Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Modelos de aprendizado profundo com transferência de aprendizagem para modelagem hidrológica

Henrique Colonese Echternacht

JUIZ DE FORA
JANEIRO, 2023

Modelos de aprendizado profundo com transferência de aprendizagem para modelagem hidrológica

Henrique Colonese Echternacht

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Luciana Conceição Dias Campos

Coorientador: Leonardo Goliatt da Fonseca

JUIZ DE FORA

JANEIRO, 2023

Modelos de aprendizado profundo com transferência de aprendizagem para modelagem hidrológica

Henrique Colonese Echternacht

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Luciana Conceição Dias Campos
Doutora em Engenharia Elétrica

Leonardo Goliatt da Fonseca
Doutor em Modelagem Computacional

Heder Soares Bernardino
Doutor em Modelagem Computacional

Alfeu Dias Martinho
Mestre em Estatística

JUIZ DE FORA
16 DE JANEIRO, 2023

Resumo

Devido ao crescente uso da inteligência artificial, torna-se cada vez mais necessário o estudo e o desenvolvimento de ferramentas eficazes para a solução dos mais diversos problemas como, por exemplo, a capacidade de prever a vazão de rios em bacias hidrográficas. Esse problema se torna relevante, uma vez que o uso dos rios tem impacto direto em diversas áreas da sociedade, como em transporte, agricultura e geração de energia. Nesse trabalho, é proposta a criação de um modelo de aprendizado profundo com transferência de aprendizagem que utiliza uma rede neural profunda, do tipo convolucional (CNN), para prever a vazão em de rios utilizando séries temporais hidrológicas. O modelo desenvolvido neste trabalho é capaz de prever a vazão do dia corrente com base no valor da vazão dos três ou sete últimos. Para isso, foram utilizas séries temporais contendo informações de duas bacias, sendo elas: bacia do Paraíba do Sul, localizado no Brasil; e bacia do Zambeze, localizado no continente africano. Como resultado, foi obtido uma redução de até 27% no tempo de treinamento perdendo somente 0.31% de desempenho para os dados do Paraíba do Sul. Já para os dados do Zambeze, foi obtida uma redução de até 48% do tempo de treinamento perdendo somente 2% de desempenho.

Palavras-chave: Redes Neurais Convolucionais, *Deep Learning*, *Transfer Learning*, Redes Neurais, Séries Temporais, Rio Paraíba do Sul, Rio Zambeze.

Abstract

Due to the rising application of artificial intelligence, it becomes more and more necessary the study and development of eficient tools to solve a variety of problems, such as the capacity to forecast rivers's streamflow inside river basins. This capacity reveals to be relevant once river's usage has direct impact in many society areas, as to mention transportation, agriculture and power generation. In this article, we propose the creation of a deep-learning model with transfer learning which uses a convolutional deep-neural network (CNN) to forecast rivers's streamflow through the use hydrological time series. The model developed here is, therefore, capable of forecast the river's streamflow one day in advance based on those of the last three or seven days. In order to do that, two time series have been used, which contained informations of two river's basins: Paraíba do Sul, in Brazil; and Zambebe, in the Africa. As a result, a reduction of up to 27% in training time was obtained, losing only 0.31% of performance for the data from Paraíba do Sul. As for the Zambezi data, a reduction of up to 48% of training time was obtained, losing only 2% of performance.

Keywords: Convolutional Neural Network, Deep Learning, Transfer Learning, Neural Networks, Time Series, Paraíba do Sul River, Zambezi River.

Agradecimentos

Gostaria de agradecer primeiramente a Deus por sempre me guiar no caminho certo e me fornecer o foco necessários para chegar até aqui. Agradeço aos meus pais pelo apoio e amizade, sem vocês com certeza não seria capaz de nada disso. As minhas avós, meus irmãos e companheira, que tanto me ajudaram nessa caminhada.

A todos os professores que tive contato ao longo dessa caminhada, especialmente a minha orientadora Luciana e coorientador Leonardo, por estarem sempre disponíveis para ajudar e ensinar.

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) pela bolsa de estudos e auxílio financeiro, que possibilitou a dedicação integral ao desenvolvimento deste trabalho de conclusão de curso e a operacionalização do estudo.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

DCC	Departamento de Ciência da Computução
UFJF	Universidade Federal de Juiz de Fora
ML	Machine Learning
TL	Trasnfer Learning
ANA	Agência Nacional das Águas
CNN	Convolutional Neural Networks
ANN	Artificial Neural Networks
MSE	Mean Squared Error
MAE	Mean Absolut Error
RMSE	Root Mean Squared Error
R^2	R^2 Score
LSTM	Long Short-term Memory

1 Apresentação do tema

Desde os primórdios da humanidade, os rios desempenham um papel essencial para o desenvolvimento das sociedades por meio de atividades fundamentais, tais como: agricultura, pecuária, abastecimento, saneamento básico, turismo e geração de energia. Segundo dados da EPE (empresa de pesquisa energética) (EPE, 2022), grande parte da energia elétrica gerada no Brasil vem de usinas hidrelétricas. Por esse motivo, o estudo dos rios torna-se relevante, sendo a capacidade de previsão de suas características, principalmente de sua vazão, um conhecimento de grande interesse.

A vazão é um dado de fundamental importância por estar conectada com quase todos os fatores ambientais, sociais e econômicos relacionados à água. O regime de vazão de um rio está ligado a alguns condicionantes (YASEEN et al., 2015), como a alteração no uso terra, por exemplo, que impacta diretamente na frequência e na intensidade das oscilações na vazão. Além disso, usinas hidrelétricas, em função do consumo energético, também podem bloquear ou liberar o volume de água, tal qual a captação de água para atividades agrícolas e industriais, afetando novamente a vazão (KELMAN, 2015).

Diante disso, as previsões a curto prazo são essenciais ao dia a dia em vários aspectos, como para determinar a vazão sob pontes, para a gestão de geração de hidrelétricas, para a irrigação de plantações, para a navegação fluvial e para a previsão de enchentes (LE et al., 2021) (EDOSSA; BABEL, 2011). Por isso, achar formas eficientes de modelar e compreender as complexas relações por trás das diversas variáveis envolvidas na vazão é de grande importância à formulação de previsões acuradas. Neste cenário, novos modelos com uso de inteligência artificial, em particular redes neurais de aprendizado profundo, surgem como uma boa alternativa para modelagem hidrológica, com potencial de gerar soluções acuradas para esse tipo de problema.

1.1 Contextualização

Devido ao avanço de tecnologias de monitoramento, hoje é possível a coleta intensiva de dados diários de uma bacia hidrográfica inteira, desde sua nascente até os menores afluentes. Como resultado, tem-se a produção de uma grande quantidade de dados históricos, que vão desde a simples medição da vazão até dados relativamente complexos, como a ocupação do solo e a quantidade de poluentes nas águas (PAULSON, 1999).

No contexo brasileiro, a Agência Nacional de Águas (ANA) é responsável pela coordenação da Rede Hidrometeorológica Nacional (RHN), um sistema que abriga 4.641 pontos de monitoramento no país (ÁGUAS, 2020). Eles são divididos em grupos: algumas estações monitoram parâmetros relacionados aos rios, como níveis, vazões, qualidade da água e transporte de sedimentos; outras, por sua vez, monitoram principalmente as chuvas. Essa grande quantidade de registros relativos às variáveis hidrológicas necessita de um acompanhamento contínuo para ser melhor aproveitada. Dessa forma, os registros históricos, antes mencionados, são contidos em séries temporais de fenômenos hídricos, os quais são valiosos para a construção de modelos que simulam o comportamento dos rios e de seus corpos de água (HERSCHY, 2014).

No contexto desse trabalho, são abordadas a Bacia hidrográfica do Zambeze, localizada na África Austral, e a Bacia hidrográfica do Paraíba do Sul, localizada no Brasil, sendo a previsão da vazão desses rios de muita importância para o desenvolvimento econômico e para a manutenção do bem-estar social local. Ademais, sua má gestão pode trazer consequências graves, como secas e inundações. Desse modo, sua previsão, com a melhor precisão possível, pode contribuir para a solução de uma grande quantidade de problemas que afetam a sociedade como um todo e melhorar o gerenciamento de recursos ambientais.

Em se tratando do âmbito global, esse trabalho pode contribuir, inclusive, para o cumprimento dos Objetivos de Desenvolvimento Sustentável sugeridos pela ONU e trabalhados localmente por seus parceiros no Brasil. São 17 objetivos ambiciosos e interconectados que abordam os principais desafios de desenvolvimento enfrentados por pessoas no Brasil e no mundo (UNICEF, 2015). Esse trabalho relaciona-se, em especial, com os seguintes objetivos:

- Fome zero, à medida que contribuirá para ganhos na agricultura e na pesca.
- Assegurar a disponibilidade e a gestão sustentável da água.
- Tornar as cidades e os assentamentos humanos inclusivos, seguros, resilientes e sustentáveis.

1.2 Descrição do Problema

A Bacia hidrográfica do Paraíba do Sul é de grande importância para o Brasil, uma vez que engloba os três maiores estados brasileiros em população: Rio de Janeiro, São Paulo e Minas Gerais. Além disso, está localizada entre os dois maiores polos industriais do país, as cidades de Rio de Janeiro e São Paulo, e comporta um complexo industrial com mais de 6000 fábricas e 120 usinas hidroelétricas em operação, que correspondem a cerca de 11% do Produto Interno Bruto (PIB) nacional (IORIS, 2008).

A bacia hidrográfica do Zambeze, por sua vez, localiza-se na região da África Austral e abrange o território de oito países: Angola, Zâmbia, Namíbia, Botswana, Zimbabwe, Tanzânia, Malawi e Moçambique. Sabe-se que a precipitação pluviométrica é o elemento mais crítico da região, onde a abundância ou insuficiência têm impacto direto. Em decorrência disso, a seca é a calamidade natural que mais afeta a bacia e, por isso, ela abriga algumas das maiores barragens do mundo: a barragem de Kariba, terceira maior do mundo; e barragem de Cahora Bassa, décima segunda maior do mundo.

Dito isso, entende-se quanto a carência de modelos de aprendizagem pode ser prejudicial, bem como a importância de abordar as bacias neste trabalho. Essa carência surge principalmente pela necessidade de monitoramento dos rios por um longo período de tempo, para obtenção de grandes quantidades de dados e o alto custo computacional de treinamento desses modelos que veem se tornando cada vez mais viável devido ao avanço computacional dos últimos anos.

Uma vez desenvolvido o modelo de aprendizagem, a previsão da vazão dos rios poderá beneficiar os países em diferentes âmbitos, sendo capaz de amenizar, ou mesmo solucionar, problemas hidrológicos e socioeconômicos.

Hoje, entretanto, existem diversas formas de se desenvolver um modelo de apren-

dizagem e, em se tratando do contexto hidrológico, é recente fazê-lo. Assim, é relevante determinar modelos de aprendizagem que apresentam bom desempenho na previsão da vazão de cada uma dessas bacias. Por isso, cada modelo proposto foi testado e avaliado até que fosse possível determinar o mais eficaz, para isso foram utilizadas diferentes métricas que serão detalhadas no capítulo 2.

1.3 Justificativa/ Motivação

Embora a quantidade de pesquisa voltada para modelos hidrológicos tenha crescido nos últimos anos, a literatura brasileira envolvendo inteligência artificial, em particular, modelos de aprendizado profundo para a previsão da vazão ainda carecem de mais pesquisas. Por este motivo, este trabalho tem como motivação o desenvolvimento de modelos de aprendizado profundo com transferência de aprendizagem com potencial de gerar resultados significativos para a pesquisa em Hidrologia.

Ademais, este trabalho propõe a aplicação de um tipo de modelo de aprendizado profundo específico, chamado Redes Neurais Convolucionais, mais conhecida na literatura como *Convolutional Neural Networks* (CNN) (KIRANYAZ et al., 2021). As CNNs possuem uma vasta gama de pesquisas voltadas para a área de classificação de imagens e reconhecimento facial; além disso, elas vêm demonstrando um bom desenpenho quando aplicadas a modelos de séries temporais, como em (SHEN; LIU; WU, 2020).

Aliado a isso, também tem-se como proposta a utilização de técnicas de transferência de aprendizagem, ou *Transfer Learning (TL)* (TAN et al., 2018), onde a CNN treinada em uma determinada base de dados é aplicada a dados de outro contexto, sem a necessidade de ser retreinada do zero.

Em suma, portanto, esse trabalho tem como motivação o desenvolvimento de modelos preditivos, usando redes neurais convolucionais e transferência de aprendizagem, voltados a aplicações de séries temporais hidrológicas.

1.4 Questões de Pesquisa

Esse trabalho tem como questão de pesquisa a comparação da eficiência do modelo de aprendizado profundo em três cenários de treinamento diferentes. No primeiro deles, a rede é treinada com dados das bacias em foco no trabalho. Já no segundo e terceiro, outra rede será criada e treinada com dados de contextos diferentes, sendo eles séries temporais de vazão da Bacia do Rio São Francisco, no Brasil, e séries temporais climáticas de Delhi, na Índia. Após isso, através de TL, reutilizaremos os modelos pré-treinados para aplicação no contexto das bacias de maior foco do trabalho. Com isso, visamos responder as seguintes perguntas: É possível obter modelos acurados e de baixo custo computacional utilizando TL? Como o contexto dos dados de pré-treinamento impacta no desempenho final do modelo?

1.5 Hipóteses de Pesquisa

Espera-se que o contexto dos dados que pré-treinaram o modelo e os que utilizarão esse conhecimento possuam uma relação direta na eficiência na transferência de aprendizagem em termos de desempenho. É provável que, quando o modelo pré-treinado é criado em um contexto similar ao modelo que irá receber sua aprendizagem, haja maior capacidade de transferência e aumento da eficiência final da rede em termos de desempenho.

1.6 Objetivos

Nesse trabalho, tem-se como objetivo principal o desenvolvimento de um modelo com transferência de aprendizagem utilizando CNNs. Desta forma, pretende-se conseguir métodos de previsão robustos que aproveitam as relações implícitas e padrões presentes nas séries temporais para gerar previsões acuradas.

Como objetivos secundários, temos:

- Avaliar o impacto do contexto dos dados de pré-treinamento do *transfer learning* na acurácia do modelo.
- Análise da economia de tempo e recurso computacional quando utilizado o *transfer*

learning.

Além disso, acredita-se que esse trabalho possa contribuir para:

- o desenvolvimento científico no cenário brasileiro.
- a sociedade no âmbito econômico, incluindo ganhos na navegação, agricultura e pesca.
- a sociedade no âmbito hidrológico, incluindo previsão de secas e enchentes.

1.7 Escopo da Pesquisa

O escopo da pesquisa teve como foco a criação de modelos eficazes utilizando TL, visando reduzir o alto custo computacional, o tempo gasto no treino e a necessidade de grandes quantidades de dados rotulados para treinamento. Para tal, fez-se o uso de duas séries temporais diferentes das em foco do trabalho descritas no capítulo 4.

Essas, foram utilizadas para pré-treinar os modelos usados para a transferência de aprendizagem. Já as séries estudadas no trabalho foram utilizadas para: treinar seus próprios modelos, usados como *baseline* dos resultados; receber através de TL os parâmetros dos modelos pré-treinados utilizando as bases do São Francisco e Delhi.

1.8 Organização do trabalho

Este trabalho está dividido em seis capítulos, organizados da seguinte maneira:

- Capítulo 1: Apresentação do problema trabalhado, relevância científica, metodologias usadas e resultados esperados.
- Capítulo 2: Apresentação de conceitos fundamentais para a compreensão do projeto, utilizados ao longo de seu desenvolvimento.
- Capítulo 3: Apresentação de trabalhos que abordam conceitos relacionados ao problema deste trabalho, mostrando seus objetivos, resultados e pontos positivos e negativos em relação ao problema aqui tratado.

- Capítulo 4: Apresentação e implementação do modelo de aprendizagem profunda aplicado, pré-processamento feito nos dados e definição dos hiperparâmetros.
- Capítulo 5: Apresentação e avaliação dos resultados obtidos durante a fase de testes.
- Capítulo 6: Apresentação das considerações finais do trabalho e informações sobre trabalhos futuros.

2 Fundamentação Teórica

2.1 Introdução

Este capítulo é destinado a introduzir os conceitos fundamentais e fornecer o conhecimento teórico necessário para o entendimento dos temas abordados nesse trabalho. Ele tem cinco seções: Redes Neurais Artificiais, Séries temporais, Convoluções matemáticas, Redes Neurais Convolucionais e Transferência de Aprendizagem.

2.2 Redes Neurais Artificiais

Dentro de inteligência artificial, o campo de redes neurais artificiais, mais referenciado na literatura como *Artificial Neural Network* (ANN), vem ganhando destaque em diversas aplicações, como reconhecimento de padrões, reconhecimento facial e predição (ABIODUN et al., 2018).

As ANNs, criadas como uma abstração do cérebro humano, são programas computacionais desenvolvidos para simular a maneira na qual o cérebro humano processa as informações. Desse modo, análogamente ao cérebro humano, elas são formadas por uma grande quantidade de componentes computacionais de processamento simples, denominados neurônios artificiais (JAIN; MAO; MOHIUDDIN, 1996).

Neurônio é o nome dado às principais células do cérebro humano, os quais são constituídos pelo corpo da célula, também chamado de soma, axônio e dendrito, apresentados na Figura 2.1. Seu funcionamento é descrito da seguinte maneira: os sinais enviados de outros neurônios são recebidos pelo dendrito, percorrendo todo o corpo do neurônio receptor e esse, por sua vez, repassa-os para o próximo neurônio através de seu axônio. A conexão feita entre dois neurônios é chamada de sinapse (ABIODUN et al., 2018).

Em se tratando das ANNs, é por meio da detecção de padrões, das relações existentes entre dados e da experiência obtida com eles que ocorre seu aprendizado ou treinamento, adquirindo seu conhecimento. Para isso, elas contam com os chamados

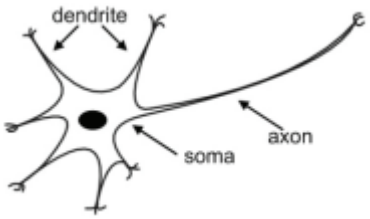

Figura 2.1: Neurônio cerebral (AGATONOVIC-KUSTRIN; BERESFORD, 2000)

neurônios artificiais, que têm como objetivo replicar a estrutura e o funcionamento de um neurônio cerebral (FENG; LU, 2019).

Assim, os neurônios artificiais são compostos de entradas, simulando os dendritos, e de uma saída, simulando o axônio, como mostrado na Figura 2.2. Além disso, cada neurônio possui uma função matemática, chamada de função de ativação, que determina o valor da saída do neurônio.

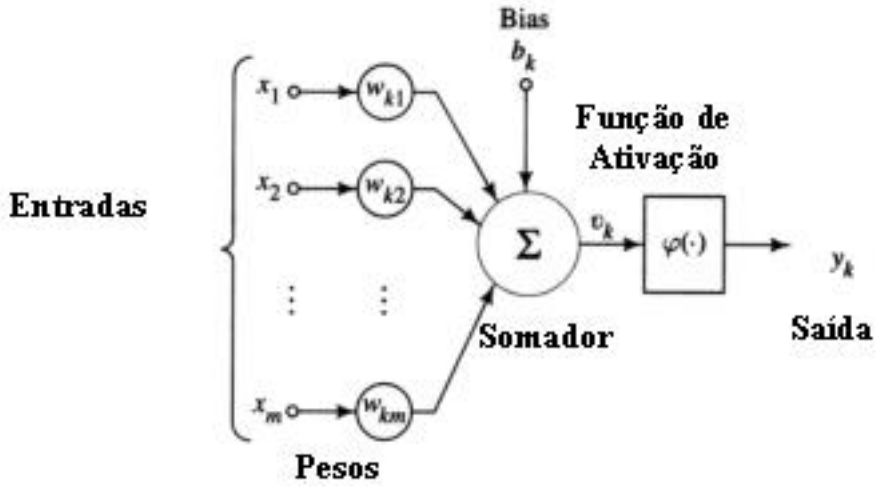

Figura 2.2: Neurônio artificial

Na prática, temos k camadas na rede, onde as entradas são representadas por x_j, os pesos de cada entrada por w_{kj}, o bias da camada como b_k e a saída do somador como v_k. Dessa forma, v_k é descrito na equação 2.1 pelo somatório da multiplicação das entradas pelos pesos somado ao bias.

$$\forall k, v_k = (\sum_{j=1}^{n} x_j w_{kj}) + b_k \tag{2.1}$$

Já as funções de ativação, um dos componentes mais importantes das ANNs, são funções matemáticas existentes em todos os neurônios e sua escolha tem impacto direto na performance da rede(KARLIK; OLGAC, 2011). Nelas, são inseridos valores das entradas ou da saída de neurônios da camada anterior multiplicados por seus pesos sinápticos. Em outras palavras, as funções são as responsáveis por transformar o nível de ativação de um neurônio em um sinal de saída.

Atualmente, uma das funções de ativação mais utilizadas nas CNNs é a unidade linear retificada, mais conhecida na literatura como *rectified linear unit* (RELU) (AGARAP, 2018). Seu funcionamento pode ser visto na Figura 2.3.

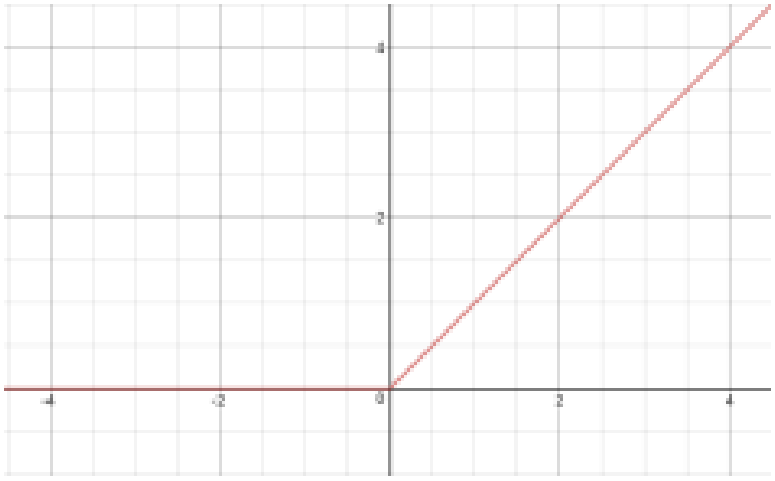

Figura 2.3: A função de ativação da unidade linear retificada (ReLU) produz 0 como saída quando x <0 e uma saída linear com inclinação de 1 quando x >0 (AGARAP, 2018)

Por fim, um conjunto de neurônios artificiais formam camadas, onde cada uma delas possui diferentes papéis, sendo que a união dessas camadas forma uma rede neural artificial. Vale lembrar, ainda, que esses conjuntos podem ser estruturados de diferentes maneiras, dando origem a diferentes tipos de redes neurais, com vantagens e desvantagens a depender do problema ao qual for aplicada (JAIN et al., 2014).

Pode-se dizer, portanto, que cada tipo de rede neural possui diferentes tipos de ligação entre suas camadas, interferindo diretamente na maneira pela qual a informação flui entre elas.

2.3 Séries Temporais

As séries temporais são conjuntos de observações armazenados de maneira específica, nas quais a ordem de acontecimento dos fatos é relevante. Em outras palavras, elas são uma sequência de observações tomadas sequêncialmente no tempo (BROWNLEE, 2017).

Por isso, é intrínseca a ela uma dependência entre a ordem de observação dos fatos, chamada de dimensão de tempo. Essa nova dimensão serve tanto como restrição do problema quanto como fonte de informações adicionais, que podem ser extraídas para auxiliar futuramente na previsão ou descrição do problema.

As séries temporais, para serem melhores compreendidas, são dividas em três principais componentes:

- Tendência

  Presente na maioria das séries temporais, determina o crescimento ou decrescimento da série em um período de tempo, apresentada na parte central da Figura 2.4.

- Sazonalidade

  Indica variações rítmicas que acontecem de forma regular e periódica dentro de um período de menos de um ano, apresentada na parte inferior da Figura 2.4.

- Ciclos

  Indica variações regulares que acontecem em um período de mais de um ano .

Dentre os ciclos, os principais são tendências e variações sazonais e, nesses, deve-se considerar, ainda, a alta correlação que tende a existir entre dados que ocorrem próximos cronologicamente.

Na Figura 2.4 está exemplificada a decomposição de uma série observada em suas componentes de tendência e sazonalidade.

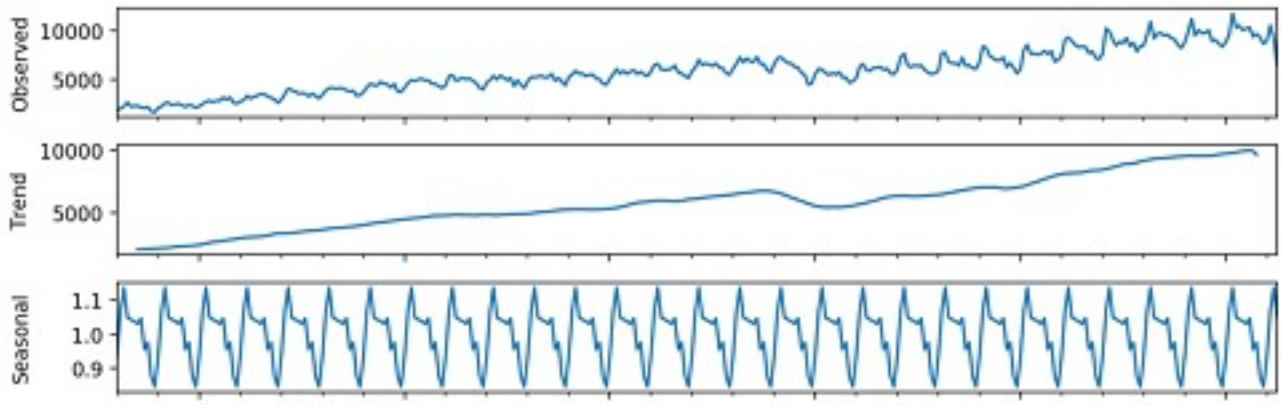

Figura 2.4: Decomposição da série temporal em Observada, Tendência, Sazonalidade (ANALYSIS; FORECASTING, 2022)

2.3.1 Análise de Séries Temporais

Há uma diferenciação quanto ao uso das séries temporais no que diz respeito ao que se espera extrair do conjunto de dados, que podem ser classificados como previsão ou descrição, modelagem ou extração de sinal (KITAGAWA, 2010).

O foco desse trabalho é na previsão de séries temporais. Quando o objetivo é fazer previsões sobre o comportamento futuro da série temporal ou somente sobre uma de suas informações, são usados modelos que, através dos dados históricos da série, conseguem prever observações futuras. Esses modelos podem ser acrescidos de informações adicionais, como outras séries temporais, para auxiliar nesse processo.

Outro fator importante a ser mencionado na previsão de séries temporais é a característica do futuro estar completamente indisponível, sendo possível apenas fazer estimativas baseadas em fatos já ocorridos. Nesse caso, a qualidade do modelo preditivo será determinada pelo desempenho de previsão da série temporal histórica (BOX et al., 2015).

2.3.2 Séries Temporais Hidrológicas

Neste trabalho, serão utilizadas séries temporais hidrológicas, as quais possuem tradicionalmente um conjunto de características bem definidas como: forte sazonalidade; duração de ciclos semelhantes; baixa tendência; e dados não lineares. Dessa forma, a exploração dessas características pode trazer bons resultados em questão de desempenho para os

modelos de aprendizado profundo.

2.4 Convolução

Na matemática, principalmente usada na análise funcional, convoluções são operações onde duas funções, $\boldsymbol{f}$ e $\boldsymbol{g}$, produzem uma terceira função $\boldsymbol{f*g}$, que expressa como o formato de $\boldsymbol{f}$ é modificado por $\boldsymbol{g}$ (WEISSTEIN, 2003). Essa nova função é definida pela integral do produto entre as duas funções iniciais, $\boldsymbol{f}$ e $\boldsymbol{g}$. Observe a Figura 2.5: em A temos as funções iniciais f e g; Em B, temos que a função g foi invertida; e em C, temos o deslocamento de g por f dada pela convolução $f*g$.

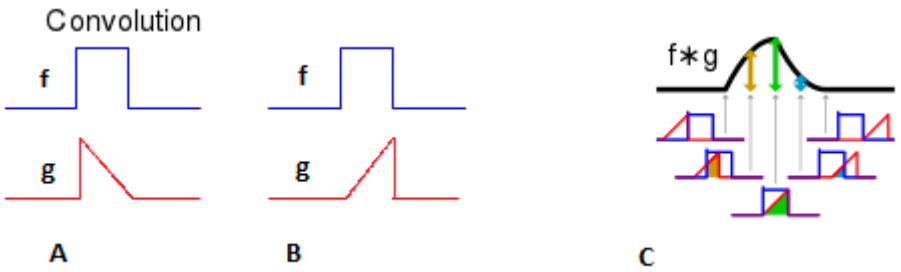

Figura 2.5: Convolução entre f e g

Na computação, a convolução pode ser aplicado em uma, duas ou três dimensões e é feito de maneira análoga. Neles, inicialmente temos uma matriz de entrada (mostrada em A na Figura 2.6), a qual, especificamente nas convoluções de 1 dimensão, é multiplicada por uma matriz de menor dimensão, chamada de *kernel* ou filtro, (mostrado em G na Figura 2.6).

Esse filtro é deslocado da esquerda para direita, de cima para baixo (mostrada na sequência A, B, C, D, E, F na Figura 2.6), em que cada multiplicação feita é somada e será atribuída a um campo da matriz de saída (mostrada em H na Figura 2.6). Nesse processo, a cada iteração, a área sombreada se desloca, multiplicando o kernel por uma área diferente da entrada e salvando o resultado na área sombreada da saída. Com isso, ao fim desse processo, temos a característica de ordenação dos dados preservada, e a saída obtida é o resultado da convolução entre a entrada e o *kernel*.

Além disso, outros dois parâmetros das convoluções de matrizes são o *padding* e

stride (MILLSTEIN, 2020). Esses, por sua vez, são importantes para definir a quantidade de iterações do processo e, como consequência, a dimensão da matriz de saída.

Assim, o valor definido para o *padding* indica se o *kernel* irá começar e parar nas bordas da matriz de entrada (*padding* = 0, demostrado na Figura 2.6 A, onde o kernel é posicionado na posição [0,0] da matriz), ou se irá começar $\boldsymbol{n}$ colunas antes e ainda irá se deslocar $\boldsymbol{n}$ colunas além da borda direita (*padding* = $\boldsymbol{n}$). Já o valor definido para o *stride* determina como será feito o deslocamento, e indica a quantidade $\boldsymbol{n}$ de colunas que o *kernel* irá se deslocar a cada iteração (na Figura 2.6, *stride* = 1, onde de A para B o kernel se deslocou uma coluna para a direita).

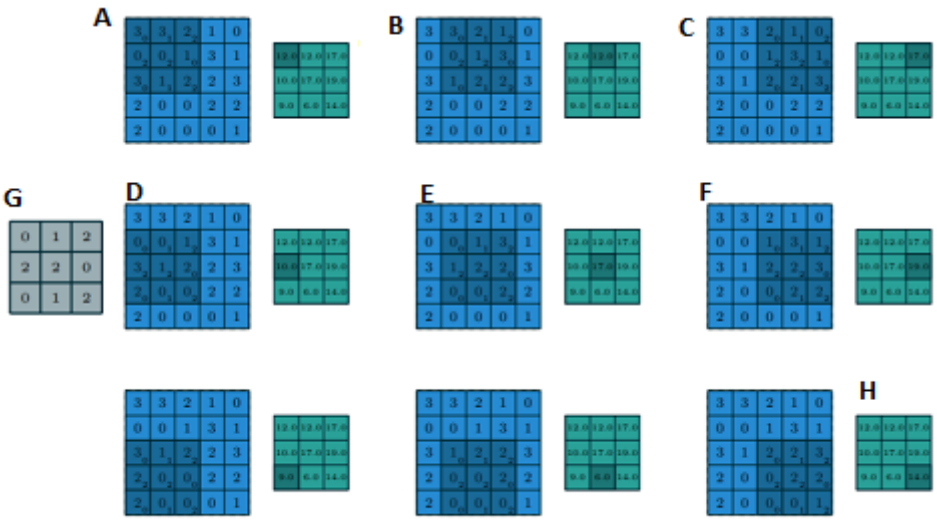

Figura 2.6: Processo de convolução em matrizes (DUMOULIN; VISIN, 2016)

Por fim, vale ressaltar que esse processo pode ser aplicado em qualquer tipo de valor de entrada, como imagens, séries temporais, ondas elétricas e sonoras etc, desde que tenham as seguintes características:

- Sejam armazenados em matrizes multidimensionais;
- Possuam um ou mais dimensões, nos quais a ordem cronológica de acontecimento dos dados é relevante;
- Possuam uma dimensão, chamada de eixo de canal, usada para acessar diferentes pontos de vista dos dados. Como exemplo, podem ser citadas as cores em imagens

coloridas, onde em uma única imagem temos uma dimensão para cada uma das 3 cores primárias, exemplificada na Figura 2.7 abaixo.

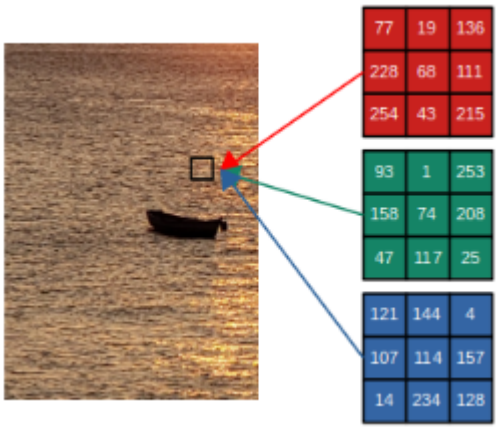

Figura 2.7: Ilustração da decomposição de imagens em matrizes das cores primárias (UFOP, 2022).

2.5 Redes Neurais Convolucionais

Vastamente usadas na área de reconhecimento de imagens, as Redes Neurais Convolucionais, mais referenciadas na literatura como *Convolutional Neural Network* (CNN) (AL-BAWI; MOHAMMED; AL-ZAWI, 2017), são uma arquitetura de aprendizado profundo, nas quais essencialmente são aplicadas convoluções recursivas para extração de padrões dos dados.

A sua criação foi inspirada no mecanismo de percepção visual natural dos seres vivos (GU et al., 2018) e sua origem foi baseada em diversos avanços científicos, desde o primeiro até o mais recente. O primeiro deles foi feito por Hubel & Wiesel, em 1959, no qual foi descoberto que, nos animais, o reconhecimento da imagem nas células do córtex cerebral se dava através da detecção de luz em campos visuais receptivos locais (HUBEL; WIESEL, 1968). O avanço mais recente, por sua vez, foi feito em 1990 por Le Cun; nele, foi publicado um artigo estabelecendo um framework para as CNNs, que, posteriormente, veio a ser melhorado, desenvolvendo uma ANN multi-camada, chamada de LeNet-5, capaz de classificar dígitos escritos a mão (LECUN et al., 1989).

Pode-se dizer que, desde então, as CNNs evoluíram bastante, em especial quanto

à ideia de campos receptivos locais. Eles são responsáveis pela capacidade dos neurônios de extraírem recursos visuais elementares nas imagens, como arestas, cantos e contornos. Assim, atualmente, são capazes de reconhecer não só digitos, como também textos escritos manualmente, objetos e faces (GU et al., 2018). Além disso, embora possuam grande eficiência na classificação de imagens visuais, as CNNs vêm obtendo bons resultados quando aplicadas ao campo das predições de séries temporais (ZHAO et al., 2017).

Além disso, vale-se ressaltar que as CNNs podem ser aplicadas em uma, duas ou três dimensões, nesse trabalho, serão aplicadas convoluções de uma dimensão, em que é necessária a transformação dos dados de entrada em sequências de 2 dimensões. Dessa maneira, as séries temporais abordadas serão convertidas em matrizes, nas quais uma dimensão indicará os valores de entrada e a outra os valores esperados, como na Figura 2.8, onde um vetor, de uma dimensão de tamanho sete, é transformado em um vetor de duas dimensões, composta pelos *Inputs* e *Labels*, no formato [6,1].

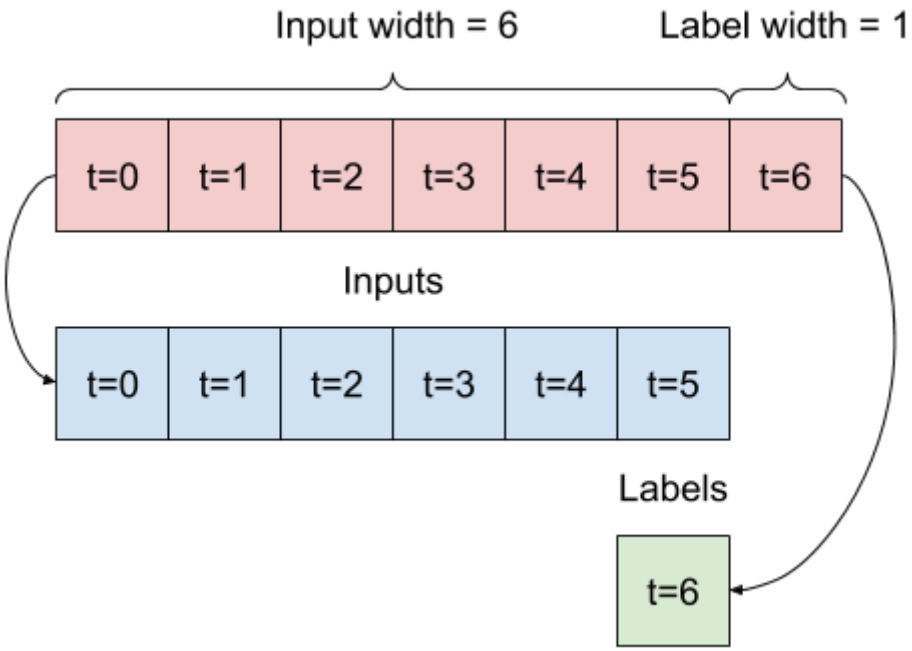

Figura 2.8: Transformação de uma única dimensão para duas dimensões (TENSORFLOW_DOCUMENTATION, 2022)

Após a transformação dos dados, sua arquitetura pode ser definida em cinco camadas (O'SHEA; NASH, 2015).

- Camada de *input*

De maneira semelhante a outras ANNs, é responsável pela entrada dos valores dos dados na rede e é exemplificada em A na Figura 2.11.

- Camada de Achatamento (*Flatten*)

  Como mostra a Figura 2.9, a camada *Flatten* tem como objetivo transformar vetores multidimensionais em unidimensionais, para que, assim, possam ser usados como entrada para outras camadas (AYYADEVARA, 2018).

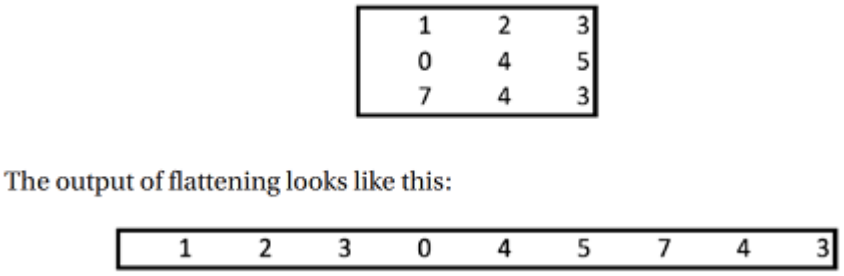

Figura 2.9: Exemplificação do processo ocorrido na camada *Flatten* (AYYADEVARA, 2018).

- Camada de convolução

  Exemplificada em B na Figura 2.11, é responsável pela aplicação do processo de convolução explicado anteriormente, onde a escolha dos hiper-parâmetros, como quantidade de filtros e kernel, variam de acordo com as dimensões da entrada e interferem diretamente no resultado de saída.

- Camada de *pooling*

  Exemplificada em C na Figura 2.11, é responsável por diminuir gradualmente a complexidade computacional e a dimensionalidade espacial da entrada. Isso ocorre através da simplificação de matrizes em um único valor. Além disso, essa simplificação pode ocorrer de três maneiras diferentes: os chamados *max pooling*, que reduzem a matriz fornecida no seu maior valor; os *min pooling*, que reduzem a matriz fornecida no seu menor valor; ou os *average pooling*, que reduzem a matriz fornecida no seu valor médio, como é ilustrado na Figura 2.10.

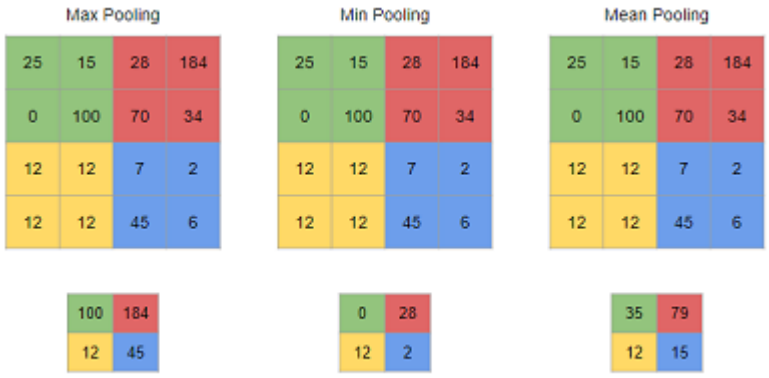

Figura 2.10: Exemplificação pooling (elaborado pelo autor)

- Camada totalmente conectada (*Dense*)

  Geralmente localizadas nas últimas camadas das redes, exemplificadas em D na Figura 2.11, são apropriadas para produzir uma maior quantidade de representação de características, tornando a classificação em diferentes classes mais fácil (SAINATH et al., 2015). Além disso, os neurônios dessa camada são ligados a todos os neurônio da próxima camada.

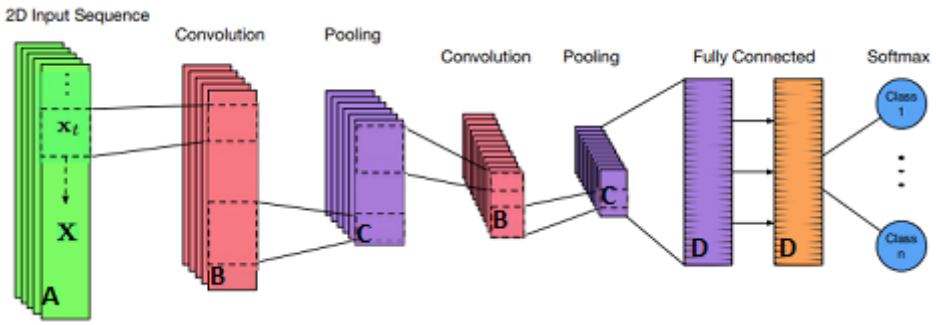

Figura 2.11: Arquitetura simplificada de CNN usando convolução 1D para classificação de imagens (TSANTEKIDIS et al., 2017)

Além dessas camadas, as CNNs ainda podem possuir mais um tipo de camada, chamada de *dropout*. Ela foi proposta recentemente como tentativa de solução do sobreajuste da rede (*over-fitting*), que consiste no processo da rede decorar os dados utilizados no seu treinamento e não ser capaz de aplicar as relações aprendidas em dados novos (BILBAO; BILBAO, 2017). Dessa forma, essa camada implementa um método de regularização, no qual estocasticamente são desativados alguns neurônios em cada caso de

treinamento. Isso interrompe o processo de co-adaptação da rede, uma vez que neurônios desativados não podem influenciar outros (WU; GU, 2015).

Vale-se citar ainda a técnica de *Early Stopping* como alternativa para evitar o *over-fiitng* da rede. Originalmente descrita por (GERSHENFELD, 1988), consiste basicamente na separação do conjunto de treinamento da rede em treino e validação, e monitoramento do erro obtido. Enquanto o erro do conjunto de treinamento diminui ao longo das épocas, espera-se que o erro do conjunto de validação comece a crescer quando a rede começar a sofrer *overfitting*. O objetivo desta técnica é interromper o treinamento precocemente quando erro da validação estiver em seu menor ponto (PRECHELT, 1998).

Na Figura 2.12, é demostrado o processo da camada de *dropout*, onde em (a) temos a rede sem o uso dessa camada, já em (b), os circulos representados com um X são neurônios que foram desativados através da utilização dessa mesma camada.

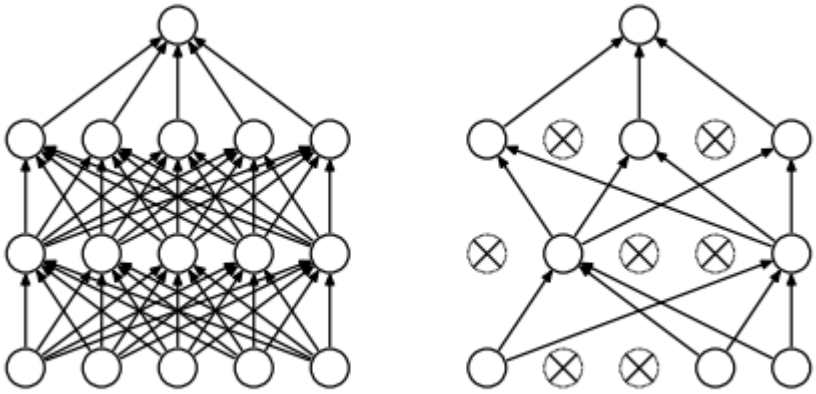

Figura 2.12: Modelo de rede neural *Dropout*. Na esquerda: uma rede neural padrão com 2 camadas. Na direita: um exemplo de reduzida, produzida pela aplicação de *dropout* na rede à esquerda, onde os neurônios com X foram desligados (SRIVASTAVA et al., 2014).

Vale ressaltar, por fim, que as CNNs que obtêm melhores resultados são aquelas que possuem uma maior quantidade de camadas de convolução, *pooling* e *dropout* sobrepostas, e consequentemente, um maior número de parêmetros treináveis. Porém aumentar o quantidade de camadas nem sempre é viável. O aumento da quantidade de camadas acarreta em 3 grandes problemas para o treinamento: 1) aumento no tempo de treinamento necessário; 2) aumento do custo computacional; 3) maior quantidade de dados necessária. (SHAZEER et al., 2017)

2.6 Transferência de Aprendizagem

A transferência de aprendizagem, ou mais comumente referênciado na literatura como *Transfer Learning* (TL), teve sua origem em 1976 com a publicação de um artigo descrevendo seu funcionamento (BOZINOVSKI, 2020). Desde então, vem demostrando ser uma solução viável para problemas enfrentados nos dias de hoje, como a necessidade de grandes quantidades de dados rotulados (TAN et al., 2018) e o alto custo computacional necessário para o treinamento eficiente das redes neurais profundas (ZHANG et al., 2021).

Isso posto, a ideia principal do TL é eliminar os problemas acima através da reutilização de modelos já treinados e do seu conhecimento pré-adquirido para solução de problemas semelhantes. Nesse sentido, são alterados somente parâmetros de poucas camadas, a fim de que a nova rede possa se ajustar ao novo contexto.

Na Figura 2.13, é demonstrada a aplicação do TL, onde uma rede com dados de outro contexto A, nesse trabalho as bases do São Francisco e Delhi, é treinada e seu conhecimento é transferido para uma segunda rede. Essa, por sua vez, recebe os parâmetros já treinados em A e retreina somente os parâmetros das últimas camadas, exemplificada na Figura 2.14, utilizando o conjunto de dados em foco, que nesse trabalho são as bases dos rios Paraíba do Sul e Zambeze.

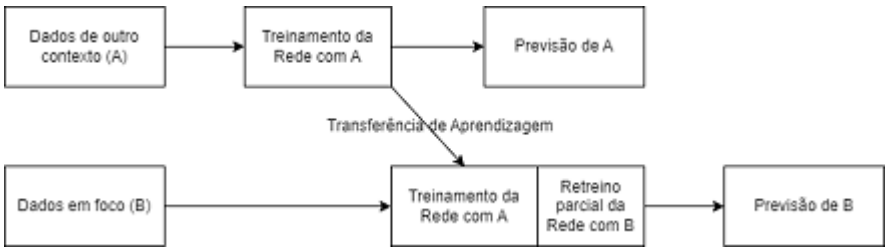

Figura 2.13: Demonstração do *Transfer Learning* (ZHANG et al., 2021).

Além disso, vale-se ressaltar a grande aplicabilidade do TL, que vem sendo utilizado em diversos cenários como predição de defeito em softwares (NAM; KIM, 2015), reconhecimento de atividade (COOK; FEUZ; KRISHNAN, 2013), entre outros.

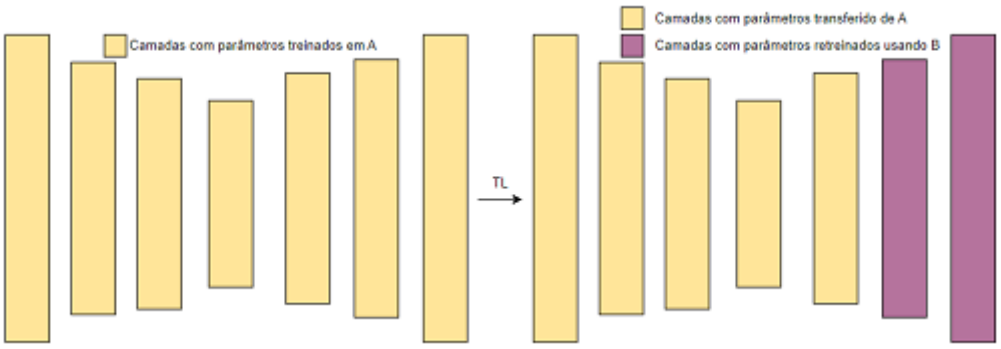

Figura 2.14: Demonstração do retreino das últimas camadas (Elaborado pelo autor).

2.7 Métricas de Avaliação

Em cenários onde desejamos avaliar a performance de modelos de predição, é desejado indicadores que representem fielmentes os erros obtidos, comparando os valores projetados para o futuro com os valores que realmente aconteceram. Dessa forma, apresentamos quatro métricas que foram utilizadas para a avaliação dos resultados obtidos a partir do modelo desenvolvido neste trabalho (CHICCO; WARRENS; JURMAN, 2021).

- *Mean Squared Error* - MSE

  A métrica MSE é uma das mais comuns no ramo aprendizado de máquina e tem seu cálculo expresso na equação 2.2, onde y é o valor ocorrido, y' é o valor previsto e n o número de dados.

$$MSE = \frac{1}{n}\sum(y_i - y_i')^2 \tag{2.2}$$

- *Root Mean Squared Error* - RMSE

  A métrica RMSE é a raiz da métrica MSE, onde a vantagem de se trabalhar com raízes quadradas é que o erro passa a ter a mesma escala do indicador trabalhado e tem seu cálculo apresentado na equação 2.3.

$$RMSE = \sqrt{MSE} \tag{2.3}$$

- *Mean Absolute Error* - MAE

A métrica MAE é calculado através do módulo da subtração obtida entre o valor ocorrido pelo valor previsto, dividido pelo número n de dados. Seu cálculo é expressado na equação 2.4, onde y é o valor ocorrido e y' é o valor previsto.

$$MAE = \frac{1}{n} \sum |y_i - y'_i| \tag{2.4}$$

- *R2 Score* - R^2

A métrica R^2 é uma medida de performance que avalia os acertos e não os erros. Onde o valor tende a estar entre 0, ruim, e 1, perfeito, indica em qual porcentagem o modelo avaliado será melhor em relação a atribuição da média como estimativa. Seu cálculo é expressado na equação 2.5 onde, y é o valor predito, y' o valor previsto e y"é a média dos valores valores previstos.

$$R^2 = 1 - \frac{\sum (y_i - y'_i)^2}{\sum (y_i - y"_i)^2} \tag{2.5}$$

2.8 Metodologia

Inicialmente, foi construído um mapeamento sistemático da literatura, no qual se teve como objetivo buscar referências sobre o uso de uma rede neural artificial específica, chamada de rede neural convolucional, comumente usada para classificação de imagens e para a previsão de séries temporais. Como resultado, foram obtidas referências mostrando não só seu vasto uso, como um melhor desempenho quando comparado a outras técnicas encontradas na literatura, como mostrado em (LIVIERIS; PINTELAS; PINTELAS, 2020).

Após a análise da literatura, sendo comprovado o alto desempenho da rede neural artificial escolhida, continuou-se a busca por referências da aplicação da rede neural convolucional em séries temporais na área de eventos naturais, como predição de luz solar, massas de ar e chuvas, vistos em (KOPRINSKA; WU; WANG, 2018).

Com isso, finalizada essa etapa, foi executado um período de aprendizado e familiarização com a linguagem Python, escolhida pelo seu fácil aprendizado e suas poderosas bibliotecas. Também foram estudadas, nesse período, a biblioteca TensorFlow (ABADI

et al., 2015), usada para machine learning, e a biblioteca Keras (CHOLLET et al., 2015), usada para redes neurais artificiais, ambas de código aberto. Todos os elementos estudados nesse período foram usados no modelo desenvolvido, usando a rede neural convolucional vista na fase de pesquisa da literatura.

Uma vez definida a técnica usada no modelo e as ferramentas, foi analisado o banco de dados disponível, onde foram contidas as séries temporais hidrológicas das bacias em foco no trabalho e das séries temporais usadas para treinar os modelos que transferiram aprendizado. Após essa análise, foi feito um processo de tratamento dos dados e separação dos conjuntos de treino e teste, detalhados no capítulo 4 e visto em A na Figura 2.15.

Em posse dos dados disponíveis tratados, foi desenvolvido o modelo de aprendizado profundo utilizando a arquitetura da rede neural convolucional, e utilizou-se a biblioteca Keras Tuner (O'MALLEY et al., 2019), para definição dos hiperparâmetros da rede, como pode ser visto em B na Figura 2.15.

Por fim, com o modelo pronto, foram feitas quatro cópias idênticas para o treinamento de cada uma das bases e a aplicação do TL nos respectivos casos, como visto em C na Figura 2.15.

2.9 Resumo

Nesse capítulo, foram introduzidos alguns conceitos essenciais para o entendimento do modelo proposto nesse trabalho. Primeiramente, foi descrito o funcionamento e a arquitetura das ANNs e, posteriormente, o mesmo foi feito paras CNNs, tipo de rede neural escolhido para o desenvolvimento do modelo proposto nesse trabalho.

Além disso, também foi introduzido o conceito de séries temporais, uma forma de armazenamento dos dados onde a ordem cronológica de acontecimento dos fatos tem relevância. Essa introdução torna-se necessária uma vez que, esse tipo de dado é utilizado como entrada para o modelo desenvolvido, contendo informações das bacias hidrográficas estudadas.

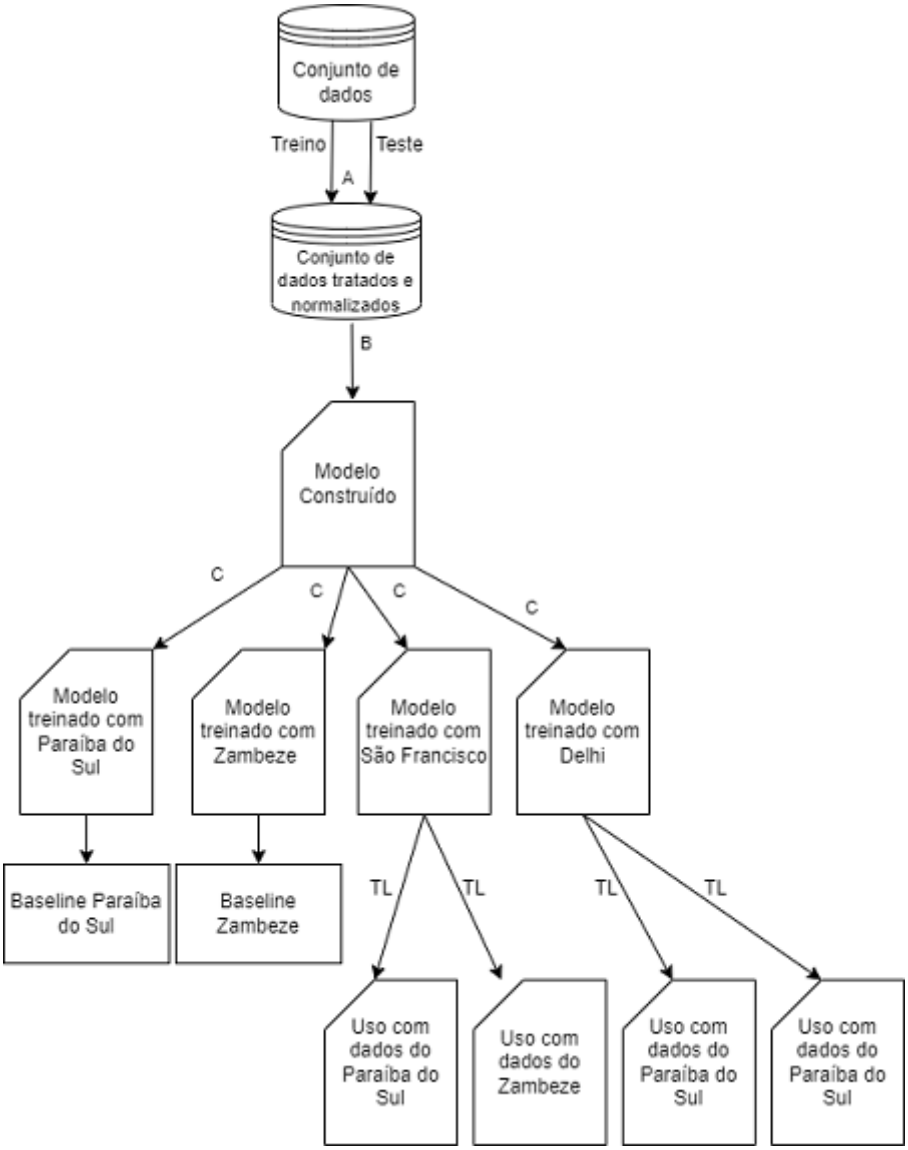

Figura 2.15: Diagrama da metodologia utilizada (Elaborado pelo autor)

3 Trabalhos Relacionados

3.1 Introdução

Esse capítulo é destinado à apresentação dos trabalhos relacionados selecionados. A escala PRISMA foi utilizada como base para avaliação crítica das revisões sistemáticas já publicadas, ajudando os autores a selecionar o melhor referencial teórico possível (GALVÃO; PANSANI; HARRAD, 2015).

Em cada trabalho citado são apresentados os objetivos, resultados alcançados, conclusões e pontos positivos e negativos em relação à ser escolhido para comparação.

3.2 Descrições dos Trabalhos Relacionados

Neste primeiro trabalho relacionado (ABREU, 2019), é proposto o modelo MultiTask-LSTM, onde são aliados o modelo recorrente de Deep Learning LSTM com a transferência de aprendizagem *MultiTask Learning*, para prever e compartilhar informações adquiridas em toda a bacia do rio. Ainda nesse trabalho, foi usado como conjunto de dados séries temporais de vazão de 45 estações de medição do Rio Paraíba do Sul.

Além disso, foram apresentadas e comparadas através da métrica MAPE, três técnicas diferentes para imputação de dados faltantes, sendo elas ARIMA, média dos dias em todos os anos e mediana da série. Também foram comparados os modelos utilizando somente LSTM e sua união com *MultiTask-Learning*.

Como resultado, foi visto que o uso do modelo MultiTask-LSTM utilizando a técnica de média dos dias do ano, que posteriormente foi aprimorada para média dos dias do mês, apresentou um desempenho quase duas vezes melhor que os alcançados com o modelo que utilizou somente LSTM.

Por fim, temos como relações de comparação positivas nesse primeiro trabalho relacionado o uso de séries temporais hidrológicas nacionais, em específico as do Paraíba do Sul e a utilização de modelos preditivos. Como relações de comparação negativas, temos

a ausência do uso de CNNs para usarmos em comparação com os resultados obtidos nesta monografia.

Já no segundo trabalho relacionado (MARTINHO et al., 2020), é proposto a aplicação de *Extreme Learning Machine* (ELM) juntamente com algoritmos genéticos para a predição da vazão na barragem de Cahora Bassa. Nesse sentido, a ideia é utilizar algoritmos genéticos para definir a melhor combinação de parâmetros internos do ELM, essencias para o bom desempenho do modelo.

Ainda nesse segundo trabalho, para avaliar o desempenho do modelo proposto, foram implementados outros três modelos de *machine learning*, sendo eles: o modelo linear *ElasticNet* (EN); regressão de vetor de suporte (SVR); e aumento de gradiente extremo (XGB). Como métricas de desempenho, foram utilizadas: R^2, RMSE, MAPE, eficiência de Nash-Sutcliffe (NSE) e eficiência Kling-Gupta (KGE).

Como resultado, foi visto que o uso de algoritmos genéticos para guiar a escolha dos parâmentos internos do ELM trouxe melhores resultados quando comparado aos outros modelos implementados. Embora tenha sido notado que o modelo proposto superestima picos de fluxo, demonstrou-se um bom potencial para a realização da previsão da vazão.

Por fim, como relações de comparação positivas nesse segundo trabalho relacionado temos o uso da série temporal da barragem de Cahora Bassa, localizada no rio Zambeze. Além disso, a avaliação da predição dos mesmos dados com outros modelos, utilizando as mesmas métricas (RMSE e R^2), serve também como referência de desempenho do modelo desenvolvido na presente monografia. Como relações de comparação negativas temos a falta de comparações com modelos que utilizem CNNs.

Como terceiro trabalho relacionado (KOPRINSKA; WU; WANG, 2018), é investigado o uso de CNNs para a previsão de séries temporias de energia. Em particular, busca-se prever a energia solar fotovoltaica e a carga de eletricidade para o dia seguinte, por meio do uso das informações de dias anteriores. Além disso, também é descrito em detalhes o funcionamento das CNNs e suas características.

Ainda nesse trabalho, são comparados os seguintes métodos, mais citados na literatura como: Convolutional neural network (CNN), Multilayer perceptron neural network

(MLP), Long short-term memory (LSTM) e Persistence model (P). Em todos os métodos foram submetidos os mesmos dados de entrada e seu desempenho foi comparado levando em consideração MSE e MAE.

Como resultado, foi visto que a CNN foi o método que obteve o melhor desempenho, obtendo resultados mais acurados em quase metade dos conjuntos de dados testados e igualando seu desempenho com a MLP na outra metade. Além disso, se sobressaíram em relação ao tempo de treinamento, levando de um a três minutos, enquanto o método LSTM levou cerca de cinco a seis horas para treinar.

Por fim, como relações de comparação positivas nesse terceiro trabalho relacionado a comparação do desempenho das CNNs aplicadas a séries temporais, quanto ao tempo de treinamento e à acurácia dos modelos. Como relações de comparação negativas, temos o uso de séries temporais que não são hidrológicas e a falta de testes que comprovem a superioridade do método CNN sobre o método MLP que, embora tenham sido citadas características indicando vantagens, não foi testado.

No quarto trabalho relacionado (SARAIVA et al., 2021), são comparados 2 modelos de ML, ANNs e SVMs, combinados com transformação *wavelet* e reamostragem de dados com o método *bootstrapping* voltados para a previsão da vazão com dados da bacia do São Francisco. Além disso, é apresentada a origem e importância dessa bacia para a região onde se encontra e para o âmbito nacional.

Ainda nele, são descritas as técnicas *wavelet* e *bootstrapping* e suas implementações, utilizadas nos dois modelos propostos, bem como as métricas de desempenho utilizadas: MSE, MAE e R^2.

Como resultado, foi observado que: a combinação dos modelos com as técnicas de *Wavelet* ou *Bootstrapping* forneceram um ganho de acurácia nos resultados, e a presença de ambas um ganho maior ainda; o modelo utilizando ANNs e as duas técnicas propostas se sobressaiu quando comparado ao modelo SVM.

Por fim, temos como relações de comparação positivas nesse quarto trabalho relacionado, o desenvolvimento de modelos preditivos voltados para vazão hidrológica, mais especificamente os da bacia do São Francisco, os quais foram usados para transferência de aprendizagem nessa monografia. Já como relações de comparação negativas, temos a

ausência do uso e/ou comparações com CNNs.

Neste quinto trabalho relacionado (SAHOO et al., 2019), é explorado um modelo de rede neural recorrente long short-term memory (LSTM-RNN), voltado para a previsão de baixas no fluxo do rio Mahanadi, localizado na Índia. Além disso, são citadas a importância da previsão de séries temporais hidrológicas, a definição de baixo fluxo e sua importância para o meio ambiente ao seu redor.

Ainda nele, é descrito como é feita a união dos modelos LSTM e *recurrent neural network* (RNN), usados para a criação de um novo modelo. Em seguida, compara-se esse novo modelo com o modelo de RNN e um modelo ingênuo, em que é assumido, nesse último citado, que o valor a ser previsto sempre será igual ao valor anterior. Essa comparação tem como objetivo avaliar o desempenho ao levar em consideração MSE e MAE e R^2.

Como resultado, foi observado que o modelo proposto obteve um desempenho significativamente melhor quando comparado com os outros modelos. Além disso, foi descrito que a habilidade de lembrar, esquecer e atualizar informações foi o principal motivo pelo qual o modelo proposto se sobressaiu.

Por fim, temos como relações de comparação positivas nesse quinto trabalho relacionado, a descrição da relevância de séries temporais hidrológicas e sua utilização voltadas para previsão de fluxo. Como relações de comparação negativas, temos a falta da utilização do modelo CNN, dificultando a comparação dos resultados com o modelo desenvolvido nesta monografia.

Neste sexto trabalho relacionado (YASEEN et al., 2015), são exploradas as aplicações de inteligência artificial voltadas para a previsão da vazão de rios, e foca em definir não só os maiores desafios envolvidos, como também as vantagens de modelos complementares.

Ainda nele, são mostrados cinco tipos de modelos para previsão da vazão, sendo eles: modelagem de redes neurais artificiais; abordagem de máquina de vetores de suporte; método de lógica fuzzy; método de computação evolucionária; e modelagem complementar *Wavelet*. No trabalho, para cada um dos modelos citados, são mostrados outros exemplos de abordagens utilizando inteligência artificial no ramo das aplicações hidrológicas, como previsão da precipitação pluviométrica, secas e alagamentos.

Como resultado, foram tiradas conclusões sobre os diversos resultados encontrados nas pesquisas utilizadas. Dentre elas, foi concluído que a presença de outras variáveis, como temperatura máxima, mínima e precipitação, possui relação direta com um melhor desempenho do modelo. Além disso, também foi observado que, na maioria dos artigos revisados, foram usados para avaliação do desempenho o R^2 ou RMSE, acrescido de outro critério de avaliação, como MSE e MAE.

Por fim, temos como pontos de comparação positivos deste sexto trabalho relacionado, a presença de uma grande quantidade de artigos avaliados e comparados, de onde foram tiradas informações relevantes sobre a previsão da vazão e referências úteis de outras abordagens envolvendo séries hidrológicas, podendo ser usadas como fonte de inspiração para o desenvolvimento de trabalhos futuros. Como pontos de comparação negativos, não temos a presença do método CNN, o qual foi utilizado no desenvolvimento desse trabalho.

E como sétimo e último trabalho relacionado (HUSSAIN et al., 2020), é explorada a previsão da vazão, um passo à frente, em três cenários diferentes (diário, semanal e mensal), usando tanto uma rede neural convolucional de uma dimensão (1D-CNN) como as ELMs. Ainda, foram descritos a importância da previsão da vazão, o motivo pelo qual foi escolhido a utilização e a comparação dos modelos 1D-CNN e ELM.

Ainda nele, foram descritas a bacia do rio Gilgit, situado no Paquistão, como área de estudo, bem como o conjunto de dados utilizados para teste. Nesses dados, os conjuntos foram não só divididos em diários, semanais e mensais, como também foram separados pela estrutura de input, sendo classificadas em inputs de 1 a 5 passos atrás.

Como objetivo, foi buscada a avaliação e a comparação do desempenho dos dois modelos propostos nos três cenários estudados. Isso foi feito atráves de quatro métricas diferentes: R^2, MSE, MAE e o erro relativo (RE).

Como resultado, temos que o modelo 1D-CNN não foi capaz de captar picos de fluxo diária e mensalmente, e o modelo ELM subestimou-os, semanal e mensalmente. Assim, embora não tenham sido obtidas boas previsões mensais, ambos apresentaram um bom desempenho para os demais períodos de tempo.

Por fim, temos como pontos de comparação positivos deste sétimo trabalho rela-

cionado, a utilização de uma CNN aplicada ao cenário de previsão de fluxo, bem como uma abordagem diferente para divisão e combinação dos dados de entrada, a qual gerou bons resultados. Como ponto de comparação negativo, temos a falta de comparações com outros modelos existentes na literatura, dificultando a verificação do desempenho do modelo proposto.

3.3 Resumo

Nesse capítulo foram apresentados artigos relacionados que apresentavam características em comum com o trabalho desenvolvido nessa monografia. A escolha dos trabalhos relacionados foi condicionada a presença de pelo menos 3 dos seguintes critérios: uso de redes neurais convolucionais, tipo de rede que foi usado nesta monografia; uso de dados de vazão hidrológica da bacia do Paraíba do Sul e/ou Zambeze, dados que foi usado nesta monografia; uso de dados de origem da natureza; desenvolvimento de modelos preditivos, tipo de modelo que fornece previsões como informação de saída, foco nesta monografia; apresentaram comparações de seus resultados com a literatura.

Entretanto, ainda que esses trabalhos relacionados se apliquem ao contexto desta monografia por lidar individualmente com parâmetros relevantes para nós, a literatura ainda carece de pesquisas que associam esses conhecimentos. Vale-se destacar ainda, a falta de referências na literatura de trabalhos que utilizem CNNs para previsão de séries temporais hidrológicas.

Dito isso, foi identificada a necessidade de se trabalhar com CNNs para previsão de dados hidrológicos, contribuindo para o progresso na predição da vazão de rios e para a literatura nacional.

4 Descrição do Modelo

4.1 Introdução

Esse capítulo é destinado à apresentação e à descrição do modelo proposto, bem como o processo de definição da arquitetura e seus hiperparâmetros.

4.2 Arquitetura

A definição da arquitetura foi feita empiricamente, através de experimentos preliminares analisando a métrica de desempenho MSE. Para cada arquitetura analisada, foram realizados 10 treinamentos com o mesmo conjunto de dados, utilizando 300 épocas e aplicando a técninca de *EarlyStopping*.

Foram analisadas arquiteturas com 3, 4, 5 e 6 camadas de convolução; 1 ou 2 camadas de *pooling* e *Dense*; 0, 2 e 5 camadas de *Dropout*. Todos os testes feitos utilizaram o conjunto de dados do Rio Paraíba do Sul, o qual foi separado em treino, validação e teste, nas porcentagens de 75, 15 e 10, respectivamente, e são apresentados na Tabela 4.1.

Testes executados utilizando os dados do Paraíba do Sul	**Quantidade de Camadas**
Camada de Convolução	3, 4, **5** e 6
Camada *Pooling*	**1** e 2
Camada *Dense*	1 e **2**
Camada *Dropout*	**0**, 2 e 5

Tabela 4.1: Tabela de resultados para definição da arquitetura.

Por fim, a arquitetura definida para o modelo possui 5 camadas de convolução, 1 camada *pooling*, 2 camadas *dense* e nenhuma camada de *dropout*, sendo apresentada em negrito na Tabela 4.1. A arquitetura supracitada foi fixada para todos os modelos para a obtenção de um parâmetro inicial e está disposta na seguinte configuração: três camadas de convolução de uma dimensão; seguidas por uma camada de *max-pooling*; duas camadas

de convolução de uma dimensão; uma camada de achatamento (*Flatten*); e, por fim, duas camadas totalmente conectadas (*Dense*), podendo ser vista na Figura 4.1.

4.3 Definição de hiperparâmetros

O processo de escolha dos hiperparâmetros foi feito através do KerasTuner da biblioteca Keras (O'MALLEY et al., 2019). Esse framework atua como otimizador das váriavies fornecidas, onde é delimitada uma faixa de possibilidades e a quantidade de épocas por combinação. Nesse trabalho, foram escolhidas como varíaveis a serem otimizadas: a quantidade de filtros para a camada de convolução de uma dimensão; o tamanho do Kernel; e a função de ativação. Além disso, também foi otimizado a quantidade de neurônios na camada *Dense* e o *Pool-size* da camada de *Pooling*. Esse processo utilizou 50 buscas por combinação e os resultados podem ser vistos na Tabela 4.2, onde são apresentadas as variáveis otimizadas e a faixa de possibilidades.

	Valor Otimizado	Faixa de Possibilidades
Quantidade de Filtros	208	[16-256]
Tamanho do Kernel	2	[1-3]
Função de Ativação	ReLU	[Linear,Sigmoid,ReLU]
Quantidade de Neurônios	192	[32-256]
Pool-size	2	[1-3]

Tabela 4.2: Tabela de resultados para a definição dos hiperparâmetros.

Através da ferramenta KerasTuner citada, foi aprendido que nem sempre uma maior quantidade de filtros ou neurônios trazem os melhores resultados, onde nesse trabalho, embora o limite máximo da faixa de possibilidades fosse 256, esses valores não geraram as melhores soluções.

Por fim, na Figura 4.1, é representado em vermelho a arquitetura definida, em azul e amarelo, a quantidade de filtros e quantidade de neurônios da camada *Dense* respectivamente, determinados pelo processo de definição dos hiperparâmetros apresentados nesta seção. Além disso, é possível ver em verde o tamanho da janela de tempo utilizada, representando a medição de 3 vazões diárias para a previsão do dia seguinte.

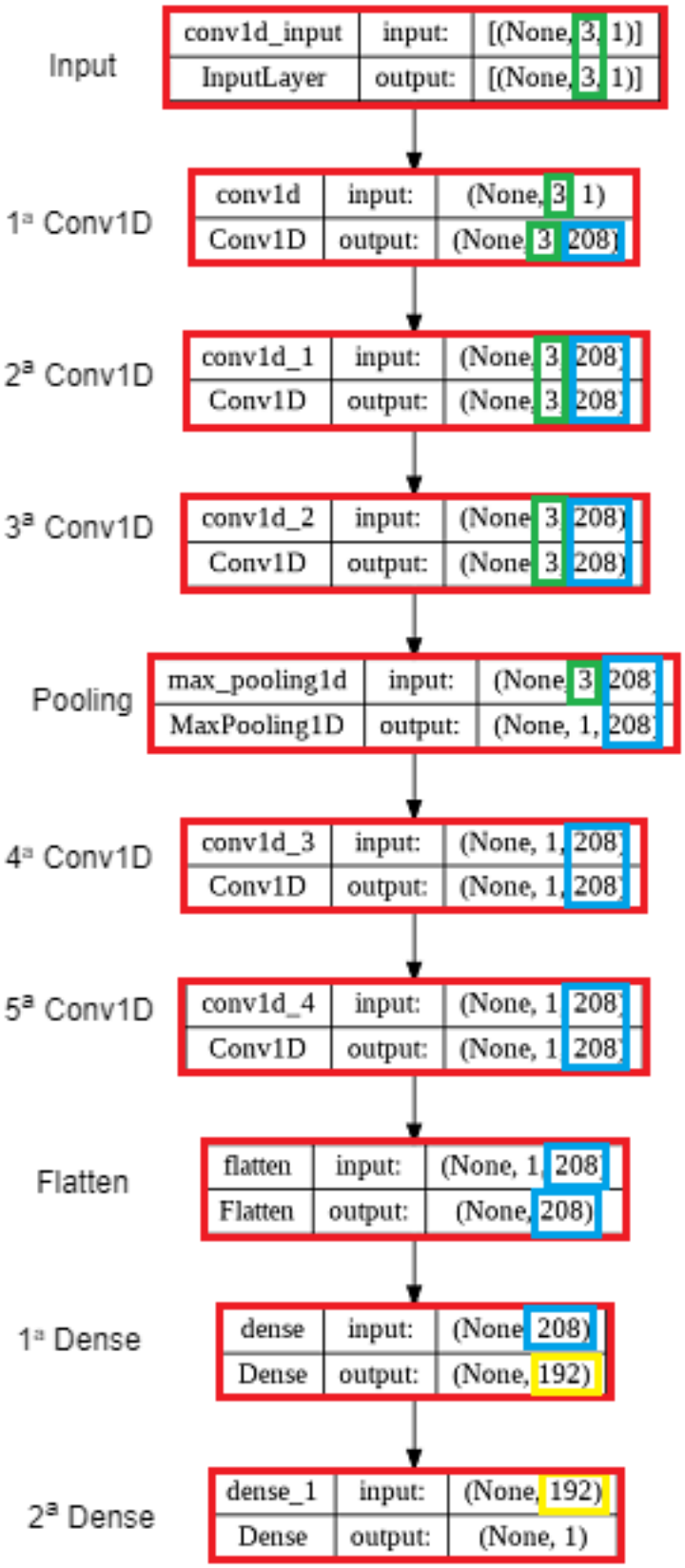

Figura 4.1: Arquitetura Modelo (Elaborado pelo autor)

4.4 Implementação do Modelo

Essa seção é destinada à descrição da implementação do modelo e a detalhes técnicos como: linguagem, funções, bibliotecas e ambiente de Desenvolvimento utilizados.

4.4.1 Ambiente de Desenvolvimento

O modelo foi desenvolvido no *Google Colaboratory* (Colab) (RESEARCH, 2022), um ambiente de desenvolvimento integrado (IDE) *Web* voltada para a linguagem Python. A escolha por essa IDE levou em conta 2 principais motivos. O primeiro deles foi o uso da linguagem Python, a qual possui uma grande quantidade de bibliotecas focadas em inteligência artificial e *machine learning*. Já o segundo se deve ao fato do Colab rodar em nuvem, fornecendo uma maior disponibilidade e eliminando a necessidade da instalação de programas e *setup* prévio.

4.4.2 Bibliotecas

Durante a implementação do modelo, diversas bibliotecas foram utilizadas, como: TensorFlow (ABADI et al., 2015), usada para algoritmos de aprendizado de máquina; Keras (CHOLLET et al., 2015), usada para implementação das CNNs; Pandas (TEAM, 2020), usada para manipulação de dados; Scikit-Learn (PEDREGOSA et al., 2011), usada para normalização dos dados; SeaBorn (WASKOM, 2021), usada para criação de gráficos estatísticos; e NumPy (HARRIS et al., 2020), usada para manipular matrizes e vetores.

4.4.3 Conjuntos de dados

Os dados utilizados nos modelos foram separados em 2 grupos diferentes, as séries temporais em foco no trabalho e as utilizadas para pré-treinamento do TL. As séries em foco no trabalho são:

- Série Temporal da Vazão da Bacia do Paraíba do Sul

  A escolha desse conjunto de dados deu-se pela importância econômica da bacia para a região onde se encontra, sendo apresentada na seção 1.2. A série temporal trabalhada possui 4748 dados, coletados entre os anos de 2002 a 2014

e é representada na Figura 4.2. Os dados representam a vazão diária em metros cúbicos por segundo, aferidos no ponto de coleta da ponte municipal em Campo dos Goytacazes, retirados do sistema da Agência Nacional das Águas (ANA) (ANA, 2022).

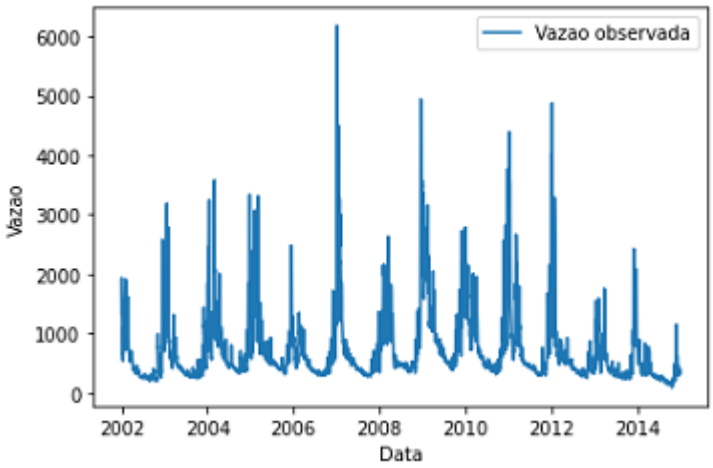

Figura 4.2: Dados da série temporal da Bacia do Paraíba do Sul coletados entre os anos de 2002 a 2014 (Elaborado pelo autor)

- Série Temporal da Vazão da Bacia do Zambeze

  A escolha desse conjunto de dados deu-se pela importância hídrica da bacia para a região onde se encontra, sendo apresentada na seção 1.2. A série temporal trabalhada possui 5844 dados, coletados entre os anos de 2003 a 2018 e é representada na Figura 4.3. Os dados representam a vazão diária em metros cúbicos por segundo, aferidos na barragem de Cahora Bassa em Moçambique, retirados da companhia hidroelétrica de Cahora Bassa do departamento de gestão ambiental de recursos hídricos (BASSA, 2022).

Já as séries temporais para pré-treinamento dos modelos utilizando TL foram:

- Série Temporal da Vazão da Bacia do São Francisco

  A escolha desse conjunto de dados se deu pela semelhança de contexto com outros 2 conjuntos de dados apresentados acima, principalmente o do Paraíba do Sul, onde o objetvivo foi utilizar dados de uma bacia nacional de mesma relevência

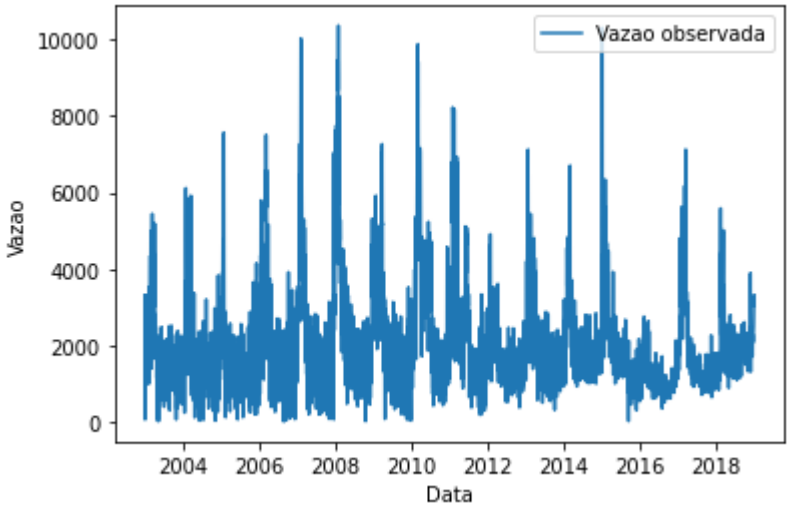

Figura 4.3: Dados da série temporal da Bacia do Zambeze coletados entre os anos de 2013 a 2018 (Elaborado pelo autor)

para a região onde se localiza. Dessa forma, foi escolhida a bacia do São Francisco, a quarta maior bacia do Brasil e principal fonte de água do nordeste. A série temporal trabalhada possui 4018 dados, coletados entre os anos de 2002 a 2012 e é apresentada na Figura 4.4. Os dados representam a vazão diária em metros cúbicos por segundo, aferidos no reservatório de Três Marias e retirados do ONS (Operador Nacional do Sistema Elétrico) (ONS, 2022).

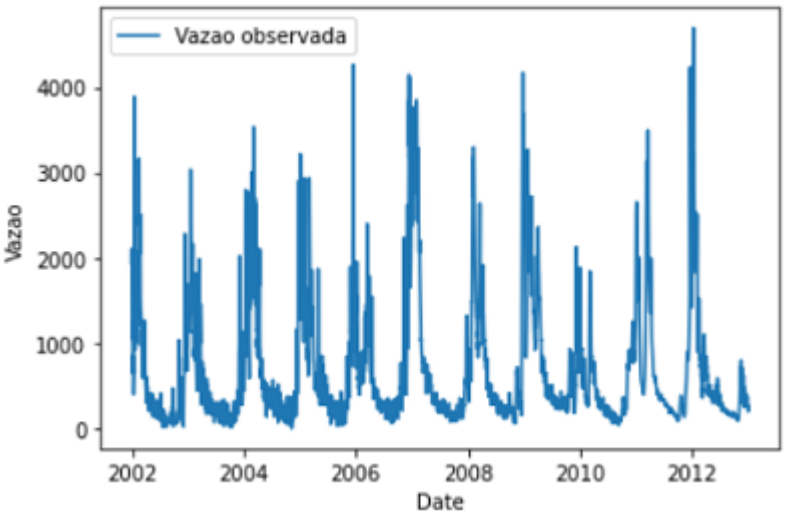

Figura 4.4: Dados da série temporal da Bacia do São Francisco coletados entre os anos de 2002 a 2012 (Elaborado pelo autor)

- Série Temporal da Temperatura da cidade de Delhi

A escolha dessa base se deu pela necessidade de analisar dados de contextos diferentes, porém ainda sendo dados coletados da natureza, como tempetarura, precipitação, entre outros. Dessa forma, foi escolhida uma base de dados climática desenvolvida pela universidade de PES (People's Education Society), em Bangalore (PES, 2022).

A série temporal trabalhada possui 1575 dados coletados entre os anos de 2013 a 2017 e é apresentada na Figura 4.5. Os dados representam a temperatura média diária em graus celsius, aferidos na cidade de Delhi na India e retirados do acervo de *datasets online* Kagle (BANGALORE, 2019).

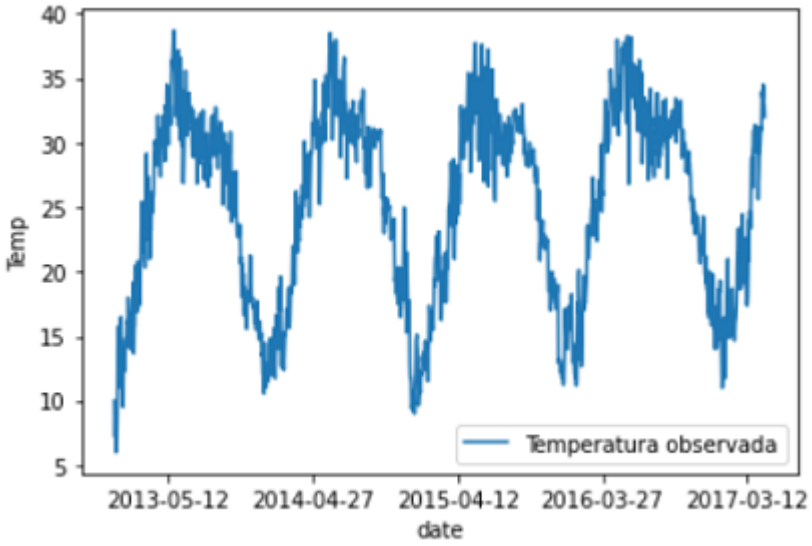

Figura 4.5: Dados da série temporal de temperatura da cidade de Delhi coletados entre os anos de 2013 a 2017 (Elaborado pelo autor)

Também foi estudado outros dois conjuntos de dados de contexto financeiro para pré-treinamento do modelo, utilizando TL heterogêneo (WEISS; KHOSHGOFTAAR; WANG, 2016), sendo eles séries temporais com valores diários das criptomoedas Bitcoin e Ethereum, retirados do site de monitoramento de criptoativos investing.com (INVESTING.COM, 2022). Porém, os resultados com esses conjuntos de dados foram insatisfatórios devido a falta de relações presentes com dados ambientais, como forte sazonalidade e pouca tendência, e por este motivo os testes com esses conjuntos de dados foram descartados.

As bases foram tratadas seguindo o seguinte processo: análise da existência ou

não de dados faltantes; transformação da coluna de data em index da tabela; exclusão de demais colunas exceto as que contem a informação desejada. Após esse processo, todos os conjuntos de dados passaram a ter apenas uma coluna além do index, e podem ser exemplificados na Figura 4.6.

Data	Vazão
2002-01-01	1337.527466
2002-01-02	1935.916748
2002-01-03	1746.648560
2002-01-04	1518.592773
2002-01-05	1330.862305

Figura 4.6: Formato das bases após o processo de tratamento (Elaborado pelo autor)

4.4.4 Janelamento de tempo

A transformação dos dados de janelas de tempo é feita através de uma função, na qual é definido um tamanho para janela de entrada (M) e para a de saída (N). Esse processo pode ser visualizado na Figura 4.7, onde foi utilizado o conjunto de dados do Paraíba do Sul, que inicialmente continha o formato (4748,1) e após a aplicação do janelamento utilizando M = 3 e N = 1 passou a ter o formato (4745,3,1).

Nesse trabalho, foram feitos testes com M = [3,7] e N = 1. Além disso, vale ressaltar que devido à necessidade de M dados para a formação da janela de entrada e N dados para a janela de saída, ao fim do processo temos (Tamanho inicial)-M-N janelas.

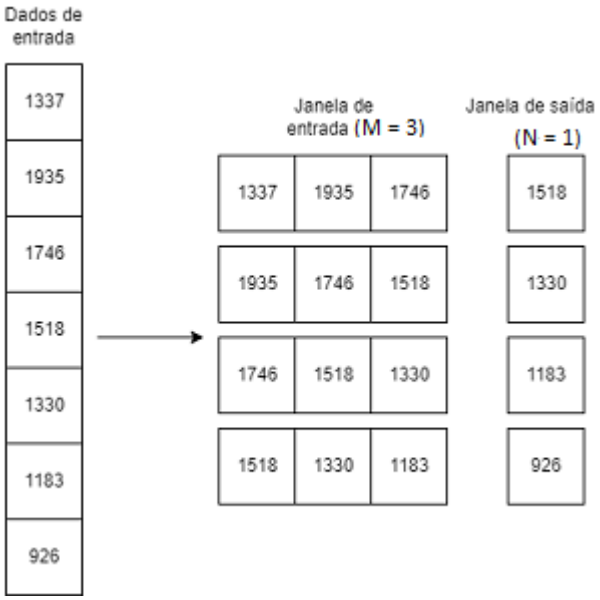

Figura 4.7: Demonstração do processo de janelamento usando dados do Paraíba do Sul (Elaborado pelo autor).

5 Estudo de Caso

Neste trabalho, dividimos o estudo de caso em 2 etapas, a de modelos para previsão e a de modelos para previsão utilizando TL. Além disso, em cada uma dessas etapas, foram avaliados 2 tipos de modelos, utilizando M = 3 e M = 7, escolhidos empíricamente para se obter parâmetros iniciais.

Durante a fase de modelos para previsão, o foco foi avaliar o desempenho dos modelos sem o uso do TL, treinados nas bases em foco no trabalho, apresentadas em A na Figura 5.1. O resultado foi obtido através de um conjunto de 10 execuções, selecionando a que apresentou um melhor valor de MSE e que posteriormente foi comparada com o desempenho dos modelos que utilizam TL.

Já durante a fase de modelos para previsão que utilizam TL, foram feitos testes para determinar a quantidade de camadas que deveriam ser retreinadas após o TL, visando obter um melhor ajuste aos novos dados. Além disso, foi avaliado o desempenho dos modelos apresentados em B e C na Figura 5.1, que podem ser separados em 2 grupos, bases de vazão hidrológica, mesmo contexto de A, e base de temperatura, contexto diferente de A. O resultado nessa etapa seguiu o mesmo processo de seleção da etapa anterior, onde também foram realizadas um conjunto de 10 execuções.

Vale ressaltar que em C na Figura 5.1, foram feitos testes preliminares para avaliar o desempenho de previsão com as bases de Bitcoin e Ethereum, porém foram excluídas devido a seus resultados insatisfatórios. Já a escolha da base de mesmo contexto se deu pela busca de bacias de alta relevância nacional.

Por fim, é feita a consolidação dos resultados, onde foram comparados os resultados da TL pré-treinada através da base de mesmo contexto e de contextos diferentes para definir qual delas conseguiu melhor desempenho. Podemos visualizar o processo feito nas duas etapas na Figura 5.2, onde temos a de modelos para previsão em vermelho e a de modelos para previsão utilizando TL em azul.

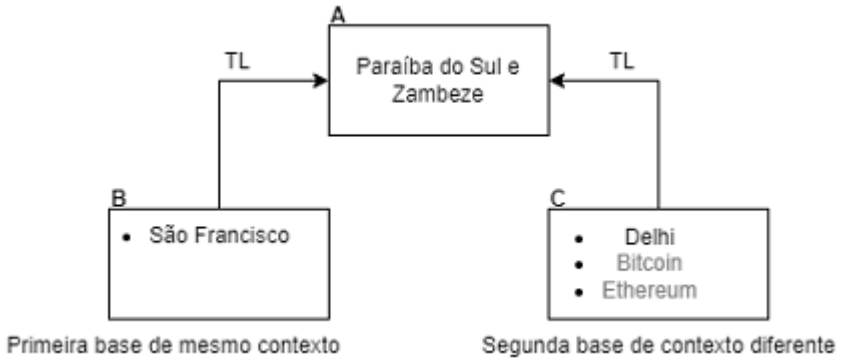

Figura 5.1: Apresentação das bases utilizadas em cada etapa (Elaborado pelo autor).

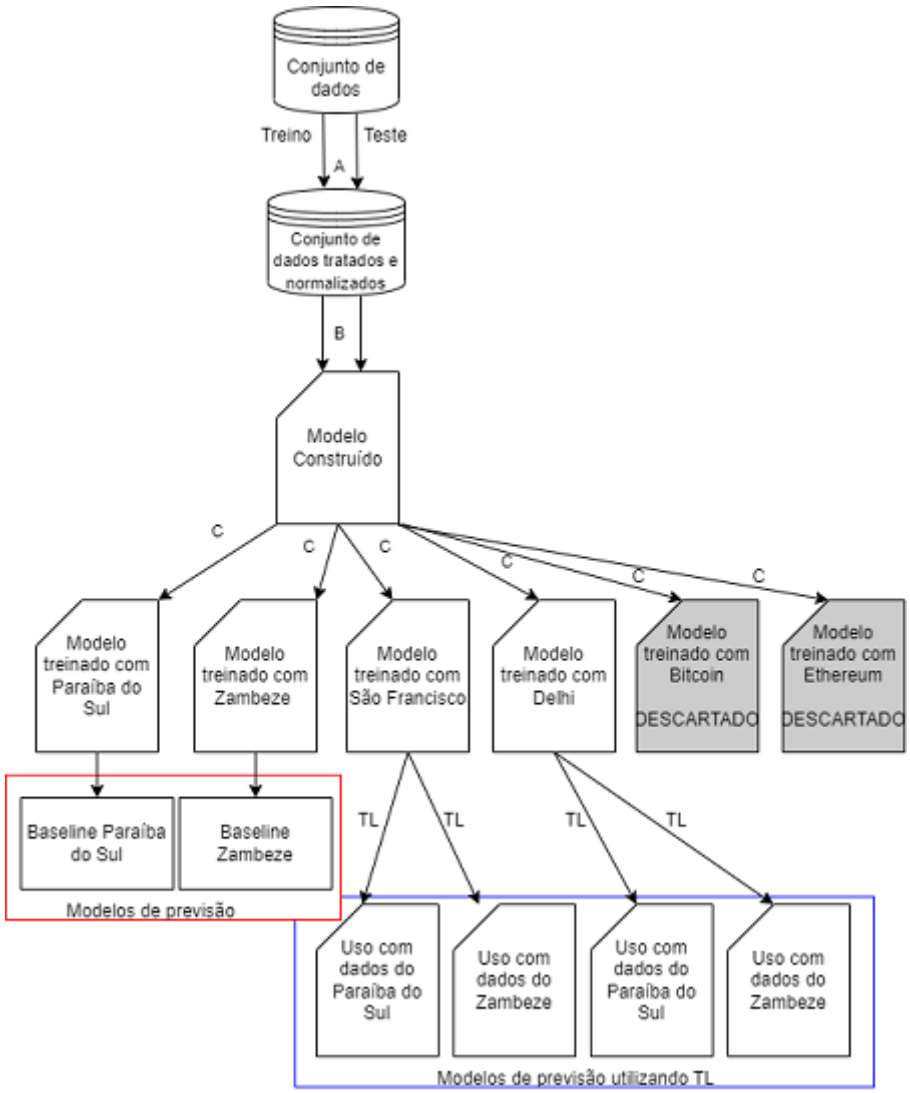

Figura 5.2: Visualização do processo realizado nas etapas de modelos de previsão e modelos de previsão utilizando TL (Elaborado pelo autor)

5.1 Modelos de previsão

Para as bases do Paraíba do Sul e Zambeze, cada modelo foi treinado 10 vezes e selecionou-se aquele que apresentou um melhor desempenho em relação a métrica MSE, com objetivo de possuir um valor de referência base sem o uso do TL.

A Tabela 5.1 apresenta os valores das métricas MSE, MAE, RMSE e R^2 da melhor execução entre as 10 realizadas com cada uma das bases de dados, tanto para o modelo que utilizou 3 dias anteriores quanto para o modelo que utilizou 7 dias anteriores, representados por M na tabela.

Para a obtenção desses resultados foi construido um modelo que utilizou a arquitetura definida na seção 4.2 e os hiperparâmetros otimizados, apresentados na Tabela 4.2.

		MSE	MAE	RMSE	R^2
Base Paraíba do Sul	M = 3	**0.00024**	**0.00085**	**0.01547**	**0.91052**
	M = 7	0.00026	0.01017	0.01642	0.89924
Base Zambeze	M = 3	0.00154	0.02733	0.03925	0.58061
	M = 7	**0.00145**	**0.02722**	**0.03809**	**0.60526**

Tabela 5.1: Resultado do valor médio das métricas MSE, MAE, RMSE e R^2 das 10 execuções para base Paraíba do Sul e Zambeze utilizando M=3 e M=7.

Como pode ser observado em negrito na Tabela 5.1, o modelo utilizando M = 3 obteve melhor desempenho com os dados da Bacia do Paraíba do Sul, já o modelo com M = 7 foi o que apresentou melhores resultados para os dados da Bacia do Zambeze.

Acredita-se que, como a base do Paraíba do Sul apresenta períodos de seca bem definidos e distantes dos picos de cheia, uma maior quantidade de informações, ou seja, uma maior janela de entrada (M), pode perturbar mais os dados no período de seca, e prejudicar o desempenho do modelo. Por esse motivo, o modelo utilizando M = 3 obteve um melhor resultado.

Já na base do Zambeze, é observado que os períodos de seca não são tão bem comportados e, por esse motivo, uma maior quantidade de informações é benéfica ao modelo. Por esse motivo, acredita-se que o modelo utilizando M = 7 se saiu melhor para essa base.

5.2 Modelos de previsão utilizando TL

Inicialmente na Tabela 5.2, são apresentados os resultados dos modelos preditivos utilizando as bases do São Francisco e Delhi, onde foi confirmada a possibilidade de aplicação do TL devido ao bom desempenho observado.

		MSE	MAE	RMSE	R^2
Base São Francisco	M = 3	0.00201	0.02211	0.04486	0.94741
	M = 7	**0.00170**	**0.02047**	**0.04126**	**0.95593**
Base Delhi	M = 3	**0.00256**	**0.03977**	**0.05064**	**0.91716**
	M = 7	0.00292	0.04221	0.05406	0.90779

Tabela 5.2: Resultado do valor médio das métricas MSE, MAE, RMSE e R^2 das 10 execuções para base São Francisco e Delhi utilizando M=3 e M=7.

Como pode ser visto na Figura 5.1, nesta etapa do trabalho vão ser treinadas duas bases de dados para aplicar TL: Bacia do São Francisco e Delhi. Depois o TL é aplicado nas bases de previsão do rio Paraíba do Sul e Zambeze, retreinando a última camada e as duas últimas camadas.

A retreinagem das 2 últimas camadas demonstrou maior desempenho para os dois valores de M utilizando a base do São Francisco e Delhi, conforme são vistos nas Tabelas 5.3, 5.4, 5.5 e 5.6.

São Francisco utilizando M = 3	Métricas	Camadas retreinadas	
		Última	Duas últimas
Paraíba do Sul	MSE	0.00038	**0.00029**
	MAE	0.01242	**0.01013**
	RMSE	0.02044	**0.01707**
	R^2	0.84280	**0.89115**
Zambeze	MSE	0.01516	**0.00158**
	MAE	0.03299	**0.02875**
	RMSE	0.04284	**0.03986**
	R^2	0.49999	**0.56773**

Tabela 5.3: Resultado dos testes executados para determinar a quantidade de camadas à retreinar utilizando M = 3 e a base do São Francisco.

São Francisco utilizando M = 7	Métricas	Camadas retreinadas	
		Última	Duas últimas
Paraíba do Sul	MSE	0.00164	**0.00070**
	MAE	0.02781	**0.02062**
	RMSE	0.02834	**0.02653**
	R^2	0.69845	**0.73700**
Zambeze	MSE	0.00172	**0.00157**
	MAE	0.02994	**0.02847**
	RMSE	0.04153	**0.03961**
	R^2	0.53048	**0.57300**

Tabela 5.4: Resultado dos testes executados para determinar a quantidade de camadas à retreinar utilizando M = 7 e a base do São Francisco.

Delhi utilizando M = 3	Métricas	Camadas retreinadas	
		Última	Duas últimas
Paraíba do Sul	MSE	0.00040	**0.00025**
	MAE	0.01097	**0.00871**
	RMSE	0.02010	**0.01572**
	R^2	0.84829	**0.90764**
Zambeze	MSE	0.00146	**0.00154**
	MAE	0.02990	**0.02846**
	RMSE	0.04127	**0.03930**
	R^2	0.53232	**0.57956**

Tabela 5.5: Resultado dos testes executados para determinar a quantidade de camadas à retreinar utilizando M = 3 e a base Delhi.

Delhi utilizando M = 7	Métricas	Camadas retreinadas	
		Última	Duas últimas
Paraíba do Sul	MSE	0.02376	**0.00130**
	MAE	0.03721	**0.03294**
	RMSE	0.04112	**0.03606**
	R^2	0.47786	**0.51402**
Zambeze	MSE	0.00204	**0.00188**
	MAE	0.03432	**0.03081**
	RMSE	0.04517	**0.04342**
	R^2	0.44475	**0.51862**

Tabela 5.6: Resultado dos testes executados para determinar a quantidade de camadas à retreinar utilizando M = 7 e a base Delhi.

5.3 Consolidação dos Resultados

Aqui, para comparar os resultados da TL pré-treinada através da base de mesmo contexto e de contextos diferentes, foi feito um processo sistemático, que incluiu: 1) Ganho de tempo; 2) Perda de eficácia do modelo comparada através do valor de MSE, MAE, RMSE e R^2 da melhor execução.

Os testes feitos podem ser separados em 2 blocos, os quais utilizaram um valor de M = 3 e M = 7. Esses diferentes valores trouxeram a possibilidade de analisar mais uma questão de pesquisa, qual o impacto do valor de M no desempenho do modelo?

No comparativo de tempo de treinamento, embora as duas aplicações de TL tenham fornecido uma redução de pelo menos 27% com M = 3 e 3% com M = 7, houve maior economia em relação ao tempo de processamento daquela cujo contexto de pré-treino era mais similar ao da base final, chegando a 54% com M = 3 e 41% com M = 7, como apresentado nas Tabelas 5.7 e 5.8.

Utilizando M = 3		Tempo de Treinamento em Segundos			
		Com TL		% Redução de tempo de treino c/ TL	
Conjunto de Dados	Sem TL	São Frascisco	Delhi	São Frascisco	Delhi
Paraíba do Sul	882.87	476.26	648.30	46%	27%
Zambeze	1149.26	531.77	597.16	54%	48%

Tabela 5.7: Resultado dos testes executados para apresentar o tempo de processamento com e sem TL utilizando M = 3.

Utilizando M = 7		Tempo de Treinamento em Segundos			
		Com TL		% Redução de tempo de treino c/ TL	
Conjunto de Dados	Sem TL	São Frascisco	Delhi	São Frascisco	Delhi
Paraíba do Sul	665.06	510.74	647.22	23%	3%
Zambeze	885.57	522.14	809.93	41%	9%

Tabela 5.8: Resultado dos testes executados para apresentar o tempo de processamento com e sem TL utilizando M = 7.

Já no comparativo em relação a qualidade da solução, foi observado que o valor de M interferiu diretamente no desempenho do TL. Utilizando M = 3, as bases de pré-treinamento cujo contexto era diferente obtiveram um melhor resultado, obtendo-se uma redução de até 48% no tempo de treinamento perdendo apenas 0.18% de desempenho em relação a métrica R^2 utilizando os dados do Zambeze. De maneira análoga, podemos observar que utilizando os dados do Paraíba do Sul, obtivemos uma redução do tempo de treinamento de 27% perdendo somente 0.31% de desempenho em relação a métrica R^2. Os melhores resultados de desempenho obtidos são apresentados em negrito na Tabela 5.9.

Para um valor de M = 7, as bases de contexto semelhantes se sobresaíram, e foi obtida uma redução de até 41% no tempo de treinamento perdendo apenas 3% em

relação a métrica R^2 utilizando os dados do Zambeze. Já utilizando os dados do Paraíba do Sul, obtivemos uma redução de até 23% do tempo de treinamento perdendo 16% de desempenho em relação a métrica R^2. Os melhores resultados obtidos são apresentados na Tabela 5.10.

Melhores Resultados Obtidos						
Utilizando M = 3			Com TL		Perda de Desempenho % Com TL	
Conjunto de Dados	Métrica	Sem TL	São Frascisco	Delhi	São Frascisco	Delhi
Paraíba do Sul	MSE	0.000239	0.000291	0.000247	22%	**3%**
	MAE	0.000859	0.010127	0.008707	1079%	**914%**
	RMSE	0.015474	0.017067	0.015721	10%	**2%**
	R^2	0.910523	0.891152	0.907643	2%	**0.31%**
Zambeze	MSE	0.001541	0.001588	0.001545	3%	**0.25%**
	MAE	0.027332	0.028746	0.028461	5%	**4%**
	RMSE	0.039259	0.039858	0.039308	1%	**0.12%**
	R^2	0.580611	0.567729	0.579558	2%	**0.18%**

Tabela 5.9: Resultado dos testes executados para apresentar o tempo de processamento com e sem TL, utilizando M = 3.

Melhores Resultados Obtidos						
Utilizando M = 7			Com TL		Perda de Desempenho % Com TL	
Conjunto de Dados	Métrica	Sem TL	São Frascisco	Delhi	São Frascisco	Delhi
Paraíba do Sul	MSE	0.000269	0.000703	0.001300	**161%**	354%
	MAE	0.010176	0.020624	0.032943	**102%**	223%
	RMSE	0.016422	0.026530	0.036063	**61%**	113%
	R^2	0.899236	0.736991	0.514022	**16%**	38%
Zambeze	MSE	0.001450	0.001569	0.001885	**8%**	29%
	MAE	0.027217	0.028474	0.030815	**4%**	8%
	RMSE	0.038088	0.039614	0.043426	**4%**	14%
	R^2	0.605257	0.572997	0.518617	**3%**	9%

Tabela 5.10: Resultado dos testes executados para apresentar o tempo de processamento com e sem TL, utilizando M = 7.

Por fim, podemos observar que para um valor de M = 3, o contexto das bases de pré-treinamento do TL não foi relevante para o desempenho final, onde bases de contexto diferente obtiveram os melhores resultados. Já para um maior valor de M = 7, o contexto dos dados de pré-treinamento teve impacto direto no desempenho final, onde o TL pré-treinado em bases de mesmo contexto se sobresairam. Além disso, é válido comentar que com o uso de TL, sempre obtivemos redução de custo computacional, sendo bem significativos em alguns casos.

6 Conclusões

Nesse trabalho, foram desenvolvidos modelos de ML utilizando TL, onde o principal objetivo era analisar e investigar o seu desempenho em relação ao tempo gasto e resultados obtidos, visando responder as seguintes questões de pesquisa:

- É possível se obter modelos acurados e de baixo custo computacional utilizando TL?
- Como o contexto dos dados de pré-treinamento impacta no desempenho final do modelo?

Ainda, vale-se ressaltar que, foram desenvolvidos modelos sem o uso do TL, que serviram como comparação tanto do tempo de treinamento quanto de desempenho. Dessa forma, podemos afirmar que o uso do TL foi capaz de oferecer uma economia no tempo de treino sem diminuir significativamente o desempenho do modelo, atingindo uma redução de até 48% no tempo de treinamento e perdendo somente 2% de desempenho em relação à métrica R^2.

Além disso, foi notado que o uso do TL pré-treinados em bases de mesmo contexto obtiveram uma maior redução no tempo de treinamento quando comparadas ao pré-treinamendo com bases de contextos diferentes, atingindo uma diferença percentual de pelo menos 6%, chegando até 32%.

6.1 Trabalhos Futuros

Como trabalhos futuros temos como meta tornar os modelos de TL desenvolvidos robustos a dados faltantes e/ou ruidosos. Também espera-se definir uma arquitetura para cada um dos modelos utilizados e avaliar o desempenho dos resultados obtidos.

Além disso, é tido como trabalho futuro a comparação dos resultados obtidos com outros trabalhos encontrados na literatura que utilizem os dados do Paraíba do Sul e Zambeze. Também espera-se testar outras janelas de tempo de entrada e de saída para os modelos, tornando possível outros tipos de análise, como por exemplo, a análise

da sazonalidade anual, onde é fornecido como janela de entrada os dados dos 365 dias anteriores.

Por fim, é tido como trabalho futuro o uso de séries temporais multivariadas (série temporal contendo mais de uma informação ao longo do tempo) como entrada para o modelo, a avaliação do TL utilizando dados de outros contextos, como energia eólica e radiação solar.

Espera-se que, com mais informações fornecidas, o modelo seja capaz de extrair e aprender uma maior quantidade de padrões, e consequentemente atingir uma maior acurácia.

Bibliografia

ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: ⟨https://www.tensorflow.org/⟩.

ABIODUN, O. I. et al. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, Elsevier, v. 4, n. 11, p. e00938, 2018.

ABREU, G. D. de. *MultiTask Learning para Previsão de Vazão*. XXX: [s.n.], 2019.

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

AGATONOVIC-KUSTRIN, S.; BERESFORD, R. Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis*, v. 22, n. 5, p. 717–727, 2000. ISSN 0731-7085. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0731708599002721⟩.

ÁGUAS, A. N. de. *HidroWeb. Sistema de informações Hidrológicas*. [S.l.]: ANA Brasília, 2020.

ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: IEEE. *2017 international conference on engineering and technology (ICET)*. [S.l.], 2017. p. 1–6.

ANA. *Sistema de Hidro-Metria*. 2022. ⟨http://www.snirh.gov.br/hidrotelemetria/Mapa.aspx⟩. [Online; accessed 17-December-2022].

ANALYSIS, R. T. S.; FORECASTING. *How To Isolate Trend, Seasonality And Noise From A Time Series*. 2022. ⟨https://timeseriesreasoning.com/contents/time-series-decomposition/⟩. [Online; accessed 17-January-2023].

AYYADEVARA, V. K. Convolutional neural network. In: *Pro Machine Learning Algorithms*. [S.l.]: Springer, 2018. p. 179–215.

BANGALORE, P. U. *Kagle data base climate changes*. 2019. ⟨https://www.kaggle.com/datasets/sumanthvrao/daily-climate-time-series-data⟩. [Online; accessed 17-December-2022].

BASSA, C. H. de C. *Company's official web site*. 2022. ⟨https://www.hcb.co.mz/⟩. [Online; accessed 17-December-2022].

BILBAO, I.; BILBAO, J. Overfitting problem and the over-training in the era of data: Particularly for artificial neural networks. In: IEEE. *2017 eighth international conference on intelligent computing and information systems (ICICIS)*. [S.l.], 2017. p. 173–177.

BOX, G. E. et al. *Time series analysis: forecasting and control*. [S.l.]: John Wiley & Sons, 2015.

BOZINOVSKI, S. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica*, v. 44, n. 3, 2020.

BROWNLEE, J. *Introduction to time series forecasting with python: how to prepare data and develop models to predict the future*. [S.l.]: Machine Learning Mastery, 2017.

CHICCO, D.; WARRENS, M. J.; JURMAN, G. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *PeerJ Computer Science*, PeerJ Inc., v. 7, p. e623, 2021.

CHOLLET, F. et al. *Keras*. [S.l.]: GitHub, 2015. ⟨https://github.com/fchollet/keras⟩.

COOK, D.; FEUZ, K. D.; KRISHNAN, N. C. Transfer learning for activity recognition: A survey. *Knowledge and information systems*, Springer, v. 36, n. 3, p. 537–556, 2013.

DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

EDOSSA, D. C.; BABEL, M. S. Application of ann-based streamflow forecasting model for agricultural water management in the awash river basin, ethiopia. *Water resources management*, Springer, v. 25, n. 6, p. 1759–1773, 2011.

EPE. *Empresa de Pesquisa Energética*. 2022. ⟨https://www.epe.gov.br/pt/abcdenergia/matriz-energetica-e-eletrica⟩. [Online; accessed 17-December-2022].

FENG, J.; LU, S. Performance analysis of various activation functions in artificial neural networks. In: IOP PUBLISHING. *Journal of physics: conference series*. [S.l.], 2019. v. 1237, n. 2, p. 022030.

GALVÃO, T. F.; PANSANI, T. d. S. A.; HARRAD, D. Principais itens para relatar revisões sistemáticas e meta-análises: A recomendação prisma. *Epidemiologia e serviços de saúde*, SciELO Public Health, v. 24, p. 335–342, 2015.

GERSHENFELD, N. An experimentalist's introduction to the observation of dynamical systems. In: *Directions in Chaos—Volume 2*. [S.l.]: World Scientific, 1988. p. 310–353.

GU, J. et al. Recent advances in convolutional neural networks. *Pattern recognition*, Elsevier, v. 77, p. 354–377, 2018.

HARRIS, C. R. et al. Array programming with NumPy. *Nature*, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: ⟨https://doi.org/10.1038/s41586-020-2649-2⟩.

HERSCHY, R. W. Chapter 10. weirs and flumes. *Streamflow Measurement; CRC Press: Boca Raton, FL, USA*, v. 3, 2014.

HUBEL, D. H.; WIESEL, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, Wiley Online Library, v. 195, n. 1, p. 215–243, 1968.

HUSSAIN, D. et al. A deep learning approach for hydrological time-series prediction: A case study of gilgit river basin. *Earth Science Informatics*, Springer, v. 13, n. 3, p. 915–927, 2020.

INVESTING.COM. *Crypto currencie monitor*. 2022. ⟨https://www.investing.com/crypto/bitcoin/historical-data⟩. [Online; accessed 17-December-2022].

IORIS, A. A. Os limites políticos de uma reforma incompleta: a implementação da lei dos recursos hídricos na bacia do paraíba do sul. *Revista Brasileira de Estudos Urbanos e Regionais*, v. 10, n. 1, p. 61–61, 2008.

JAIN, A. K.; MAO, J.; MOHIUDDIN, K. M. Artificial neural networks: A tutorial. *Computer*, IEEE, v. 29, n. 3, p. 31–44, 1996.

JAIN, L. C. et al. A review of online learning in supervised neural networks. *Neural computing and applications*, Springer, v. 25, n. 3, p. 491–509, 2014.

KARLIK, B.; OLGAC, A. V. Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, Citeseer, v. 1, n. 4, p. 111–122, 2011.

KELMAN, J. Water supply to the two largest brazilian metropolitan regions. *Aquatic Procedia*, Elsevier, v. 5, p. 13–21, 2015.

KIRANYAZ, S. et al. 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, Elsevier, v. 151, p. 107398, 2021.

KITAGAWA, G. *Introduction to time series modeling*. [S.l.]: Chapman and Hall/CRC, 2010.

KOPRINSKA, I.; WU, D.; WANG, Z. Convolutional neural networks for energy time series forecasting. In: IEEE. *2018 international joint conference on neural networks (IJCNN)*. [S.l.], 2018. p. 1–8.

LE, X.-H. et al. Comparison of deep learning techniques for river streamflow forecasting. *IEEE Access*, IEEE, v. 9, p. 71805–71820, 2021.

LECUN, Y. et al. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, v. 2, 1989.

LIVIERIS, I. E.; PINTELAS, E.; PINTELAS, P. A cnn–lstm model for gold price time-series forecasting. *Neural computing and applications*, Springer, v. 32, n. 23, p. 17351–17360, 2020.

MARTINHO, A. et al. Extreme learning machine with evolutionary parameter tuning applied to forecast the daily natural flow at cahora bassa dam, mozambique. In: _____. [S.l.: s.n.], 2020. p. 255–267. ISBN 978-3-030-63709-5.

MILLSTEIN, F. *Convolutional neural networks in Python: beginner's guide to convolutional neural networks in Python*. [S.l.]: Frank Millstein, 2020.

NAM, J.; KIM, S. Heterogeneous defect prediction. In: *Proceedings of the 2015 10th joint meeting on foundations of software engineering*. [S.l.: s.n.], 2015. p. 508–519.

O'MALLEY, T. et al. *KerasTuner*. 2019. ⟨https://github.com/keras-team/keras-tuner⟩.

ONS. *Operador nacional do sistema elétrico*. 2022. ⟨https://www.ons.org.br/Paginas/resultados-da-operacao/historico-da-operacao/dados_hidrologicos_vazoes.aspx⟩. [Online; accessed 17-December-2022].

O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

PAULSON, R. W. Hydrologic data collection for river basin management. In: _____. *Data Sharing for International Water Resource Management: Eastern Europe, Russia and the CIS*. Dordrecht: Springer Netherlands, 1999. p. 21–44. ISBN 978-94-017-1209-5. Disponível em: ⟨https://doi.org/10.1007/978-94-017-1209-5_3⟩.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PES. *PES University Web Site*. 2022. ⟨https://pes.edu/⟩. [Online; accessed 17-December-2022].

PRECHELT, L. Early stopping-but when? In: *Neural Networks: Tricks of the trade*. [S.l.]: Springer, 1998. p. 55–69.

RESEARCH, G. *Google*. 2022. ⟨https://colab.research.google.com/notebooks/intro.ipynb⟩. [Online; accessed 17-December-2022].

SAHOO, B. B. et al. Long short-term memory (lstm) recurrent neural network for low-flow hydrological time series forecasting. *Acta Geophysica*, Springer, v. 67, n. 5, p. 1471–1481, 2019.

SAINATH, T. N. et al. Convolutional, long short-term memory, fully connected deep neural networks. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2015. p. 4580–4584.

SARAIVA, S. V. et al. Daily streamflow forecasting in sobradinho reservoir using machine learning models coupled with wavelet transform and bootstrapping. *Applied Soft Computing*, Elsevier, v. 102, p. 107081, 2021.

SHAZEER, N. et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

SHEN, F.; LIU, J.; WU, K. Multivariate time series forecasting based on elastic net and high-order fuzzy cognitive maps: a case study on human action prediction through eeg signals. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 29, n. 8, p. 2336–2348, 2020.

SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

TAN, C. et al. A survey on deep transfer learning. In: SPRINGER. *International conference on artificial neural networks*. [S.l.], 2018. p. 270–279.

TEAM, T. pandas development. *pandas-dev/pandas: Pandas*. Zenodo, 2020. Disponível em: ⟨https://doi.org/10.5281/zenodo.3509134⟩.

TENSORFLOW_DOCUMENTATION. 2022. ⟨https://www.tensorflow.org/tutorials/structured_data/time_series⟩.

TSANTEKIDIS, A. et al. Forecasting stock prices from the limit order book using convolutional neural networks. In: *2017 IEEE 19th Conference on Business Informatics (CBI)*. [S.l.: s.n.], 2017. v. 01, p. 7–12.

UFOP, D. *Deep Learning E Industria 4.0: Introdução Às Redes Neurais Convolucionais*. 2022. ⟨http://www2.decom.ufop.br/imobilis/deep-learning-e-a-industria-4-0-introducao-as-redes-neurais-convolucionais/⟩. [Online; accessed 17-December-2022].

UNICEF. *Objetivos de Desenvolvimento Sustentável, Ainda é possível mudar 2030*. 2015. ⟨https://www.unicef.org/brazil/objetivos-de-desenvolvimento-sustentavel⟩. [Online; accessed 17-December-2022].

WASKOM, M. L. seaborn: statistical data visualization. *Journal of Open Source Software*, The Open Journal, v. 6, n. 60, p. 3021, 2021. Disponível em: ⟨https://doi.org/10.21105/joss.03021⟩.

WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. *Journal of Big data*, SpringerOpen, v. 3, n. 1, p. 1–40, 2016.

WEISSTEIN, E. W. Convolution. *https://mathworld. wolfram. com/*, Wolfram Research, Inc., 2003.

WU, H.; GU, X. Towards dropout training for convolutional neural networks. *Neural Networks*, v. 71, p. 1–10, 2015. ISSN 0893-6080. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0893608015001446⟩.

YASEEN, Z. M. et al. Artificial intelligence based models for stream-flow forecasting: 2000–2015. *Journal of Hydrology*, Elsevier, v. 530, p. 829–844, 2015.

ZHANG, Q. et al. Waste image classification based on transfer learning and convolutional neural network. *Waste Management*, Elsevier, v. 135, p. 150–157, 2021.

ZHAO, B. et al. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, BIAI, v. 28, n. 1, p. 162–169, 2017.