UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO ACADÊMICO DO AGRESTE
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA CIVIL E AMBIENTAL

ERICKSON JOHNY GALINDO DA SILVA

PREDIÇÃO DE VAZÃO AFLUENTE DA BARRAGEM DE JUCAZINHO UTILIZANDO TÉCNICAS DE APRENDIZADO DE MÁQUINA

Caruaru
2024

ERICKSON JOHNY GALINDO DA SILVA

PREDIÇÃO DE VAZÃO AFLUENTE DA BARRAGEM DE JUCAZINHO UTILIZANDO TÉCNICAS DE APRENDIZADO DE MÁQUINA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil e Ambiental da Universidade Federal de Pernambuco, Centro Acadêmico do Agreste, como requisito parcial para obtenção do título de mestre em Recursos Hídricos. Área de concentração: Tecnologia Ambiental.

Orientador: Artur Paiva Coutinho

Coorientador: Saulo de Tarso Marques Bezerra

Caruaru

2024

Catalogação na fonte:
Bibliotecária – Nasaré Oliveira - CRB/4 - 2309

S586p Silva, Erickson Johny Galindo da.
Predição de vazão afluente da barragem de Jucazinho utilizando técnicas de aprendizado de máquina. / Erickson Johny Galindo da Silva. – 2024.
66 f.; il.: 30 cm.

Orientador: Artur Paiva Coutinho.
Coorientador: Saulo de Tarso Marques Bezerra.
Dissertação (Mestrado) – Universidade Federal de Pernambuco, CAA, Programa de Pós- Graduação em Engenharia Civil e Ambiental, 2024.
Inclui Referências.

1. Máquina de vetores de suporte. 2. Redes neurais (Computação). 3. Modelagem hidrológica. 4. Chuvas. 5. Vazão. I. Coutinho, Artur Paiva (Orientador). II. Bezerra, Saulo de Tarso Marques (Coorientador). III. Título.

CDD 620 (23. ed.) UFPE (CAA 2024-030)

ERICKSON JOHNY GALINDO DA SILVA

PREDIÇÃO DE VAZÃO AFLUENTE DA BARRAGEM DE JUCAZINHO UTILIZANDO TÉCNICAS DE APRENDIZADO DE MÁQUINA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil e Ambiental da Universidade Federal de Pernambuco, Centro Acadêmico do Agreste, como requisito parcial para obtenção do título de mestre em Recursos Hídricos. Área de concentração: Tecnologia Ambiental.

Aprovado em: 15/04/2024.

BANCA EXAMINADORA

_____
Profº. Dr. Artur Paiva Coutinho (Orientador)
Universidade Federal de Pernambuco

_____
Profº. Dr. Alessandro Romario Echevarria Antunes (Examinador Externo)
Universidade Federal de Pernambuco

_____
Profº. Dr. Antonio Celso Dantas Antonino (Examinador Externo)
Universidade Estadual de Pernambuco

Dedico este trabalho a todos que contribuíram direta ou indiretamente para a realização desse projeto. Nesta dedicatória, faço menção especial aos meus pais, Sandra e Haroldo.

AGRADECIMENTOS

Agradeço à Universidade Federal de Pernambuco - Centro Acadêmico do Agreste.

Agradeço ao Programa de Pós-Graduação em Engenharia Civil e Ambiental da Universidade Federal de Pernambuco.

Agradeço aos meus professores da área de Recursos Hídricos, Prof. Saulo, Prof. Almir e Prof. Artur.

Agradeço ao meu orientador, Prof. Artur.

Agradeço ao coorientador, Prof. Saulo.

Agradeço à FACEPE.

Agradeço à COMPESA.

RESUMO

A história secular da construção de barragens, desde a Barragem Saad el-Kafara até a expansão global na década de 1950, destaca a importância dessas estruturas na gestão dos recursos hídricos. A Barragem de Jucazinho, construída em 1998, surgiu como resposta à escassez de água no Agreste, em Pernambuco. Embora tenha enfrentado colapso em 2016, a barragem se recuperou em 2020 após intervenções da concessionária de água local. Nesse contexto, a confiabilidade de modelos de predição de vazão afluente em barragens torna-se crucial para os gestores. Este estudo propôs um modelo hidrológico baseado em inteligência artificial com o objetivo de gerar séries de vazão e avaliar a adaptabilidade desses modelos na operação da Barragem de Jucazinho. A metodologia adotada utilizou o Python 3.7.13 no ambiente Jupyter Notebook 6.4.12, aproveitando a clareza, simplicidade e extensa biblioteca da linguagem. A normalização dos dados entre 0 e 1 foi aplicada para evitar o predomínio de variáveis com valores elevados. O modelo, baseado em aprendizado de máquina, empregou máquinas de vetores de suporte (MVS), floresta aleatória (FA) e redes neurais artificiais (RNA) fornecidas pela biblioteca Sklearn. A seleção das estações de monitoramento ocorreu por meio do portal HIDROWEB, da Agência Nacional de Águas e Saneamento (ANA), utilizando a correlação de Spearman para identificar a relação entre precipitação e vazão. A avaliação do desempenho dos modelos envolveu análise gráfica e critérios estatísticos como o coeficiente de eficiência do modelo de Nash-Sutcliffe (NSE), o Percentual de Viés (PBIAS), o Coeficiente de Determinação (R^2) e a Razão do Desvio Padrão da Raiz da Média (RSR). Os resultados dos coeficientes estatísticos para os dados de teste indicaram um desempenho insatisfatório para as previsões de longo prazo (8, 16 e 32 dias à frente), revelando uma tendência de queda na qualidade do ajuste com o aumento do horizonte de previsão. O modelo de MVS destacou-se, obtendo os melhores índices de NSE, PBIAS, R^2 e RSR. Os resultados gráficos dos modelos MVS mostraram uma subestimação dos valores de vazão com o aumento do horizonte de previsão, devido à sensibilidade da MVS a padrões complexos nas séries temporais. Por outro lado, os modelos de FA e RNA mostraram uma hiperestimação dos valores de vazão à medida que o número de dias previstos aumentou, atribuído principalmente ao *overfitting*. Em síntese, este estudo destaca a relevância da inteligência artificial na previsão de vazão para a gestão eficiente de barragens, especialmente em cenários de escassez hídrica.

A escolha adequada dos modelos e a garantia de dados de entrada confiáveis são cruciais para a obtenção de previsões precisas, contribuindo para a segurança hídrica e a operação efetiva de barragens como a de Jucazinho.

Palavras-chave: Máquina de vetores de suporte. Florestas aleatórias. Redes neurais. Modelagem hidrológica. Chuva. Vazão.

ABSTRACT

The centuries-old history of dam construction, from the Saad el-Kafara Dam to global expansion in the 1950s, highlights the importance of these structures in water resource management. The Jucazinho Dam, built in 1998, emerged as a response to the scarcity of water in the Agreste region of Pernambuco, Brazil. Although it faced collapse in 2016, the dam recovered in 2020 after interventions by local water utility. In this context, the reliability of influent flow prediction models in dams becomes crucial for managers. This study proposed a hydrological model based on artificial intelligence aiming to generate flow series and evaluate the adaptability of these models in the operation of the Jucazinho Dam. The methodology adopted used Python 3.7.13 in the Jupyter Notebook 6.4.12 environment, taking advantage of the clarity, simplicity and extensive library of the language. Data normalization between 0 and 1 was applied to avoid the predominance of variables with high values. The model, based on machine learning, employed support vector regression (SVM), random forest (RF) and artificial neural networks (ANN) provided by the Sklearn library. The selection of the monitoring stations took place via Brazilian National Water and Sanitation Agency's (ANA) HIDROWEB portal, using Spearman's correlation to identify the relationship between precipitation and flow. The evaluation of the performance of the models involved graphical analysis and statistical criteria such as Nash–Sutcliffe model efficiency coefficient (NSE), the Percentage of Bias (PBIAS), the Coefficient of Determination (R^2) and Root Mean Standard Deviation Ratio (RSR). The results of the statistical coefficients for test data indicated an unsatisfactory performance for long-term predictions (8, 16 and 32 days ahead), revealing a downward trend in the quality of the fit with the increase in the forecast horizon. The ANN model stood out, obtaining the best indices of NSE and R^2. The graphical results of the SVM models showed an underestimation of the flow values with the increase of the forecast horizon, due to the sensitivity of the SVM to complex patterns in the time series. On the other hand, the RF and ANN models showed a hyperestimation of the flow values as the number of forecast days increased, mainly attributed to overfitting. In summary, this study highlights the relevance of artificial intelligence in flow prediction for the efficient management of dams, especially in water scarcity scenarios. Proper choice of models and ensuring reliable input data are crucial for obtaining accurate forecasts, contributing to the water security and effective operation of dams such as Jucazinho.

Keywords: Support vector machine. Random forests. Neural networks. Hydrological modeling. Rainfall. Flow.

SUMÁRIO

1 INTRODUÇÃO

Diversos arqueólogos consideram a barragem Saad el-Kafara como uma das primeiras do mundo. Ela provavelmente foi construída no reinado de Khufu, rei do Egito, por volta de 2900-2877 a.C. (JANSEN, 1980). Desde então, há uma longa tradição de construção de barragens que se estende por milênios, com propósitos diversos, tais como, controlar enchentes, prover água para consumo humano, irrigação e dessedentação de animais, e, mais recentemente na história das barragens, para fins industriais e geração de energia elétrica.

À medida que se chegou à década de 1950, período de grande expansão das populações e economias globais, as barragens começaram a ser cada vez mais consideradas como uma solução para atender às crescentes demandas de água e energia. Desde então, de acordo com o *World Commission on Dams* (WCD, 2000), foram erguidas pelo menos 45.000 grandes barragens em todo o mundo, estando quase metade dos rios globais com pelo menos uma barragem de grande porte em seu curso.

No Brasil, no município de Surubim, no estado de Pernambuco, foi inaugurada em 1998 a Barragem Jucazinho, localizada na bacia hidrográfica do Capibaribe e barrando o rio de mesmo nome. Ela foi construída com intuito de minimizar o cenário de escassez de água para abastecimento no agreste pernambucano e para o controle de cheias no rio Capibaribe.

De acordo com a ficha técnica da APAC (2020), a barragem possui capacidade de acumulação de 204,82 milhões de metros cúbicos de água na elevação 292 metros da soleira do vertedouro, e, como registra a ficha resumo da ANA (2017), possui 80% da demanda total de retirada para abastecimento urbano, 12% para abastecimento rural, 7% para dessedentação animal e 1% para irrigação. Esse montante de água possui relevância devido ao grande número de municípios que atende, um total de 15 cidades.

O Plano Hidroambiental da Bacia Hidrográfica do Capibaribe da SRH (2010) destaca as potencialidades econômicas dos municípios atendidos pela barragem, sendo nove pertencentes ao Polo Têxtil e de Confecções (Caruaru, Cumaru, Frei Miguelinho, Santa Maria do Cambucá, Riacho das Almas, Surubim, Toritama, Passira e Santa Cruz do Capibaribe), três com relevantes atividades agropecuárias (Casinhas,

Vertente do Lério e Vertentes), dois pertencentes ao Polo Moveleiro e Turístico (Gravatá e Bezerros) e um com forte indústria de turismo termal (Salgadinho).

Entretanto, mesmo com os esforços da Companhia Pernambucana de Saneamento (COMPESA) para combater a estiagem prolongada, a barragem colapsou em 2016, quando atingiu percentual de acumulação abaixo de 0,01%, e só voltou a recuperar-se quatro anos depois, em 2020, saindo do pré-colapso e atingindo percentual de acumulação acima de 1% (COMPESA, 2020).

Visando garantir a eficiência, e, por conseguinte, o funcionamento do sistema de abastecimento de água sem colapso do reservatório, Geogakakos et al. (2001) verificaram que a confiabilidade dos modelos de predição de vazão afluente, associados a sistemas de decisão adaptativos, são atributos chave para beneficiar substancialmente a performance do reservatório.

Nas últimas décadas, diversos modelos de inteligência artificial estão sendo utilizados por pesquisadores na predição de vazões afluentes devido à grande acurácia e flexibilidade de considerar processos físicos com todas as suas características. Entre os métodos utilizados destacam-se as redes neurais artificiais (RNA) (Nourani *et al*., 2013; Noori *et al.*, 2016), florestas aleatórias (Hussain *et al.*, 2020; Kashani *et al.*, 2016) e máquinas de vetores de suporte (MVS) (Sudheer *et al.*, 2014; Rasouli *et al.*, 2012; Yaseen *et al.*, 2016).

Com base na literatura, há modelos que se adequam de uma melhor maneira a depender da bacia hidrográfica. He et al. (2014) compararam modelos de RNA, Sistemas de Inferência Neuro-Fuzzy Adaptativo e MVS na predição de vazão de rios de uma região montanhosa semiárida e verificaram que o modelo MVS possui a melhor performance. Enquanto Tongal et al. (2018) verificaram os modelos de RNA, MVS e FA para quatro rios dos Estados Unidos, concluindo que o modelo RNA performou melhor em três sub-bacias e o modelo MVS foi melhor em apenas uma. Hussain et al. (2020) compararam os modelos Perceptron Multicamadas, MVS e FA para predição de vazão do rio Hunza no Paquistão, concluindo que o modelo FA performou melhor que os modelos MVS e PMC.

Portanto, a motivação desse estudo é a entrega de um modelo hidrológico para o gerenciamento de reservatórios, baseado em técnicas emergentes de inteligência artificial, agregando à literatura adequações de aprendizado de máquina para a predição de vazões afluentes e otimizando a operação de reservatórios de água.

A partir desse contexto, o presente trabalho objetiva efetuar a geração estocástica de séries de vazões afluentes para a Barragem de Jucazinho, na bacia hidrográfica do Capibaribe, avaliando a adaptabilidade de três modelos de aprendizado de máquina e a qualidade dos ajustes.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Avaliar a adaptabilidade dos modelos de aprendizado de máquina de vetores de suporte, floretas aleatórias e redes neurais, para gerar séries sintéticas de vazões afluentes para a Barragem de Jucazinho.

2.2 OBJETIVOS ESPECÍFICOS

- Analisar a influência na qualidade dos ajustes com o aumento da quantidade de dias de previsão; e,
- Verificar a qualidade dos ajustes, a partir de métricas estatísticas, e determinar o melhor modelo para as variáveis hidrológicas em estudo.

3 REVISÃO DA LITERATURA

3.1 MODELOS HIDROLÓGICOS

As ciências que se dedicam ao estudo da água na Terra exploram a sua distribuição, circulação, propriedades físicas e químicas e interação com o meio ambiente, incluindo os seres vivos. A hidrologia engloba todos esses tópicos, entretanto, mais precisamente, foca nas fases da água na Terra, estudando o ciclo hidrológico, que é a circulação infinita de água entre a Terra e a atmosfera (CHOW *et al*., 1988).

De forma geral, o ciclo hidrológico pode ser descrito a partir dos fenômenos de precipitação, interceptação em folhas e caules, transpiração dos vegetais, escoamento superficial e evaporação em qualquer tempo e local. E para simular o comportamento desses fenômenos dentro de uma bacia hidrográfica, usa-se os modelos hidrológicos, que são uma ferramenta para prever condições futuras, simular situações hipotéticas e avaliar impactos de alterações (TUCCI, 2001; ENOMOTO, 2004).

Com o intuito de prever o pico do escoamento de uma precipitação em uma bacia hidrográfica, os modelos iniciais utilizavam relações empíricas entre a vazão máxima e a área da bacia. Por partirem de dados observados em regiões específicas, não podiam ser utilizados com precisão em outras regiões. A reviravolta aconteceu na metade do século XIX quando o engenheiro irlandês John Edmund Mulaney publicou uma nova metodologia, sendo esta considerada um divisor de águas na modelagem hidrológica (TUCCI, 2005; TODINI, 2007).

Esta metodologia recebeu o nome de Método Racional de Mulaney (1851) por contrapor os métodos empíricos da época e propôs que a vazão de pico do escoamento superficial ocorreria para uma chuva com tempo de duração igual ao tempo de concentração da bacia, que corresponde ao tempo da trajetória de uma gota de água do ponto hidrológico mais distante do exultório até esse. Dessa forma, o tempo de concentração da bacia opera como sendo o tempo necessário para que toda a bacia contribua simultaneamente na seção de análise do exultório.

Como relata Fayal (2008), os próximos avanços significativos aconteceram a partir de 1930 com grandes países desenvolvendo seus programas de pesquisas hidrológicas. Entretanto, nessa época, os modelos tratavam componentes isolados do

ciclo hidrológico e não ele como um todo. Destaca-se o Hidrograma Unitário de Sherman (1932), o Modelo da Infiltração de Horton (1939) e o Modelo de Escoamento em Rios de McCarthy (1938).

A propulsão dos modelos hidrológicos aconteceu concomitantemente ao avanço da disponibilidade de computadores a partir do final da década de 1950. Com isso, os modelos passaram a trabalhar uma análise baseada em conceitos da física ao invés de análises estatísticas e empíricas (GALIZONI, 2021). As décadas de 60 e 70 foram marcadas por modelos com características singulares, como o Standford IV (Crawford, 1966). Já na década de 70, Ibbit (1970) propôs a estimativa dos parâmetros de um modelo hidrológico através da calibração automática, que envolveu o ajuste de várias combinações de séries de dados e uma análise minuciosa dos efeitos dos erros no processo de ajuste. Na década de 80, surgiram modelos mais eficientes com menor número de funções e parâmetros, como o IPH II (TUCCI *et al.*, 1981) e o Smap (LOPES, 1999). A partir da década de 90, com o avanço dos modelos climáticos globais (MCG), os hidrólogos verificaram que as quadrículas dos MCG eram muito maiores, logo, incompatíveis com as bacias usualmente simuladas na hidrologia, portanto, o desafio passou a ser a criação de modelos acoplados climáticos, hidrológicos e ambientais (TUCCI, 2005). Em meados de 2010, começaram a se popularizar os modelos para predição de variáveis hidrológicas a partir de métodos de inteligência artificial, devido à capacidade desses modelos em se adaptar a processos não-lineares e a capacidade de lidar efetivamente com um grande conjunto de dados dinâmicos e com picos concentrados (NOURANI, 2014).

Desde então, a modelagem hidrológica vem acompanhando os avanços da tecnologia, como o aumento da disponibilidade de dados hidrometeorológicos, a facilitação do levantamento de modelos digitais do terreno e a utilização de inteligências artificiais.

3.2 CLASSIFICAÇÃO DOS MODELOS HIDROLÓGICOS

Os modelos podem ser classificados com relação ao tipo de análise matemática das variáveis utilizadas na modelagem (estocástico ou determinístico), as relações entre as variáveis (conceitual, empírico e semiconceitual) e a quantidade numérica dos dados das variáveis (discreto ou contínuo) (MAIDMENT, 1993; VERTESSY *et al*., 1993; TUCCI, 2005).

Um modelo estocástico considera a aleatoriedade na análise de pelo menos uma das variáveis analisadas, utilizando conceitos da estatística probabilística, por essa razão, é também chamado de modelo probabilístico. Ele utiliza formas matemáticas para representar a distribuição da probabilidade de uma variável aleatória, que são os números reais que representam os resultados do modelo. Em contraponto, um modelo determinístico, também chamado de modelo físico, utiliza equações matemáticas para que a partir do conjunto de variáveis de entrada conhecidas seja possível obter um único conjunto de saídas (FAYAL, 2008). Exemplificando, considerando um simples modelo que relaciona chuva e vazão, o modelo estocástico irá determinar a equação da distribuição de probabilidade (binomial, Poisson, normal etc.) que melhor se adequa à variável aleatória em estudo, enquanto um modelo determinístico irá determinar a equação que relaciona as duas variáveis (Método Racional de Mulaney,1851).

Os modelos conceituais baseiam-se nas equações dos processos físicos (precipitação, evaporação, escoamento superficial etc.) que acontecem no fenômeno que está sendo estudado, portanto, são fisicamente embasados. Já o modelo empírico irá relacionar as variáveis através de equações matemáticas, sem embasar fisicamente o modelo (FAYAL, 2008). O modelo semiconceitual une ambos os conceitos, sendo este embasado em conceitos físicos, mas com parâmetros determinados a partir de relações matemáticas sem fisicidade. Por exemplo, o modelo semiconceitual Método Racional (Mulaney, 1851) relaciona fisicamente a área de contribuição e a intensidade da precipitação na bacia com a vazão máxima de escoamento, através de um coeficiente não físico determinado como coeficiente de escoamento.

Na matemática, uma variável discreta é finita e possui um número contável de observações, podendo ser entendida graficamente como um conjunto de pontos, enquanto uma variável contínua é infinita e pode ser entendida graficamente como uma curva contendo um número infinito de pontos. Dessa maneira, o modelo hidrológico discreto trata de eventos pontuais, como um evento de cheia que aconteceu em uma bacia hidrográfica, enquanto um modelo contínuo descreve toda a série de eventos, contemplando as secas e cheias, envolvendo épocas diferentes do ciclo hidrológico da bacia hidrográfica.

3.3 MACHINE LEARNING APLICADO À PREDIÇÃO DE VAZÕES

Com o objetivo de obter uma compreensão abrangente sobre o uso da inteligência artificial na predição de vazões em rios e bacias, realizou-se uma revisão sistemática da literatura especializada, visando identificar as pesquisas mais relevantes.

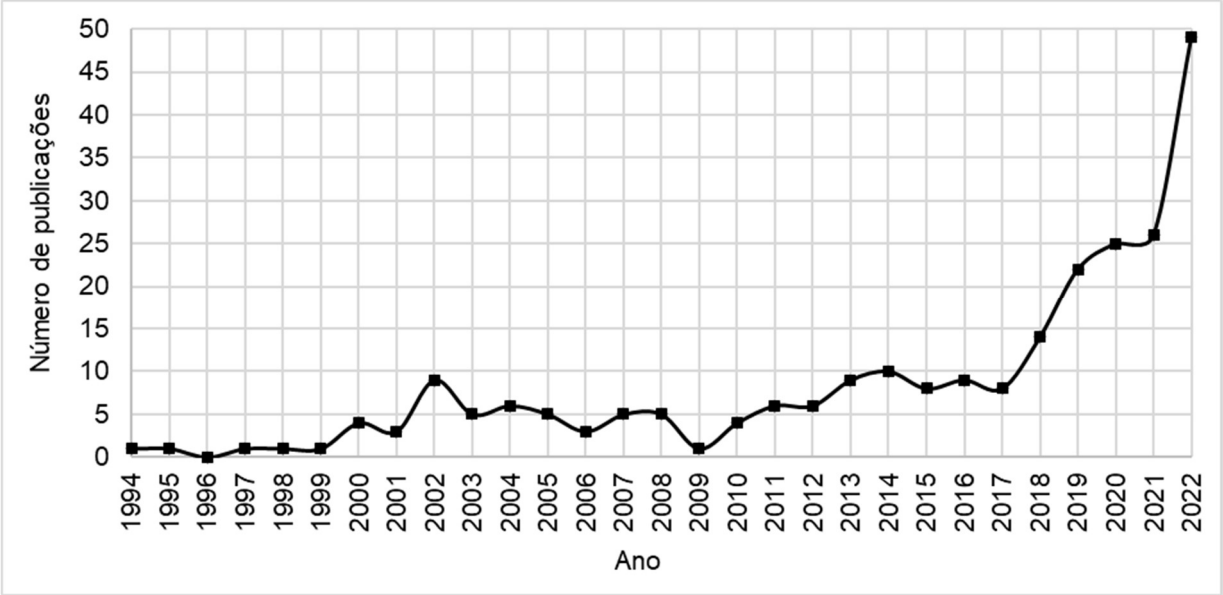
O processo de seleção e estudo dos trabalhos foi dividido em três etapas distintas: coleta de trabalhos em plataformas de pesquisa, seleção dos estudos pertinentes e uma análise detalhada da premissa de cada artigo utilizando o *software VOSviewer*. Por meio desse método, foi possível obter uma análise abrangente e aprofundada sobre a implementação e otimização de algoritmos para a predição de vazões.

Na seleção dos artigos relevantes, inicialmente foram utilizadas as palavras-chave "*machine learning*", "*support vector machine*", "*random forest*" e "*neural network*" na plataforma de pesquisa Scopus (*www.scopus.com*). A fim de refinar os parâmetros da pesquisa, foram adicionadas as palavras-chave "*forecasting*", "*prediction*", "*flow*", "*outflow*", "*streamflow*" e "*rainfall*". A busca foi limitada a artigos de periódicos em inglês.

Por meio do *software* VOSviewer, foi possível identificar os principais autores e referências com base na frequência de citações e na quantidade de links criados durante o processamento do *software*. Essa análise proporcionou uma visualização dos padrões de ocorrência e relevância das informações obtidas.

Ao analisar a Figura 1, que representa a quantidade de publicações sobre o tema em periódicos ao longo do tempo, observa-se um aumento significativo de trabalhos publicados a partir de 2017, atingindo o ápice em 2022. Esse padrão de crescimento reforça o caráter inovador da pesquisa e ressalta a importância do tema.

Figura 1 – Quantidade de publicações da temática em relação ao tempo.



Fonte: Autor (2024).

Analisando a Tabela 1, que apresenta em ordem decrescente as fontes científicas mais citadas, observa-se que o *Journal of Hydrology*, *Water Resources Research* e o *Journal of Hidrologic Engineering* são as fontes que lideram em termos de citações entre os artigos analisados. Isso indica que essas publicações têm uma grande relevância e reconhecimento no campo da pesquisa de predição de vazão de rios utilizando algoritmos de aprendizagem de máquina.

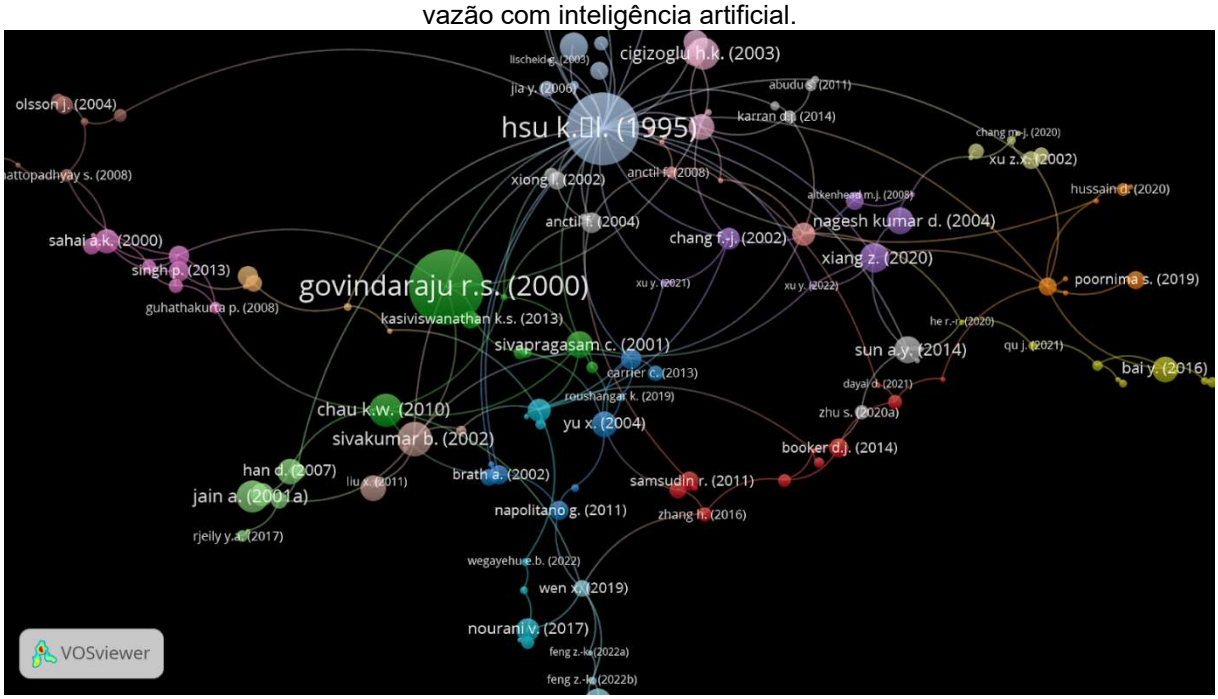
Tabela 1 – Fontes científicas mais citadas.

Fonte	Documentos	Citações	Força total do link
Jornal of Hydrology	49	2521	57
Water Resources Research	9	1649	42
Journal of Hydrologic Engineering	12	1561	18
Journal of Hydroinformatics	17	1163	37
Water Resources Management	26	901	27
Hydrological Sciences Jornal	10	678	9
Atmospheric Environment	3	440	0
Hydrological Processes	8	432	18
Hydrology and Earth System Sciences	11	410	24
Stochastic Environmental Research	8	371	15
Climate Dynamics	3	269	12
Metereology and Atmospheric Physics	4	177	7
Journal of Hydrometereology	4	168	1
International Journal of Climatology	2	157	3
Advances in Water Resources	2	157	0
Remote Sensing	9	150	1
Acta Geophysica	3	123	9
Journal of Earth System Science	4	121	4
Journal of the American Water Resources	4	118	5

A Figura 2 representa a rede das principais referências, em querroe a centralidade dos trabalhos é indicada pelos círculos, sendo que quanto maior o tamanho do círculo, maior o índice de centralidade da referência. Além disso, a quantidade de citações do artigo é representada pelo tamanho da fonte do trabalho, indicando a relevância do artigo. A centralidade, como mencionado, mede o grau em que um trabalho faz parte dos caminhos que conectam um par aleatório de trabalhos na rede (CHEN, 2006).

Em outras palavras, um trabalho com alta centralidade tende a estar no meio de conexões entre diferentes grupos de trabalhos. Isso indica que um trabalho com alta centralidade desempenha um papel importante na integração de diferentes ideias, conceitos ou abordagens na área de estudo. Esses trabalhos atuam como "pontes" entre diferentes grupos de trabalhos, permitindo a transferência de conhecimento e a interação entre diferentes perspectivas e linhas de pesquisa.

Figura 2 – Rede de citações gerada no VOSviewer com as principais referências de predição de vazão com inteligência artificial.



Fonte: Autor (2024).

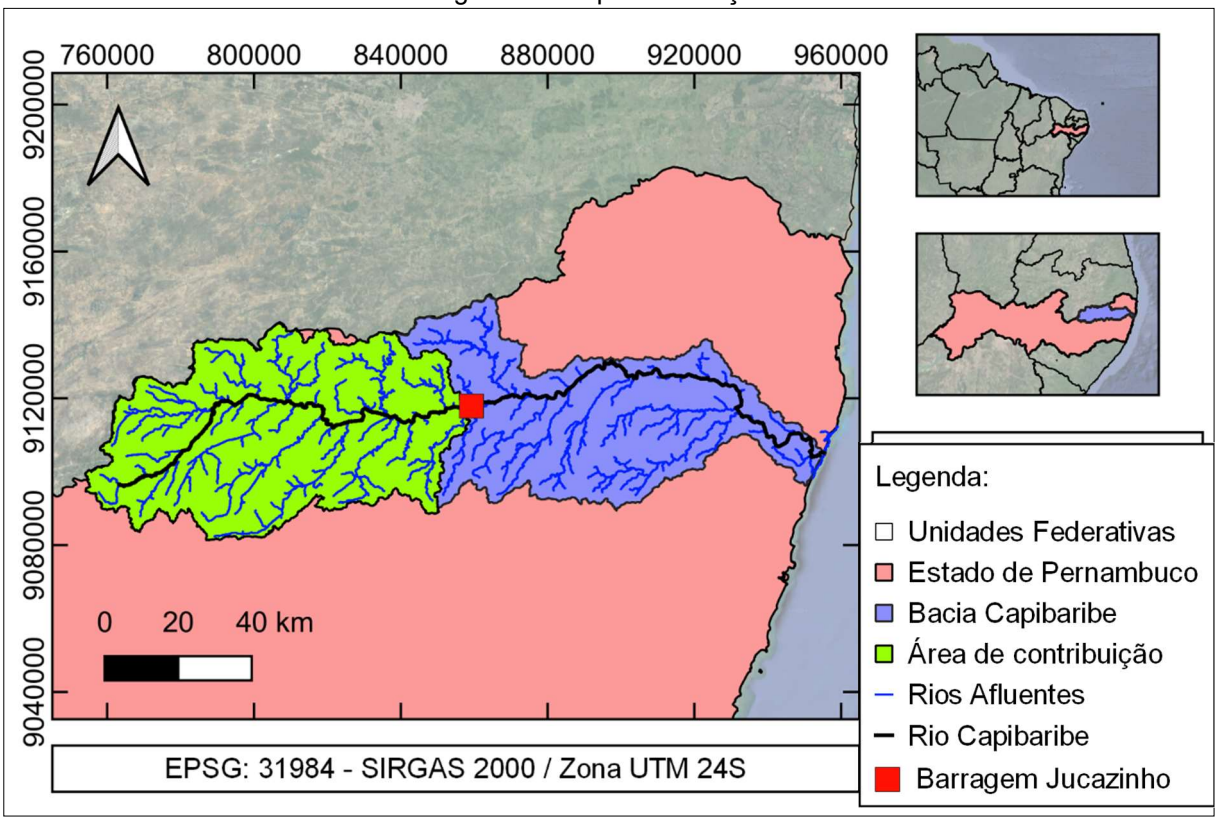
A representação visual da rede permite identificar as referências mais centrais citadas na área de predição de vazão de rios utilizando algoritmos de aprendizagem de máquina. Essas referências possuem um papel importante na geração de mais estudos na área e indicam trabalhos que têm impacto e reconhecimento significativos na comunidade científica.

Ao avaliar a centralidade da Figura 1, reforça-se a importância dos trabalhos de Govindaraju (2000) e HSU (1995) na centralidade da pesquisa. Esses estudos foram inovadores ao propor o uso de redes neurais artificiais para resolver problemas específicos na área de hidrologia, incluindo a predição de vazões, modelagem hidrológica e predição do processo chuva-vazão e fundamentais para o desenvolvimento de muitas pesquisas subsequentes na área. Eles abriram caminho para a aplicação de algoritmos de inteligência artificial na hidrologia e demonstraram sua eficácia na modelagem e predição de fenômenos complexos. A sua notoriedade indica que são referências importantes e influentes na comunidade científica, destacando-se como marcos na área de predição de vazão e no uso de algoritmos de inteligência artificial nesse contexto.

4 ÁREA DE ESTUDO

Localizada no município de Surubim, em Pernambuco, na bacia hidrográfica do Capibaribe, a barragem de Jucazinho barra o rio que também se chama Capibaribe, como se pode verificar no mapa de situação da Figura 3. Sua construção foi iniciada em 1995 e finalizada em 1998, e, na época, houve a desapropriação de mais de dois mil hectares nas áreas ribeirinhas, envolvendo cerca de 5.000 pessoas (GIRÃO, 2004).

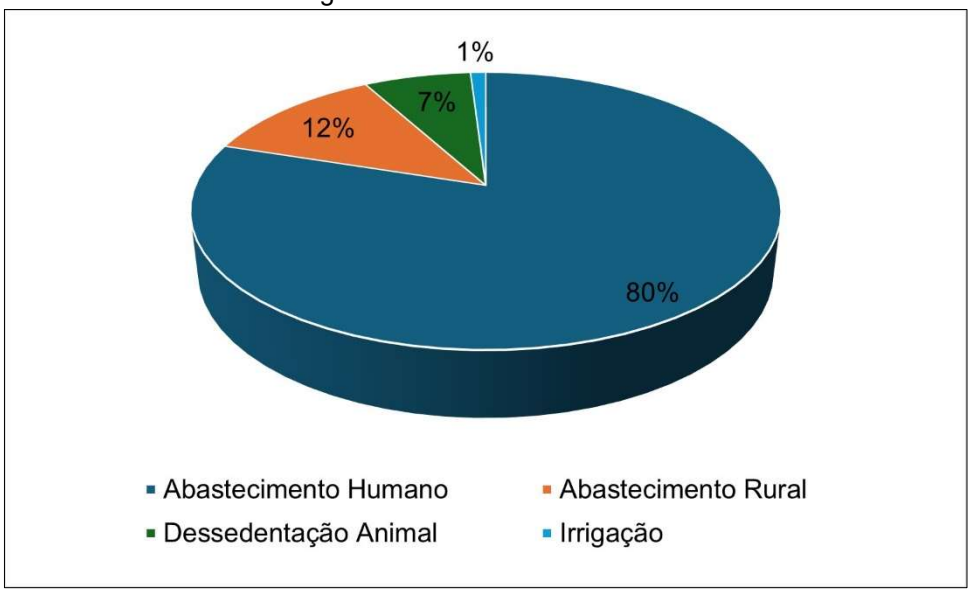
Figura 3 – Mapa de situação.



Fonte: Autor (2024).

Um dos motivos da construção da barragem foi o cenário de escassez de água para abastecimento no agreste pernambucano. Com a construção de Jucazinho, que tem capacidade máxima de acumulação de 245,26 milhões de metros cúbicos de água na cota máxima *maximorum* de 295 metros, 21 municípios puderam ser atendidos, o que impactou a vida de aproximadamente 800 mil habitantes. Como consta no Figura 4, 80% do uso da barragem é para abastecimento humano (SRH, 2010), além disso, ela é utilizada para piscicultura, técnicas pecuárias e agrícolas.

Figura 4 – Demandas totais de retirada.



Fonte: SRH (2010).

O outro motivo da construção da barragem foi o Plano de Controle de Cheias no Capibaribe, que consistiu na construção de diversas barragens a fim de proteger a Região Metropolitana de Recife (distante cerca de 135 km), Paudalho e Limoeiro de cheias históricas como as que aconteceram entre os anos de 1960 e 1980. A barragem possui um volume para amortecimento de enchentes de 100 x 106 m³ (SRH, 2010).

Seu tipo construtivo é gravidade em concreto compactado a rolo e possui um vertedouro central com degraus (*stepped spillway*), com uma bacia de dissipação do tipo salto de esqui, como apresenta a Figura 5. Existem também dois vertedouros laterais, conectados às ombreiras da barragem. Contém uma galeria, com acesso nas duas ombreiras, que se estende por todo o maciço da barragem. Acima do vertedouro central existe uma ponte que conecta as ombreiras. Tem uma tomada d'água para abastecimento e outra para liberação da vazão ecológica a jusante, ambas com uma tubulação de 2,0 m de diâmetro, com redução para 1,5 m. Na Tabela 2 é possível verificar a ficha técnica da barragem, com informações disponibilizadas pelo SHR (2010) e dados mensurados por Neves *et al*. (2021).

Figura 5 – Vista do paramento de jusante da barragem de Jucazinho.

Fonte: Autor (2021).

Tabela 2 – Ficha técnica da barragem de Jucazinho.

Nome	Dimensão	Unidade
Barramento		
Latitude	07° 57’ 59,39’’ S	-
Longitude	35° 44’ 3,16’’ W	-
Área de drenagem incremental	2.865,60	km^2
Área de drenagem total	4.149,90	km^2
Volume máximo*	204,82	hm^3
Volume mínimo	0,29	hm^3
Volume útil	326,75	hm^3
Nível de água máximo operacional	292,00	m
Nível de água mínimo operacional	253,00	m
Cota do fundo do lago	238,00	m
Coroamento		
Comprimento	442,00	m
Largura*	8,00	m

Nome	Dimensão	Unidade
Cota	299,00	m
Vertedor Principal		
Comprimento*	170,00	m
Cota da soleira	292,00	m
Distância entre a soleira e o coroamento	7,00	m
Vazão máxima*	5.446,69	$m^3 \cdot s^{-1}$
Vertedores Laterais		
Comprimento*	57,00	m
Cota da soleira*	295,00	m
Vazão máxima*	1.291,3	$m^3 \cdot s^{-1}$
Galeria		
Comprimento*	2	m
Largura*	2	m
Cota*	250	m
Vazão máxima*	2,72	$m^3 \cdot s^{-1}$

Fonte: SHRE (2010) e Neves *et al*. (2021)*.

5 METODOLOGIA

5.1 AMBIENTE DE TRABALHO

O modelo foi desenvolvido em *Python* versão 3.7.13, por meio da aplicação web *Jupyter Notebook* versão 6.4.12. O *Python* é uma linguagem de programação de alto nível, interpretada e orientada a objetos, que tem uma sintaxe clara e simples, ao mesmo tempo que oferece recursos avançados. O *Jupyter Notebook* é uma ferramenta de exploração de dados e prototipação rápida, pois permite que o usuário execute e teste o código de forma interativa, em pequenos blocos chamados de células. Cada célula pode ser executada separadamente, permitindo que o usuário verifique os resultados passo a passo e faça alterações no código em tempo real. As simulações foram realizadas em um notebook com processador Intel Core i7-8550U, com 1,80 GHz e 8 GB de memória RAM.

Uma das principais vantagens do Python é a grande quantidade de bibliotecas disponíveis, que permitem aos desenvolvedores aproveitar o trabalho de outros programadores para acelerar o processo de desenvolvimento e aumentar a funcionalidade do software. Para o aprendizado de máquina, foram utilizados os modelos de regressão vetores de suporte, árvores de decisão e redes neurais oferecidos pela biblioteca Sklearn. Também foram utilizadas funções das bibliotecas Numpy, Pandas, Matplotlib e Scipy que possuem uma breve descrição de suas respectivas funcionalidades no Quadro 1. Os códigos elaborados na linguagem de programação Python para desenvolvimento dessa dissertação, constam, na íntegra, no APÊNDICE A.

Quadro 1 – Descrição das bibliotecas utilizadas.

Biblioteca	Descrição
Numpy	Oferece uma estrutura de dados chamada de "array" que permite armazenar e manipular grandes conjuntos de dados homogêneos.
Pandas	O principal objeto fornecido pelo Pandas é o DataFrame, que pode ser visto como uma tabela de dados bidimensional, semelhante a uma planilha do Excel ou a uma tabela em um banco de dados relacional.
Matplotlib	Permite a criação de gráficos estáticos, gráficos interativos e visualizações personalizadas de dados de forma simples e eficaz.
Scipy	Fornece um conjunto de módulos que abrangem áreas como álgebra linear, otimização, interpolação, processamento de sinais, estatísticas, processamento de imagens, solução de equações diferenciais etc.
Hydroeval	Avalia a qualidade dos ajustes entre dados de vazões medidos e simulados.

5.2 DADOS DE ENTRADA

O princípio "*garbage in, garbage out*" (GIGO) refere-se à ideia fundamental de que a qualidade da saída de um sistema de processamento de dados é diretamente influenciada pela qualidade dos dados de entrada. Em outras palavras, se informações imprecisas, incompletas ou inadequadas forem inseridas em um sistema, é inevitável que a saída resultante também seja imprecisa ou de baixa qualidade. Esse princípio destaca a importância crítica da entrada de dados confiáveis e de alta qualidade para garantir resultados precisos e úteis em qualquer processo de computação ou tomada de decisão.

O GIGO serve como um lembrete essencial de que, independentemente da sofisticação do sistema ou algoritmo, a qualidade dos dados iniciais desempenha um papel crucial na eficácia global do processo. Portanto, um passo inicial importante para desenvolver o modelo hidrológico de previsão de vazão é a seleção das séries históricas de estações de monitoramento de precipitação e vazão que serão utilizadas.

A consulta das estações foi feita a partir do portal HIDROWEB, desenvolvido pela Agência Nacional de Águas e Saneamento Básico (ANA), que é uma plataforma on-line que fornece informações e dados relacionados aos recursos hídricos no Brasil. Esse sistema disponibiliza dados hidrometeorológicos, hidrográficos e de qualidade da água em tempo real, provenientes de uma extensa rede de estações de monitoramento distribuída pelo país. A partir da consulta foi identificada na área de contribuição da barragem Jucazinho a presença de 32 postos de monitoramento pluviométrico e 2 postos de monitoramento fluviométrico, conforme apresenta o Quadro 2.

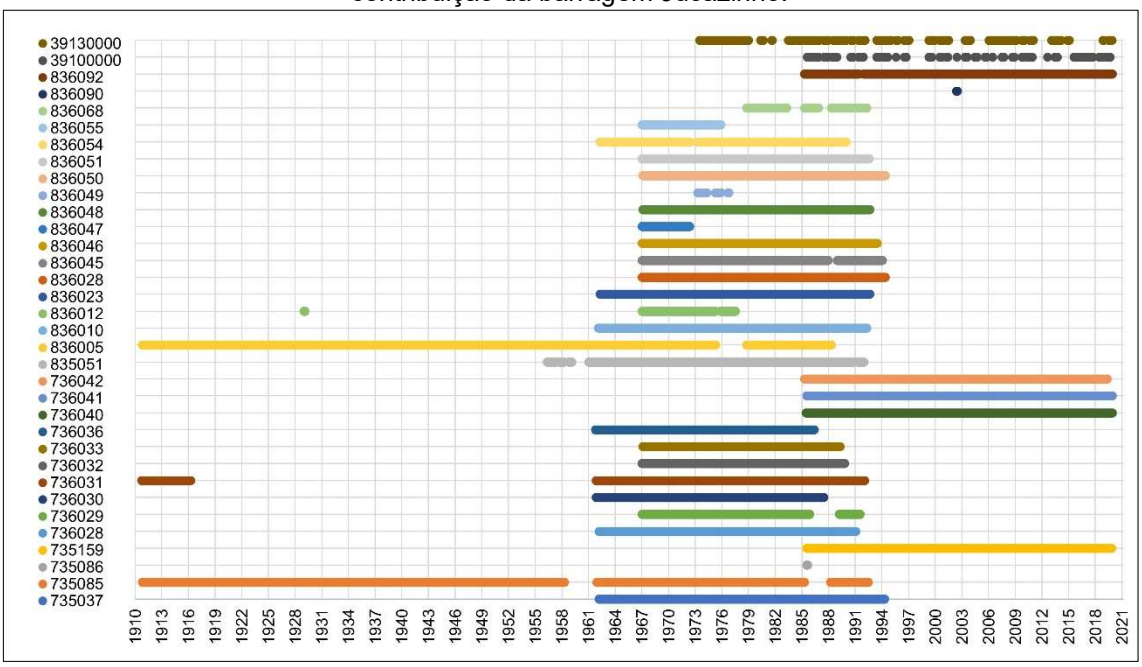
Quadro 2 – Estações pluviométricas e fluviométricas na área de contribuição da barragem Jucazinho.

Índice	Nome	Código	Latitude	Longitude	Tipo
1	Algodão do Manso (DNOCS)	735037	-7,96670	-35,8833	Pluviométrica
2	Vertentes	735085	-7,91670	-35,9833	Pluviométrica
3	Vertentes	735086	-7,91670	-35,9833	Pluviométrica
4	Vertentes	735159	-7,91000	-35,9886	Pluviométrica
5	Jataúba (Jatobá)	736028	-7,96667	-36,4833	Pluviométrica
6	Poço Fundo	736029	-7,93333	-36,3333	Pluviométrica
7	Santa Cruz do Capibaribe	736030	-7,95000	-36,2000	Pluviométrica
8	Taquaritinga do Norte	736031	-7,90000	-36,0500	Pluviométrica

Índice	Nome	Código	Latitude	Longitude	Tipo
9	Sítio Mulungu	736032	-7,88333	-36,3833	Pluviométrica
10	Sítio Salgado	736033	-7,96667	-36,4167	Pluviométrica
11	Vila do Pará	736036	-7,85000	-36,3667	Pluviométrica
12	Jataúba	736040	-7,98640	-36,5006	Pluviométrica
13	Santa Cruz do Capibaribe	736041	-7,96190	-36,2022	Pluviométrica
14	Taquaritinga do Norte	736042	-7,90390	-36,0469	Pluviométrica
15	Sítio Barriguda	835051	-8,10000	-35,8667	Pluviométrica
16	Brejo da Madre de Deus	836005	-8,15000	-36,3833	Pluviométrica
17	Carapotos (Riacho Doce)	836010	-8,13333	-36,0667	Pluviométrica
18	Fazenda Nova	836012	-8,16667	-36,2000	Pluviométrica
19	Mandacaia	836023	-8,10000	-36,2833	Pluviométrica
20	Passagem do Tó	836028	-8,10000	-36,5167	Pluviométrica
21	Serra do Vento	836045	-8,23333	-36,3667	Pluviométrica
22	Sítio Apolinário	836046	-8,08333	-36,4500	Pluviométrica
23	Sítio Canhoto	836047	-8,15000	-36,5833	Pluviométrica
24	Sítio Lagoa do Félix	836048	-8,16670	-36,5667	Pluviométrica
25	Sítio Logradouro	836049	-8,16667	-36,1833	Pluviométrica
26	Sítio Muquem	836050	-8,10000	-36,6000	Pluviométrica
27	Sítio Severo	836051	-8,13333	-36,5500	Pluviométrica
28	Toritama (Torres)	836054	-8,01667	-36,0667	Pluviométrica
29	Xucuru (Aldeia Velha)	836055	-8,23333	-36,5833	Pluviométrica
30	Brejo da Madre de Deus	836068	-8,15000	-36,3833	Pluviométrica
31	Toritama	836090	-8,01667	-36,0667	Pluviométrica
32	Brejo da Madre de Deus	836092	-8,14560	-36,3703	Pluviométrica
33	Santa Cruz do Capibaribe	39100000	-7,9619	-36,2022	Fluviométrica
34	Toritama	39130000	-8,0128	-36,0578	Fluviométrica

Para seleção das estações a serem utilizadas no presente estudo, inicialmente foram filtradas as estações pluviométricas e fluviométricas que possuem registros no mesmo intervalo de tempo. Como pode ser verificado na Figura 6, as estações fluviométricas 39100000 e 39130000 possuem registros no mesmo intervalo de tempo das estações pluviométricas 735159, 736040, 736041, 736042 e 836092. Além disso, essas estações são as que possuem registros de dados mais recentes.

Figura 6 – Disponibilidade de dados das estações pluviométricas e fluviométricas da área de contribuição da barragem Jucazinho.



Fonte: Autor (2024).

Um modelo hidrológico de predição de vazão deve ter uma boa correlação entre chuva e vazão como dado de entrada devido à relação intrínseca entre esses dois fatores no ciclo hidrológico. A chuva é o principal impulsionador do escoamento superficial e da recarga de águas subterrâneas, exercendo influência direta sobre os níveis de vazão dos rios. Uma correlação significativa entre os dados de chuva e vazão permite que o modelo capture de maneira mais precisa as respostas do sistema hidrológico às condições meteorológicas, melhorando assim a precisão das previsões de vazão.

Foi utilizada a correlação de Spearman para verificar a correlação entre precipitação e vazão porque ela é uma medida não paramétrica que não faz suposições sobre a distribuição dos dados e é mais robusta quanto às relações não lineares. Isso é especialmente relevante em contextos hidrológicos, nos quais as relações entre precipitação e vazão podem ser complexas e não seguir um padrão linear.

No cálculo da correlação de Spearman, inicialmente, para cada uma das duas variáveis, os dados são classificados em ordem crescente e são atribuídos *ranks* (posições) com base na ordem após a classificação. Posteriormente, o fator de correlação é calculado através da Equação 1. O resultado ρ varia entre -1 e 1, no qual

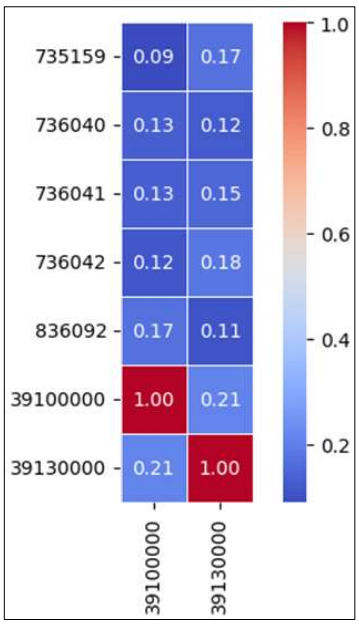
-1 indica uma correlação negativa perfeita, 1 indica uma correlação positiva perfeita e 0 indica ausência de correlação monotônica.

$$\rho = 1 - \left(\frac{6\sum_{i=1}^{n} {d_i}^2}{n(n^2-1)}\right) \qquad \text{Equação 1}$$

Tal que d_i e n são, respectivamente, a diferença de ranks entre a série original e a série classificada em ordem crescente para a i-ésima observação e o número total de observações.

Portanto, para verificar qual das estações seria utilizada, foi verificado qual combinação de estação fluviométrica e pluviométrica possuía a maior correlação. A partir do gráfico da correlação de Spearman, apresentado na Figura 7, é possível verificar que a estação pluviométrica 736042 e a estação fluviométrica 39130000 foram as estações que apresentaram a maior correlação com ρ = 0,18, dessa maneira, essas foram as estações utilizadas para o desenvolvimento desse estudo.

Figura 7 – Gráfico da Matriz de Correlação.



Fonte: Autor (2024).

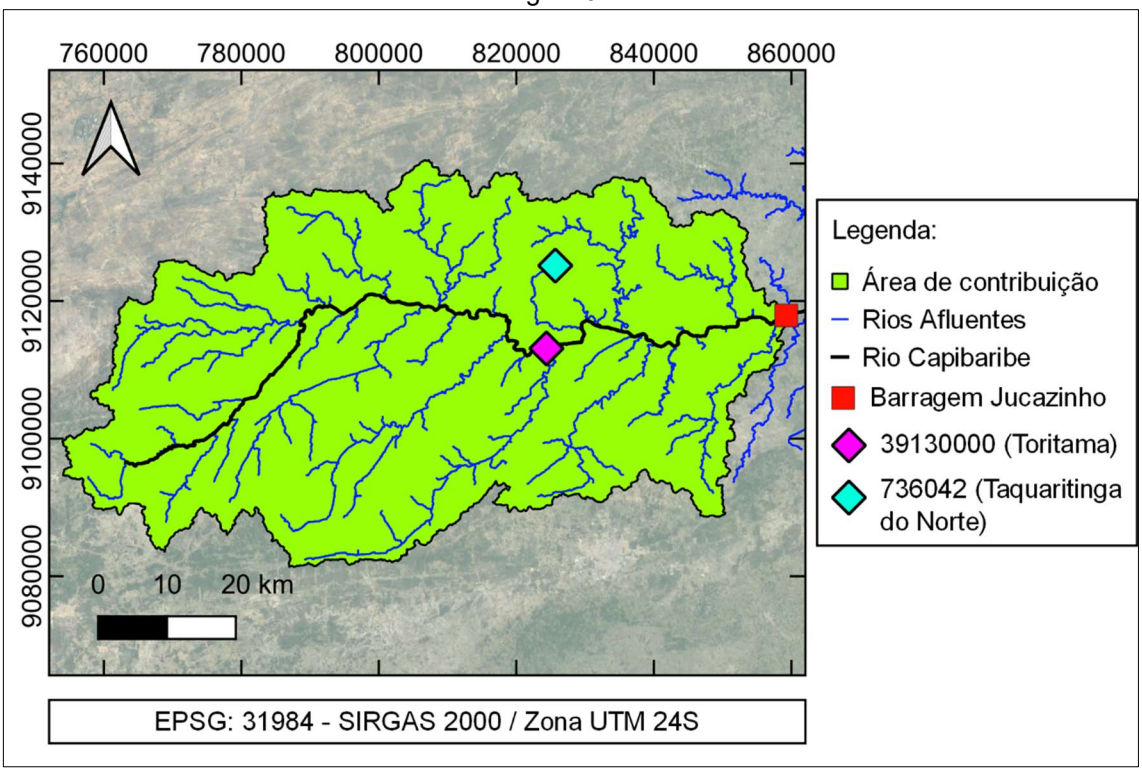
Os dados gerais de ambas as estações e a localização delas na área de contribuição da barragem Jucazinho constam, respectivamente, na Tabela 3 e Figura

8. Em virtude do início e fim de ambas as séries, foram utilizados os registros de 01/01/1986 a 01/06/2023 para desenvolvimento do modelo hidrológico.

Tabela 3 – Dados gerais das estações pluviométrica e fluviométrica utilizadas.

Informações	Estação Pluviométrica	Estação Fluviométrica
Nome da Estação	Taquaritinga do Norte	Toritama
Código	736042	39130000
Bacia	3 – Atlântico, trecho NO/NE	3 – Atlântico, trecho NO/NE
Sub-bacia	39 – Rios Capibaribe, Ipojuca, Una, Goiana, Mundaú, Paraíba do Meio, Coruripe, Sirinhaém, São Miguel e Camaragibe	39 – Rios Capibaribe, Ipojuca, Una, Goiana, Mundaú, Paraíba do Meio, Coruripe, Sirinhaém, São Miguel e Camaragibe
Município	Taquaritinga do Norte	Santa Cruz do Capibaribe
Estado	Pernambuco	Pernambuco
Responsável	ANA	ANA
Operadora	CPRM	CPRM
Latitude	-7,9039	-8,0128
Longitude	-36,0469	-36,0578
Altitude (m)	785	376
Área de drenagem (km²)	-	2450
Distância até a barragem Jucazinho (km)	34,31	35,16
Início da série	01/01/1986	01/01/1973
Fim da série	30/06/2023	01/06/2023
Tamanho da série (anos)	36,5	49,5

Figura 8 – Localização das estações pluviométrica e fluviométrica utilizadas na área de contribuição da barragem Jucazinho.

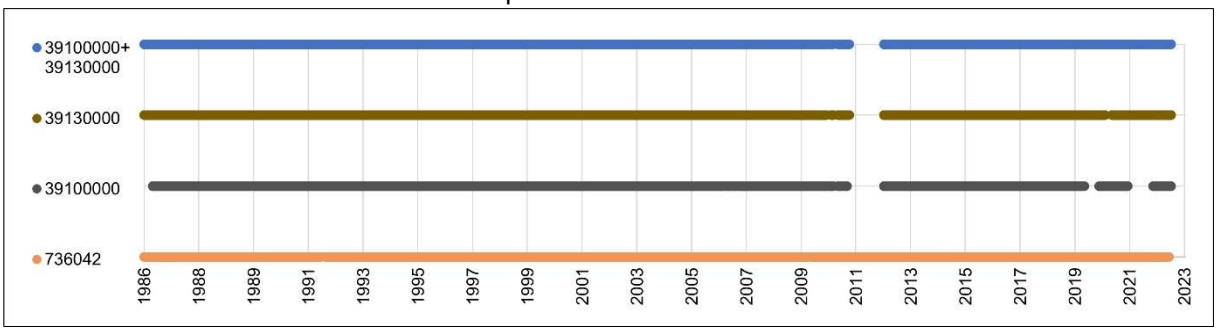


Fonte: Autor (2024).

Os dados disponíveis das estações 736042 e 39130000 constam na Figura 9. Com relação aos dados de vazão, para o preenchimento das falhas da estação 39130000, foram utilizados os dados da estação 39100000 multiplicados pelo fator 1,57, referente à razão entre a área de drenagem de 2.450 km² da estação 39130000 sob a área de drenagem de 1.560 km² da estação 39100000, obtidas através dos dados do portal HIDROWEB da ANA.

Ressalta-se que ambas as estações se situam no Rio Capibaribe, que é o rio principal da Bacia Hidrográfica do Capibaribe, barrado pela barragem Jucazinho, e que a quantidade de falhas é insignificativa se comparada com a série total, possibilitando, sem grandes prejuízos ao modelo, esse tipo de preenchimento de falhas.

Figura 9 – Disponibilidade de dados das estações pluviométricas e fluviométricas utilizadas no presente estudo.



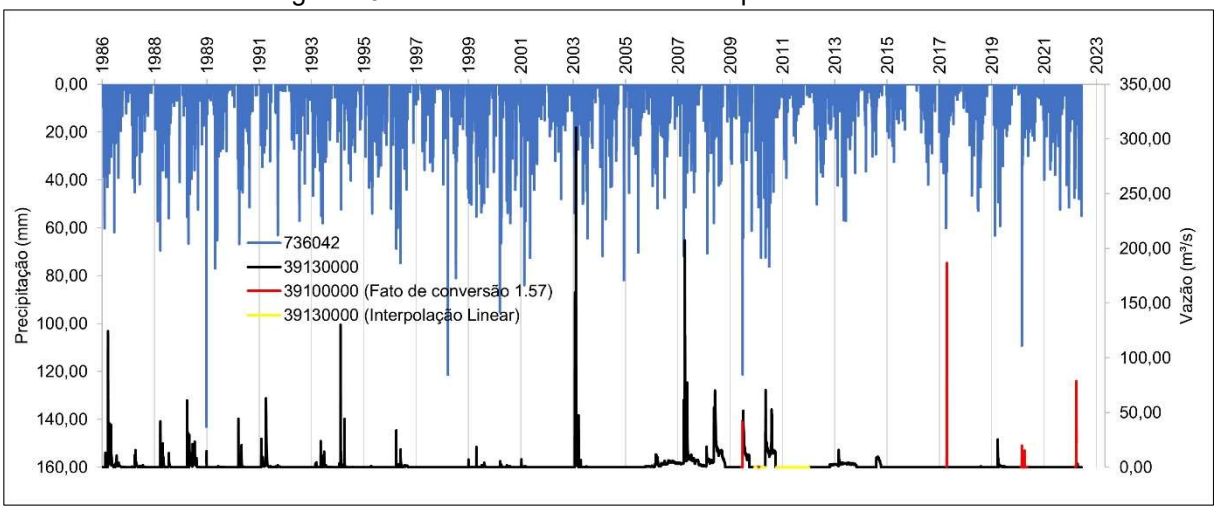
Fonte: Autor (2024).

O restante dos dados ausentes, tanto de vazão, quanto de precipitação, para as estações 736042 e 39130000, foram interpolados linearmente. A série com o preenchimento de falhas é apresentada na Figura 10, com um total de 13668 dias. Inicialmente, as estações 736042 e 39130000 apresentaram, respectivamente, 0,61% e 5,38% de falhas nessa quantidade de dias. Após a utilização dos dados da estação 39100000 multiplicados pelo fato 1,57, a porcentagem de falhas da estação 39130000 caiu para 4,15%. Por fim, ambas as séries apresentaram os 13668 dias de registro sem falhas.

Ressalta-se que a interpolação linear não é a metodologia adequada para o preenchimento de falhas diárias de uma série de precipitação ou vazão, sobretudo por se tratar de um preenchimento para um período com baixa precipitação e vazão, com valores iguais a zero ou próximos disso, é aceitável a aplicação do método sem associação de erros significativos nos resultados das séries preenchidas.

Enfatiza-se que a estação fluviométrica 391300000 dista cerca de 35,16 km da Barragem Jucazinho. Portanto, a série de vazão gerada pelos modelos de inteligência artificial treinados a partir dos dados apresentados na Figura 10 deverá ser multiplicada pelo fator 1,69 para aplicação prática de gestão do reservatório. Esse fator é referente à razão entre a área de drenagem de 4.149,90 km² da Barragem Jucazinho sob a área de drenagem de 2.450,00 km² da estação 39130000.

Figura 10 – Série de dados utilizada no presente estudo.



Fonte: Autor (2024).

A Figura 11 apresenta a média dos acumulados mensais de todos os anos da bacia de contribuição. Verifica-se que chove pouco na área, com aproximadamente quatro meses com chuva e oito meses de estiagem, indicando que a bacia hidrográfica do Capibaribe não possui memória hidrológica.

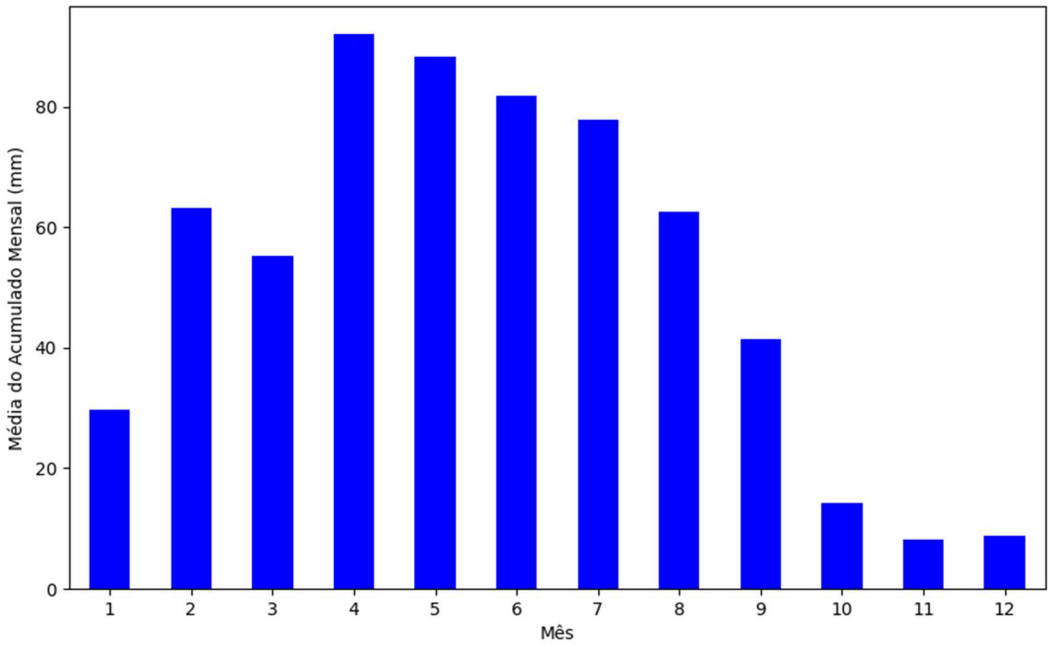
A memória hidrológica de uma bacia hidrográfica representa sua capacidade de armazenar e liberar água ao longo do tempo, respondendo às condições climáticas. Ela pode variar significativamente de acordo com as características geográficas e climáticas de uma região

Em uma bacia hidrográfica, a água da chuva pode ser retida temporariamente nos solos, infiltrar para o lençol freático, fluir superficialmente para rios e lagos, ou ainda ser armazenada em reservatórios artificiais. A forma como essa água é armazenada e liberada ao longo do tempo influencia diretamente o comportamento hidrológico da bacia, afetando aspectos como o regime de vazão dos rios, a disponibilidade de água para abastecimento humano, agricultura, geração de energia, entre outros.

Em regiões áridas, onde a precipitação é escassa e irregular, a memória hidrológica pode ser relativamente curta, com poucos reservatórios naturais ou artificiais para armazenar água. Já em regiões de clima mais úmido, com chuvas frequentes e abundantes, a memória hidrológica tende a ser mais longa, devido à presença de uma maior quantidade de reservatórios naturais.

Os modelos hidrológicos enfrentam desafios em bacias sem memória hidrológica, porque podem ter respostas menos previsíveis, prejudicando a capacidade do modelo de capturar eventos climáticos anômalos, tendo em vista que a variabilidade temporal na retenção e liberação de água influencia diretamente a resposta hidrológica.

Figura 11 – Média do acumulado mensal de todos os anos da série.



Fonte: Autor (2024).

5.3 CONSTRUÇÃO DO MODELO

Para o desenvolvimento do modelo foram utilizadas as variáveis de entrada apresentadas no Quadro 3, que predizem uma sequência de próximos passos a partir de uma sequência de observações passadas. Considerando que $Q_{(t-1)}$ representa a vazão no tempo anterior a $Q_{(t)}$, as vazões com retardo de 1, 2, 3, …, 32 dias com relação a t-1 foram denominadas, respectivamente, $Q_{(t-1)}$, $Q_{(t-2)}$, …, $Q_{(t-32)}$, assim como, as vazões com adianto de 1, 2, 3, …, 32 dias com relação a t foram denominadas,

respectivamente, $Q_{(t+1)}$, $Q_{(t+2)}$, …, $Q_{(t+32)}$. A mesma lógica de nomenclatura foi utilizada para as precipitações.

Dessa maneira, no primeiro cenário, a predição da vazão do próximo 1 dia foi realizada a partir de dados prévios de 1 dia de vazão e precipitação. No segundo cenário, foram considerados dados prévios de 2 dias de vazão e precipitação predizendo os próximos 2 dias de vazão. Nos outros cenários, as mesmas lógicas anteriores foram utilizadas, porém com 4, 8, 16 e 32 dias de dados. Os modelos foram agrupados em:

- Grupo C): Predição de curto prazo, para 1, 2 e 4 dias de predição;
- Grupo L): Predição de longo prazo, para 8, 16 e 32 dias de predição.

Quadro 3 – Estrutura do modelo para predição das vazões.

Modelos	Variáveis de Entrada	Variável da Saída
C – 1	$P_{(t-1)}$, $Q_{(t-1)}$	$Q_{(t)}$
C – 2	$P_{(t-1)}$, $P_{(t-2)}$, $Q_{(t-1)}$, $Q_{(t-2)}$	$Q_{(t)}$, $Q_{(t+1)}$
C – 4	$P_{(t-1)}$, $P_{(t-2)}$, …, $P_{(t-4)}$, $Q_{(t-1)}$, $Q_{(t-2)}$, …, $Q_{(t-4)}$	$Q_{(t)}$, $Q_{(t+1)}$, $Q_{(t+2)}$, $Q_{(t+3)}$
L – 8	$P_{(t-1)}$, $P_{(t-2)}$, …, $P_{(t-8)}$, $Q_{(t-1)}$, $Q_{(t-2)}$, …, $Q_{(t-8)}$	$Q_{(t)}$, $Q_{(t+1)}$, …, $Q_{(t+7)}$
L – 16	$P_{(t-1)}$, $P_{(t-2)}$, …, $P_{(t-16)}$, $Q_{(t-1)}$, $Q_{(t-2)}$, …, $Q_{(t-16)}$	$Q_{(t)}$, $Q_{(t+1)}$, …, $Q_{(t+15)}$
L – 32	$P_{(t-1)}$, $P_{(t-2)}$, …, $P_{(t-32)}$, $Q_{(t-1)}$, $Q_{(t-2)}$, …, $Q_{(t-32)}$	$Q_{(t)}$, $Q_{(t+1)}$, …, $Q_{(t+31)}$

Os modelos foram construídos de forma ordenada, sem misturar a ordem cronológica dos pares contendo as variáveis de entrada e saída, e recursiva, repetindo-se valores nesses pares. Utilizando como exemplo o modelo C - 1, os pares são apresentados no Quadro 4.

Quadro 4 – Ordenação e recursividade dos modelos, utilizando como exemplo o modelo C - 1.

Par	Entrada	Saída
1	$P_{(t-1)}$, $P_{(t-2)}$, $Q_{(t-1)}$, $Q_{(t-2)}$	$Q_{(t)}$, $Q_{(t+1)}$
2	$P_{(t)}$, $P_{(t-1)}$, $Q_{(t)}$, $Q_{(t-1)}$	$Q_{(t+1)}$, $Q_{(t+2)}$
3	$P_{(t+1)}$, $P_{(t)}$, $Q_{(t+1)}$, $Q_{(t)}$	$Q_{(t+2)}$, $Q_{(t+3)}$
...	...	
n	$P_{(t+n-2)}$, $P_{(t+n-3)}$, $Q_{(t+n-2)}$, $Q_{(t+n-3)}$	$Q_{(t+n-1)}$, $Q_{(t+n)}$

Nota-se que $P_{(t-2)}$, $Q_{(t-2)}$ e $Q_{(t+1)}$ estão repetidos nos conjuntos de entrada e saída. Dessa maneira, valores históricos são usados tanto para prever os valores futuros da vazão, quanto para fornecer informações sobre os padrões passados que influenciam essas previsões.

Em um contexto de curto prazo, geralmente abrangendo períodos de até uma semana, a predição de vazão é essencial para a tomada de decisões imediatas. Isso inclui o controle em tempo real do fluxo de água, a prevenção de enchentes e a gestão

do nível da água do reservatório. A capacidade de antecipar eventos climáticos intensos ou mudanças repentinas nas condições hidrológicas permite uma resposta rápida, como a liberação controlada de água para evitar inundações ou o ajuste imediato do volume armazenado para atender à demanda atual.

Por outro lado, em uma perspectiva de longo prazo, geralmente abrangendo períodos maiores que uma semana, a predição de vazão é crucial para o planejamento estratégico. Isso permite a adaptação gradual às condições hidrológicas sazonais, contribuindo para a gestão sustentável dos recursos hídricos ao longo do tempo.

Com relação a sua classificação, os modelos hidrológicos baseados em inteligência artificial podem ser adaptados para gerar tanto previsões probabilísticas, quanto determinísticas. Como o modelo gera previsões únicas e específicas com base em condições iniciais e parâmetros definidos, é razoável classificá-lo como determinístico.

Os dados de entrada e saída foram normalizados entre 0 e 1 para prevenir a predominância de variáveis com valores altos, o que é comum em modelos de aprendizado de máquina. Ressalta-se que a normalização foi feita por variável, por exemplo, a normalização de $Q_{(t-1)}$ considera a série de dados apenas de $Q_{(t-1)}$. Nesse processo foi utilizada a função MinMaxScaler da biblioteca Sklearn, que realiza a transformação através da Equação 2 abaixo:

$$x_n{}' = \frac{x_n - x_{min}}{x_{max} - x_{min}} \qquad \text{Equação 2}$$

Tal que x_n é o enésimo valor da série de dados, $x_n{}'$ é o valor de x_n após a normalização, e x_{min} e x_{max} são, respectivamente, os valores mínimo e máximo da série de dados.

Para aplicação dos modelos de inteligência artificial, os dados foram divididos em treinamento e teste utilizando a função traintestsplit da biblioteca sklearn do python e foi feita uma análise de sensibilidade considerando as seguintes proporções:

- 65% pra treinamento e 35% para teste (65-35);
- 70% para treinamento e 30% para teste (70-30);
- 75% para treinamento e 25% para teste (75-25);
- 80% para treinamento e 25% para teste (80-20).

5.3.1 Definição dos hiperparâmetros

Hiperparâmetros são parâmetros externos que não são aprendidos diretamente pelo modelo durante o treinamento, mas precisam ser especificados antes do início do processo de treinamento. Esses parâmetros influenciam o comportamento global do modelo e afetam o desempenho e a eficácia do treinamento. Em contraste, os parâmetros são os pesos que o modelo ajusta durante o treinamento para fazer previsões com base nos dados.

Devido à série temporal ser de grande escala, aumentando significativamente o custo computacional e objetivando obter um ponto de referência inicial para avaliar a adaptabilidade dos modelos de aprendizado de máquina, o presente trabalho utilizou os valores padrão de hiperparâmetros. Esses valores, por serem genéricos e funcionarem bem em uma variedade de situações, proporcionam eficiência computacional, evitando a necessidade inicial de experimentação extensiva.

5.3.1.1 Máquinas de vetores de suporte

No modelo de máquinas de vetores de suporte, foi utilizada a função SVR da biblioteca Sklearn, cuja lista contém os hiperparâmetros com seus respectivos tipos de variáveis e valores padrão (*defaut values*), os quais estão descritos no Quadro 5.

Quadro 5 – Hiperparâmetros da função SVR da biblioteca Sklearn.

Parâmetro	Tipo da variável	Valor Padrão
kernel	{‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’} or callable	’rbf’
degree	int	3
gamma	{‘scale’, ‘auto’} or float	Scal
coef0	float	0.0
tol	float	1e-3
C	float	1.0
Epsilon	float	0.1
shrinking	bool	True
cache_size	float	200
Verbose	bool	False
max_iter	int	-1

O modelo de aprendizado de máquina SVR é uma extensão do algoritmo de MVS para tarefas de regressão. O SVM original foi desenvolvido inicialmente para problemas de classificação, mas a variante SVR foi adaptada para lidar com a previsão de valores numéricos em vez de classes.

A lógica central por trás do SVR é a mesma do SVM, que envolve a busca por um hiperplano ótimo que melhor se ajusta aos dados de treinamento. No entanto, ao contrário do SVM de classificação, no qual o objetivo é encontrar um hiperplano que separe as classes de forma eficiente, o SVR busca um hiperplano que otimize a previsão de valores contínuos.

5.3.1.2 Floresta Aleatória

No modelo de floresta aleatória, foi utilizada a função RandomForestRegressor da biblioteca Sklearn, cuja lista contém os hiperparâmetros com seus respectivos tipos de variáveis e valores padrão (*defaut values*), os quais estão descritos no Quadro 6.

Quadro 6 – Hiperparâmetros da função RandomForestRegressor da biblioteca Sklearn.

Parâmetro	Tipo da variável	Valor Padrão
n_estimators	int	100
criterion	string: {“squared_error”, “absolute_error”, “friedman_mse”, “poisson”}	squared_error
max_depth	int	None
min_samples_split	int or float	2
min_samples_leaf	int or float	1
min_weight_fraction_leaf	float	0.0
max_features	string: {“sqrt”, “log2”, None}, int or float	1.0
max_leaf_nodes	int	None
min_impurity_decrease	float	0.0
bootstrap	bool	True
oob_score	bool	False
n_jobs	int	None
random_state	int, RandomState or None	None
verbose	Int	0
warm_start	bool	False
ccp_alpha	non-negative float	0.0
max_samples	int or float	None

O modelo de aprendizado de máquina RandomForestRegressor reside na construção de um *ensemble* de Árvores de Decisão, um componente fundamental da FA. Cada árvore é treinada de maneira independente, utilizando amostragens aleatórias tanto das instâncias do conjunto de dados quanto das características em cada divisão de nó. Essa abordagem aleatória contribui para a diversidade entre as árvores e, consequentemente, para a robustez do modelo.

Durante o processo de treinamento, cada árvore realiza previsões individuais para as instâncias do conjunto de dados de acordo com as decisões tomadas em suas estruturas. A previsão final do RandomForestRegressor é obtida pela média dessas previsões, resultando em uma estimativa mais estável e menos suscetível a *overfitting*.

O *overfitting* é um fenômeno comum em aprendizado de máquina, no qual um modelo se adapta excessivamente aos detalhes específicos dos dados de treinamento, perdendo a capacidade de generalização para novos dados. Isso ocorre quando o modelo é muito complexo em relação à complexidade inerente dos dados, capturando padrões irrelevantes, ruído ou variações específicas do conjunto de treinamento.

Como resultado, o desempenho do modelo pode ser excelente nos dados de treinamento, mas falha ao enfrentar novos dados, prejudicando sua capacidade de fazer previsões precisas em situações do mundo real.

5.3.1.3 Redes Neurais Artificiais

No modelo de redes neurais, foi utilizada a função MLPRegressor da biblioteca Sklearn, cuja lista contém os hiperparâmetros com seus respectivos tipos de variáveis e valores padrão (*defaut values*), os quais estão descritos no Quadro 7.

Quadro 7 – Hiperparâmetros da função MLPRegressor da biblioteca Sklearn.

Parâmetro	Tipo da variável	Valor Padrão
hidden_layer_sizes	array-like of shape(n_layers - 2,)	(100,)
activation	{‘identity’, ‘logistic’, ‘tanh’, ‘relu’}	’relu’
solver	{‘lbfgs’, ‘sgd’, ‘adam’}	’adam’
alph	float	0.0001
batch_size	int	auto
learning_rate	{‘constant’, ‘invscaling’, ‘adaptive’}	’constant’

learning_rate_init	float	0.001
power_t	float	0.5
max_iter	int	200
shuffle	bool	True
random_state	int	None
tol	float	1e-4
verbose	bool	False
warm_start	bool	False
momentum	float	0.9
nesterovs_momentum	bool	True
early_stopping	bool	False
validation_fraction	float	0.1
beta_1	float	0.9
beta_2	float	0.999
epsilon	float	1e-8
n_iter_no_change	int	10
max_fun	int	15000

O modelo de aprendizado de máquina MLPRegressor pertence à categoria de redes neurais artificiais conhecidas como *Perceptrons Multicamadas* (MLPs). Esse modelo é especificamente designado para tarefas de regressão, sendo capaz de realizar previsões de valores numéricos com base em dados de entrada.

A estrutura fundamental do MLPRegressor é composta por múltiplas camadas de neurônios, cada uma conectada às camadas adjacentes. Essa arquitetura permite que o modelo capture relações complexas entre as variáveis de entrada e saída. Ao contrário de modelos lineares simples, o MLPRegressor é capaz de aprender padrões não lineares nos dados.

Durante o treinamento, o MLPRegressor utiliza um processo iterativo conhecido como retropropagação (*backpropagation*). Esse processo envolve a passagem para frente (*forward pass*) das entradas através da rede para gerar previsões, e em seguida, a comparação dessas previsões com os valores reais para calcular o erro. O erro é então retropropagado pela rede, ajustando os pesos das conexões entre neurônios para minimizá-lo na próxima iteração.

5.4 AVALIAÇÃO DO MODELO

Os modelos foram avaliados quanto ao seu desempenho por meio de uma análise gráfica e através de critérios estatísticos. O propósito da avaliação foi verificar a qualidade dos resultados da calibração e da validação, comparando os dados de vazão simulados pelos modelos com os dados reais observados.

A análise gráfica adotada buscou verificar a tendência da correlação dos dados observados e preditos ao longo do acréscimo da vazão e também com aumento de dias de acordo com os modelos.

Os critérios estatísticos adotados foram a eficiência de modelagem Nash e Sutcliffe (1970) limitado (NSE), o Percentual de Tendências (PBIAS), o coeficiente de determinação (R^2) e a razão (RSR) entre a Raiz do Erro Quadrático Médio (RMSE) e o Desvio Padrão dos dados observados (σ_{obs}). Esses critérios estão descritos em equações, especificamente, da Equação 3 a Equação 6, respectivamente.

$$NSE = 1 - \left(\frac{\sum_{i=1}^{n}(Q_i^{sim} - Q_i^{obs})^2}{\sum_{i=1}^{n}(Q_i^{obs} - \overline{Q_{obs}})^2}\right) \quad \text{Equação 3}$$

$$PBIAS = \frac{\sum_{i=1}^{n} Q_i^{obs} - Q_i^{sim}}{\sum_{i=1}^{n} Q_i^{obs}} * 100 \quad \text{Equação 4}$$

$$R^2 = \frac{(n * \left(\sum_{i=1}^{n} Q_i^{obs} * Q_i^{sim}\right) - \sum_{i=1}^{n} Q_i^{obs} * Q_i^{sim})^2}{\left(n * \sum_{i=1}^{n} Q_i^{obs^2} - \left(\sum_{i=1}^{n} Q_i^{obs}\right)^2\right) * \left(n * \sum_{i=1}^{n} Q_i^{sim^2} - (\sum_{i=1}^{n} Q_i^{sim})^2\right)} \quad \text{Equação 5}$$

$$\mathrm{RSR} = \frac{\mathrm{RMSE}}{\sigma_{\mathrm{med}}} = \frac{\sqrt{\sum_{\mathrm{i=1}}^{\mathrm{n}}\left(Q_i^{obs} - Q_i^{sim}\right)^2}}{\sqrt{\sum_{\mathrm{i=1}}^{\mathrm{n}}\left(Q_i^{obs} - \overline{Q_{obs}}\right)^2}} \quad \text{Equação 6}$$

Tal que Q_i^{obs} e Q_i^{sim} são, respectivamente, os i-nésimos valores das séries de dados observados e simulados e $\overline{Q_{obs}}$ e $\overline{Q_{sim}}$ são, respectivamente, os valores médios das séries de dados observados e simulados.

O NSE varia de -∞ a 1, sendo 1 o representativo do valor ótimo. Os valores entre 0 e 1 são vistos como níveis de performance aceitáveis, entretanto, valores ≤0 indicam que a média observada é um preditor melhor que o valor simulado, indicando uma má performance do modelo.

O valor ótimo do PBIAS é 0, sendo valores positivos ou negativos, com magnitudes baixas, representativos de uma boa performance. Valores positivos indicam que o modelo subestimou valores medidos, enquanto valores negativos indicam que o modelo superestimou valores medidos.

O R^2 estima a correlação entre os valores medidos e simulados, variando de 0 a 1, com o valor 1 representando a concordância perfeita.

O RSR varia do valor ideal de 0, o que indica RMSE = 0, até um grande valor positivo. Portanto, quanto menor o RSR, menor será o RMSE e melhor será o desempenho da simulação.

Na análise quantitativa foi utilizada a classificação de Moriasi et al. (2007), apresentada na Tabela 4. A classificação utilizou a avaliação de uma modelagem com frequência mensal e, portanto, para simulações com valores diários pode-se considerar que valores de NSE acima de 0,36 ainda são satisfatórios.

Tabela 4 – Classificação dos coeficientes de eficiência da modelagem.

Classificação	**NSE**	**PBIAS**
Muito Boa	0,75 < NSE ≤ 1,00	PBIAS ≤ ±10
Boa	0,65 < NSE ≤ 0,75	±10 < PBIAS ≤ ±15
Satisfatória	0,50 < NSE ≤ 0,65	±15 < PBIAS ≤ ±25
Insatisfatória	NSE ≤ 0,50	PBIAS ≥ ±25
Classificação	R^2	RSR
Muito Boa	0,8 < R^2 ≤ 1,0	0,0 < RSR ≤ 0,5
Boa	0,7 < R^2 ≤ 0,8	0,5 < RSR ≤ 0,6
Satisfatória	0,6 < R^2 ≤ 0,7	0,6 < RSR ≤ 0,7
Insatisfatória	R^2 ≤ 0,6	RSR ≥ 0,6

Fonte: Adaptado de *Moriasi* et al. (2007).

A análise gráfica foi realizada através do gráfico de dispersão para todos os modelos e através do gráfico das leituras ao longo do tempo para os modelos C-1 e L – 32. Sendo que o gráfico das leituras ao longo do tempo para o modelo L – 32 foi criado considerando apenas o primeiro, o décimo quinto e o último dia dos pares, tendo em vista que os valores se repetem.

Ambos os gráficos são representações visuais da relação entre as duas variáveis. No gráfico de dispersão, cada ponto representa uma observação comparando o eixo x, representando a variável independente, e o eixo y, representando a variável dependente. O padrão geral dos pontos no gráfico pode fornecer *insights* sobre a natureza da relação entre as variáveis. Se os pontos seguirem uma tendência linear ou não, isso fornece informações sobre a qualidade do ajuste do modelo.

No gráfico das leituras ao longo do tempo, cada ponto apresenta uma observação comparando o eixo x, representando as datas, e o eixo y, representando as vazões medidas no posto pluviométrico e previstas pelo modelo. Quanto mais próximas as curvas geradas, melhor é a qualidade do ajuste do modelo. A vantagem desse tipo de representação gráfica é poder identificar se o modelo está tendo dificuldades de prever os máximos ou os mínimos da série, além de identificar retardos entre os gráficos.

6 RESULTADOS E DISCUSSÕES

Os resultados dos coeficientes estatísticos de eficiência dos modelos constam na Tabela 5 para os dados de teste. Observa-se que, no geral, todos os modelos apresentaram desempenho insatisfatório para predição de longo prazo, que contempla 8, 16 e 32 dias para qualquer uma das divisões de treinamento e teste (65-35, 70-30, 75-25, 80-20). Além disso, com o aumento de dias de predição há uma clara tendência de decréscimo da qualidade do ajuste.

É importante destacar que apesar da baixa correlação entre as estações pluviométrica e fluviométrica utilizadas nesse estudo, os modelos de inteligência artificial MSV, RNA e FA apresentaram resultados estatísticos aceitáveis para as predições de curto prazo, que incluem 1, 2 e 4 dias de predição.

Cheng et al. (2020) adotaram Redes Neurais Artificiais (RNA) e Memória de Curto e Longo Prazo (LSTM) para prever vazões em escalas diárias e mensais. Para a previsão de vazão mensal, foi utilizada uma abordagem de previsão recursiva. Os dois modelos foram treinados e validados utilizando conjuntos de dados de precipitação e vazão coletados nas bacias dos rios Nan e Ping, na Tailândia, cobrindo o período compreendido entre 1974 a 2014. Os principais achados do estudo destacam que tanto as RNA quanto a MCLP podem fornecer previsões precisas diárias, de até 20 dias.

Não é incomum que a performance de modelos de previsão de vazão utilizando inteligência artificial diminua com o aumento do horizonte de predição. Essa comparação sugere que a precisão da previsão de vazão a longo prazo pode variar significativamente dependendo do método de modelagem e da abordagem adotada.

Tabela 5 – Critérios estatísticos do treinamento.

	SVM				RF				NN			
NSE	65-35	70-30	75-25	80-20	65-35	70-30	75-25	80-20	65-35	70-30	75-25	80-20
C-1	0,54	0,83	0,91	0,94	0,70	0,87	0,85	0,82	0,73	0,92	0,92	0,93
C-2	0,40	0,75	0,85	0,91	0,66	0,85	0,80	0,69	0,71	0,87	0,88	0,79
C-4	0,27	0,63	0,78	0,86	0,57	0,63	0,72	0,56	0,64	0,76	0,77	0,71
C-8	0,16	0,47	0,68	0,80	0,50	0,49	0,62	0,48	0,53	0,60	0,70	0,59
C-16	0,08	0,30	0,52	0,70	0,37	0,35	0,50	0,20	0,41	0,37	0,57	0,38
C-32	0,01	0,17	0,41	0,57	0,21	0,05	0,06	-1,81	0,24	0,15	0,42	0,00
	SVM				**RF**				**NN**			
PBIAS	65-35	70-30	75-25	80-20	65-35	70-30	75-25	80-20	65-35	70-30	75-25	80-20
C-1	-27,73	-10,17	-0,09	5,52	-9,63	1,15	7,23	9,04	-20,12	8,42	-11,74	20,32
C-2	-37,38	-16,13	-4,18	2,61	-5,61	1,27	9,64	13,25	-11,94	3,61	-15,1	6,78

C-4	-48,06	-23,69	-7,71	0,94	-13,05	5,97	16,49	20,56	-19,35	6,67	61,92	37,23
C-8	-56,63	-32,24	-13,74	-1,23	-22,89	8,78	23,27	31,75	-16,22	-0,04	-1,77	55,23
C-16	-62,5	-41,70	-23,98	-6,69	-34,88	15,48	34,72	49,17	-24,01	15,22	39,53	22,71
C-32	-70,99	-51,05	-30,33	-14,68	-44,57	23,63	82,32	128,36	-35,57	12,38	53,87	63,46

R²	**SVM**				**RF**				**NN**			
	65-35	70-30	75-25	80-20	65-35	70-30	75-25	80-20	65-35	70-30	75-25	80-20
C-1	0,54	0,83	0,91	0,94	0,70	0,87	0,85	0,82	0,73	0,92	0,92	0,93
C-2	0,40	0,75	0,85	0,91	0,66	0,85	0,80	0,69	0,71	0,87	0,88	0,79
C-4	0,27	0,63	0,78	0,86	0,57	0,63	0,72	0,56	0,64	0,76	0,77	0,71
C-8	0,16	0,47	0,68	0,80	0,50	0,50	0,62	0,48	0,53	0,60	0,70	0,59
C-16	0,08	0,30	0,52	0,70	0,37	0,35	0,50	0,20	0,41	0,37	0,57	0,38
C-32	0,01	0,17	0,41	0,57	0,21	0,05	0,03	-1,81	0,24	0,15	0,41	0,00

RSR	**SVM**				**RF**				**NN**			
	65-35	70-30	75-25	80-20	65-35	70-30	75-25	80-20	65-35	70-30	75-25	80-20
C-1	0,67	0,41	0,30	0,25	0,55	0,36	0,38	0,42	0,52	0,28	0,28	0,27
C-2	0,78	0,50	0,38	0,30	0,58	0,38	0,44	0,55	0,54	0,35	0,34	0,46
C-4	0,86	0,61	0,47	0,37	0,66	0,61	0,53	0,66	0,60	0,49	0,48	0,54
C-8	0,92	0,73	0,56	0,44	0,70	0,71	0,61	0,72	0,69	0,63	0,55	0,64
C-16	0,96	0,84	0,69	0,55	0,80	0,80	0,71	0,90	0,77	0,79	0,65	0,79
C-32	0,99	0,91	0,77	0,66	0,89	0,97	0,97	1,68	0,87	0,92	0,76	1,00

O decréscimo na qualidade dos ajustes com o aumento da quantidade de dias previstos é atribuído à incerteza crescente em relação às condições futuras e à acumulação de erros ao longo do horizonte de previsão. De acordo com os critérios estatísticos, o modelo que melhor se adaptou à série foi o SVM 80-20, que resultou nos melhores índices estatísticos de NSE, PBIAS, R^2 e RSR, conforme Tabela 6.

Al-Mukhtar (2019) investigou a modelagem e previsão de sedimentos no rio Tigris em Bagdá, um parâmetro influente na poluição dos corpos d'água. Três métodos de inteligência artificial (florestas aleatórias, máquinas de vetores de suporte e redes neurais artificiais) foram empregados utilizando dados de vazão observados (m^3/s) e concentração de sedimentos suspensos (mg/l) coletados entre 1962–1981 e 2000–2010. Os resultados preditivos dos três métodos avaliados foram analisados com base nos coeficientes R^2, RMSE e NSE e indicaram que florestas aleatórias apresentaram o melhor desempenho.

Essas constatações destacam a importância de considerar a variação de desempenho entre diferentes métodos de previsão, bem como a necessidade de avaliar adequadamente as incertezas associadas aos modelos, especialmente em cenários de previsão de longo prazo.

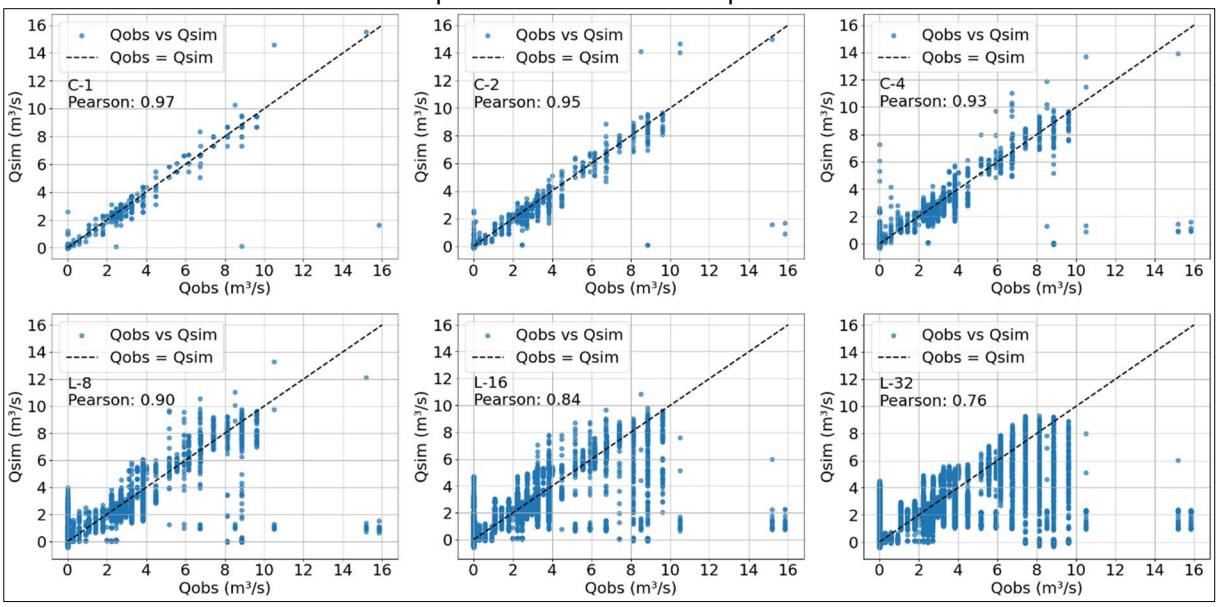
Tabela 6 – Melhores critérios estatísticos do treinamento.

C – 1	**NSE**	**PBIAS**	**R²**	**RSR**
	SVM 80-20	SVM 75-25	SVM 80-20	SVM 80-20
C - 2	NSE	PBIAS	R²	RSR
	SVM 80-20	RF 70-30	SVM 80-20	SVM 80-20
C - 4	NSE	PBIAS	R²	RSR
	SVM 80-20	SVM 80-20	SVM 80-20	SVM 80-20

Os resultados gráficos dos modelos são apresentados da Figura 12 a Figura 17. Os modelos demoraram 90% do tempo de execução do código no treinamento, ou seja, o custo computacional do treinamento é significativamente maior que o necessário para utilizar o modelo treinado.

Nos modelos de MVS, com o aumento dos dias houve uma subestimação dos valores de previsão de vazão (pontos aproximando-se do eixo x). Apesar de ser eficaz em modelar relações não lineares, o modelo MVS é sensível a padrões complexos nas séries temporais. Por essa razão, à medida que o horizonte de previsão aumentou o modelo não conseguiu prever as mudanças significativas nos padrões de vazão, como pode ser verificado na Figura 12.

Figura 12 – Gráficos de dispersão dos testes com Máquina de Vetores de Suporte com 80% dos dados para treinamento e 20% para teste.

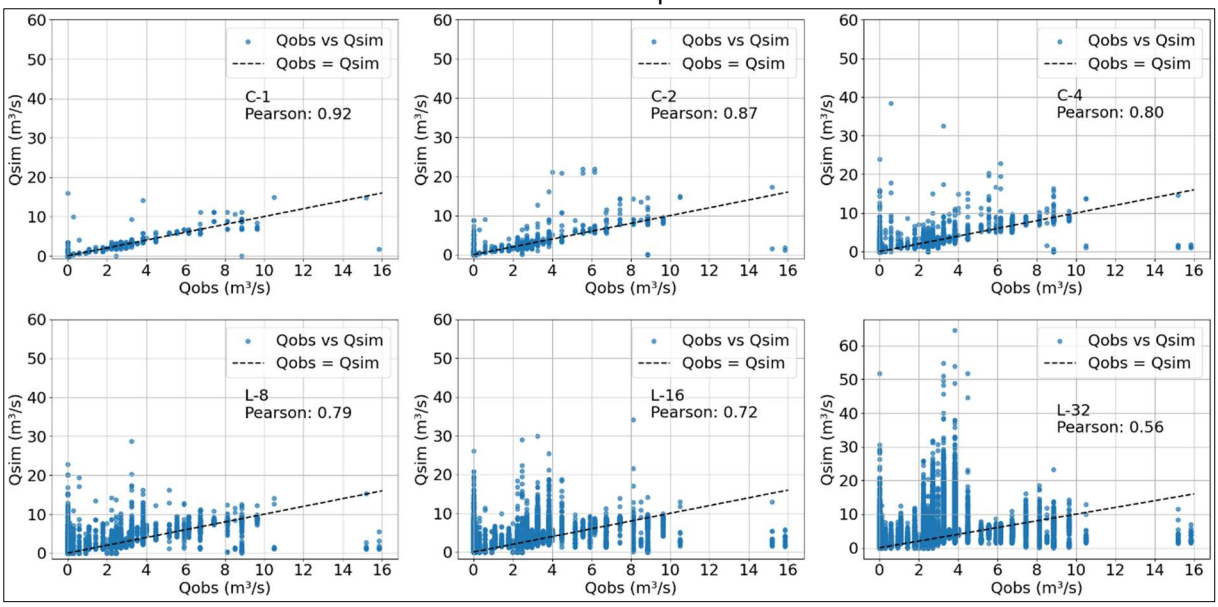


Fonte: Autor (2024).

No modelo de FA e RNA, aconteceu o contrário do modelo MVS, pois com o aumento dos dias houve uma hiperestimação dos valores de previsão de vazão (pontos aproximando-se do eixo y), conforme apresenta a Figura 13 e a Figura 14. Isso se dá principalmente pelo *overfitting*, em que os modelos se ajustaram

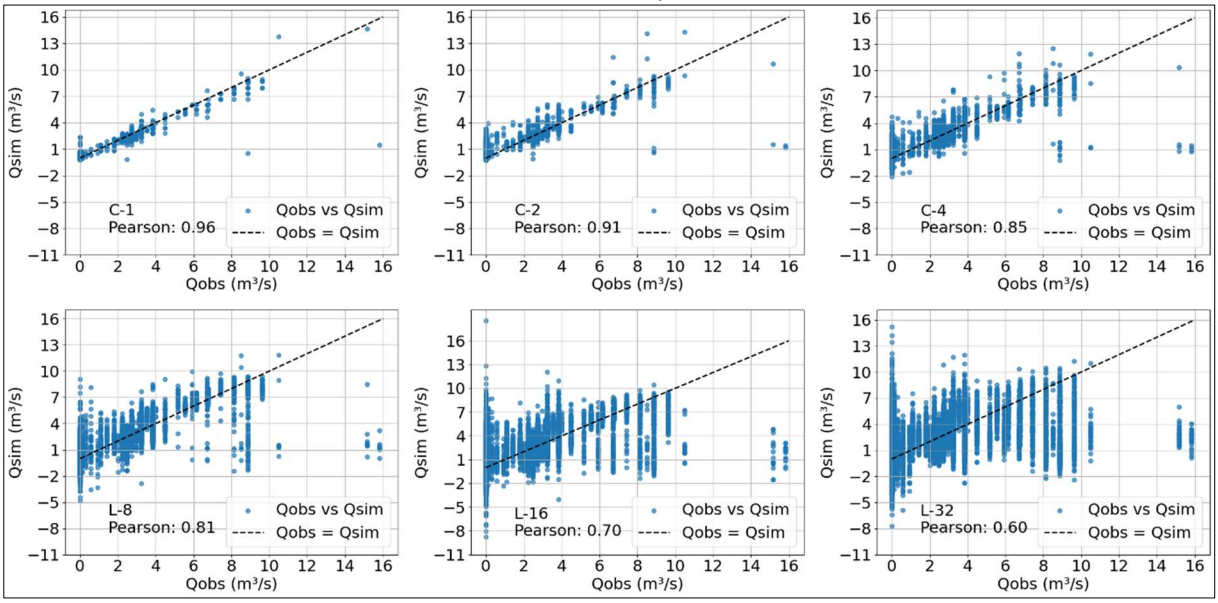
demasiadamente aos dados de treinamento, incorporando ruídos e padrões transitórios que não são representativos do comportamento real da vazão.

Figura 13 – Gráficos de dispersão dos testes com Florestas Aleatórias com 80% dos dados para treinamento e 20% para teste.



Fonte: Autor (2024).

Figura 14 – Gráficos de dispersão dos testes com Redes Neurais Artificiais com 80% dos dados para treinamento e 20% para teste.



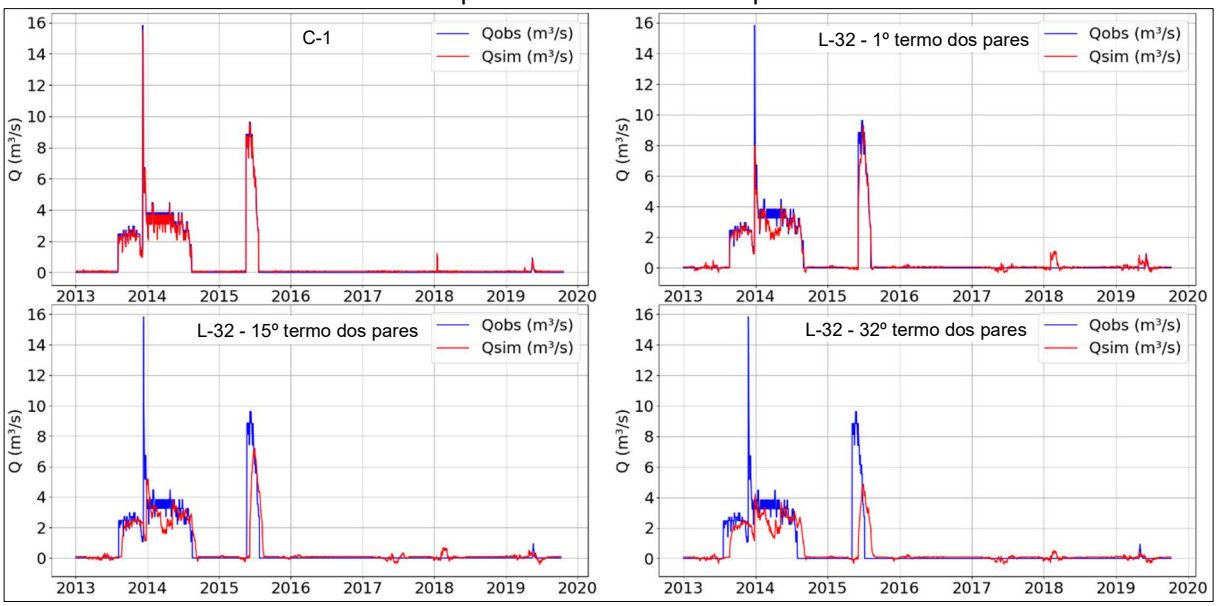
Fonte: Autor (2024).

Han et al. (2007) aplicaram MVS sobre a bacia hidrográfica de *Bird Creek* para a previsão de enchentes e verificaram que, assim como os modelos de redes neurais artificiais, o SVM também sofre com problemas de *underfitting* e *overfitting*, sendo o

overfitting mais prejudicial do que o *underfitting*. O estudo também revela um resultado interessante na resposta do SVM a diferentes entradas de chuva, no qual chuvas mais leves geraram respostas muito diferentes das chuvas mais intensas, tal qual ocorreu no presente trabalho.

No gráfico das leituras ao longo do tempo, apresentado na Figura 15, é possível verificar a subestimação dos picos citada anteriormente para os modelos com máquina de vetores de suporte. O modelo C - 1 conseguiu predizer melhor os picos do que o modelo L – 32. Verifica-se, também, que há um retardo entre os gráficos da vazão observada e medida para o modelo L – 32 no decorrer do aumento do termo nos pares de *input* e *output*. Esse retardo explica os baixos coeficientes estatísticos, tendo em vista que o erro acumulado é somado.

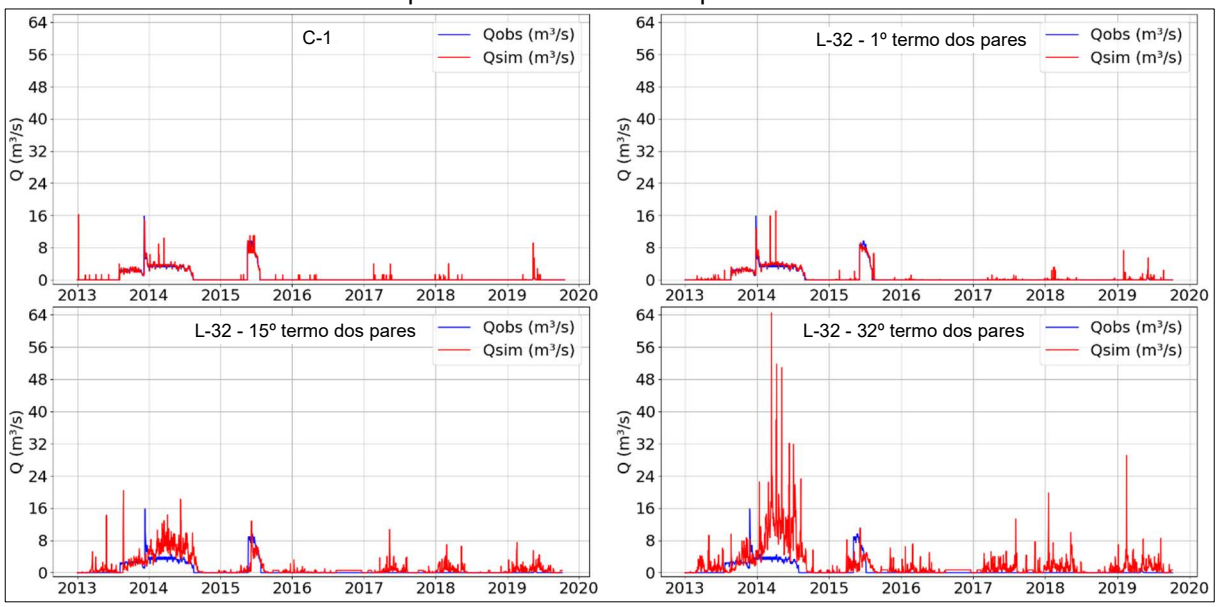
Figura 15 – Gráfico das leituras ao longo do tempo com Máquina de Vetores de Suporte com 80% dos dados para treinamento e 20% para teste.



Fonte: Autor (2024).

De acordo com a Figura 16 e com a Figura 17, os modelos de Florestas Aleatórias apresentaram resultados semelhantes aos de Redes Neurais Artificiais, porém os ruídos gerados oscilaram muito acima da vazão medida. Assim como ocorreu com o modelo de Máquina de Vetores de Suporte, também há um retardo entre os gráficos da vazão observada e medida para o modelo L – 32 no decorrer do aumento do termo nos pares de *input* e *output*.

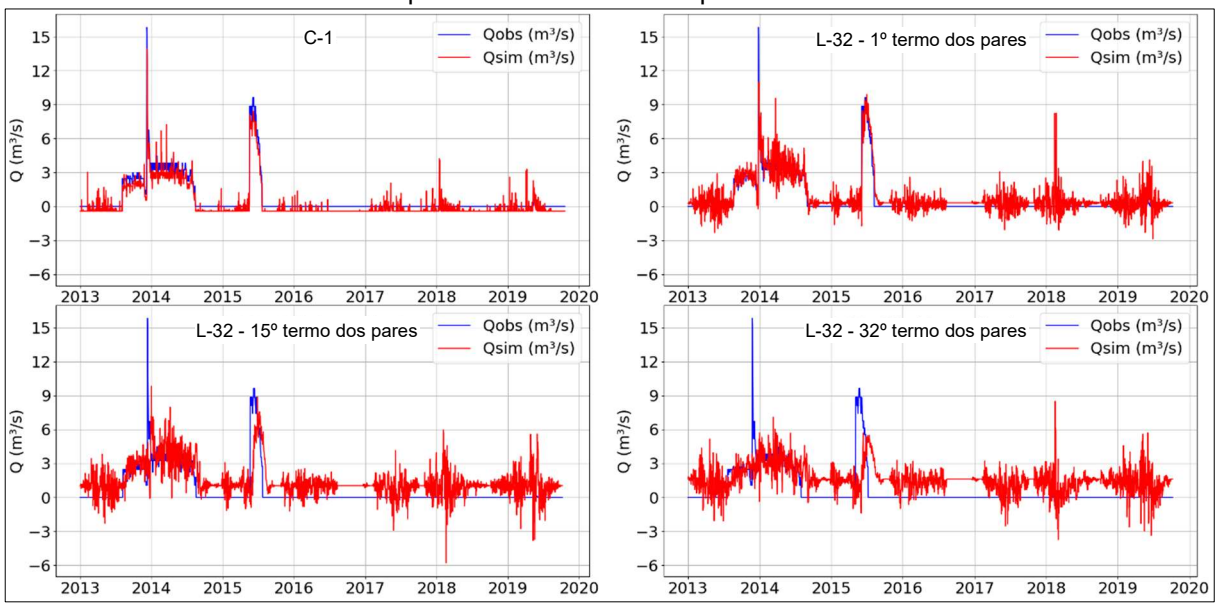
Figura 16 – Gráfico das leituras ao longo do tempo com Florestas Aleatórias com 80% dos dados para treinamento e 20% para teste.



Fonte: Autor (2024).

Os modelos de Redes Neurais Artificiais conseguiram representar bem os picos de vazão, conforme Figura 17, sobretudo geraram ruídos que oscilaram os valores, que deveriam ser zero, entre valores ligeiramente maiores ou menores que isso, chegando a prever valores negativos de vazão. Assim como ocorreu com ambos os modelos apresentados anteriormente, também há um retardo entre os gráficos da vazão observada e medida para o modelo L – 32 no decorrer do aumento do termo nos pares de *input* e *output*.

Figura 17 – Gráfico das leituras ao longo do tempo com Redes Neurais Artificiais com 80% dos dados para treinamento e 20% para teste.



Fonte: Autor (2024).

Os resultados confirmam que a bacia hidrográfica do Capibaribe não possui memória hidrológica. Recomenda-se em trabalhos futuros a aplicação de Redes Neurais Artificiais em diferentes bacias hidrográficas para verificação da adaptabilidade do modelo de acordo com a memória hidrológica de cada bacia.

7 CONCLUSÕES

A presente pesquisa teve como objetivo geral gerar séries sintéticas de vazões afluentes para a barragem de Jucazinho, localizada no Agreste de Pernambuco, a partir de modelos estocásticos. Os objetivos específicos consistiram em avaliar a adaptabilidade dos modelos de aprendizado de máquina MVS, FA e RN para gerar as séries sintéticas de vazões, analisar a influência na qualidade dos ajustes com o aumento da quantidade de dias e previsão e verificar a qualidade dos ajustes realizados pelos modelos utilizando métricas estatísticas.

Com base nos resultados obtidos e discussões apresentadas, conclui-se que:

- Todos os modelos apresentaram desempenho satisfatório para predição de curto prazo, que contempla 1, 2 e 4 dias, e insatisfatório para predição de longo prazo, que contempla 8, 16 e 32 dias;
- Os resultados gráficos dos modelos de Máquinas de Vetores de Suporte (MVS) evidenciaram uma subestimação dos valores de vazão com o

aumento do horizonte de previsão, devido à sensibilidade do MVS a padrões complexos nas séries temporais;

- Os modelos de Floresta Aleatória (FA) e Redes Neurais Artificiais (RNA) apresentaram uma hiperestimação dos valores de vazão à medida que o número de dias de previsão aumentava, atribuída principalmente ao *overfitting*;
- Todos os modelos, com o aumento de dias de predição, tiveram uma tendência de decréscimo da qualidade do ajuste, justificada principalmente pelo retardo das predições que gerou acúmulo de erros;
- De acordo com os critérios estatísticos, o modelo que melhor se adaptou à série foi o MVS, índices estatísticos de NSE, PBIAS, R^2 e RSR;
- Mesmo em situações nas quais os dados são escassos, os modelos de inteligência artificial MVS, FA e RNA possuem potencial para serem aplicados para predição de curto prazo; e,
- A bacia hidrográfica do Capibaribe não possui memória hidrológica, impactando no treinamento do modelo. Recomenda-se em trabalhos futuros a aplicação de Redes Neurais Artificiais em diferentes bacias hidrográficas para verificação da adaptabilidade do modelo de acordo com a memória hidrológica de cada bacia.

REFERÊNCIAS

AGÊNCIA NACIONAL DAS ÁGUAS - ANA. **Reservatórios do semiárido brasileiro**: Hidrologia, balanço hídrico e operação. Brasília: ANA, 2017. 178 p. v. ANEXO E.

AGÊNCIA PERNAMBUCANA DE ÁGUAS E CLIMA - APAC. **Ficha técnica com cota volume**. [*S. l.*], 24 set. 2020. Disponível em: https://www.apac.pe.gov.br/images/media/1602284842_JUCAZINHO%20NOVO.pdf. Acesso em: 20 nov. 2022.

AL-MUKHTAR, Mustafa. Random forest, support vector machine, and neural networks to modelling suspended sediment in Tigris River-Baghdad. **Environmental monitoring and assessment**, v. 191, n. 11, p. 673, 2019.

ANA. **Reservatórios do semiárido brasileiro**: Hidrologia, balanço hídrico e operação. Brasília: [s.n.], 2017. 178 p. ANEXO E.

ANDREU, J.; CAPILLA, J.; SANCHIS, E. **AQUATOOL**: A computer-assisted support system for water resources research management including conjunctive use. In: Decision support systems. Springer-Verlag Berlin Heidelberg, 1991.

APAC. **Ficha técnica com cota volume**, 24 setembro 2020. Disponivel em: <https://www.apac.pe.gov.br/images/media/1602284842_JUCAZINHO%20NOVO.pdf>. Acesso em: 20 novembro 2022.

ASCE TASK COMMITTEE ON APPLICATION OF ARTIFICIAL NEURAL NETWORKS IN HYDROLOGY. Artificial neural networks in hydrology. II: Hydrologic applications. **Journal of Hydrologic Engineering**, v. 5, n. 2, p. 124-137, 2000.

BERGSTRA, James; BENGIO, Yoshua. Random search for hyper-parameter optimization. **Journal of machine learning research**, v. 13, n. 2, 2012.

CAMARGO, A. P.; SENTELHAS, P. C. **Avaliação do desempenho de diferentes métodos de estimativa da evapotranspiração potencial no Estado de São Paulo, Brasil**. Revista Brasileira de Agrometeorologia, v. 5, n. 1, p. 89-97, 1997.

CHAN, Simon; TRELEAVEN, Philip. Continuous model selection for large-scale recommender systems. In: **Handbook of statistics**. Elsevier, 2015. p. 107-124.

CHEN, C. M. CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. Journal of the American Society for Information Science and Technology, v. 57, n. 3, p. 359-377, 2006.

CHENG, M. *et al*. Long lead-time daily and monthly streamflow forecasting using machine learning methods. **Journal of Hydrology**, v. 590, p. 125376, 2020.

CHOW, V. T.; MAIDMENT, D. R.; MAYS, L. W. **Applied Hydrology**. Singapura: McGraw-Hill, 1988.

COMPANHIA PERNAMBUCANA DE SANEAMENTO - COMPESA. **Sistema Jucazinho registra acúmulo recorde e COMPESA reduz o rodízio na Região Agreste**. Recife, 20 mar. 2020. Disponível em: https://servicos.compesa.com.br/sistema-jucazinho-registra-acumulo-recorde-e-compesa-reduz-o-rodizio-na-regiao-do-agreste/. Acesso em: 20 nov. 2022.

CONTRERAS, Pablo *et al*. Influence of random forest hyperparameterization on short-term runoff forecasting in an andean mountain catchment. **Atmosphere**, v. 12, n. 2, p. 238, 2021.

CRAWFORD, Norman H.; LINSLEY, Ray K. **Digital Simulation in Hydrology:** Stanford Watershed Model 4. 1966.

D.; VEITH, T. L. **Model Evaluation Guidelines for Systematic Quantification of Accuracy in Watershed Simulations**. Transactions of the ASABE, v. 50, n. 3, p. 885–900, 2007.

ENOMOTO, C. F. **Método para elaboração de mapas de inundação**: Estudo de caso na bacia do Rio Palmital, Paraná. 2004. Dissertação (Mestrado Recursos Hídricos) - Universidade Federal do Paraná, Curitiba, 2004.

FAYAL, M. A. A**. Previsão de Vazão por Redes Neurais Artificiais e Transformada Wavelet**. 2008. Dissertação (Mestrado em Engenharia Elétrica) - Pontífica Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2008.

FIRAT, MAHMUT. Comparison of artificial intelligence techniques for river flow forecasting. **Hydrology and Earth System Sciences**, v. 12, n. 1, p. 123-139, 2008.

GALIZONI, T. **Previsão de vazões naturais afluentes médias semanais para usinas hidrelétricas**. 2021. Dissertação (Mestre em Engenharia Hídrica) - Universidade Federal de Itajubá, Itajubá, 2021.

GEOGAKAKOS, K. P.; GEOGAKAKOS, J. S.; YAO, H. Hydrologic Forecasting and Water Resources Management for Folsom Lake Watershed in California. **Hydrologic Forecasting–Lake Folson**, California, 2001.

GIRÃO, L. C. P. **Uma análise da contribuição dos programas básicos ambientais como instrumento de gestão ambiental para a barragem de Jucazinho localizada no município de Surubim/PE**. Dissertação (Mestrado) - Universidade Federal de Pernambuco, [S. l.], 2004.

GOVINDARAJU, R. S. Task committee on application of artificial neural networks in hydrology, artificial neural networks in hydrology. ii: hydrologic application. **J. Hydrol. Eng**, v. 5, n. 2, p. 124-136, 2000.

HASANPOUR KASHANI, Mahsa et al. Rainfall-Runoff simulation in the Navrood river basin using truncated volterra model and artificial neural networks. **Journal of Watershed Management Research**, v. 6, n. 12, p. 1-10, 2016.

HAN, D.; CHAN, L.; ZHU, N. Flood forecasting using support vector machines. **Journal of hydroinformatics**, v. 9, n. 4, p. 267-276, 2007.

HE, Zhibin et al. A comparative study of artificial neural network, adaptive neuro fuzzy inference system and support vector machine for forecasting river flow in the semiarid mountain region. **Journal of Hydrology**, v. 509, p. 379-386, 2014.

HORTON, R. E. Analysis of runoff-plot experiments with varying infiltration capacity. **Transactions American Geophysical Union**, Washington, p. 693-711, 1939.
HSU, Kuo-lin; GUPTA, Hoshin Vijai; SOROOSHIAN, Soroosh. Artificial neural network modeling of the rainfall-runoff process. Water resources research, v. 31, n. 10, p. 2517-2530, 1995.

HSU, K.; GUPTA, H. V.; SOROOSHIAN, S. Artificial neural network modeling of the rainfall-runoff process. **Water resources research**, v. 31, n. 10, p. 2517-2530, 1995.

HUSSAIN, Dostdar; KHAN, Aftab Ahmed. Machine learning techniques for monthly river flow forecasting of Hunza River, Pakistan. **Earth Science Informatics**, v. 13, p. 939-949, 2020.

IBBIT, R. P. **Systematic parameter fitting for conceptual models of catchment hydrology**. Imperial College of Science and Technology, University of London. London. 1970.

JANSEN, R. B. **Dams and Public Safety**: A Water Resources Technical Publication. Denver: United States Printing Office, 1980.

LABADIE, J.W. 1987. Otimização da operação de projetos agrícolas. Brasília: PRONI. Learning model versus conventional machine learning algorithms for streamflow forecasting. **Water Resources Management**, v. 35, n. 12, p. 4167-4187, 2021.

LIMA, H. V. C.; LANNA, A. E. L. **Modelos para operação de sistemas de reservatórios**: Atualização do estado da arte. Revista Brasileira de Recursos Hídricos, v. 10, n. 3, p. 5-22, 2005.

LOPES, J. E. G. Escola Politécnica da USP - Departamento de Engenharia Hidráulica e Ambiental. **Manual Smap**, 1999. Disponivel em: <http://pha.poli.usp.br/default.aspx?id=76&link_uc=disciplina>. Acesso em: 06 jun. 2023.

MAASS, A. *et al*. **Design of water-resource systems**. Cambridge: Harvard University Press, 1962.

MAIDMENT, D. R. GIS and hydrologic modeling. **Environmental modeling with GIS**, New York, p. 147-167, 1993.

MATTIUZI, C. D. P. **Gestão integrada dos recursos hídricos**: Alocação otimizada com uso conjunto de água superficial e subterrânea para redução da escassez

hídrica na bacia do Rio Santa Maria / RS. Universidade Federal do Rio Grande do Sul. Porto Alegre, p. 93. 2018.

MCCARTHY, G. T. The unit hydrograph and flood routing. **Conference of North Atlantic Division**, US Army Corps of Engineers, New London, 1938.

MORIASI, D. N.; ARNOLD, J. G.; VAN LIEW, M. W.; BINGNER, R. L.; HARMEL, R. **Model evaluation guidelines for systematic quantification of accuracy in watershed simulations**. Transactions of the ASABE, v. 50, n. 3, p. 885-900, 2007.

MULVANY, T. J. **On the use of self-registering rain and flood gauges in making observations of the relations of rain fall and flood discharges in a given catchment**. Institution of Civil Engineers of Ireland, v. 4, n. 2, p. 18-33, 1851.

NASH, J. E.; SUTCLIFFE, J. V. River flow forecasting through conceptual models part I - A discussion of principles. **Journal of Hydrology**, v. 10, n. 3, p. 282-290, 1970.

NEVES, Y. T.; RODRIGUES, A. B.; CABRAL, J. J. S. P. **Modelagem computacional do rompimento hipotético da barragem de Jucazinho no estado de Pernambuco (Brasil)**. Revista DAE, São Paulo, v. 69, ed. 230, p. 167-182, 25 mar. 2021.

NOORI, Navideh; KALIN, Latif. Coupling SWAT and ANN models for enhanced daily streamflow prediction. **Journal of Hydrology**, v. 533, p. 141-151, 2016.

NOURANI, Vahid *et al*. **Applications of hybrid wavelet–artificial intelligence models in hydrology: a review**. Journal of Hydrology, v. 514, p. 358-377, 2014.

PROBST, Philipp; WRIGHT, Marvin N.; BOULESTEIX, Anne-Laure. Hyperparameters and tuning strategies for random forest. **Wiley Interdisciplinary Reviews: data mining and knowledge discovery**, v. 9, n. 3, p. e1301, 2019.

PURCELL, P. J. **Design of Water Resources Systems**. Londres: Thomas Telford, 2003.

RAHIMZAD, Maryam et al. Performance comparison of an LSTM-based deep learning model versus conventional machine learning algorithms for streamflow forecasting. **Water Resources Management**, v. 35, n. 12, p. 4167-4187, 2021.

RASOULI, Kabir; HSIEH, William W.; CANNON, Alex J. Daily streamflow forecasting by machine learning methods with weather and climate inputs. **Journal of Hydrology**, v. 414, p. 284-293, 2012.

RENNÓ, C. D.; SOARES, J. V. Discretização espacial de bacias hidrográficas. **Simpósio Brasileiro de Sensoriamento Remoto**, v. 10, p. 485-492, 2001.

SECRETARIA DE RECURSOS HÍDRICOS - SRH. **Plano Hidroambiental da Bacia Hidrográfica do Capibaribe**: Cenários tendenciais e sustentáveis. Recife: PROJETEC - BRLi, 2010. 190 p. v. TOMO II.

SHERMAN, L. K. Streamflow from rainfall by unit-graph method. **Engineering News-Record**, v. 108, p. 501-505, 1932.

SILVA, A. C. M.; SANTANA, C. F. D.; BRITO, V. C.; SILVA, E. C.; RIBEIRO, R. R. F.; PEREIRA, G. E. S. **Otimização do sistema de abastecimento de água da Barragem de Jucazinho através de programação linear.** Revista Ibero-Americana de Ciências Ambientais, v. 10, n. 5, p. 101-113, 2019 apud MATTIUZI, 2018.

SRH. **Plano Hidroambiental da Bacia Hidrográfica do Capibaribe**: Cenários tendenciais e sustentáveis. Recife: PROJETEC - BRLi, 2010. TOMO II.

SUDHEER, Ch et al. A hybrid SVM-PSO model for forecasting monthly streamflow. **Neural Computing and Applications**, v. 24, p. 1381-1389, 2014.

TODINI, E. Hydrological catchment modelling: Past, present and future. **Hydrology and Earth System Sciences**, Itália, v. 11, n. 1, p. 468–482, 17 jan. 2007.

TODINI, E. **Hyrological catchment modelling**: Past, present and future. Hydrology and Earth System Sciences, Itália, v. 11, n. 1, 17 janeiro 2017. 468-482.

TONGAL, Hakan; BOOIJ, Martijn J. Simulation and forecasting of streamflows using machine learning models coupled with base flow separation. **Journal of hydrology**, v. 564, p. 266-282, 2018.

TUCCI, C. E. M. **Hidrologia**: Ciência e aplicação. Porto Alegre: UFRGS, 2001.

TUCCI, C. E. M. **Modelos Hidrológicos**. 2. ed. Porto Alegre: UFRGS, 2005.

TUCCI, C. E. M.; ORDONEZ, J. S.; SIMÕES, M. L. **Modelo Matemático Precipitação-Vazão IPH II**. Simpósio Brasileiro de Recursos Hídricos. Fortaleza: [s.n.]. 1981.

VERTESSY, R. A.; ELSENBEER, H. Distributed modeling of storm flow generation in an Amazonian rain forest catchment: effects of model parameterization. **Water Resources Research**, v. 35, n.7, p. 2173-2187, 1999.

VILLALOBOS-ARIAS, Leonardo et al. Evaluating hyper-parameter tuning using random search in support vector machines for software effort estimation. In: **Proceedings of the 16th ACM International Conference on Predictive Models and Data Analytics in Software Engineering**. 2020. p. 31-40.

WORLD COMMISSION ON DAMS. **Dams and development: A new framework for decision-making**: The report of the world commission on dams. Earthscan, 2000.

YASEEN, Zaher Mundher; KISI, Ozgur; DEMIR, Vahdettin. Enhancing long-term streamflow forecasting and predicting using periodicity data component: application of artificial intelligence. **Water resources management**, v. 30, p. 4125-4151, 2016.

ZÖRNIG, P. Introdução à programação não linear. Brasília: UNB, 2011.

APÊNDICE A

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import matplotlib.dates as mdates

from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.preprocessing import MinMaxScaler

from scipy.stats import pearsonr, spearmanr

##
# LEITURA DOS DADOS DE ENTRADA
##

file_path = '/content/39130000x736042.csv' # Lendo a planilha csv
df = pd.read_csv(file_path)
df_interpolated = pd.read_csv(file_path)
for data_frame in [df, df_interpolated]: # Convertendo a coluna 'Data' para o formato de data e configurando como índice
    data_frame['Data'] = pd.to_datetime(data_frame['Data'])
    data_frame.set_index('Data', inplace=True)

##
# INTERPOLANDO LINEARMENTE FALHAS NA SÉRIE
##

df_interpolated['736042'] = df_interpolated['736042'].interpolate() # Interpolando linearmente os valores em branco para ambas as colunas
df_interpolated['39130000'] = df_interpolated['39130000'].interpolate()
```

```
        ##
        # PLOTANDO OS DADOS DE ENTRADA (MEDIDO VS INTERPOLADO)
        ##

        fig, ax1 = plt.subplots(figsize=(15, 6)) # Criando uma figura e os eixos
        ax1.plot(df.index, df['736042'], color='b', label='Precipitação - P (mm)', zorder=2) # Configurando o eixo y esquerdo de precipitação
        ax1.plot(df_interpolated.index, df_interpolated['736042'], color='black', linestyle='--', label='P Interpolada', zorder=1, linewidth=0.9)
        ax1.set_xlabel('Ano')
        ax1.set_ylabel('Precipitação (mm)', color='b')
        ax1.tick_params('y', colors='b')
        ax2 = ax1.twinx() # Adicionando o eixo y direito de vazão
        ax2.plot(df.index, df['39130000'], color='r', label='Vazão - Q (m³/s)', zorder=2)
        ax2.plot(df_interpolated.index, df_interpolated['39130000'], color='black', linestyle='--', label='Q Interpolada', zorder=1, linewidth=1)
        ax2.set_ylabel('Vazão (m³/s)', color='r')
        ax2.tick_params('y', colors='r')
        for axis in [ax1, ax2]: # Configurando os eixos para mostrar apenas anos no formato yyyy
            axis.xaxis.set_major_locator(mdates.YearLocator())
            axis.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
        ax1.set_ylim(bottom=0, top=200) # Configurando o valor máximo no eixo y da precipitação
        ax1.invert_yaxis() # Invertendo a ordem no eixo y da precipitação
        for axis in [ax1, ax2]: # Rotacionando rótulos no eixo x
            plt.setp(axis.get_xticklabels(), rotation=90, ha='right')
        for axis in [ax1, ax2]: # Adicionando grade e título
            axis.xaxis.grid(True, linestyle='--', alpha=0.7)
        plt.title('Leituras de Precipitação e Vazão ao Longo do Tempo')
        fig.legend(loc='upper left', bbox_to_anchor=(0.8, 0.55)) # Adicionando legenda
        plt.tight_layout() # Ajustando layout
        plt.show() # Mostrando o gráfico
```

```
        ##
        # PLOTANDO OS DADOS DE ENTRADA (TREINO VS VALIDAÇÃO)
        ##

        y_train, y_valid = train_test_split(df_interpolated[['39130000']], test_size=0.3,
shuffle=False) # Dividindo os dados em conjunto de treinamento e validação

        fig, ax1 = plt.subplots(figsize=(15, 6)) # Criando uma figura e eixos
        ax1.plot(df_interpolated.index, df_interpolated['736042'], 'b-',
label='Precipitação - P (mm)') # Configurando o eixo y esquerdo de precipitação
        ax1.set_xlabel('Ano')
        ax1.set_ylabel('Precipitação (mm)', color='b')
        ax1.tick_params(axis='y', colors='b')
        ax2 = ax1.twinx() # Adicionando o eixo y direito de vazão
        ax2.plot(y_train.index, y_train['39130000'], 'r-', label='Vazão Treino - Qtr (m³/s)')
        ax2.plot(y_valid.index, y_valid['39130000'], 'k-', label='Vazão Validação - Qva
(m³/s)')
        ax2.set_ylabel('Vazão (m³/s)', color='r')
        ax2.tick_params(axis='y', colors='r')
        ax1.xaxis.set_major_locator(mdates.YearLocator()) # Configurando os eixos
para mostrar apenas anos no formato yyyy
        ax1.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
        ax1.set_ylim(bottom=0, top=200) # Configurando o valor máximo no eixo y da
precipitação
        ax1.invert_yaxis() # Invertendo a ordem no eixo y da precipitação
        plt.setp(ax1.get_xticklabels(), rotation=90, ha='right') # Rotacionando rótulos no
eixo x
        plt.setp(ax2.get_xticklabels(), rotation=90, ha='right')
        ax1.xaxis.grid(True, linestyle='--', alpha=0.7) # Adicionando grade e título
        ax2.xaxis.grid(True, linestyle='--', alpha=0.7)
        plt.title('Leituras de Precipitação e Vazão ao Longo do Tempo')
        fig.legend(loc='upper left', bbox_to_anchor=(0.72, 0.55)) # Adicionando
legenda
        plt.tight_layout() # Ajustando layout
```

```
plt.show() # Mostrando o gráfico

##
# ORGANIZANDO O DATAFRAME
##

num_colunas = 32 #Definindo a quantidade de dias de predição
df_reorganizado = pd.concat( # Criando o DataFrame com colunas de precipitação (P) e vazão (Q)
    [df_interpolated['736042'].shift(-i) for i in range(1, num_colunas + 1)] +
    [df_interpolated['39130000'].shift(-i) for i in range(1, num_colunas * 2 + 1)],
    axis=1
)
colunas_renomeadas = [f'P{i}' for i in range(1, num_colunas + 1)] + [f'Q{i}' for i in range(1, num_colunas * 2 + 1)] # Renomeando as colunas de acordo com o padrão desejado
df_reorganizado.columns = colunas_renomeadas
df_reorganizado = df_reorganizado.dropna().reset_index(drop=True) # Descartando as linhas que contêm valores nulos resultantes do shift
X = df_reorganizado.iloc[:, :num_colunas * 2] # Criando DataFrame de entrada (X) e saída (y)
y = df_reorganizado.iloc[:, num_colunas * 2:num_colunas * 3]  # Escolhendo a coluna específica para prever
X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.3, shuffle=False) #Dividindo os dados em conjuntos de treinamento e validação

# Normalizando os dados
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_valid_scaled = scaler.transform(X_valid)
y_train_scaled = scaler.fit_transform(y_train)
y_valid_scaled = scaler.transform(y_valid)

##
```

```
# PREDIÇÃO DE VAZÃO SUPPORT VECTOR MACHINE
##

from sklearn.svm import SVR

svm_model = SVR() # Criando um modelo SVM
multioutput_svm_model = MultiOutputRegressor(svm_model) # Criando um modelo MultiOutputRegressor com SVM
multioutput_svm_model.fit(X_train_scaled, y_train_scaled) # Treinando o modelo
y_valid_pred_scaled = multioutput_svm_model.predict(X_valid_scaled) # Fazendo previsões no conjunto de validação
y_valid_pred = scaler.inverse_transform(y_valid_pred_scaled) # Voltando para a escala original
y_valid = y_valid.to_numpy()


##
# PREDIÇÃO DE VAZÃO RANDOM FOREST
##

#from sklearn.ensemble import RandomForestRegressor

#rf_model = RandomForestRegressor() # Criando um modelo Random Forest
#rf_model.fit(X_train_scaled, y_train_scaled) # Treinando o modelo na versão normalizada dos dados
#y_valid_pred_scaled = rf_model.predict(X_valid_scaled) # Fazendo previsões no conjunto de validação normalizado
#y_valid_pred = scaler.inverse_transform(y_valid_pred_scaled) # Voltando para a escala original
#y_valid = y_valid.to_numpy()

##
# PREDIÇÃO DE VAZÃO ARTIFICIAL NEURAL NETWORK
```

```
        ##

        #from sklearn.neural_network import MLPRegressor

        #mlp_model = MLPRegressor() # Criando um modelo MLPRegressor
        #mlp_model.fit(X_train_scaled, y_train_scaled) # Treinando o modelo
        #y_valid_pred_scaled = mlp_model.predict(X_valid_scaled) # Fazendo previsões no conjunto de validação
        #y_valid_pred = scaler.inverse_transform(y_valid_pred_scaled) # Voltando para a escala original
        #y_valid = y_valid.to_numpy()


        ##
        # CLASSIFICANDO AS MÉTRICAS DOS RESULTADOS
        ##

        def calculate_nse(y_true, y_pred): # Definindo funções para calcular as métricas
            nse = 1 - np.sum((y_true - y_pred) ** 2) / np.sum((y_true - np.mean(y_true)) ** 2)
            return nse
        def calculate_pbias(y_true, y_pred):
            pbias = np.sum((y_pred - y_true) / np.sum(y_true)) * 100
            return pbias
        def calculate_rsr(y_true, y_pred):
            rsr = np.sqrt(np.sum((y_true - y_pred) ** 2) / np.sum((y_true - np.mean(y_true)) ** 2))
            return rsr
        def classify_nse(nse): # Definindo funções para classificar as métricas
            if 0.75 <= nse <= 1.0:
                return "MUITO BOA"
            elif 0.65 <= nse < 0.75:
                return "BOA"
```

```
    elif 0.50 <= nse < 0.65:
        return "SATISFATÓRIA"
    else:
        return "INSATISFATÓRIA"
def classify_pbias(pbias):
    abs_pbias = abs(pbias)
    if abs_pbias <= 10:
        return "MUITO BOA"
    elif 10 < abs_pbias <= 15:
        return "BOA"
    elif 15 < abs_pbias <= 25:
        return "SATISFATÓRIA"
    else:
        return "INSATISFATÓRIA"
def classify_r2(r2):
    if 0.8 <= r2 <= 1.0:
        return "MUITO BOA"
    elif 0.7 <= r2 < 0.8:
        return "BOA"
    elif 0.6 <= r2 < 0.7:
        return "SATISFATÓRIA"
    else:
        return "INSATISFATÓRIA"
def classify_rsr(rsr):
    if 0 <= rsr <= 0.5:
        return "MUITO BOA"
    elif 0.5 < rsr <= 0.6:
        return "BOA"
    elif 0.6 < rsr <= 0.7:
        return "SATISFATÓRIA"
    else:
        return "INSATISFATÓRIA"

nse = calculate_nse(y_valid, y_valid_pred) # Calculando as métricas
```

```
        pbias = calculate_pbias(y_valid, y_valid_pred)
        r2 = r2_score(y_valid, y_valid_pred)
        rsr = calculate_rsr(y_valid, y_valid_pred)

        classification_df = pd.DataFrame({ # Classificando as métricas
            'NSE': [classify_nse(nse)],
            'PBIAS (%)': [classify_pbias(pbias)],
            'R²': [classify_r2(r2)],
            'RSR': [classify_rsr(rsr)]
        })

        results_df = pd.DataFrame({ # Arredondando os resultados numéricos para duas casas decimais
            'NSE': [nse],
            'PBIAS (%)': [pbias],
            'R²': [r2],
            'RSR': [rsr]
        }).round(2)

        combined_df = pd.concat([results_df, classification_df], keys=['Resultados', 'Classificação']) # Concatenando os DataFrames
        print(combined_df) # Exibindo o DataFrame combinado

        ##
        # VERIFICANDO A CORRELAÇÃO DOS RESULTADOS
        ##

        pearson_corr, _ = pearsonr(y_valid.flatten(), y_valid_pred.flatten()) # Calculando as correlações
        spearman_corr, _ = spearmanr(y_valid.flatten(), y_valid_pred.flatten())
        print(f"Correlação de Pearson: {pearson_corr}") # Imprimindo as correlações
        plt.figure(figsize=(8, 8)) # Plotando o gráfico y_valid vs y_valid com a reta x=y
        plt.scatter(y_valid, y_valid_pred, alpha=0.7, label='Qobs vs. Qpred (m³/s)')
```

```
        plt.plot([y_valid.min(), y_valid.max()], [y_valid.min(), y_valid.max()], 'k--', lw=2, label='Qobs = Qpred')
        plt.title('Qobs vs. Qpred (m³/s)')
        plt.xlabel('Qobs (m³/s)')
        plt.ylabel('Qpred (m³/s)')
        plt.grid(True)
        plt.legend() # Adicionando legenda
        middle_y = (y_valid.min() + y_valid.max()) / 2 # Calculando a posição média do eixo y
        plt.text(y_valid.min(), middle_y, f'Correlação Pearson: {pearson_corr:.2f}', verticalalignment='center') # Adicionando texto com os valores das correlações no meio do eixo y
        plt.show()
```