for biomass gasification modeling by considering monotonic relationships. *Bioresource Technology*, Elsevier, v. 369, p. 128472, 2023. Citado na página 47.

ROBERT, C. Machine learning, a probabilistic perspective. Taylor & Francis, 2014. Citado na página 10.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citado na página 18.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group, v. 323, n. 6088, p. 533, 1986. Citado na página 22.

RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. England: Pearson Education Limited, 2016. 1132 p. Citado na página 12.

SABINO, M.; SOUZA, A. P. d. Preenchimento de falhas em dados meteorológicos usando o programa gapmet no estado de mato grosso, brasil. *Revista Brasileira de Engenharia Agrícola e Ambiental*, SciELO Brasil, v. 27, p. 149–156, 2022. Citado na página 6.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, IBM, v. 3, n. 3, p. 210–229, 1959. Citado na página 10.

SIROIS, A. The effects of missing data on the calculation of precipitation-weighted-mean concentrations in wet deposition. *Atmospheric Environment. Part A. General Topics*, Elsevier, v. 24, n. 9, p. 2277–2288, 1990. Citado na página 6.

SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. *Statistics and computing*, Springer, v. 14, p. 199–222, 2004. Citado na página 18.

STEEL, R.; TORRIE, J.; DICKEY, D. Principles and procedures of statistics: A biometric approach, pp: 663–666 mcgraw-hill book co. *New York, NY, USA*, 1997. Citado na página 10.

STOSIC, D.; STOSIC, D.; LUDERMIR, T.; OLIVEIRA, W. de; STOSIC, T. Foreign exchange rate entropy evolution during financial crises. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 449, p. 233–239, 2016. Citado na página 5.

SWAIN, M.; DASH, S. K.; DASH, S.; MOHAPATRA, A. An approach for iris plant classification using neural network. *International Journal on Soft Computing*, Academy & Industry Research Collaboration Center (AIRCC), v. 3, n. 1, p. 79, 2012. Citado na página 12.

TCHAKONTE, S.; NANA, P.-A.; TAMSA, A. A.; TCHATCHO, N. L. N.; KOJI, E.; ONANA, F. M.; AJEAGAH, G. A. Using machine learning models to assess

the population dynamic of the freshwater invasive snail physa acuta draparnaud, 1805 (gastropoda: Physidae) in a tropical urban polluted streams-system. *Limnologica*, Elsevier, v. 99, p. 126049, 2023. Citado na página 2.

TEEGAVARAPU, R. S.; CHANDRAMOULI, V. Improved weighting methods, deterministic and stochastic data-driven models for estimation of missing precipitation records. *Journal of Hydrology*, Elsevier, v. 312, n. 1-4, p. 191–206, 2005. Citado na página 6.

THOBER, S.; KUMAR, R.; WANDERS, N.; MARX, A.; PAN, M.; RAKOVEC, O.; SAMANIEGO, L.; SHEFFIELD, J.; WOOD, E. F.; ZINK, M. Multi-model ensemble projections of european river floods and high flows at 1.5, 2, and 3 degrees global warming. *Environmental Research Letters*, IOP Publishing, v. 13, n. 1, p. 014003, 2018. Citado na página 1.

TOSUNOĞLU, F.; HANAY, S.; ÇINTAŞ, E.; ÖZYER, B. Monthly streamflow forecasting using machine learning. *Erzincan University Journal of Science and Technology*, Erzincan Binali Yildirim University, v. 13, n. 3, p. 1242–1251, 2020. Citado na página 2.

WANG, S.; SANKARAN, S.; PERDIKARIS, P. Respecting causality is all you need for training physics-informed neural networks. *arXiv preprint arXiv:2203.07404*, 2022. Citado na página 26.

WEN, J.; YANG, J.; JIANG, B.; SONG, H.; WANG, H. Big data driven marine environment information forecasting: a time series prediction network. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 29, n. 1, p. 4–18, 2020. Citado na página 1.

WIDROW, B.; HOFF, M. E. et al. Adaptive switching circuits. In: NEW YORK. *IRE WESCON convention record*. [S.l.], 1960. v. 4, n. 1, p. 96–104. Citado na página 18.

WNDERLEY, H. S. Variabilidade espacial e preenchimento de falhas de dados pluviométricos para o estado de alagoas. *Revista Brasileira de Meteorologia*, SciELO Brasil, v. 27, n. 3, 2012. Citado na página 5.

XU, Y.; LIU, X.; CAO, X.; HUANG, C.; LIU, E.; QIAN, S.; LIU, X.; WU, Y.; DONG, F.; QIU, C.-W. et al. Artificial intelligence: A powerful paradigm for scientific research. *The Innovation*, Elsevier, v. 2, n. 4, 2021. Citado na página 1.

YING, X. An overview of overfitting and its solutions. In: IOP PUBLISHING. *Journal of physics: Conference series*. [S.l.], 2019. v. 1168, p. 022022. Citado na página 8.

YULE, G. U. Vii. on a method of investigating periodicities disturbed series, with special reference to wolfer's sunspot numbers. *Phil. Trans. R. Soc. Lond. A*, The Royal Society, v. 226, n. 636-646, p. 267–298, 1927. Citado na página 5.

ZHANG, H.; ZHANG, L.; JIANG, Y. Overfitting and underfitting analysis for deep learning based end-to-end communication systems. In: IEEE. *2019 11th international conference on wireless communications and signal processing (WCSP)*. [S.l.], 2019. p. 1–6. Citado na página 8.

ZHANG, L.; HAN, J.; WANG, H.; CAR, R.; WEINAN, E. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Physical review letters*, APS, v. 120, n. 14, p. 143001, 2018. Citado na página 3.

ANEXO I

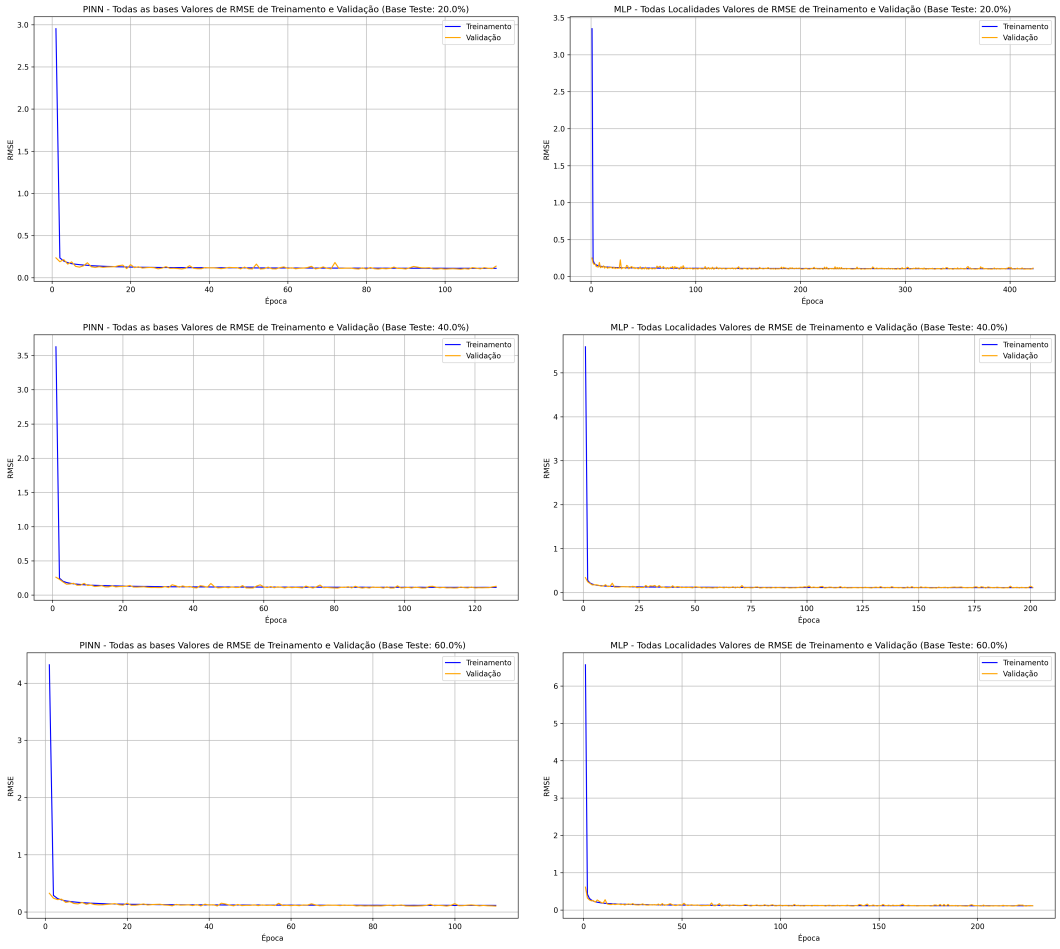


Figura 21: Tempo de Treinamento por Localidade e Reamostragem

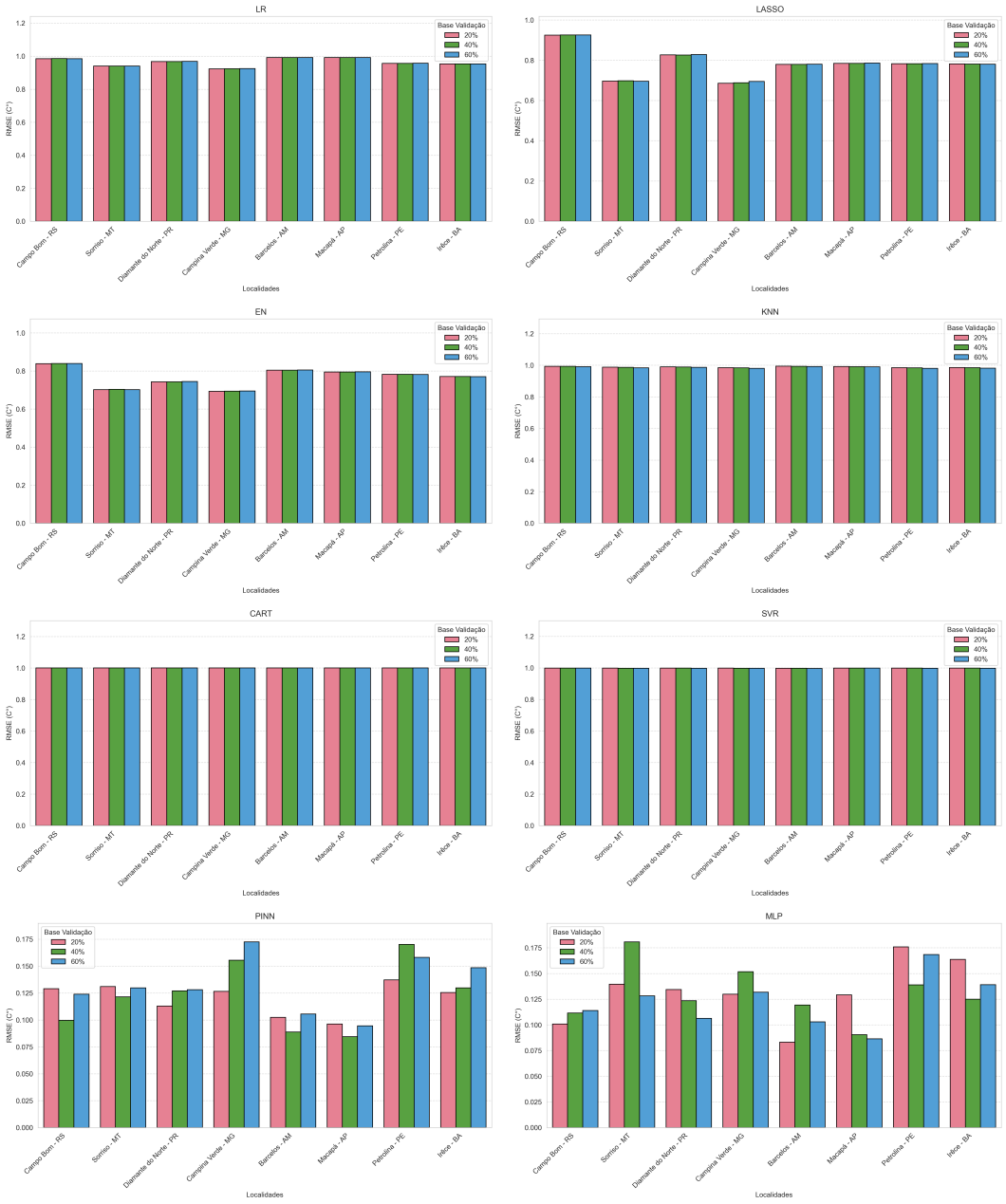


Figura 22: Valores de RMSE(°C) da fase de treinamento por Localidade e Reamostragem

Figura 23: Tempo de Treinamento por Localidade e Reamostragem