

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

Aplicação de Deep Learning no Preenchimento de
Falhas em Dados Micrometeorológicos

Anisio Alfredo da Silva Junior

Orientador: Prof. Dr. Raphael de Souza Rosa Gomes

Cuiabá - MT
Fevereiro/2019

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

Aplicação de Deep Learning no Preenchimento de
Falhas em Dados Micrometeorológicos

Anisio Alfredo da Silva Junior

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

Prof. Dr. Raphael de Souza Rosa Gomes

Cuiabá, MT

Fevereiro/2019

Dados Internacionais de Catalogação na Fonte.

D111a da Silva Junior, Anisio Alfredo.
APLICAÇÃO DE DEEP LEARNING NO PREENCHIMENTO DE FALHAS EM
DADOS MICROMETEOROLÓGICOS / Anisio Alfredo da Silva Junior. -- 2019
67 f. : il. color. ; 30 cm.

Orientador: Raphael de Souza Rosa Gomes.
Dissertação (mestrado) - Universidade Federal de Mato Grosso, Instituto de Física,
Programa de Pós-Graduação em Física Ambiental, Cuiabá, 2019.
Inclui bibliografia.

1. Inteligência artificial. 2. Tratamentos de dados. I. Título.

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

Permitida a reprodução parcial ou total, desde que citada a fonte.

UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE FÍSICA
Programa de Pós-Graduação em Física Ambiental

FOLHA DE APROVAÇÃO

TÍTULO: APLICAÇÃO DE *DEEP LEARNING* NO PREENCHIMENTO DE FALHAS EM DADOS MICROMETEOROLÓGICOS

AUTOR: ANÍSIO ALFREDO DA SILVA JÚNIOR

Dissertação de Mestrado defendida e aprovada em 26 de fevereiro de 2019, pela comissão julgadora:


Prof. Dr. Raphael de Souza Rosa Gomes
Orientador
Instituto de Computação – UFMT


Prof. Dr. Josiel Maimone de Figueiredo
Examinador Interno
Instituto de Computação – UFMT


Prof. Dr. Carlo Ralph De Musis
Examinador Interno
Universidade de Cuiabá/UNIC


Prof. Dr. Thiago Meirelles Ventura
Examinador Externo
Instituto de Computação – UFMT

*“Importante não é ver o que ninguém
nunca viu, mas sim, pensar o que nin-
guém nunca pensou sobre algo que todo
mundo vê.”*

(Schopenhauer)

Agradecimentos

- O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.
- A Deus por tudo.
- A minha mãe, eterna heroína.
- Aos meus irmãos Leonardo e Cláudia, companheiros para toda vida.
- A minha namorada Leilane, sempre presente.
- A minhas amigas Lucielly e Paloma, pelo constante apoio.
- Ao professor Raphael de Souza Rosa Gomes pela paciência e compreensão durante a orientação. Cresci muito com este trabalho, e sou grato pelo seu exemplo de profissional sempre disponível e dedicado.
- Ao professor José de Souza Nogueira, sua esposa, professora Marta Cristina de Jesus Albuquerque Nogueira, e aos demais professores do corpo docente que fazem do PPGFA um ambiente acolhedor e vivo.
- Aos amigos e colegas da Pós-Graduação em Física Ambiental, em especial a: Marlus Sabino, Kelly Sousa Romera, Jonh Billy da Silva, Ana Clara Alves Justi, Bruno Santos Abdalla, Elio Santos Almeida Junior, Lucas Douglas Rothmund, Luís Philippe de Arruda Lima, Rayanna de Oliveira Costa, Luciene de Mello Taques, Luiz Eduardo Barros da Guia, Lougans Peterson Sette, Dalila Morgana de Souza Mützenberg, Danielle Cristine Stenner

Nassarden, Keylyane Santos da Silva, Lucas Peres Angelini, Jamson Justi, José Guilherme Roquette e Antonio Ramos Correia.

- Ao professor Luís Craco, por me aconselhar em fazer o mestrado.
- Ao professor Geraldo, por sempre ajudar.
- A Jô, Soilce, Flair e Cesário, pelo importante trabalho que realizam no programa.
- A todas as pessoas que direta ou indiretamente contribuíram para a realização deste trabalho.

SUMÁRIO

LISTA DE FIGURAS	I
LISTA DE ABREVIATURAS	III
RESUMO	IV
ABSTRACT	V
1 INTRODUÇÃO	1
1.1 Problemática	1
1.2 Justificativa	2
2 FUNDAMENTAÇÃO TEÓRICA	3
2.1 Dados e Séries Temporais Micrometeorológicas	3
2.2 Métodos de Preenchimento de Falhas	5
2.2.1 Modelos Estatísticos	5
2.2.1.1 Regressão Linear	5
2.2.1.2 Regressão Linear Múltipla	6
2.2.1.3 Média Móvel	7
2.2.2 Modelos de Aprendizado de Máquina	7
2.2.2.1 Redes Neurais Artificiais	9
2.2.2.2 Parâmetros Externos da Rede	12
2.2.2.3 Deep Learning	13
2.3 Seleção de Variáveis e Separação em Dados de Treino e Teste	16

3	MATERIAIS E MÉTODOS	19
3.1	Dados Utilizados	19
3.2	Desenvolvimento do Modelo de Deep Learning	21
3.2.1	Preparação dos Dados	22
3.2.2	Seleção das Variáveis	23
3.2.3	Reamostragem	23
3.2.4	Modelo de Deep Learning para Regressão	24
3.2.5	Análise do Modelo e Comparação com Outros Métodos	24
4	Apresentação e Análise dos Resultados	26
4.1	Modelo da Rede Neural DL	26
4.1.1	Resultados do Modelo de DL	28
4.1.1.1	Análise Estatística	32
4.1.1.2	Tempo de Execução	34
4.2	Regressão Linear Múltipla e Média Móvel	35
4.2.1	Regressão Linear Múltipla	35
4.2.2	Média Móvel Simples	36
4.3	Comparativo entre os Métodos	37
4.4	Discussão dos Resultados	39
5	CONSIDERAÇÕES FINAIS	41
5.1	Trabalhos Futuros	41
	REFERÊNCIAS	43

LISTA DE FIGURAS

1	Série temporal horária da temperatura do ar.	4
2	Regressão (A): ajustar uma curva aos dados. Clusterização (B): agrupamento dos dados. Classificação (C): encontrar uma fronteira de decisão.	8
3	Uma rede neural com um único neurônio. Nessa configuração, a saída do neurônio é a saída geral da rede neural.	10
4	Comportamento da função ReLU.	11
5	Rede neural multicamada.	11
6	Timeline da Deep Learning.	13
7	Relação entre IA e deep learning.	14
8	Visualização dos passos para formação da matriz C pelo processo de convolução entre as matrizes A e B	15
9	Particionamento dos dados em dados de treino e teste.	17
10	k -fold, onde $k = 4$	18
11	Localização das quatro estações meteorológicas selecionadas pertencen- tes ao INMET.	20
12	Etapas de desenvolvimento do presente estudo.	21
13	Fluxograma do algoritmo de DL.	27
14	Relação da média do MAE individual para cada porcentagem de falhas com a quantidade de testes efetuados pelo modelo de DL.	29
15	Relação entre a quantidade de falhas e o MAE para cada variável de estudo, indicando o comportamento da média e mediana.	30

16	Séries temporais das variáveis de estudo (temperatura do ar, umidade relativa, ponto de orvalho, pressão, radiação e velocidade do vento) coletadas da estação do INMET, localidade Sorriso- MT, no período de novembro/2014 à janeiro/2015. Ilustradas: na forma original, com 50% de falhas e após o processo de preenchimento de falhas.	31
17	Distribuição dos valores do MAE para os grupos: I, II, III, IV e V (Tabela 8).	34
18	Distribuição do tempo médio de execução.	35
19	Comparativo entre DL, RLM e média móvel para o preenchimento de falhas nos dados com a presença de diferentes quantidades de falhas para as variáveis: temperatura do ar, umidade relativado do ar, ponto de orvalho, pressão atmosférica, radiação solar e velocidade do vento.	38

LISTA DE ABREVIATURAS

CNN	Redes Neurais Convolucionais	14
CUDA	<i>Compute Unified Device Architecture</i>	21
DL	<i>Deep Learning</i>	2
INMET	Instituto Nacional de Meteorologia	19
MAE	<i>Mean Absolute Error</i>	22
MLP	<i>Multilayer Perceptron</i>	11
MMS	Média Móvel Simples	36
ReLU	<i>Rectified Linear Unit</i>	10
RLM	Regressão Linear Múltipla	6
RNA	Redes Neurais	9
RNN	Redes Neurais Recorrentes	15
ST	Série Temporal	1

RESUMO

SILVA JUNIOR, A. A. Aplicação de Deep Learning no Preenchimento de Falhas em Dados Micrometeorológicos. Cuiabá, 2019, 67f. Dissertação (Mestrado em Física Ambiental) - Instituto de Física, Universidade Federal de Mato Grosso.

O objetivo deste trabalho é estudar a utilização de Deep Learning no preenchimento de falhas em séries de dados microclimáticos, para isso, utilizou dados de estações automáticas do INMET, localizadas nas cidades de Campina Verde -MG, Sorriso -MT, Diamante do Norte -PR e Campo Bom -RS, coletados no período de novembro de 2014 à janeiro de 2015, compostos pelas variáveis: radiação solar, velocidade do vento, ponto de orvalho, umidade relativa do ar, pressão atmosférica e temperatura do ar. O experimento consistiu na construção de um modelo de aprendizado de máquina e análise de quatro fatores básicos, como a quantidade de camadas ocultas utilizadas, quantidade de neurônios, quantidade de variáveis independentes utilizadas na estimativa da variável alvo e, por fim, os impactos causados pela porcentagem de falhas contida na série. Para a análise estatística foram utilizados os testes de Kruskal-Wallis e Mann-Whitney na análise de variância, e o teste de Nemenyi na comparação em pares, além de comparações com outros modelo amplamente utilizados na literatura. Os resultados demonstraram que a quantidade de 50 e 100 neurônios causa impacto significativo na resposta da rede, que a utilização de apenas uma variável independente tem diferença significativa e em comparação a o uso de mais, e que a porcentagem de falhas influenciou apenas o preenchimento das séries temporais de ponto de orvalho. Conclui-se que o modelo desenvolvido apresenta resultados significantes para o preenchimento de falhas de temperatura, umidade relativa e ponto de orvalho.

Palavras-chave: inteligência artificial, tratamento de dados.

ABSTRACT

SILVA JUNIOR, A. A. Application of Profound Learning in Filling in Micrometeorological Data. Cuiabá, 2019, 67f. Dissertation (Master's Degree in Environmental Physics) – Institute of Physics, Federal University of Mato Grosso, Cuiabá.

This work aimed to study the use of Deep Learning in gap filling in microclimatic data series. The databases used belong to automatic stations of the Institute of Meteorology (INMET), located in the cities of Campina Verde - MG, Sorriso -MT, Diamante do Norte -PR and Campo Bom -RS, collected from November 2014 to January 2015, composed by the following variables: solar radiation, wind speed, dew point, relative humidity, atmospheric pressure and air temperature. The experiment consisted in the construction of a model of machine learning and analysis of four basic factors, such as the number of hidden layers, number of neurons, number of independent variables used in the estimation of the target variable, and finally the impacts caused by the percentage of failures contained in the series. For the statistical analysis, Kruskal-Wallis and Mann-Whitney tests were used in the analysis of variance, Nemenyi test to paired comparison, and comparisons with other models widely used in the literature. The results showed that the number of 50 and 100 neurons does have a significant impact on the network response, that the use of only one independent variable has a significant difference in comparison to the use of more, and the percentage of failures influenced only the filling of the point time series of dew. It is concluded that the developed model presents good results for filling temperature, relative humidity and dew point faults.

Keywords: *artificial intelligence, data correction*

Capítulo 1

INTRODUÇÃO

1.1 Problemática

Ao redor do mundo, ações antrópicas têm provocado grandes alterações no meio ambiente e têm desencadeado um cenário de extrema preocupação. Estudos na área ambiental, como emissões de poluentes, efeito estufa, desmatamento, entre outros assuntos ganham cada vez mais notoriedade no cenário mundial, por se tratar de temas com relevante importância para sustentabilidade da vida no planeta. Uma crescente quantidade de pesquisadores direcionam suas pesquisas nessa direção, com o intuito de obterem maior compreensão e informações sobre os sistemas microclimáticos.

A veracidade das informações adquiridas através da pesquisa científica, fundamenta-se na qualidade dos dados coletados. Avanços na tecnologia possibilitaram a modelagem cada vez mais precisa de fenômenos reais através da automatização de processos de medições ao longo do tempo, o que gera uma maior quantidade de informação. Em contrapartida, aumenta a possibilidade do aparecimento de falhas, que podem ser oriundas do próprio equipamento utilizado, devido a um mal funcionamento por defeito na fabricação, exposição a intempéries, falha na calibração, falta de manutenção, ausência de energia elétrica, transmissão de dados mal sucedida, entre outros.

A existência de falhas na Série Temporal (ST) pode levar a resultados inexatos, que se distanciam do real comportamento do fenômeno estudado, levando o pesquisador a suposições equivocadas. Uma alternativa frente a esse problema é o uso de um processo de preenchimento que possibilite o aproveitamento da série em sua totalidade temporal, de forma a impedir que parte dos dados seja desconsiderada. A utilização de um método eficiente e fidedigno de preenchimento de falhas, é uma preocupação recorrente em diversas áreas, pois o mesmo deve

estimar os dados faltantes com considerável precisão ao comportamento real.

1.2 Justificativa

Trabalhar com dados micrometeorológico isentos de falhas é de grande importância para o desenvolvimento de estudos climáticos de confiabilidade, pois representam o fenômeno observado com maior integridade. Muitos pesquisadores ao se depararem com a presença de falhas em seus dados buscam por métodos eficazes de preenchimento e acabam se deparando com grande variedade de opções.

Obter informação sobre a comparação entre determinados métodos de preenchimento de falhas, não é uma tarefa fácil, principalmente quando específica para a área de Ciências Ambientais, que carece de pesquisas que utilizem a tecnologia vigente expondo as vantagens e desvantagens de seu emprego.

Este estudo tem como objetivo geral a utilização de *Deep Learning* (DL), uma área dentro do aprendizado de máquina, como alternativa atual para o preenchimento de falhas em séries temporais. DL vem ganhando destaque nos últimos anos, considerada o estado da arte, essa técnica promissora aprende a partir dos dados fornecidos, possibilitando um resultado mais próximo da realidade.

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

- Aquisição e preparação dos dados micrometeorológicos;
- Estudo sobre a quantidade de variáveis independentes a ser utilizada no preenchimento.
- Criação de um modelo de DL regressivo, para estimar valores no preenchimento de falhas;
- Treinamento do modelo criado;
- Avaliar a acurácia dos resultados.
- Avaliar o modelo de DL com outros modelos presentes na literatura.

Capítulo 2

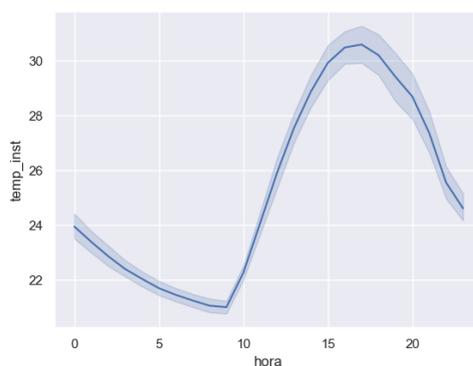
FUNDAMENTAÇÃO TEÓRICA

2.1 Dados e Séries Temporais Micrometeorológicas

Para o estudo de fenômenos meteorológicos, faz-se necessário o aferimento de variáveis microclimáticas. Para tal, utiliza-se diversos instrumentos, com sensores diversificados, para leitura específica de cada variável microclimática, como temperatura do ar, umidade do ar, velocidade do vento, radiação, entre outras. Com isso a leitura desses instrumentos de medição instalados, são armazenadas em sistemas de aquisição de dados.

O agrupamento sequencial desses dados, corresponde a uma *ST*, que podem ser definidas 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 (1927), uma *ST* deveria ser vista simplesmente como a realização de um processo estocástico. Cada uma das observações podem ser consideradas como pontos em um gráfico de duas dimensões, onde o eixo das ordenadas, determinam as medições dos dados e o eixo das abscissas, delimitam em que momento discreto do tempo tais medições foram aferidas. A Figura 1 ilustra um exemplo de série temporal com 2112 medições da temperatura do ar no decorrer de três meses.

Figura 1: Série temporal horária da temperatura do ar.



Fonte: do autor

Séries temporais se diferem em tendência e sazonalidade. A tendência capta elementos de longo prazo relacionados com a série em si, conforme aponta 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 a análise de séries temporais, à 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). A análise também pode estar alinhada ao almejo da 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 mal uso ou defeitos de fábrica, intempéries, falta de manutenção ou até mesmo erros na manipulação dos sensores. Falhas caracteriza ausência de dados, formando lacunas na ST. Dessa forma pode levar a afirmações tendenciosas, pois uma série com falhas pode se distanciar do real comportamento do fenômeno estudado. Nesse caso 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 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 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. Diante disto, falhas em uma ST micrometeorológica é um problema iterado, que pode prejudicar a pesquisa.

2.2 Métodos de Preenchimento de Falhas

Falhas ou inconsistências podem gerar dificuldades, dúvidas ou imprecisão na análise dos dados (BABA et al., 2014). Vários métodos são empregados com o intuito de obter maior exatidão nas estimativas de dados ausentes. Dentre eles podem ser destacados os métodos baseados em estatística (RIHBANE, 2018) e, também, os métodos baseados em Inteligência Artificial (VENTURA, 2012).

2.2.1 Modelos Estatísticos

2.2.1.1 Regressão Linear

A análise de regressão tem por objetivo descrever através de um modelo matemático, as relações existentes entre duas ou mais variáveis, a partir de n observações dessas variáveis (HAIR et al., 2014, p. 715 – 716). A tarefa de encontrar uma curva parametrizada, que corresponda aproximadamente a um conjunto de dados é chamada de regressão, ou seja, caracteriza-se regressão linear, quando o modelo de regressão é linear nos parâmetros ajustados. Na regressão linear, há um relacionamento funcional entre variáveis correlacionadas. Um parâmetro denominado de variável independente é utilizado na predição de outro, que recebe o nome de variável dependente, de modo linear, como na Equação 2.1 (RITZMAN et al., 2004).

$$y = \alpha + b(x) \quad (2.1)$$

Onde y é a variável dependente, α denota a intersecção no eixo y , b a inclinação da reta e x a variável independente. Modelos de regressão linear são frequentemente utilizados na exploração da relação entre um resultado contínuo e variáveis independentes (SCHMIDT et al., 2014; AUSTIN; LAUPACIS, 2011).

O coeficiente de correlação, o erro-padrão e o intervalo de confiança são indicadores da qualidade da regressão. O coeficiente de correlação r mostra a relação entre as variáveis dependente e independente. Valores positivos indicam variação no mesmo sentido das variáveis. Valores negativos indicam variáveis inversamente relacionadas. Um valor nulo significa que não existe relação. Quanto mais próximo r estiver de 1, maior será a correlação.

O coeficiente de determinação R^2 pode ser utilizado para indicar a intensidade da correlação. Valores próximos a um são desejáveis e indicam que as variáveis dependente e independente estão fortemente correlacionadas (NETO, 2002). Em outras palavras, indica quanto o modelo foi capaz de explicar os dados originais. O coeficiente de determinação é dado pela Equação 2.2:

$$R^2 = \frac{1 - u}{v} \quad (2.2)$$

onde u é a soma residual dos quadrados e v é a soma total dos quadrados.

Embora a melhor pontuação possível é 1, pode ter resultados negativos, indicando um modelo inversamente proporcional. Um modelo constante que sempre prevê o valor esperado de y , desconsiderando as variáveis independentes, obteria uma pontuação igual a zero.

2.2.1.2 Regressão Linear Múltipla

A Regressão Linear Múltipla (RLM) envolve duas ou mais variáveis, ou seja, uma única variável dependente y e duas ou mais variáveis independentes ou explanatórias ou covariáveis ou regressoras. A teoria é uma extensão da análise de regressão linear simples no qual tem por objetivo estabelecer uma equação que possa ser usada para prever valores de y para valores dados das diversas variáveis independentes. A finalidade das variáveis independentes adicionais é melhorar a capacidade de predição em confronto com a regressão linear simples (LEVINE et al., 2005). O modelo genérico é dado pela Equação 2.3 quando aplicado a uma amostra de tamanho n .

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} + \varepsilon_i, i = 1, \dots, n \quad (2.3)$$

onde:

n é o tamanho da amostra;

y_i é a observação da variável dependente para a i -ésima posição;

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 $\beta = (\beta_0, \beta_1, \beta_2, \dots, \beta_k)$, por isso a descrição de linearidade.

2.2.1.3 Média Móvel

Médias móveis são frequentemente usadas para analisar séries temporais. Segundo Idris (2014, p. 167–168), uma média móvel simples especifica uma janela de dados que é previamente visualizada, e que é calculada a cada vez que a janela desliza para frente através de um período, conforme a Equação 2.4:

$$SMM = \frac{\sum_{i=1}^n a_i}{n} \quad (2.4)$$

onde a_i é a variável no período i e n o número de períodos incluídos em cada média.

Dependendo do caso, as medidas mais recentes têm uma maior importância para a previsão do que as mais antigas. Dessa forma, é possível aplicar o método da média móvel ponderada, cada período tem seu próprio peso e a soma dos pesos tem que ser igual a 1 (MOREIRA, 2017, p. 312).

2.2.2 Modelos de Aprendizado de Máquina

O aprendizado de máquina é um subcampo da IA. Definido por Robert (2014) como um conjunto de métodos que podem detectar automaticamente padrões em dados e, em seguida, usá-los para previsão de dados futuros ou executar outros tipos de decisões sob incerteza.

Cada técnica de aprendizado de máquina usa modelos matemáticos específicos, ou seja, uma representação matemática do comportamento de sistemas, para armazenar e aplicar o que aprendeu. Essas técnicas podem melhorar o desempenho dos modelos, recebendo mais e mais dados para cada vez mais aprofundar seu aprendizado, afim de alcançar uma tarefa específica. Primeiramente, o conjunto de técnicas de aprendizado de máquina é usado em domínios problemáticos nos quais o desenvolvimento de uma solução é inviável com programação explícita ou mesmo manualmente (SAMUEL, 1959). Já que os padrões podem ser difíceis de encontrar pela percepção humana quando há uma grande quantidade de dados para analisar, os algoritmos de aprendizado de máquina podem ser úteis para procurar regras e exceções, e até mesmo substituir programas baseados em regras.

A vantagem na utilização de modelos de aprendizado de máquinas frente a modelos estatísticos, é que estes são orientados a dados, isto é, a priori, não há necessidade de compreender o comportamento do conjunto de dados que será utilizado para o treinamento do algoritmo. Compreensão esta que abrange a

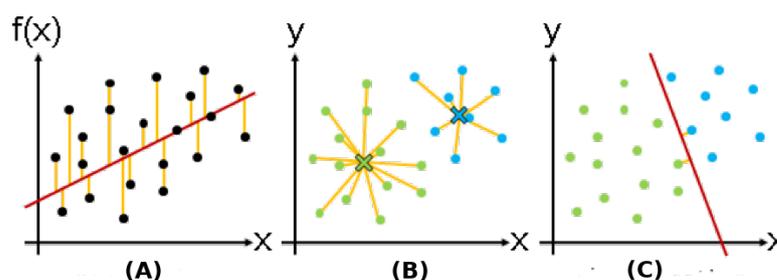
ciência sobre a linearidade ou não do modelo, cabendo este ajuste ao algoritmo de forma automática (MICHALSKI et al., 2013).

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 regressão (figura 2A): em que se espera que o modelo de aprendizado de máquina gere um valor real para algum exemplo, tentando ajustar uma curva aos dados. Para elaboração do presente trabalho foi desenvolvido um modelo de DL para a tarefa de regressão.
- tarefas de clusterização (figura 2B): em que o modelo de aprendizado de máquina tenta agrupar exemplos em grupos por suas semelhanças de recursos.
- tarefas de classificação (figura 2C): 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 2: Regressão (A): ajustar uma curva aos dados. Clusterização (B): agrupamento dos dados. Classificação (C): encontrar uma fronteira de decisão.



Fonte: Modificado de Nextview(<https://www.nextview.nl>)

A medida de desempenho P é necessária para saber o quanto uma técnica de aprendizado de máquina está realizando uma tarefa T , ou seja, P deve ter uma métrica de avaliação bem definida e diretamente relacionada a tarefa.

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

Conforme descrito a experiência E , que pode ser definida como um conjunto de exemplos, e que é utilizada para o aprendizado do modelo, de forma que quanto mais experiência houver disponível para o aprendizado, mais o modelo aprenderá e obterá melhor desempenho. O aprendizado de máquina pressupõe três categorias de processo de aprendizagem: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço. Esta divisão proposta baseia-se na experiência disponível durante a aprendizagem, mas cada processo de aprendizagem tem suas particularidades (RUSSELL; NORVIG, 2016). Com base nos dados e no domínio do problema, disponíveis neste trabalho, foi adotado o aprendizado supervisionado.

Na categoria de aprendizado supervisionado, o algoritmo tenta ajustar o modelo a partir de dados na qual a saída correta já é conhecida, tendo a ideia de que existe uma relação entre a entrada e a saída. Cada exemplo é um par que consiste em um conjunto de entradas e um valor de resposta correspondente. Considerando isso, é possível descrever exemplos em termos de pares (X, y) , onde X é um vetor de características e y um valor associado, correspondente ao valor que o modelo deve produzir dado X . Assim, a aprendizagem supervisionada envolve observar vários exemplos (X, y) e ajustar o modelo para prever y de X .

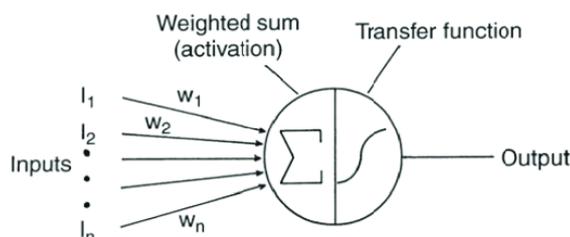
2.2.2.1 Redes Neurais Artificiais

Redes Neurais Artificiais (RNA) foram criadas com base no cérebro humano, imitando a função dos neurônios biológicos e suas conexões, simulando a forma com que o cérebro processa dados, identificando e memorizando padrões por meio das sinapses, que podem ser descritas como uma atualização dos pesos das conexões entre neurônios (LOESCH; SARI, 1996).

Pela definição de Principe et al. (2000), RNA são máquinas de aprendizado distribuídas, adaptativas e geralmente não-lineares construídas a partir de várias unidades de processamento independentes. A Figura 3 ilustra uma rede neural com um único neurônio artificial, exemplificando suas entradas $l_1, l_2 \dots l_n$, com

seus correspondentes pesos $w_1, w_2 \dots w_n$, soma ponderada das entradas e a função de ativação.

Figura 3: Uma rede neural com um único neurônio. Nessa configuração, a saída do neurônio é a saída geral da rede neural.



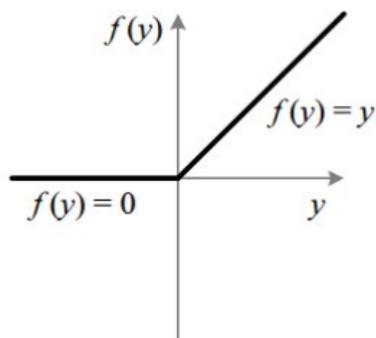
Fonte: (MISHRAA et al., 2016)

Na equação 2.5, w_i é o peso de cada sinapse i , que representa a ponderação relativa da mesma, enquanto l_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.

$$\sum_{i=1}^n w_i l_i \quad (2.5)$$

A função de ativação, também chamada de função de transferência, é uma função objetiva escalar parametrizada predefinida que afeta o comportamento da rede, restringindo a saída do neurônio. Existem diversas funções matemáticas que são utilizadas como função de ativação, a exemplo a *Rectified Linear Unit* (ReLU), uma função não linear bastante utilizada por garantir um treinamento muito mais rápido que seus equivalentes não-lineares (JIANG et al., 2018). Seu comportamento consiste em ser zero para valores negativos, e crescer linearmente para valores positivos, conforme é possível constatar na Figura 4 (GULLI; PAL, 2017). No exemplo citado, l_n é o n -ésimo exemplo de X e, ajustando w_n , é possível que a saída seja igual a y_n , a n -ésima saída esperada. O ajuste do parâmetro w , é uma tentativa de minimizar o erro entre a saída real y do exemplo X e a saída da rede neural.

Figura 4: Comportamento da função ReLU.

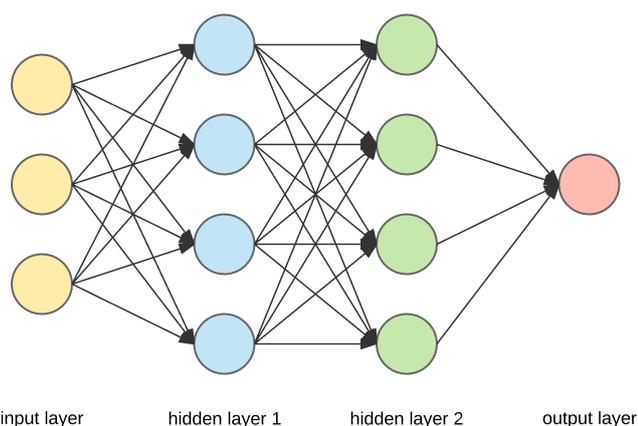


Fonte: modificada de Gulli e Pal (2017)

Para um modelo com um único neurônio, a saída é de um único valor. Nas redes neuronais multicamadas, a saída dos neurônios se torna a entrada para outros neurônios. De acordo com Cybenko (1989), uma rede com mais de uma camada possibilita a aproximação de qualquer função matemática. Existem várias arquiteturas de redes neurais, e elas são nomeadas de acordo com suas características e funcionalidades.

Uma arquitetura regular é a Perceptron Multicamadas (MLP, do inglês *Multilayer Perceptron*), na qual os neurônios são dispostos lado a lado em camadas, e a saída de cada camada é entrada para a próxima camada. Como ilustrado na Figura 5.

Figura 5: Rede neural multicamada.



Fonte: Dertat, 2017 <https://towardsdatascience.com>

As camadas internas são denominadas como camadas ocultas, pois não há interação direta com essas camadas, mas apenas com as camadas de entrada e saída. A arquitetura é completamente conectada, os neurônios de uma camada

c , estão conectados com cada neurônio da camada sucessora $c + 1$. As conexões entre os neurônios representam os pesos e regulam a influência de um neurônio em seus sucessores. A estrutura das conexões pode ser alterada, mas é difícil medir como essas alterações afetam o desempenho da rede. O número de entradas na camada de entrada corresponde a quantidade de variáveis que serão usadas para alimentar a rede neural (FACELI et al., 2011).

Após a especificação da arquitetura da rede neural, é necessário definir o algoritmo de treinamento da rede, que consiste em uma redução não linear irrestrita, em que os pesos sinápticos da rede são repetitivamente atualizados para minimizar o erro médio quadrático entre o y e a saída obtida no neurônio de saída. Do ponto de vista estatístico, o treinamento seria estimar os pesos (LONG et al., 1990).

Um algoritmo comumente utilizado para treinamento supervisionado de redes multicamadas é o de retropropagação (RUMELHART et al., 1986), baseado em gradiente descendente, sua aplicação requer a escolha de uma função de ativação contínua, diferenciável e, de preferência, não decrescente. Necessário também estabelecer um conjunto de parâmetros externos da rede. O próximo tópico é destinado a explicação destes parâmetro.

2.2.2.2 Parâmetros Externos da Rede

Fora a definição de configurações internas como quantidade de neurônios e camadas a serem utilizadas no modelo, é preciso estabelecer alguns parâmetros externos (*hyperparameters*) como: épocas, função de custo, otimizador da descida do gradiente e taxa de aprendizado. A configuração desses parâmetros pode ter impacto decisivo para a capacidade de generalização da rede (PATTERSON; GIBSON, 2017). A seguir é feita uma explicação de cada parâmetro:

- Épocas: Quantidade de iterações do algoritmo, pode ser definida como tamanho de ciclos de treinamento definido por um número inteiro (CARPENTER et al., 1992).
- Função de custo: é responsável por calcular a diferença entre o resultado fornecido pela rede e o esperado. Para isso é necessário fazer uma média de todos os exemplos, utilizando uma função que sempre retorne um valor positivo. Como alternativa a REMQ pode ser utilizada.
- Otimizador da descida do gradiente: Como o próprio nome sugere, tem como finalidade a otimização da função de ativação. Um método assaz

utilizado é o Adam (KINGMA; BA, 2014), por ser mostrar bastante eficiente, requer apenas gradientes de primeira ordem com pouca necessidade de consumo de memória.

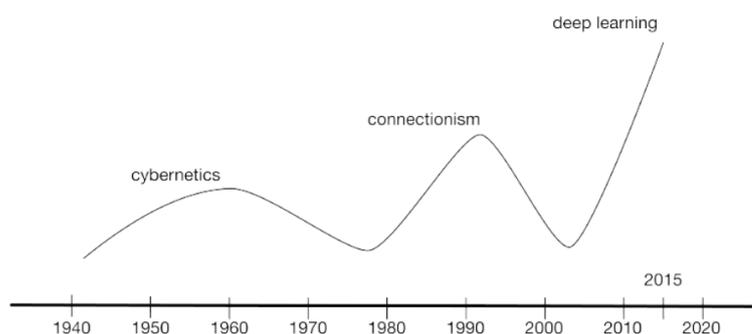
- Taxa de aprendizado: a definição de valor para essa taxa requer alguns cuidados e conhecimento da tarefa a ser desempenhada. Valores muito pequenos tornam o treinamento lento, enquanto muito grandes podem provocar divergência do processo de treinamento (KLEINBAUM; KLEIN, 1994).

Chacón (2014) afirma que a seleção dos *hyperparameters* que são mais adequados não é uma tarefa fácil, pois a quantidade de possibilidades à serem escolhidas é geralmente muito alta e, em muitos casos, uma avaliação de todas as combinações possíveis não é prática, levando a um ajuste arbitrário ou subjetivo.

2.2.2.3 Deep Learning

Definida por três marcos na pesquisa de redes neurais, a DL teve seu início nos anos 1940-1960 onde era conhecida como cibernética, começando com o perceptron (ROSENBLATT, 1958) para treinar um único neurônio. Posteriormente a abordagem conexionista do período entre 1980 e 1995, com a criação do algoritmo de retropropagação para treinar uma rede neural com uma ou duas camadas ocultas. Por fim, mais recentemente por volta de 2006, onde o termo DL começou propriamente a ser utilizado (HINTON et al., 2006). A Figura 6 ilustra a ordem cronológica dos acontecimentos.

Figura 6: Timeline da Deep Learning.

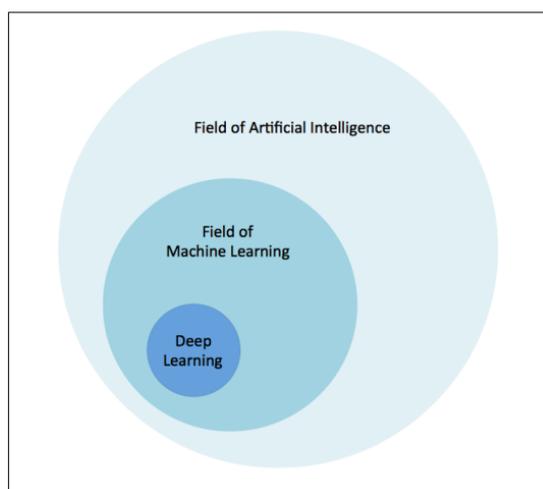


Fonte: (GOODFELLOW et al., 2016)

DL, do inglês aprendizagem profunda, é uma área pertencente a IA (Figura 7). Definida por Bengio (2009) como um tipo particular de aprendizado de máquina, que alcança grande poder e flexibilidade, aprendendo a representar

o mundo como uma hierarquia aninhada de conceitos e representações, estabelecendo cada conceito de forma mais abstratas, computados em termos menos abstratos. Os modelos profundos são bons em descobrir recursos automaticamente de maneira hierárquica obtidos pela composição de módulos simples, mas não lineares, possibilitando que funções muito complexas possam ser aprendidas (LECUN et al., 2015).

Figura 7: Relação entre IA e deep learning.



Fonte: (PATTERSON; GIBSON, 2017)

Outra perspectiva sobre DL, é permitir que o computador aprenda um programa com várias etapas. Cada camada da representação pode ser considerada como um estado da memória do computador depois de executar outro conjunto de comandos em paralelo. Redes com maior profundidade podem executar mais instruções em sequência, oferecendo grande poder, já que as instruções posteriores podem se referir aos resultados das anteriores (GOODFELLOW et al., 2016, p. 5). Dessa forma o computador consegue entender o significado de dados brutos de entrada, como uma imagem representada por uma matriz dos valores de pixel e identificar formas, como no caso das Redes Neurais Convolucionais (CNN) .

As CNN são redes neurais de arquitetura multicamadas, nas quais as camadas sucessivas são projetadas para aprender progressivamente recursos de nível mais alto, até a última camada (HUANG; LECUN, 2006). Consideradas como a primeira abordagem de DL bem-sucedida, em que muitas camadas de uma estrutura hierárquica são treinadas com êxito de maneira eficaz, aprendendo automaticamente as características salientes da imagem e retornando uma boa precisão de reconhecimento (DENG et al., 2014). Projetadas para o processamento de dados bidimensionais, como imagens e vídeos, são fundamentadas conforme o próprio

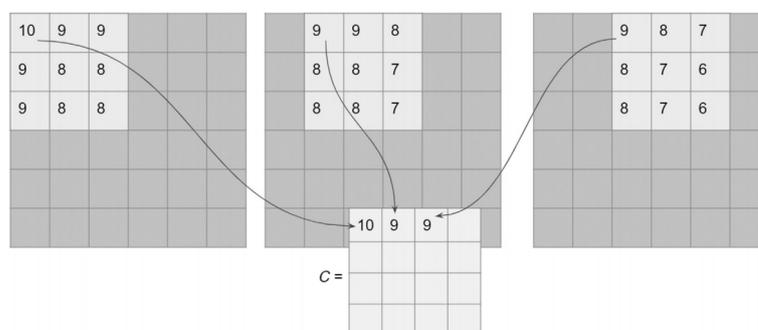
nome sugere no princípio da convolução discreta, ou seja, uma simples operação matemática entre duas matrizes (A e B) resultando em uma terceira (C) Equação 2.6.

$$C[l][c] = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} \sum_{h=0}^{m-1} A[i][j]B[k][h] \quad (2.6)$$

Onde l e c são respectivamente as posições da linha e coluna da matriz C , n a quantidade de linhas e m a de colunas.

Para facilitar o entendimento, considere a Figura 8, a matriz B desliza sobre a matriz maior A da esquerda (superior) para a direita (inferior). Em cada posição sobreposta, os elementos correspondentes de A e B são multiplicados e todos os produtos são adicionados para obter o elemento correspondente para C . A matriz de saída resultante C será menor em dimensão que A porém maior que B (KOTU; DESHPANDE, 2018, p. 325–335). A Matriz A é tipicamente uma imagem bruta em que cada célula na matriz é um valor de pixel. A matriz B é chamada de filtro que, quando convolvida com a imagem bruta, resulta em uma nova que realça apenas alguns recursos da imagem bruta.

Figura 8: Visualização dos passos para formação da matriz C pelo processo de convolução entre as matrizes A e B .



Fonte: (KOTU; DESHPANDE, 2018)

Em CNN, os estados em cada camada são organizados de acordo com uma estrutura de grade espacial. Esses relacionamentos espaciais são herdados de uma camada para a próxima porque cada valor de recurso é baseado em uma pequena região espacial na camada anterior (AGGARWAL, 2018, p. 318). Assim, as convoluções são úteis para identificar e detectar recursos básicos em imagens, como bordas.

Outro tipo bastante utilizado são as Redes Neurais Recorrentes (RNN), um método de destaque dentro de DL pelo seu uso generalizado, em que a profundidade pode ser tão grande quanto o comprimento da sequência de dados de

entrada (DENG et al., 2014, p. 220–222). Estas redes encontram aplicação em qualquer situação que envolva uma componente temporal nos dados. Amplamente utilizadas na análise de sentimento envolvida em uma série de palavras que compõem uma frase, reconhecimento do sujeito em uma sentença (GRAVES et al., 2013) e tradução linguística (BAHDANAU et al., 2014).

Existem outros métodos de DL, cada qual com uma usabilidade diferente, com relação a profundidade da rede (quantidade de camadas) não existe uma ideia generalista. Para Kotu e Deshpande (2018, p. 324) um entendimento comum é que qualquer rede com três ou mais camadas ocultas entre as camadas de entrada e saída é considerada profunda. Já para (AREL et al., 2010; WASON, 2018) DL é uma aplicação de redes neurais com quatro ou mais camadas de nós entre entrada e saída.

Para Goodfellow et al. (2016, p. 8), não existe uma definição estrita sobre a profundidade que um modelo requer para ser qualificado como “profundo”. No entanto, DL pode ser considerada com segurança como o estudo de modelos que envolvem uma maior quantidade de funções ou conceitos aprendidos em comparação a aprendizagem tradicional de máquina. A ideia de aprender a representação certa para os dados, fornece uma perspectiva sobre o aprendizado profundo.

2.3 Seleção de Variáveis e Separação em Dados de Treino e Teste

O objetivo principal de um modelo de aprendizado de máquina é ser eficaz em exemplos não vistos. Uma maneira de avaliar o quão bem o algoritmo aprendeu a realizar uma tarefa, é verificar se o mesmo mantém um bom desempenho ao se deparar com exemplos no qual nunca teve contato, isto é, dados que não foram utilizados no treinamento.

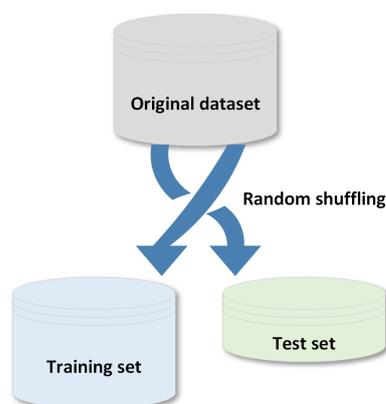
Caso um algoritmo de aprendizado de máquina fosse avaliado com o mesmo conjunto de dados utilizado no treinamento, obteria um desempenho considerável. Mas as previsões feitas em novos dados seriam inconsistentes, pois os dados de treinamento poderiam não ser representativos, ou seja, o modelo tornaria-se específico para determinados dados. Caracterizando assim um problema de *overfitting* (sobre ajuste).

O *overfitting* ocorre quando perde-se a capacidade de generalização, pois o aprendizado converge ao comportamento específico dos dados de treinamento e não é mais capaz de representar o comportamento geral dos mesmos (DIETTERICH, 1995). Nessa linha pode ocorrer também o *underfitting* (sub ajuste) que

nada mais é do que a generalização, ou seja, a previsão não é suficiente para representar de forma geral o comportamento do problema, representando-o apenas de forma simplificada (MJALLI et al., 2007).

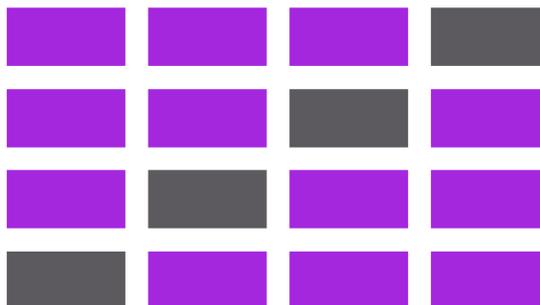
A avaliação é uma estimativa que pode ser usada para falar sobre o quanto o algoritmo pode realmente fazer na prática. Para avaliação, uma boa prática é efetuar a reamostragem, que consiste em dividir o conjunto de dados em subconjuntos de treinamento e teste; o primeiro a ser usado para treinar o modelo e o segundo para testar seus desempenhos conforme descreve Brownlee (2016, 57–58). A Figura 9 a seguir é uma representação deste processo.

Figura 9: 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 de teste é perdido para o conjunto de treinamento, ou seja, pode ocorrer dos dados de teste conterem informações valiosas para o treinamento. Uma maneira de minimizar esse problema é a utilização da validação cruzada (BENGIO; GRANDVALET, 2004), que consiste na divisão dos dados em K segmentos de igual tamanho, e, a partir disto, um segmento é utilizado para teste e os $k - 1$ restantes são utilizados para treinamento. Este processo é realizado k vezes alternando de forma circular o subconjunto de teste, em cada vez é calculada a acurácia do modelo, depois é feita a média de todas as k vezes. Para esta validação é dado o nome de k -fold, a Figura 10 ilustra esse processo.

Figura 10: k -fold, onde $k = 4$.

Fonte: do autor.

Outro fator importante a ser levado em consideração é a escolha das variáveis utilizadas no treinamento do modelo de aprendizado de máquina, pois têm uma grande influência no desempenho que o mesmo pode alcançar. Variáveis irrelevantes ou parcialmente relevantes podem impactar negativamente o desempenho do modelo. De acordo com Hawkins (2004), deve-se sempre levar em consideração o princípio da parcimônia, no qual diz que se um modelo de regressão com dois preditores for suficiente para explicar y , não mais do que esses dois preditores devem ser usados.

A seleção de variáveis é um processo no qual as variáveis são separadas automaticamente dos dados, levando em consideração os que mais contribuem para a variável de previsão ou saída. Ter variáveis irrelevante pode diminuir a precisão de muitos modelos, especialmente de algoritmos lineares, como regressão linear e logística. É possível enumerar três benefícios ao se aplicar a seleção de variáveis antes da modelagem: redução de *overfitting*, melhora 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.

Capítulo 3

MATERIAIS E MÉTODOS

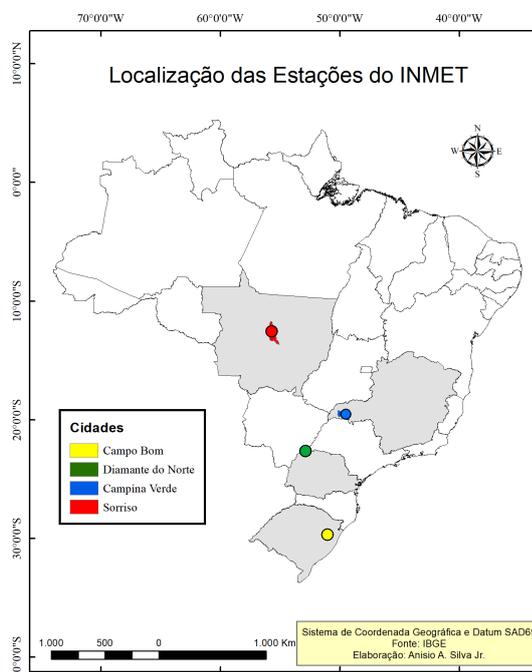
Neste capítulo é descrita a origem e composição da base de dados utilizada, bem como foi feita a seleção das variáveis independentes e a separação da mesma em base de treinamento e base de validação. É descrito também como foi todo o desenvolvimento do modelo de deep learning proposto, a linguagem utilizada e a definição da arquitetura, além de descrever quais e como foram os testes realizados no presente trabalho.

3.1 Dados Utilizados

Os dados utilizados neste trabalho são provenientes de quatro estações meteorológicas, pertencentes ao Instituto Nacional de Meteorologia (INMET), disponíveis em sua própria base de dados¹. As estações escolhidas estão localizadas em quatro estados diferentes, nas cidades de Campina Verde - MG, Sorriso - MT, Diamante do Norte - PR e Campo Bom - RS. Essas informações podem ser conferidas na Figura 11.

¹Base de dados INMET: <http://www.inmet.gov.br>

Figura 11: Localização das quatro estações meteorológicas selecionadas pertencentes ao INMET.



As estações foram escolhidas com considerável espaçamento territorial, para garantir variabilidade entre as séries temporais, uma vez que isso seja imprescindível para avaliar o aprendizado do modelo. Perante isso, cada região possui sua própria base de dados, o que possibilita o estudo individual da mesma.

As séries de dados foram coletadas no período de novembro de 2014 à janeiro de 2015. De acordo com INMET (2011), a leitura foi realizada a cada cinco segundos, no final da décima segunda leitura, é armazenada a média das mesmas. Para emissão do relatório online, foi adotada a média horária. Os dados são compostos pelas as variáveis microclimáticas: radiação solar, velocidade do vento, ponto de orvalho, umidade relativa do ar, pressão atmosférica e temperatura do ar. Na Tabela 1 é possível ver os valores dos desvios padrão de cada variável por região.

Tabela 1: Séries microclimáticas e seus respectivos desvios padrão (σ), por região.

σ Variáveis	Camp. Verde	Sorriso	Dia. do Norte	Campo Bom
Radiação Solar (kJ/m^2)	1202,505	1206,202	1276,763	1049,376
Vel. do Vento (m/s)	1,438	0,892	1,631	0,766
Pont. de Orvalho ($^{\circ}C$)	1,916	3,338	2,602	0,900
Umidade do ar (%)	19,193	18,538	16,505	13,559
Press. atm. (hPa)	1,946	3,982	2,302	1,968
Temp. do ar ($^{\circ}C$)	4,204	4,590	3,693	3,269

3.2 Desenvolvimento do Modelo de Deep Learning

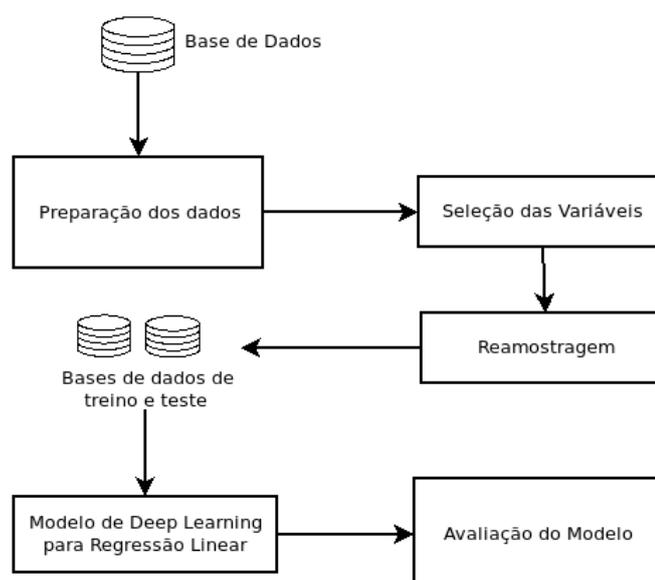
No desenvolvimento computacional do presente trabalho, foi utilizada a linguagem de programação *Python*, juntamente com a plataforma de desenvolvimento *Spyder*. A escolha da linguagem se deve ao fato de se tratar de uma linguagem de propósito geral, que pode ser utilizada tanto para pesquisa e desenvolvimento quanto para produção. Para que o processamento ocorresse em paralelo na GPU e CPU, também foi empregada a interface de programação de aplicativos CUDA (*Compute Unified Device Architecture*) desenvolvida pela empresa multinacional de tecnologia incorporada Nvidia.

O computador utilizado no processamento dos testes possui as seguintes características:

- Sistema operacional: Ubuntu 18.04.1 LTS X86_64
- CPU: Intel i7 928 2.66GHz Cache 8mb 4 Cores
- GPU: NVIDIA GeForce GTX 980
- Memória RAM: 6Gb

A Figura 12 contém todas as etapas de desenvolvimento que, no decorrer trabalho, serão explicadas individualmente com suas respectivas peculiaridades.

Figura 12: Etapas de desenvolvimento do presente estudo.



Fonte:do autor

Para verificação de desempenho foi utilizado o Erro Médio Absoluto, MAE do inglês *Mean Absolute Error* (Equação 3.1), medida de precisão comum para tarefas de regressão aplicadas às ciências climáticas e ambientais utilizada para mensurar a proporção de exemplos que o modelo identifica corretamente. É o erro médio absoluto entre o valor previsto por um modelo e o valor observado para um exemplo. Não apenas informa se o modelo está errado, mas também, a que distância o valor de saída é do valor esperado. Dessa maneira, indica como o modelo falha na estimativa da variabilidade das medições em torno da média e mede a variação nos valores estimados em torno dos medidos.

$$MAE = n^{-1} \sum_{i=1}^n |P_i - O_i| \quad (3.1)$$

onde P_i é o valor estimado, O_i o valor observado e n é a quantidade de valores observados.

Outra medida de performance utilizada, é a média do erro quadrático (MSE do inglês *mean squared error*) (Equação 3.2), amplamente utilizada em análise de processamento de resposta (WANG; BOVIK, 2009).

$$MSE = n^{-1} \sum_{i=1}^n (P_i - O_i)^2 \quad (3.2)$$

3.2.1 Preparação dos Dados

A preparação dos dados é uma etapa importante para o aprendizado de máquina, pois muitas das variáveis divergem em valores, o que pode tornar a tarefa de aprendizado muito custosa.

O dimensionamento das variáveis independentes (X) é uma etapa comum do aprendizado de máquina, utilizado com o intuito de evitar que recursos com valores grandes possam ter mais influência nos resultados finais (GARRETA; MONCECCHI, 2013, p. 10–12). A Equação 3.3 é utilizada para essa finalidade:

$$z = \frac{X - u}{s} \quad (3.3)$$

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

3.2.2 Seleção das Variáveis

Para seleção de variáveis foi utilizando o coeficiente de determinação, atribuindo pesos as variáveis independentes, formando assim um "rank" dos coeficientes. Uma vez os pesos definidos, ocorre a seleção das variáveis com maior importância.

O presente trabalho testou gradativamente todas as variáveis preditoras, de maneira que a primeira seleção considera apenas a variável de maior peso, para aplicação do modelo de DL, na seleção seguinte a segunda variável independente é adicionada com a anterior, e assim por sequência até a seleção de todas as variáveis. Dessa forma é possível mensurar o quão bem o modelo se comporta com diferentes quantidades de variáveis.

3.2.3 Reamostragem

Para cada região, a etapa de reamostragem considerou quatro configurações díspares: A, B, C e D. Cada uma representado uma porcentagem da base de dados original, conforme a Tabela 2.

Tabela 2: Método de reamostragem utilizado. Cada letra representa uma configuração adotada, os valores são percentuais da base de dados original.

Reamostragem	A	B	C	D
Base de Treinamento	95%	85%	70%	50%
Base de Teste	5%	15%	30%	50%

Esse critério foi utilizado para simular falhas em uma situação real. Se a base trabalhada possuir 5% de falhas, o restante dos dados será utilizado para treinamento do modelo de DL. Possibilitando uma análise do preenchimento com o porcentagem de escassez de dados de treinamento aumentando entre 5% à 50%.

Na separação, os dados foram selecionados de forma pseudoaleatória, com a utilização de uma semente de geração de números aleatórios, definida por um número inteiro aplicado ao algoritmo gerador de números aleatórios Mersenne Twister (MATSUMOTO; NISHIMURA, 1998), que garante a replicação da aleatoriedade, toda vez que a mesma semente for utilizada, ou seja, sempre produzirá os mesmos resultados, inclusive o erro de arredondamento.

A utilização de uma seleção pseudoaleatória, assegura a reprodutibilidade da amostra, uma vez que a mesma é utilizada para diversas configurações de reamostragem. Para a validação do experimento foi adotada a validação cruzada, utilizando o método k -fold com $k = 6$.

3.2.4 Modelo de Deep Learning para Regressão

Para a construção do modelo de DL, foram utilizadas as bibliotecas científicas da linguagem python: Keras² e Tensorflow³. O modelo de DL proposto, recebe uma quantidade l de entradas, podendo variar, conforme especificado anteriormente, e tem apenas um neurônio na camada de saída. Os parâmetros externos de treinamento permaneceram imutáveis ao longo de todos os testes, conforme a Tabela 3.

Tabela 3: Configuração dos parâmetros externos de treinamento.

Parâmetros	Valores
Épocas	50
Função de custo	MSE
Otimizador de desc. gradiente	Adam
Taxa de Aprendizagem	0,001

Na configuração interna, duas estruturas foram utilizadas, uma com o intuito de avaliar a quantidade de neurônios, e outra a proporção da profundidade. Para isso, foram feitos testes com duas e quatro camadas ocultas, com variações de 50 e 100 neurônios por camada.

Multiplicando pela quantidade de configurações resultantes da seleção de variáveis e reamostragem, foi obtido um montante de 10.050 composições, 2.511 por região.

3.2.5 Análise do Modelo e Comparação com Outros Métodos

Na análise do modelo, considerou-se quatro fatores:

- Quantidades de variáveis independentes necessárias para o preenchimento.
- Quantidade de neurônios.
- Profundidade da rede.
- Porcentagem de falhas presente na série.

A análise consistiu em saber se os quatro fatores citados tinham influência sobre o resultado. Para tal, foram utilizados testes de variância não paramétricos, uma vez que os dados não atendem às suposições de normalidade da ANOVA,

²<https://keras.io/>

³<https://www.tensorflow.org/>

mas são contínuos, independentes e as variâncias são homogêneas (ELLIOTT; HYNAN, 2011). O teste de Shapiro-Wilk (ROYSTON, 1992) foi utilizado na confirmação da normalidade dos dados.

Foram também empregados os testes: Mann-Whitney (BIRNBAUM, 1956), na comparação de até duas distribuições, e kruskal-Wallis (CHAN; WALMSLEY, 1997) nos que envolviam três ou mais. Na categoria de duas estão os testes sobre a quantidade de camadas e neurônios. Na de três distribuições à análise da porcentagem de falhas e quantidade de variáveis independentes. Para saber se algum dos grupos se diferem entre si, foi utilizado o teste de Nemenyi (NEMENYI, 1962), um teste post-hoc, que consiste em fazer comparações em pares.

Após a análise do modelo, o mesmo foi comparado com outros modelos existentes na literatura, no caso a média móvel e regressão linear múltipla. Para saber qual apresenta melhores resultados, o estimador MAE foi utilizado na avaliação, comparando o valor obtido pela rede (\bar{y}) e o valor real (y).

Capítulo 4

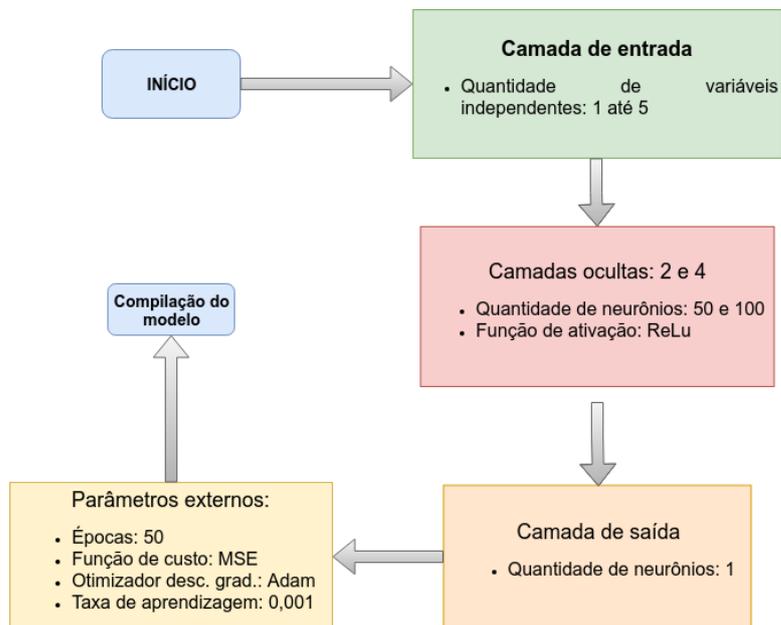
Apresentação e Análise dos Resultados

4.1 Modelo da Rede Neural DL

A Figura 13 é o fluxograma correspondente a fração do programa desenvolvido, que representa a configuração do modelo de DL, no qual inicialmente trabalha com duas camadas ocultas, e posteriormente recebe a adição de mais duas. Na camada de entrada é informada a quantidade de variáveis independentes a serem utilizadas. Em cada camada é informada a quantidade de neurônios, bem como a função de ativação. Também nessa mesma fração, estão inclusos os parâmetros externos de treinamento.

A função ReLU foi utilizada por proporcionar facilidade na otimização, por se tratar de uma função não linear e não ativar todos os neurônios ao mesmo tempo.

Figura 13: Fluxograma do algoritmo de DL.



Fonte: do autor

O código 4.1 é o modelo de DL propriamente dito, com as definições de profundidades, parâmetros internos e externos.

Código 4.1: DL

```

1  ## Biblioteca utilizadas ##
2  import tensorflow as tf
3  from tensorflow import keras
4
5  model = keras.Sequential([
6  keras.layers.Dense(neuronios, input_dim = input),
7  keras.layers.Dense(neuronios, activation=tf.nn.relu),
8  keras.layers.Dense(neuronios, activation=tf.nn.relu),
9  if camada >2:
10     keras.layers.Dense(neuronios, activation=tf.nn.relu),
11     keras.layers.Dense(neuronios, activation=tf.nn.relu),
12
13  keras.layers.Dense(1, activation=tf.nn.relu)])
14
15  ## Compile model
16  model.compile(optimizer=tf.train.AdamOptimizer(learning_rate=0.001),
17               loss= 'mean_squared_error', metrics=[ 'mean_squared_error' ])
18
19  ## Fit the model
20  model.fit(X_train, y_train, epochs= 50, batch_size=1, verbose=1)
  
```

4.1.1 Resultados do Modelo de DL

As Tabelas 4, 5, 6 e 7 contêm a média e desvio padrão dos valores do MAE, para cada variável de estudo, utilizando o modelo de DL com duas e quatro camadas ocultas. Cabe ressaltar que os valores presentes nas tabelas são referentes as quatro regiões de estudo.

Tabela 4: Média e desvio padrão(σ) dos resultados do MAE para testes realizados com o modelo de DL, com duas camadas e 50 neurônios cada, no preenchimento de falhas para as variáveis: temperatura do ar (T), umidade do ar (U), ponto de orvalho (PO), pressão atmosférica (P), radiação (R_g) e velocidade do vento (V_v)

MAE - 2 Camadas 50 neurônios												
Falhas	T (°C)		U (%)		PO (°C)		P (hPa)		R_g (kJ/m ²)		V_v (m/s)	
	\bar{t}	σ	\bar{u}	σ	\bar{p}_o	σ	\bar{p}	σ	\bar{r}_g	σ	\bar{v}_v	σ
5%	0,7156	1,1497	1,5821	2,6388	0,4464	0,6171	2,5938	0,8365	427,3865	106,1494	0,7453	0,3020
15%	0,7449	1,1981	1,5826	2,6118	0,4528	0,5985	2,6920	0,9010	427,5727	88,8386	0,7537	0,3053
30%	0,7274	1,1851	1,5787	2,6117	0,5052	0,6029	2,7153	0,8608	430,3409	91,0599	0,7647	0,2985
50%	0,7356	1,1782	1,6009	2,5970	0,5393	0,6085	2,8432	0,9550	435,7982	89,3507	0,7700	0,2976

Tabela 5: Média e desvio padrão(σ) dos resultados do MAE para testes realizados com o modelo de DL, com duas camadas e 100 neurônios cada, no preenchimento de falhas para as variáveis: temperatura do ar (T), umidade do ar (U), ponto de orvalho (PO), pressão atmosférica (P), radiação (R_g) e velocidade do vento (V_v)

MAE - 2 Camadas 100 neurônios												
Falhas	T (°C)		U (%)		PO (°C)		P (hPa)		R_g (kJ/m ²)		V_v (m/s)	
	\bar{t}	σ	\bar{u}	σ	\bar{p}_o	σ	\bar{p}	σ	\bar{r}_g	σ	\bar{v}_v	σ
5%	0,7396	1,1490	1,6017	2,5371	0,4198	0,6091	3,0332	1,1697	425,6442	105,2818	0,7403	0,2983
15%	0,7387	1,1986	1,6588	2,6639	0,4194	0,5908	2,9438	1,1425	425,0890	89,4686	0,7492	0,3063
30%	0,7540	1,1985	1,6266	2,5870	0,4390	0,6301	2,9783	1,2120	427,8634	90,8660	0,7665	0,2996
50%	0,7661	1,1962	1,6220	2,5576	0,4734	0,6230	3,3824	1,8589	434,4053	88,7610	0,7665	0,2974

Tabela 6: Média e desvio padrão(σ) dos resultados do MAE para testes realizados com o modelo de DL, com quatro camadas e 50 neurônios, no preenchimento de falhas para as variáveis: temperatura do ar (T), umidade do ar (U), ponto de orvalho (PO), pressão atmosférica (P), radiação (R_g) e velocidade do vento (V_v)

MAE - 4 Camadas 50 neurônios												
Falhas	T (°C)		U (%)		PO (°C)		P (hPa)		R_g (kJ/m ²)		V_v (m/s)	
	\bar{t}	σ	\bar{u}	σ	\bar{p}_o	σ	\bar{p}	σ	\bar{r}_g	σ	\bar{v}_v	σ
5%	0,7403	1,1396	1,6527	2,4876	0,4120	0,5890	4,3779	3,2616	417,8670	108,4431	0,7392	0,3015
15%	0,7858	1,1903	1,6530	2,4093	0,4114	0,6080	5,1344	3,5427	417,5503	91,4538	0,7536	0,3069
30%	0,7976	1,2069	1,6663	2,5045	0,4506	0,6454	4,9320	3,4455	419,6947	90,2062	0,7636	0,2965
50%	0,8078	1,2138	1,6911	2,5013	0,4662	0,6008	4,7603	2,6558	426,4806	90,8245	0,7731	0,2965

Tabela 7: Média e desvio padrão(σ) dos resultados do MAE para testes realizados com o modelo de DL, com quatro camadas contendo 100 neurônios cada uma, no preenchimento de falhas para as variáveis: temperatura do ar (T), umidade do ar (U), ponto de orvalho (PO), pressão atmosférica (P), radiação (R_g) e velocidade do vento (V_v)

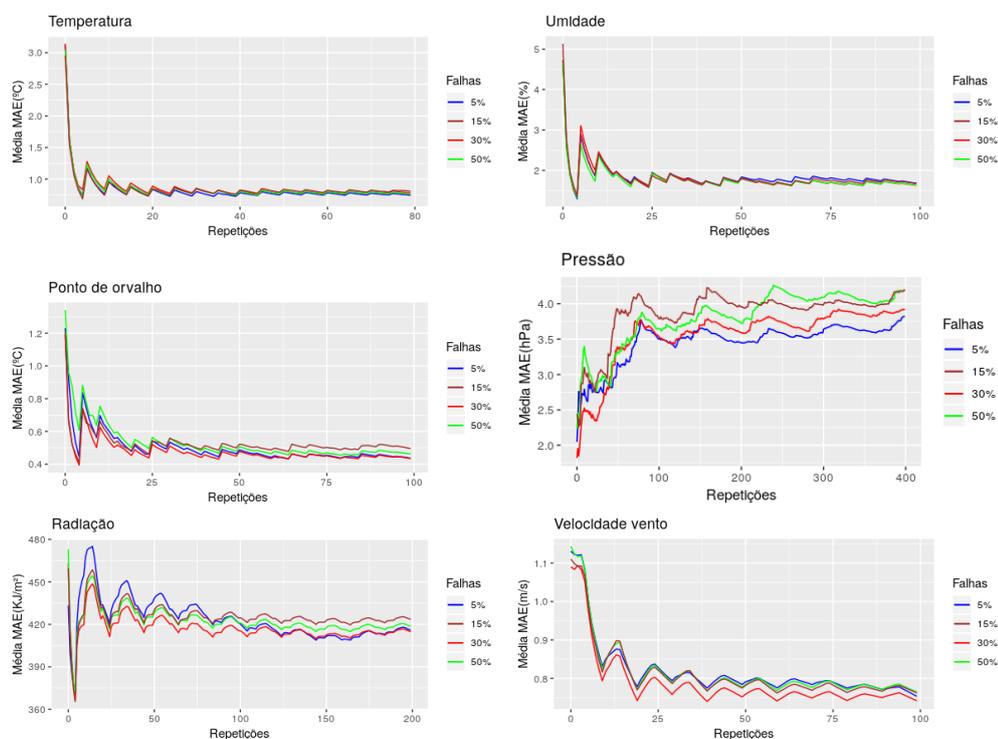
MAE - 4 Camadas 100 neurônios												
Falhas	T (°C)		U (%)		PO (°C)		P (hPa)		R_g (kJ/m ²)		V_v (m/s)	
	\bar{t}	σ	\bar{u}	σ	\bar{p}_o	σ	\bar{p}	σ	\bar{r}_g	σ	\bar{v}_v	σ
5%	0,7790	1,1451	1,7375	2,4723	0,4245	0,6126	5,5229	3,7829	412,5552	109,5641	0,7534	0,3016
15%	0,7932	1,2172	1,6973	2,4493	0,4245	0,6196	4,9135	3,2017	406,9007	88,4773	0,7484	0,3031
30%	0,7907	1,1839	1,7676	2,4638	0,4253	0,6078	6,0948	4,8901	413,0313	93,9254	0,7566	0,2945
50%	0,8097	1,1626	1,8152	2,6583	0,4444	0,6445	5,8194	4,5901	422,8068	93,3042	0,7676	0,3013

Observando as tabelas, o aumento da quantidade de camadas ocultas, não trouxe melhorias nos resultados para todas as variáveis, como pode ser constatado

nos valores da média da pressão atmosférica, temperatura do ar e umidade. Já para as variáveis ponto de orvalho e radiação é perceptível uma sutil melhoria nos valores da média. Também é possível constatar, com exceção das variáveis velocidade do vento, radiação e pressão, uma grande variação nos dados. O preenchimento de séries temporais de velocidade do vento demonstrou um baixo desempenho e insensibilidade a alteração da quantidade de camadas ocultas e presença de falhas.

A Figura 14 representa uma análise por variável para a relação entre a quantidade de testes efetuados pelo modelo e a média do MAE, revelando a quantidade de testes ideal para avaliação, que consiste na aproximação da convergência do MAE ao longo das repetições, o que acarreta em uma maior acurácia para o modelo. Com base nessa avaliação conclui-se que para a temperatura é necessário um valor mínimo de 20 repetições, para a umidade e ponto de orvalho esse valor chega a 80 repetições, para radiação seriam necessárias 150 repetições, 100 para a velocidade do vento e para a pressão atmosférica o valor de 400 repetições se mostrou insuficiente, o que caracteriza um estudo mais específico para determinação de tal valor.

Figura 14: Relação da média do MAE individual para cada porcentagem de falhas com a quantidade de testes efetuados pelo modelo de DL.

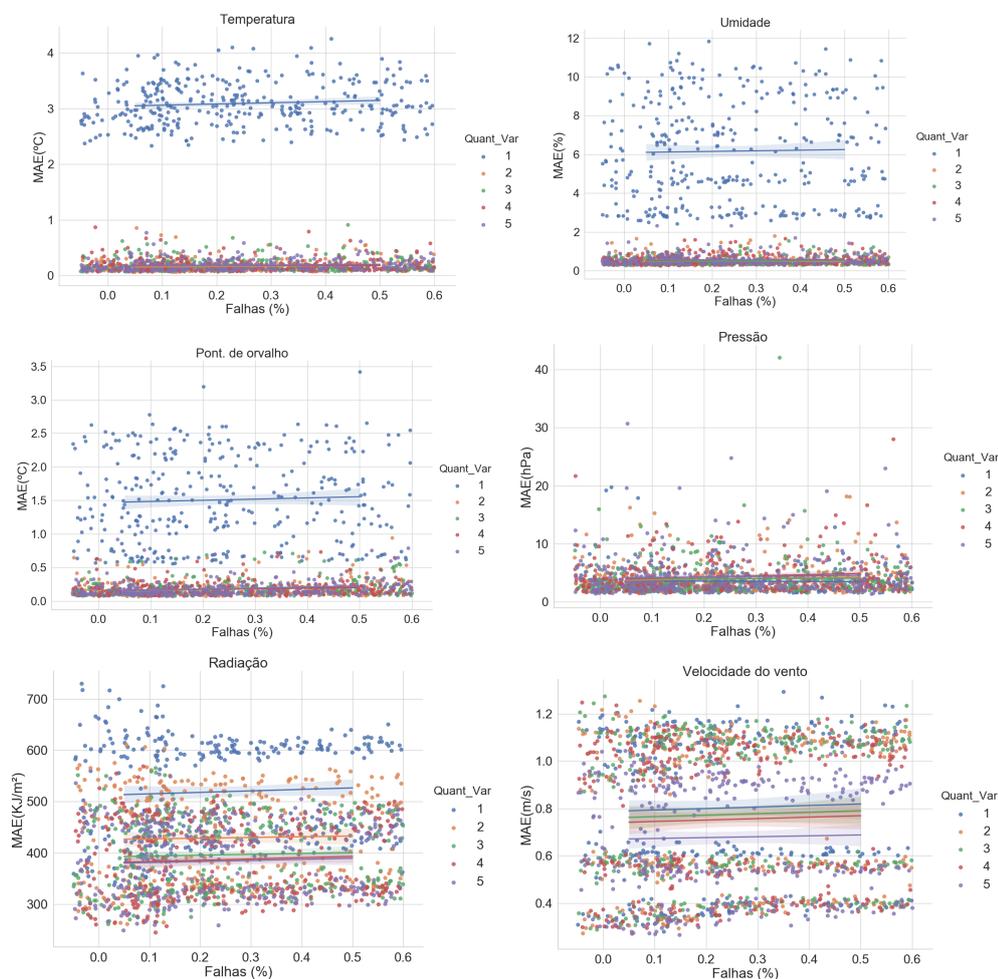


Fonte: do autor

Na Figura 15 é possível constatar o comportamento da média do MAE em relação a quantidade de variáveis independentes utilizadas no preenchimento e a porcentagem de falhas presentes nos dados. Evidenciando que os preenchimentos que utilizaram somente uma variável independente, não foram capazes de uma boa aproximação com o valor esperado, e algumas vezes deslocaram a média para cima, como no caso das variáveis temperatura do ar, umidade, ponto de orvalho e radiação.

Para a pressão do ar, a quantidade de variáveis independentes indicou não demonstrar variância na média. Nota-se que no gráfico da velocidade do vento, que a utilização das cinco variáveis independentes obteve melhores resultados.

Figura 15: Relação entre a quantidade de falhas e o MAE para cada variável de estudo, indicando o comportamento da média e mediana.

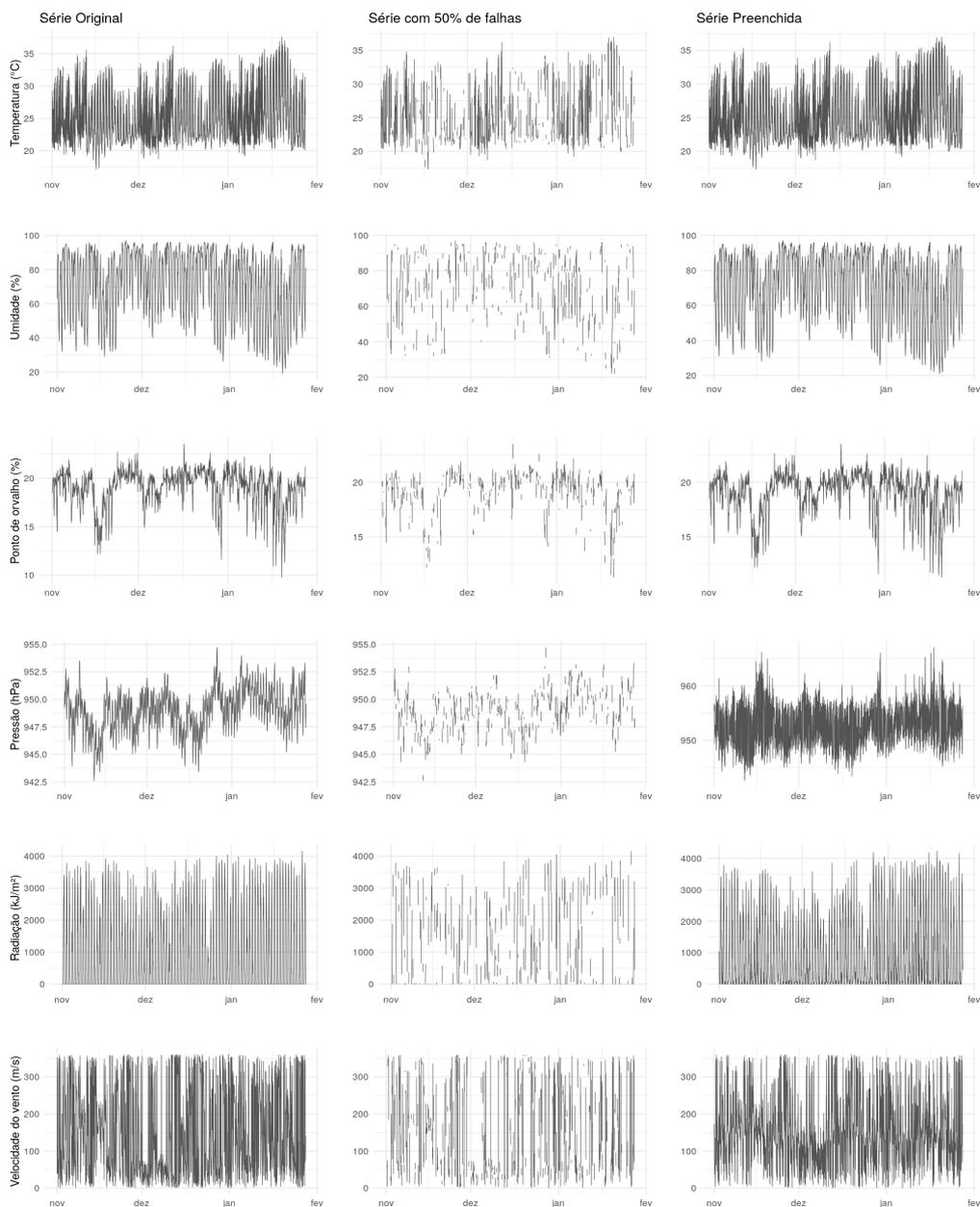


Fonte: do autor

A Figura 16 ilustra as séries temporais, das variáveis microclimáticas, da cidade de Sorriso, apresentadas em três diferentes momentos. No primeiro da

esquerda para direita, a série está em sua totalidade, ou seja, sem a presença de falhas. No segundo apresenta 50% de falhas nos dados e no último momento apresenta os dados após o processo de preenchimento efetuado pelo o modelo de DL.

Figura 16: Séries temporais das variáveis de estudo (temperatura do ar, umidade relativa, ponto de orvalho, pressão, radiação e velocidade do vento) coletadas da estação do INMET, localidade Sorriso- MT, no período de novembro/2014 à janeiro/2015. Ilustradas: na forma original, com 50% de falhas e após o processo de preenchimento de falhas.



Fonte: do autor

4.1.1.1 Análise Estatística

De forma geral, considerando os valores do MAE das cinco regiões, para analisar a quantidade de variáveis independentes utilizadas no preenchimento falhas, a configuração da Tabela 8 foi adotada.

Tabela 8: Agrupamento de acordo com a quantidade de variáveis independentes utilizadas no preenchimento de falhas.

Grupos de variáveis independentes	
Grupos	Quantidade de variáveis independentes
I	1
II	2
III	3
IV	4
V	5

O teste de Kruskal-Wallis acusou evidências de diferença significativa entre os valores de MAE, apresentando p-valor $< 0,0001$. O teste de Nemenyi revelou que em todos os cinco grupos, apenas o Grupo I apresentou diferença significativa em relação a todos os outros grupos. Na análise geral os testes de comparação de distribuição não detectaram diferença significativa para a quantidade de neurônios e porcentagem de falhas presentes na série.

Continuando com a análise geral, considerando dessa vez a quantidade de camadas agrupadas conforme a Tabela 9. O teste de Mann-Whitney acusou diferença significativa entre as distribuições apresentando p-valor $< 0,0001$.

Tabela 9: Agrupamento de acordo com a quantidade de camadas utilizadas no preenchimento de falhas.

Grupos divididos por quantidade de camadas	
Grupos	Quantidade de camadas ocultas
I	2
II	4

Na análise por variável, considerando apenas o valor do MAE em diferentes profundidades da rede neural (Tabela 9). O teste de Mann-Whitney detectou diferença significativa em todas as variáveis, exceto para a velocidade do vento, conforme informações da Tabela 10.

Tabela 10: Valores de p-valor para o Teste de Mann-Whitney individual para cada variável, comparando o valor do MAE e a profundidade da rede.

Teste de Mann-Whitney para Profundidade da Rede	
Variáveis	P-valor
Temperatura	5,63E-09
Umidade	1,004E-13
Ponto de orvalho	0,0008E-2
Pressão	2,2E-16
Radiação	0,0005E-3
Velocidade do vento	0,9852

O mesmo teste foi feito para a quantidade de neurônio, apresentando valores de p-valor significativos para as variáveis de umidade do ar, ponto de orvalho e pressão, conforme a Tabela 11.

Tabela 11: Valores de p-valor para o Teste de Mann-Whitney individual para cada variável, comparando o valor do MAE e a quantidade de neurônios.

Teste de Mann-Whitney para Quantidade de neurônios por camada	
Variáveis	P-valor
Temperatura	0,0673
Umidade	0,912E-05
Ponto de orvalho	0,001174
Pressão	9,525E-06
Radiação	0,201
Velocidade do vento	0,8436

Para a porcentagem de falhas presente na série, o teste de Kruskal-Wallis acusou variância significativa apenas para o ponto de orvalho (p-valor $< 0,0001$). Que após verificado com o teste de Nemenyi, que comparou as distribuições dos grupos da Tabela 12, apresentou evidências de diferença significativa entre o Grupo I e IV (p-valor $< 0,0001$), e Grupo II e IV com p-valor de 0,0003.

Tabela 12: Agrupamento de acordo com a quantidade de falhas presente nos dados de treinamento.

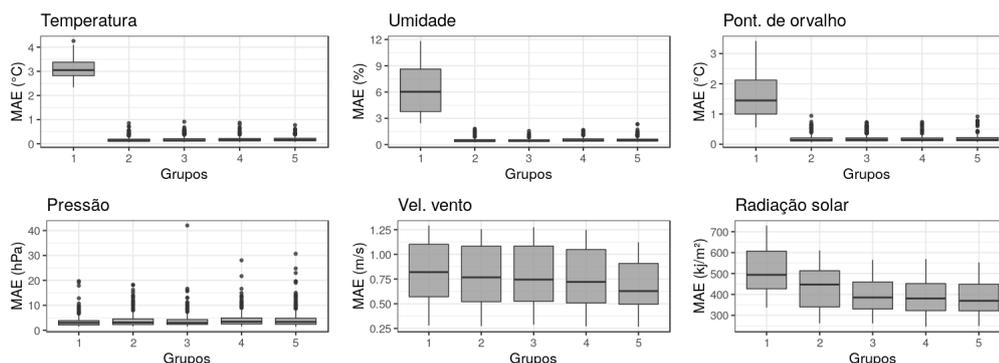
Grupos divididos por quantidade de falhas presentes na série	
Grupos	Falhas (%)
I	5
II	15
III	30
IV	50

Ainda na análise por variável, o teste de Kruskal-Wallis para a quantidade de variáveis independentes (X) utilizadas no preenchimento, apresentou variância significativa para todos os grupos da Tabela 8. O teste de Nemenyi revelou que em todas as variáveis, o Grupo I, que representa o preenchimento apenas com uma variável independente, apresentou diferença significativa em relação a todos os outros grupos, exceto para pressão atmosférica na qual apresentou diferença

entre o Grupo I e IV, e Grupo I e V, e para a velocidade do vento que apresentou diferença significativa entre o Grupo V em relação a todos os outros.

A radiação também apresentou diferença significativa entre os Grupos II e III, II e IV, II e V. A figura 17 contém informações sobre a distribuição dos dados para cada grupo.

Figura 17: Distribuição dos valores do MAE para os grupos: I, II, III, IV e V (Tabela 8).



Fonte: do autor

4.1.1.2 Tempo de Execução

O tempo de execução necessário para a realização de todos os testes revelou que a quantidade de falhas presentes na série é inversamente proporcional ao tempo de execução, pois, o aumento de falhas corresponde a diminuição da quantidade de dados. A Tabela 13 contém informações referentes a todas as modalidades de testes efetuadas:

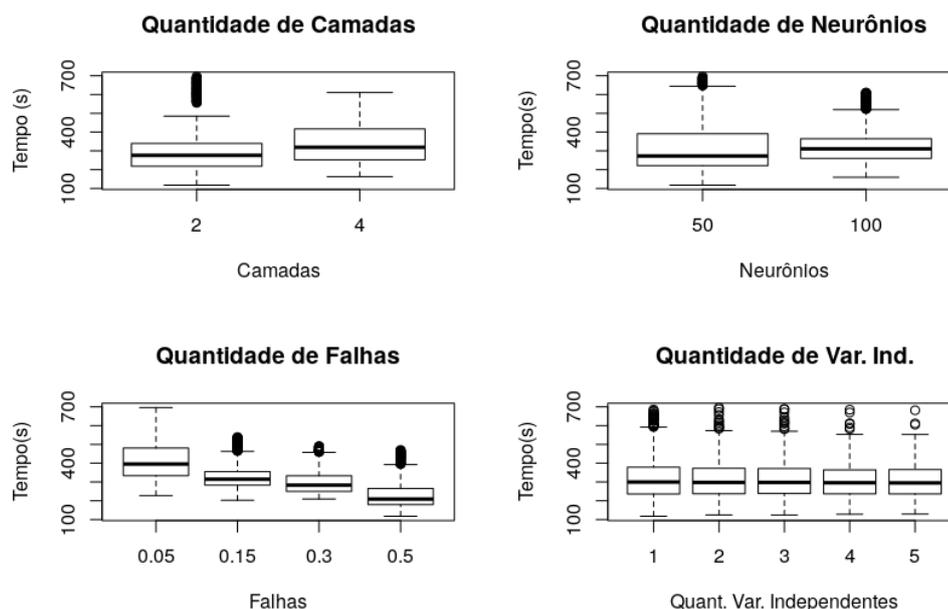
Tabela 13: Tempo de execução por teste.

Testes	Tempo (s)
2 camadas	290,3641
4 camadas	337,5736
50 neurônios	311,0778
100 neurônios	320,0282
5% de falhas	408,9449
15% de falhas	327,4024
30% de falhas	300,6253
50% de falhas	228,5328
Com 1 var. ind.	316,2971
Com 2 var. ind.	315,2982
Com 3 var. ind.	315,3667
Com 4 var. ind.	312,6333
Com 5 var. ind.	314,8645

onde o Tempo corresponde a soma da etapa de treinamento e validação de cada configuração utilizada no presente trabalho.

A análise de variância revelou existir diferenças significativas no tempo de execução em relação a quantidade de camadas ocultas (Tabela 9), sendo que ao aumentar a quantidade de camadas eleva-se o tempo de execução. Da mesma forma, a quantidade de neurônios utilizada por camada oculta também mostrou diferença significativa entre os grupos definidos na Tabela 12. Já para a quantidade de variáveis independentes (Tabela 8) não houve variância entre os grupos.

Figura 18: Distribuição do tempo médio de execução.



Fonte: do autor

Considerando o modelo treinado, ou seja, com os pesos definidos, para o preenchimento de falhas DL gasta em média 40 segundos. O tempo aproximado de execução total do experimento foi aproximadamente de 861,6 horas.

4.2 Regressão Linear Múltipla e Média Móvel

4.2.1 Regressão Linear Múltipla

O Código 4.2 foi desenvolvido para o preenchimento de falhas utilizando a RLM, assim como no modelo de DL também permitiu analisar a quantidade de falhas presentes nos dados.

Código 4.2: Regressão Linear Múltipla

```

1 ## Biblioteca utilizada ##
2 from sklearn.linear_model import LinearRegression
3
4 ## Modelo ##
5 regr = LinearRegression() # cria o modelo
6 regr.fit(X_train, y_train) # treina o modelo utilizando a
   reamostragem de treino
7
8 ## Valor previsto ##
9 ypred = regr.predict(X_test) # testa o modelo utilizando a
   reamostragem de teste

```

A Tabela 14 contém informações sobre a avaliação do preenchimento de falhas utilizando RLM, assim como para o modelo DL, o estimador de erro MAE foi utilizado. Os valores são resultantes da média das quatro regiões. O modelo apresentou bons resultados, com ênfase nos obtidos da pressão atmosférica. Os dados tiveram pouca variabilidade.

Tabela 14: Média e desvio padrão(σ) dos resultados do MAE para testes realizados com RLM, no preenchimento de falhas para as variáveis: temperatura do ar (T), umidade do ar (U), ponto de orvalho (PO), pressão atmosférica (P), radiação (R_g) e velocidade do vento (V_v)

Falhas	MAE (RLM)											
	T (°C)		U (%)		PO (°C)		P (hPa)		R_g (kJ/m ²)		V_v (m/s)	
	\bar{t}	σ	\bar{u}	σ	\bar{p}_o	σ	\bar{p}	σ	\bar{r}_g	σ	\bar{v}_v	σ
5%	0,9827	1,0325	2,9126	2,3543	0,7249	0,5980	1,7788	0,3700	532,6423	113,7615	0,8107	0,2758
15%	1,0424	1,1226	2,9266	2,3014	0,7445	0,5962	1,8090	0,4165	521,6186	82,4899	0,7744	0,3080
30%	1,0304	1,1007	2,9010	2,2559	0,7313	0,5824	1,8760	0,4917	531,6042	85,5808	0,8022	0,3184
50%	1,0187	1,0907	2,8532	2,1986	0,7239	0,5699	1,8759	0,4867	524,5980	88,4295	0,7975	0,3168

A quantidade de falhas presente nos dados não teve impacto sobre os resultados. O tempo médio de execução foi aproximadamente de 3 minutos e 54 segundos.

4.2.2 Média Móvel Simples

Por se tratar de um método univariado, a Média Móvel Simples consiste em uma alternativa simples e rápida. A grande dificuldade de aplicação desse método é encontrar o tamanho da janela a ser utilizada. Os teste efetuados neste trabalho, adotaram uma janela de duas medições, uma medição antes de ocorrer a falha e outra após a falha, uma vez que os dados do INMET são horários.

Os resultados obtidos dos testes com média móvel podem ser encontrados na Tabela 15.

O resultados demonstraram um bom resultado para a pressão atmosférica e velocidade do vento, e uma sensibilidade para a presença de falhas. Quanto

Tabela 15: Média e desvio padrão(σ) resultados do MAE para testes realizados com média móvel, no preenchimento de falhas para as variáveis: temperatura do ar (T), umidade do ar (U), ponto de orvalho (PO), pressão atmosférica (P), radiação (R_g) e velocidade do vento (V_v)

MAE (Média Móvel)												
Falhas	T (°C)		U (%)		PO (°C)		P (hPa)		R_g (kJ/m ²)		V_v (m/s)	
	\bar{t}	σ	\bar{u}	σ	\bar{p}_o	σ	\bar{p}	σ	\bar{r}_g	σ	\bar{v}_v	σ
5%	1,2457	0,1325	5,1844	0,6732	0,3917	0,0968	0,5664	0,0438	387,5637	24,4697	0,3607	0,1409
15%	1,3770	0,1274	5,6829	0,5870	0,4283	0,0897	0,6526	0,0545	452,7047	25,1514	0,3846	0,1870
30%	1,2456	0,0738	5,3295	0,3661	0,5158	0,0975	0,7482	0,1130	382,8285	30,3198	0,4092	0,1622
50%	1,7233	0,2053	7,2143	1,0401	0,6476	0,1012	0,9548	0,1155	565,5915	47,6890	0,6144	0,2228

maior a quantidade de falhas melhor foi o resultado. Esse comportamento não é ao todo anômalo, pois sugere que a janela escolhida não é a melhor indicada, uma vez que ao se aumentar as lacunas de dados, indiretamente ocorre um aumento na janela adotada.

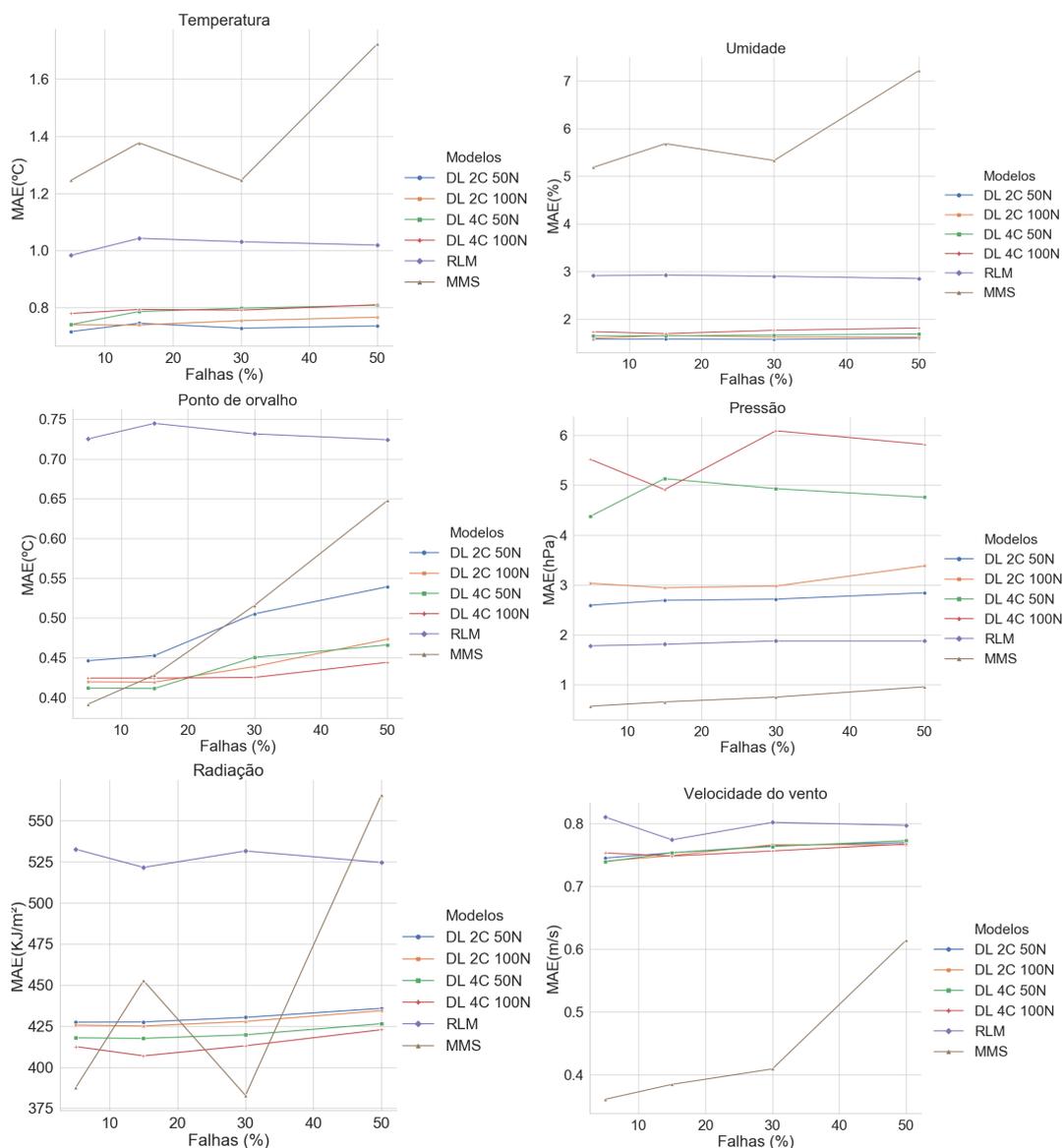
O preenchimento evidenciou uma série com pouca variação, por ser construída em torno da tendência central próxima, o que sugere apenas uma aproximação do valor real e não uma replicação do comportamento do mesmo. O tempo médio de execução foi aproximadamente de 58 segundos.

4.3 Comparativo entre os Métodos

Com a obtenção de todos os dados, foi possível fazer um comparativo entre os três métodos utilizados, utilizando o valor do MAE.

A Figura 19 representa o comparativo entre os três métodos para o preenchimento de falhas, com diferentes quantidades de falhas, para cada variável.

Figura 19: Comparativo entre DL, RLM e média móvel para o preenchimento de falhas nos dados com a presença de diferentes quantidades de falhas para as variáveis: temperatura do ar, umidade relativo do ar, ponto de orvalho, pressão atmosférica, radiação solar e velocidade do vento.



Para a temperatura o modelo de DL mostrou excelentes resultados, a configuração com 2 camadas ocultas com 50 neurônios obteve os menores resultados do MAE, com uma discrepância de aproximadamente média de $0,35^{\circ}\text{C}$ em relação ao método de RLM, e $\approx 1,3^{\circ}\text{C}$ em comparação com a média móvel. Também não aparentou sofrer impacto com a variação da quantidade de falhas presente na série, como já descrito nas análises estatísticas. Assim como a temperatura, o modelo de DL apresentou excelentes resultados para umidade relativa do ar e ponto de orvalho.

No preenchimento de falhas em séries temporais de pressão atmosférica, o modelo de DL não obteve o melhores resultados, ficando abaixo da média móvel

e RLM. Para velocidade do vento a média móvel apresentou o menor índice de erro, o que não é ao todo inesperado, pois valores mais próximos da falha descreve melhor esse fenômeno. Para a radiação o modelo obteve resultados medianos.

Em geral o modelo de média móvel adotado demonstrou uma grande sensibilidade a quantidade de falhas contidas na ST, podendo gerar variações bruscas nos resultados. Já o modelo de RLM adotado obteve um melhor preenchimento com relação ao modelo DL apenas para a ST de pressão atmosférica.

4.4 Discussão dos Resultados

O modelo criado desempenhou bem sua função de preenchimento de falhas, apresentando para quatro das seis variáveis um bom ajuste. Para formulação dos resultados, contou com 10.050 testes, sendo 2.511 por região e 1675 por variáveis. Proporcionando assim uma análise sólida sobre quatro aspectos básicos:

- Profundidade da rede neural: O estudo considerou duas configurações, a primeira configuração com duas camadas ocultas e a segunda com o dobro de camadas. Os resultados apontaram que a profundidade tem impacto direto, na resposta da rede, para o preenchimento de falhas em séries temporais de cinco variáveis (temperatura do ar, umidade relativa do ar, radiação, pressão e ponto de orvalho). O aumento de camadas, em alguns casos, não demonstrou melhorias na resposta da rede. Esse efeito, supostamente pode estar ligado ao uso fixo dos parâmetros externos da rede, como por exemplo a taxa de aprendizado e épocas. Talvez ao se elevar a quantidade de camadas ocultas, paralelamente deve-se considerar alterações na totalidade da estrutura do modelo de DL.
- Quantidade de falhas presentes na série temporal: Com exceção das séries temporais de ponto de orvalho, não houve impacto no preenchimento de falhas em dados contendo entre 5% à 50% de falhas. O que demonstra um fator positivo, que caracteriza a DL, que é o fato dela aprender com os dados, ou seja, isso prova que o modelo foi capaz de entender o comportamento da variável e replica-lo em sua resposta, estimando a variável pretendida com base nos dados de outras variáveis.
- Quantidade de variáveis independentes: O estudo mostrou que somente uma variável independente não é suficiente para o aprendizado da rede. Os teste utilizaram até cinco variáveis, mostrando que entre 2 e 5 não há diferenças significativas.

- Quantidade de neurônios: O estudo considerou 50 ou 100 neurônios por camada oculta, os resultados mostraram haver diferença significativa entre os dois grupos, para as variáveis umidade relativa do ar, ponto de orvalho e pressão atmosférica.

No trabalho Ventura (2015), no qual foi criado um framework, em que umas das funções é o preenchimento de falhas utilizando redes neurais, o mesmo obteve melhores resultados na estimativa da radiação, o que pode ser devido ao uso da informação horária de cada medição, uma vez que a radiação sofre atenuação devido ao ângulo horário.

Outros métodos, comumente utilizados na literatura, como a regressão linear múltipla e média móvel, mostraram desempenho inferior, exceto o segundo que obteve melhores resultados na estimativa da velocidade do vento e pressão atmosférica, levando em consideração que o mesmo é um método univariado e que operou com uma janela de duas medições após a falha, o que para fenômenos que ocorrem em curto prazo, pode acarretar em uma representação mais próxima da realidade. Outro ponto a ser considerado é que os modelos presentes na literatura apresentaram baixa variação dos dados. O trabalho desenvolvido por Rihbane (2018), defende um preenchimento de falhas no qual ocorre a preservação da variância da série, pois dessa forma a mesma não se distanciaria do real comportamento do fenômeno. Em uma análise superficial o modelo de DL apresentou uma maior variação, mas se faz necessário um aprofundamento maior, para fazer um comparativo mais coerente neste sentido.

DL apresentou como desvantagem o tempo de execução, o que pode ser um agravante para a sua aplicação, já que os outros métodos testados no presente trabalho mostraram uma execução de extrema rapidez quando comparados.

Portanto, a utilização de deep learning no preenchimento de falhas em dados microclimáticos, provou ser uma área promissora, que atende as expectativas, podendo evoluir ao longo do tempo e que deve ser explorada em novos trabalhos.

Capítulo 5

CONSIDERAÇÕES FINAIS

Falhas presentes nos dados, se não preenchidas, podem comprometer a informação à ser minerada, levando a suposições que muitas vezes se distanciam da realidade do fenômeno medido. A importância da resolução desse problema motivou a realização deste trabalho.

Na busca de novos métodos de preenchimento de falhas, o presente estudo foi pautado exclusivamente na análise de um proposto modelo de DL, fundamentada nos resultados de diversas configurações possíveis e comparação com outros modelos presentes na literatura.

Os resultados não apenas mostraram uma boa estimativa das variáveis alvo, mas também proporcionaram uma análise da composição estrutural do modelo. É possível inferir que a rede neural criada, produto deste estudo, consegue executar a tarefa de preenchimento de falhas em séries temporais microclimáticas, obtendo melhor desempenho em três das variáveis testadas, em comparação a dois modelos comumente utilizados na literatura, porém, apresentou ser custosa em relação ao tempo de execução.

5.1 Trabalhos Futuros

Nem todos os aspectos da rede neural foram explorados. Existem inúmeras combinações a serem estudadas, que podem trazer melhorias para o preenchimento, a exemplo disso o estudo das configurações dos parâmetros externos e seus impactos. Fatores como este podem servir como sugestão para trabalhos futuros e serem melhores investigados.

Outra proposta é a investigação da utilização da quantidade estimada de testes para cada variável. Embora exija um alto custo computacional e tempo de execução, traria informações importantes para futuras aplicações de DL na área

de Ciências Ambientais.

REFERÊNCIAS

AGGARWAL, C. C. *Neural networks and deep learning*. Yorktown Heights, NY, USA: Springer, 2018. 497 p. Citado na página 15.

AREL, I.; ROSE, D. C.; KARNOWSKI, T. P. et al. Deep machine learning—a new frontier in artificial intelligence research. *IEEE computational intelligence magazine*, The University of Tennessee, v. 5, n. 4, p. 13–18, 2010. Citado na página 16.

AUSTIN, P. C.; LAUPACIS, A. A tutorial on methods to estimating clinically and policy-meaningful measures of treatment effects in prospective observational studies: a review. *The international journal of biostatistics*, De Gruyter, v. 7, n. 1, p. 1–32, 2011. Citado na página 5.

BABA, R. K.; VAZ, M. S. M. G.; COSTA, J. da. Correção de dados agrometeorológicos utilizando métodos estatísticos. *Revista Brasileira de Meteorologia*, SciELO Brasil, v. 29, n. 4, 2014. Citado na página 5.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. Citado na página 16.

BENGIO, Y. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, Now Publishers, Inc., v. 2, n. 1, p. 117, 2009. Citado na página 13.

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 17.

BIRNBAUM, Z. W. On a use of the mann-whitney statistic. 1956. Citado na página 25.

BONACCORSO, G. *Machine Learning Algorithms*. Birmingham – Mumbai: Packt Publishing Ltd, 2017. 331 p. Citado na página 17.

BROWNLEE, J. *Machine Learning Mastery with Python*. Melbourne, Australia: Machine Learning Mastery Pty Ltd, 2016. 256 p. Citado na página 17.

CARPENTER, G. A.; GROSSBERG, S.; MARKUZON, N.; REYNOLDS, J. H.; ROSEN, D. B. Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on neural networks*, IEEE, v. 3, n. 5, p. 698–713, 1992. Citado na página 12.

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 4.

CHACÓN, R. B. *Automatic problem-specific hyperparameter optimization and model selection for supervised machine learning*. Tese (Doutorado), 2014. Citado na página 13.

CHAN, Y.; WALMSLEY, R. P. Learning and understanding the kruskal-wallis one-way analysis-of-variance-by-ranks test for differences among three or more independent groups. *Physical therapy*, Oxford University Press, v. 77, n. 12, p. 1755–1761, 1997. Citado na página 25.

CHATFIELD, C. *The analysis of time series: an introduction*. New York: CRC press, 1996. 352 p. Citado na página 4.

CHIBANA, E. Y.; FLUMIGNAN, D.; MOTA, R. G. Estimativa de falhas em dados meteorológicos. *Congresso Brasileiro de Agroinformática*, v. 9, 2005. Citado na página 4.

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 11.

DENG, L.; YU, D. et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, Now Publishers, Inc., v. 7, n. 3–4, p. 197–387, 2014. Citado 2 vezes nas páginas 14 e 16.

DIETTERICH, T. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, ACM, v. 27, n. 3, p. 326–327, 1995. Citado na página 16.

ELLIOTT, A. C.; HYNAN, L. S. A sas® macro implementation of a multiple comparison post hoc test for a kruskal–wallis analysis. *Computer methods and programs in biomedicine*, Elsevier, v. 102, n. 1, p. 75–80, 2011. Citado na página 25.

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 12.

GARRETA, R.; MONCECCHI, G. *Learning scikit-learn: machine learning in python*. Birmingham – Mumbai: Packt Publishing Ltd, 2013. 100 p. Citado na página 22.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. Massachusetts: MIT press Cambridge, 2016. v. 1. Citado 3 vezes nas páginas 13, 14 e 16.

GRAVES, A.; MOHAMED, A.-r.; HINTON, G. Speech recognition with deep recurrent neural networks. In: IEEE. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vancouver, 2013. p. 6645–6649. Citado na página 16.

GULLI, A.; PAL, S. *Deep Learning with Keras*. Birmingham – Mumbai: Packt Publishing Ltd, 2017. Citado 2 vezes nas páginas 10 e 11.

HAIR, J. F. J.; BLACK, W. C.; BABIN, B. J.; ANDERSON, R. E. *Multivariate data analysis*. 7. ed. Edinburgh Gate Harlow, Reino Unido: Pearson Education Limited, 2014. 734 p. Citado na página 5.

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 18.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, MIT Press, v. 18, n. 7, p. 1527–1554, 2006. Citado na página 13.

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 4.

HUANG, F.; LECUN, Y. Large-scale learning with svm and convolutional netw for generic object recognition. 2006. Citado na página 14.

IDRIS, I. *Python data analysis*. Birmingham – Mumbai: Packt Publishing Ltd, 2014. 319 p. Citado na página 7.

INMET, N. T. N. S. Rede de estação meteorológicas automáticas do inmet. *Nota Técnica do INMET, Rio de Janeiro*, 2011. Acesso: 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 20.

JIANG, X.; PANG, Y.; LI, X.; PAN, J.; XIE, Y. Deep neural networks with elastic rectified linear units for object recognition. *Neurocomputing*, Elsevier, v. 275, p. 1132–1139, 2018. Citado na página 10.

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 4.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado na página 13.

KLEINBAUM, D.; KLEIN, M. Logistic regression: a self-learning text springer-verlag. *New York, NY*, 1994. Citado na página 13.

- KOTU, V.; DESHPANDE, B. *Data Science: Concepts and Practice*. second. Washington: Morgan Kaufmann, 2018. 568 p. Citado 2 vezes nas páginas 15 e 16.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015. Citado na página 14.
- LEVINE, D.; STEPHAN, D.; KREHBIEL, T.; BERENSON, M. Estatística-teoria e aplicações usando o microsoft excel em português. 3^a edição. *Rio de Janeiro: LTC, Edição, 819p*, 2005. Citado na página 6.
- LOESCH, C.; SARI, S. T. *Redes neurais artificiais: fundamentos e modelos*. Blumenau - Santa Catarina: EDIFURB, 1996. 166 p. Citado na página 9.
- LONG, J. R.; GREGORIOU, V. G.; GEMPERLINE, P. J. Spectroscopic calibration and quantitation using artificial neural networks. *Analytical Chemistry*, ACS Publications, v. 62, n. 17, p. 1791–1797, 1990. Citado na página 12.
- 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 23.
- 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 8.
- MISHRAA, S. K.; KUMARB, S.; KUMARC, S.; SINGHD, R. K. Optimization of process parameters for α -amylase production using artificial neural network (ann) on agricultural wastes. *Curr Trends Biotechnol Pharm*, v. 10, n. 3, p. 248–260, 2016. Citado na página 10.
- MITCHELL, T. M. *Machine learning (mcgraw-hill international editions computer science series)*. McGraw-Hill, 1997. Citado na página 8.
- 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 17.
- MOREIRA, D. A. *Administração da produção e operações*. 2. ed. São Paulo: Editora Saraiva, 2017. 217 p. Citado na página 7.
- MORETTIN, P. A.; TOLOI, C. *Análise de séries temporais*. Universidade de São Paulo: Edgard Blucher, 2006. 474 p. Citado na página 3.
- NEMENYI, P. Distribution-free multiple comparisons. In: INTERNATIONAL BIOMETRIC SOC 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210. *Biometrics*. Wahington, DC, 1962. v. 18, n. 2, p. 263. Citado na página 25.

NETO, P. L. d. O. C. Estatística. Edgard Blücher, São Paulo, n. 15, p. 266, 2002. Citado na página 6.

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 4.

PATTERSON, J.; GIBSON, A. *Deep Learning: A Practitioner's Approach*. San Francisco, CA: "O'Reilly Media, Inc.", 2017. 538 p. Citado 2 vezes nas páginas 12 e 14.

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 9.

RIHBANE, F. E. C. *Preenchimento de Falhas de Séries Micrometeorológicas Utilizando Técnicas Estatísticas Combinadas*. 71 p. Tese (Doutorado) — Programa de Pós-graduação em Física Ambiental–UFMT, Cuiabá, dezembro 2018. Citado 2 vezes nas páginas 5 e 40.

RITZMAN, L. P.; KRAJEWSKI, L. J.; LEE, J. *Administração da Produção e Operações; Tradução Roberto Galman, Revisão Técnica Carlos Eduardo Mariano da Silva*. São Paulo: São Paulo-Pearson Prentice Hall, 2004. Citado na página 5.

ROBERT, C. Machine learning, a probabilistic perspective. Taylor & Francis, 2014. Citado na página 7.

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 13.

ROYSTON, P. Approximating the shapiro-wilk w-test for non-normality. *Statistics and Computing*, Springer, v. 2, n. 3, p. 117–119, 1992. Citado na página 25.

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 12.

RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. England: Pearson Education Limited, 2016. 1132 p. Citado na página 9.

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 7.

SCHMIDT, A. F.; GROENWOLD, R. H.; KNOL, M. J.; HOES, A. W.; NIELEN, M.; ROES, K. C.; BOER, A. D.; KLUNGEL, O. H. Exploring interaction

effects in small samples increases rates of false-positive and false-negative findings: results from a systematic review and simulation study. *Journal of Clinical Epidemiology*, Elsevier, v. 67, n. 7, p. 821–829, 2014. Citado na página 5.

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 4.

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 4.

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 4.

VENTURA, T. M. *Preenchimento de falhas de dados micrometeorológicos utilizando técnicas de inteligência artificial*. 73 p. Dissertação (Mestrado) — Programa de Pós-graduação em Física Ambiental–UFMT, Cuiabá, 2012. Citado na página 5.

VENTURA, T. M. *Criação de um Ambiente Computacional para Detecção de Outliers e Preenchimento de Falhas em Dados Meteorológicos*. 96 p. Tese (Doutorado) — Programa de Pós-graduação em Física Ambiental–UFMT, Cuiabá, 2015. Citado na página 40.

WANG, Z.; BOVIK, A. C. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, IEEE, v. 26, n. 1, p. 98–117, 2009. Citado na página 22.

WASON, R. Deep learning: Evolution and expansion. *Cognitive Systems Research*, Elsevier, v. 52, p. 701–708, 2018. Citado na página 16.

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 4.

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 3.