

PREVISÃO DO PERFIL DE PRESSÃO ADIMENSIONALIZADA EM RES-SALTOS HIDRÁULICOS POR ALGORÍTMOS DE APRENDIZADO DE MÁQUINA COM DADOS EXPERIMENTAIS E CFD

André Luiz Andrade Simões¹

Gabriella Laura Peixoto Botelho²

Renata Brito de Oliveira³

Resumo

Este trabalho apresenta uma abordagem baseada em aprendizado de máquina (ML) para prever o perfil da pressão média adimensionalizada (p^*) ao longo do ressalto hidráulico. Foram utilizados dados experimentais e numéricos obtidos por simulações com CFD, que, após interpolação por LOWESS, alimentaram o treinamento de quatro modelos regressivos: Random Forest (RF), Redes Neurais Artificiais (MLP), XGBoost (XGB) e Stacking Regressor (STK). Os modelos foram avaliados nas abordagens global e por número de Froude (Fr), empregando as métricas R^2 , MSE e MAE. Os resultados indicaram que o Random Forest apresenta desempenho robusto e consistente, com alta acurácia e coerência física. Embora o modelo Stacking tenha obtido as melhores métricas em alguns casos, foram observadas discrepâncias não condizentes com o comportamento hidráulico esperado, especialmente para valores elevados de x/h_j . Como resultado, foi implementada uma ferramenta que permite a geração de novos perfis de p^* para valores intermediários de Fr , com base na inferência estatística dos modelos treinados. **Palavras-chave:** Aprendizado de máquina, Ressalto hidráulico, Random Forest, Pressão média ao longo do ressalto hidráulico, Comportamento hidráulico.

Abstract

This study proposes a machine learning (ML)-based approach to predict the profile of the dimensionless mean pressure (p^*) along the hydraulic jump. Experimental data and numerical results from CFD simulations were interpolated using LOWESS and used to train four regression models: Random Forest (RF), Multilayer Perceptron (MLP), XGBoost (XGB), and Stacking Regressor (STK). The models were evaluated under global training and per-Froude-number approaches, using R^2 , MSE, and MAE as performance metrics. Results showed that the Random Forest model provided robust and consistent performance, with high accuracy and physical plausibility. Although the Stacking model achieved the best quantitative indicators in some cases, physically inconsistent discrepancies were observed for higher x/h_j values. As a practical outcome, a computational tool was implemented to enable the generation of new p^* profiles for intermediate Fr values based on statistical

¹Universidade Federal da Bahia (UFBA), andre.simoes@ufba.br

²Universidade Federal do Recôncavo da Bahia (UFRB), gbotelho@ufrb.edu.br

³Universidade Federal da Bahia (UFBA), rbritoengenharia@outlook.com

inference from trained models. **Keywords:** Machine learning, Hydraulic jump, Random forest, Average pressure across the hydraulic jump, Hydraulic behavior

Introdução

O ressalto hidráulico é um fenômeno caracterizado pela transição abrupta de um escoamento supercrítico para um regime subcrítico em canais abertos. Essa transição apresenta aplicação prática em estruturas hidráulicas de dissipação de energia e unidades de mistura em estações de tratamento (Ikinciogullari, 2019; Pakgar F, Daneshfaraz, Joudi 2016). Em sistemas extravasores de barragens, como aquele ilustrado na Figura 1, o ressalto é formado na bacia de dissipação de energia, que deve ser capaz de confinar o escoamento supercrítico e parte ou a totalidade do ressalto, de tal maneira que o rio, a jusante da bacia de dissipação, não sofra erosões expondo a fundação da barragem. A caracterização espacial e estatística de grandezas associadas ao ressalto, como as flutuações de pressão, é essencial para a previsão do seu comprimento adimensional, frequentemente representado como L_j/h_j , onde $h_j = h_2 - h_1$ é a altura do ressalto hidráulico, definida como a diferença entre as profundidades conjugadas, como indicado na Figura 1.

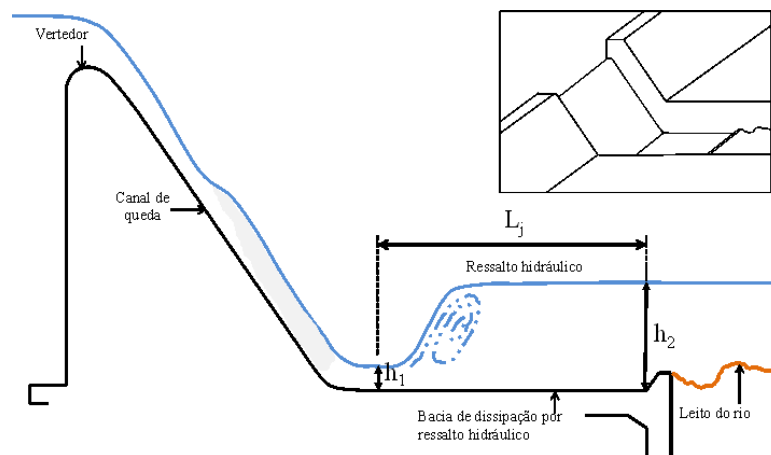


Figura 1: Sistema extravasor com bacia de dissipação por ressalto hidráulico.

Fonte: adaptado de Santos et al. (2022)

Com o trabalho de Marques *et al.* (1997), foi possível estabelecer e consolidar o critério do uso de transdutores de pressão para a determinação da posição final do ressalto baseada no comportamento estatístico dos sinais obtidos para a pressão em função do espaço e do tempo. Nesse contexto, há interesse também na determinação do perfil adimensional do ressalto, expresso pela pressão média adimensional (equação 1) ao longo do ressalto, como forma de caracterização do fenômeno.

$$p * \frac{p_m - h_1}{h_j} \quad (1)$$

Santos *et al.* (2022) propuseram novas formas de modelagem do ressalto com o uso de equações adimensionais empíricas e validação com dados experimentais. No referido

trabalho, os autores empregaram a dinâmica dos fluidos computacional (CFD – *Computational Fluid Dynamics*), resolvendo numericamente as equações de conservação de massa, Navier-Stokes com médias de Reynolds e o modelo de turbulência $k - \epsilon$, em duas dimensões. Os perfis obtidos para p^* em função de x/h_j foram comparados aos dados experimentais de diferentes autores, tendo sido constatada uma tendência global semelhante, como pode ser visto na Figura 2.

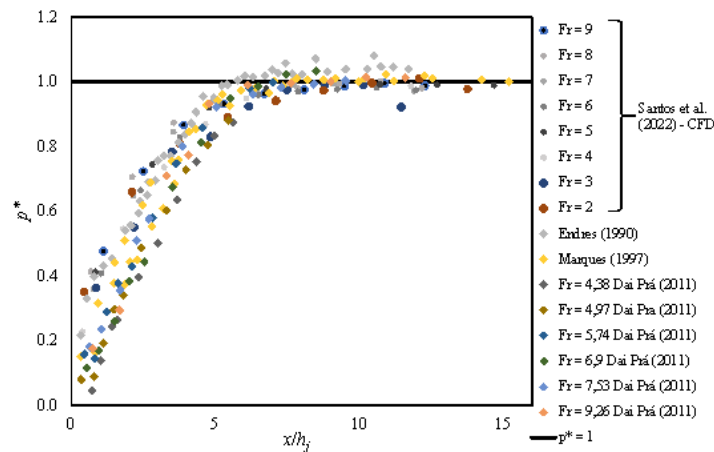


Figura 2: Comparação entre CFD (Santos et al., 2022) e dados experimentais.

Fonte: adaptado de Santos et al. (2022)

Com os avanços recentes em técnicas de aprendizado de máquina (ML), vislumbra-se a possibilidade de empregar modelos de regressão não linear para prever a evolução de p^* ao longo do resalto com maior densidade de dados e menor custo computacional. Este trabalho explora essa abordagem por meio da interpolação de dados experimentais e numéricos, treinamento de modelos de regressão multivariada e análise das métricas obtidas por diferentes algoritmos. Tendo em vista a necessidade de uma quantidade de pontos (dados para treinamento) relativamente alta, a abordagem de unir dados experimentais aos resultados numéricos obtidos via CFD pode ser promissora, uma vez que o aumento de dados assim proposto tem uma base física sólida na Mecânica dos Fluidos.

Objetivo

O objetivo deste trabalho é avaliar a capacidade de diferentes algoritmos de aprendizado de máquina em prever o perfil da pressão média adimensionalizada (p^*) ao longo do resalto hidráulico, em função da razão entre a coordenada longitudinal e a altura do resalto (x/h_j), a partir de dados experimentais e de simulações numéricas com CFD. Para isso, foram definidos como objetivos específicos: (i) interpolar os dados experimentais e numéricos disponíveis com o método LOWESS, a fim de gerar conjuntos de dados uniformemente distribuídos para o treinamento dos modelos; (ii) treinar e comparar os algoritmos MLP, Random Forest, XGBoost e Stacking, considerando métricas de erro e

coeficiente de determinação; (iii) avaliar o desempenho dos modelos em duas abordagens distintas — global e por valores discretos do número de Froude; e (iv) implementar uma ferramenta computacional que permita ao usuário gerar perfis preditos de p^* para valores de Fr intermediários àqueles utilizados no treinamento.

Material e Método

Modelagem com aprendizado de máquina

Neste estudo, a previsão da pressão média adimensional ao longo do ressalto hidráulico, que foi realizada por meio de modelos de aprendizado de máquina treinados com dados experimentais obtidos por Dai Prá (2011), e dados numéricos provenientes de simulações com CFD, conforme descrito por Santos *et al.* (2022). Os modelos utilizaram como variáveis de entrada o número de Froude supercrítico e a posição adimensional ao longo do ressalto, representada pela razão entre a coordenada longitudinal e a diferença de profundidade entre os conjugados. Os custos de experimentos e de simulações via CFD são significativamente superiores ao custo computacional das predições por aprendizado de máquina. Além disso, a integração de dados experimentais e simulados amplia a base de treinamento, justificando a adoção dessa abordagem.

Foram implementados quatro algoritmos de regressão supervisionada: Rede Neural Artificial (MLP), Random Forest (RF), XGBoost (XGB) e Stacking Regressor (STK). As implementações foram realizadas utilizando a biblioteca Scikit-learn (Pedregosa *et al.*, 2011), com o apoio da biblioteca XGBoost (Chen e Guestrin, 2016) para o modelo XGB. Todos os dados foram normalizados previamente com o método StandardScaler da biblioteca Scikit-learn (Pedregosa *et al.*, 2011), de modo a garantir média zero e desvio padrão unitário para cada variável de entrada e saída, favorecendo a estabilidade numérica dos modelos.

A rede neural do tipo MLPRegressor foi configurada com máximo de 2000 épocas de treinamento, ativação do tipo ReLU e otimização baseada no algoritmo Adam (Kingma e Ba, 2014), sem o uso de camadas ocultas explicitamente configuradas no código original. O modelo Random Forest foi implementado com 100 árvores (`n_estimators=100`) e semente de aleatoriedade fixa (`random_state=42`), sendo adequado para problemas com dados ruidosos e estruturas não lineares (Breiman, 2001). O algoritmo XGBoost foi utilizado na versão XGBRegressor com função objetivo `reg:squarederror` e semente de aleatoriedade igual a 42. O modelo de empilhamento (STK) foi estruturado com os três algoritmos anteriores como modelos base (RF, MLP e XGB), cujas saídas foram integradas por um modelo meta do tipo Random Forest, conforme a abordagem proposta por Wolpert (1992).

Para todos os treinamentos realizados, os dados foram divididos em 60% para treino e 40% para teste, utilizando `train_test_split` com `random_state=42`. As métricas R^2 , MSE e MAE foram calculadas exclusivamente sobre o conjunto de teste. Isso garante

uma avaliação imparcial e evita viés por sobreajuste.

As predições geradas por modelo foram avaliadas com base em três métricas: erro quadrático médio (MSE), erro absoluto médio (MAE) e coeficiente de determinação (R^2). Os algoritmos foram treinados e avaliados segundo duas abordagens complementares: (i) treinamento global, que utiliza o conjunto completo de dados de forma integrada; e (ii) treinamento por subconjuntos separados para cada valor discreto do número de Froude (Fr), permitindo a análise localizada do desempenho preditivo em diferentes condições de escoamento.

A base de dados empregada é formada por 300 pontos experimentais e 400 pontos originados com CFD. Após a interpolação LOWESS individual para cada curva, os dados foram integrados em uma única base unificada com as seguintes colunas: número de Froude (relação entre forças inerciais e gravitacionais); x/h_j (posição adimensional ao longo do escoamento), p^* (pressão adimensionalizada) e a Origem (Experimental ou CFD). Essa base foi utilizada para treinar todos os modelos, tanto na abordagem global quanto na abordagem por Fr . A partir de resultados experimentais e considerações teóricas, sabe-se que há forte correlação entre p^* e x/h_j . Sabe-se também que a forma da função $p^*(x/h_j)$ também está relacionada ao número de Froude, como pode ser visto em Santos et al. (2022), de tal maneira que a relação entre essas variáveis também foi objeto de análise. Todo tratamento e análises de dados foram feitos utilizando o Python (PYTHON SOFTWARE FOUNDATION, 2025), os códigos foram apresentados no Anexo A, Figura 10.

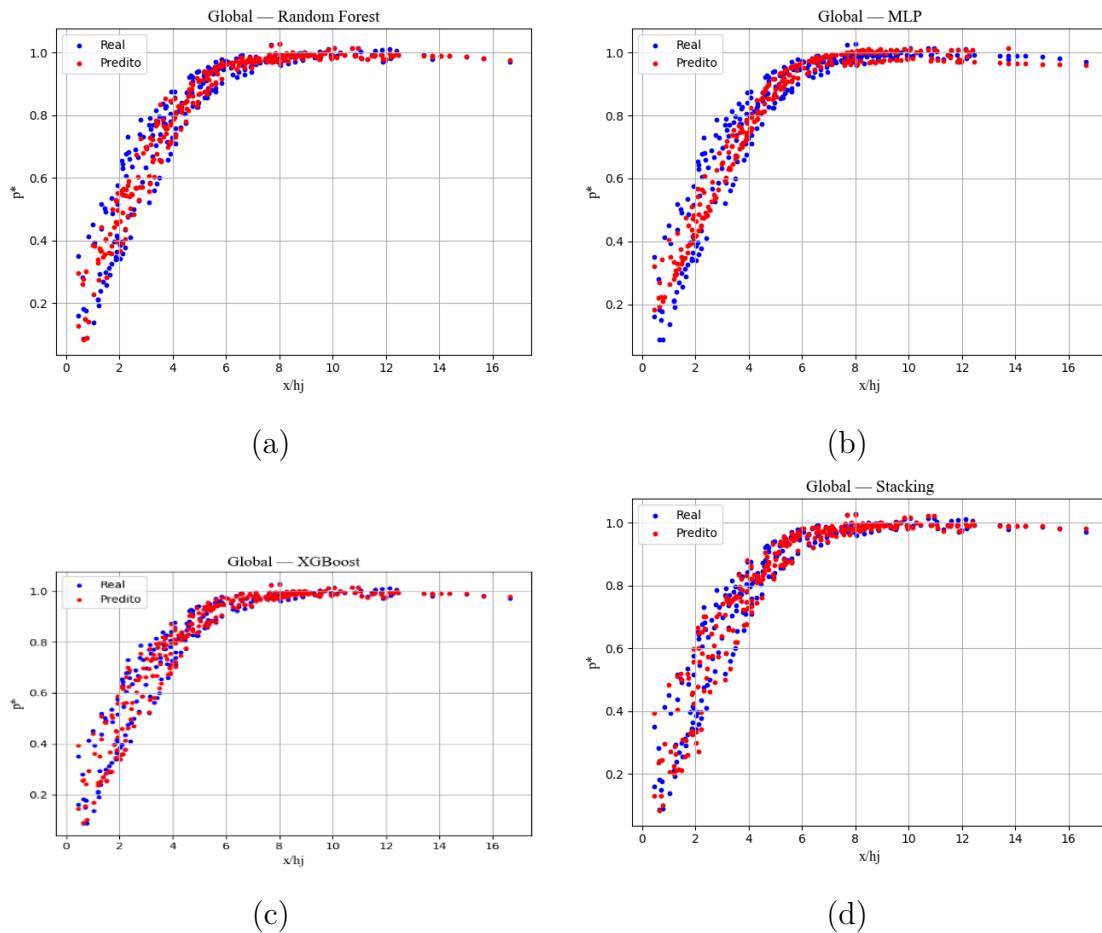
Resultados e Discussão

Para uma primeira visualização e análise qualitativa preliminar, a Figura 3 apresenta a comparação entre os valores reais de p^* (em azul) e os valores previstos (em vermelho) por quatro modelos de aprendizado de máquina — Random Forest (a), MLP (b), XGBoost (c) e Stacking (d) — utilizando como variáveis de entrada o número de Froude (Fr_1) e a posição adimensional x/h_j . Visualmente, observa-se que todos os modelos foram capazes de reproduzir com razoável fidelidade o perfil crescente de p^* , especialmente nas regiões intermediárias e de estabilização, em que os valores de p^* se aproximam do valor máximo da série (em torno de 1,0). As discrepâncias mais evidentes entre os dados reais e os valores previstos concentram-se nas regiões iniciais do ressalto, próximas de x/h_j igual a zero, onde há maior variabilidade nos dados e transições abruptas.

O modelo Random Forest (a) apresenta previsões suavizadas e uma leve tendência à subestimação de p^* na faixa inicial, mas mantém boa aderência ao comportamento global. O modelo MLP (b), por outro lado, mostra-se mais sensível às oscilações locais, com maior dispersão nos valores previstos em regiões de transição, identificando padrões de forma menos específica, tal como se houvessem padrões binários com agrupamentos

densos das soluções. O XGBoost (c) apresenta bom desempenho especialmente na região de crescimento acentuado de p^* , com boa aproximação aos dados reais entre x/h_j igual a 2 e 6. O modelo Stacking (d), ao combinar as estimativas dos modelos base, oferece um ajuste visualmente mais estável e coerente, com menor dispersão e boa representação das variações tanto locais quanto globais. A consistência desse modelo ao longo de toda a faixa de x/h_j sugere sua adequação como modelo de referência para as análises subsequentes, especialmente em contextos que envolvem dados híbridos provenientes de experimentos e simulações numéricas.

Figura 3: Comparação entre todos os dados experimentais, simulações via CFD e os resultados modelados: (a) Random Forest; (b) MLP; (c) XGBoost; (d) Stacking.



Fonte: OS AUTORES, 2025

Para uma análise detalhada da capacidade dos modelos empregados ao problema em questão, foram analisados os resultados de predições para cada número de Froude separadamente. Como pode ser visto na Figura 4, os resultados obtidos com o modelo Random Forest demonstram elevada qualidade preditiva tanto para os dados simulados via CFD quanto para os dados experimentais, com erros relativos baixos para a maior parte do domínio. Nos casos provenientes de CFD ($Fr = 2$ a 9), observa-se uma excelente sobreposição entre os valores reais e os preditos ao longo de todo o perfil adimensional de pressão.

Mesmo nas regiões iniciais do ressalto, onde o gradiente da variável é mais acentuado, o modelo manteve boa fidelidade, embora com maior desvio relativo. Pequenas dispersões locais foram observadas para Fr intermediários (como $Fr = 6$ e 7), mas sem comprometer a tendência geral, evidenciando a robustez do Random Forest na representação de padrões não lineares. Para os casos experimentais ($Fr = 4.38$ a 9.26), o desempenho do modelo permaneceu satisfatório, com boa aderência às curvas reais, mesmo diante da maior variabilidade típica de dados físicos. A previsão dos perfis de p^* foi consistente ao longo de toda a extensão do ressalto, com pequenas diferenças pontuais concentradas nos primeiros valores de x/h_j .

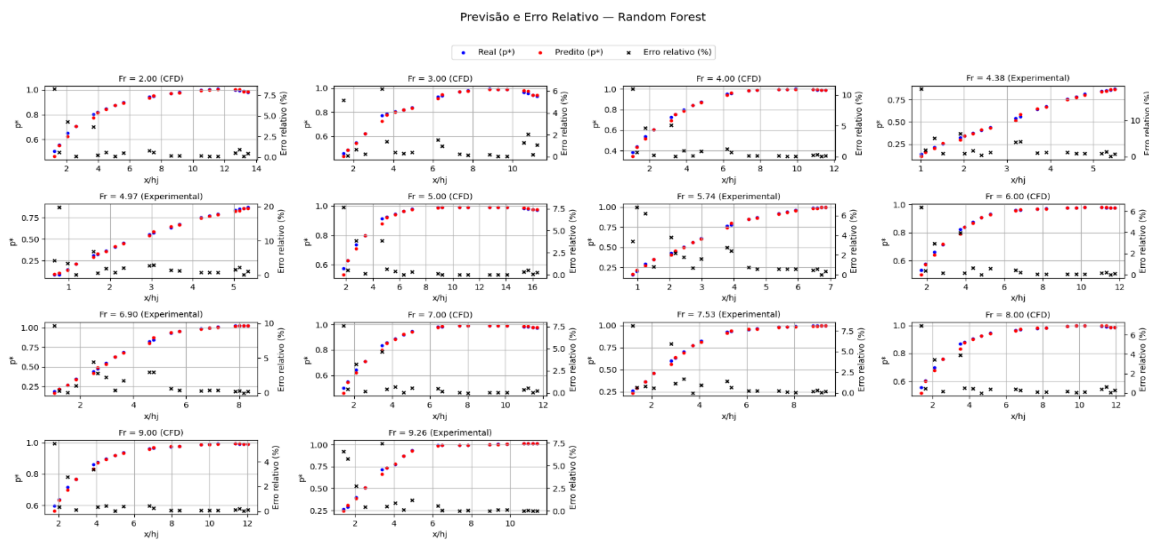


Figura 4: Comparações com o modelo Random Forest para cada número de Froude.

Fonte: OS AUTORES, 2025

Os resultados obtidos com o MLP, apresentados na Figura 5, indicam que o modelo foi capaz de capturar de forma razoável a tendência global dos dados. No entanto, observam-se desvios para valores elevados de x/h_j . Essa limitação, visível em alguns perfis, sugere desempenho inferior ao do Random Forest em termos de aderência local, mesmo que a resposta geral se mantenha coerente com o comportamento esperado da variável p^* .

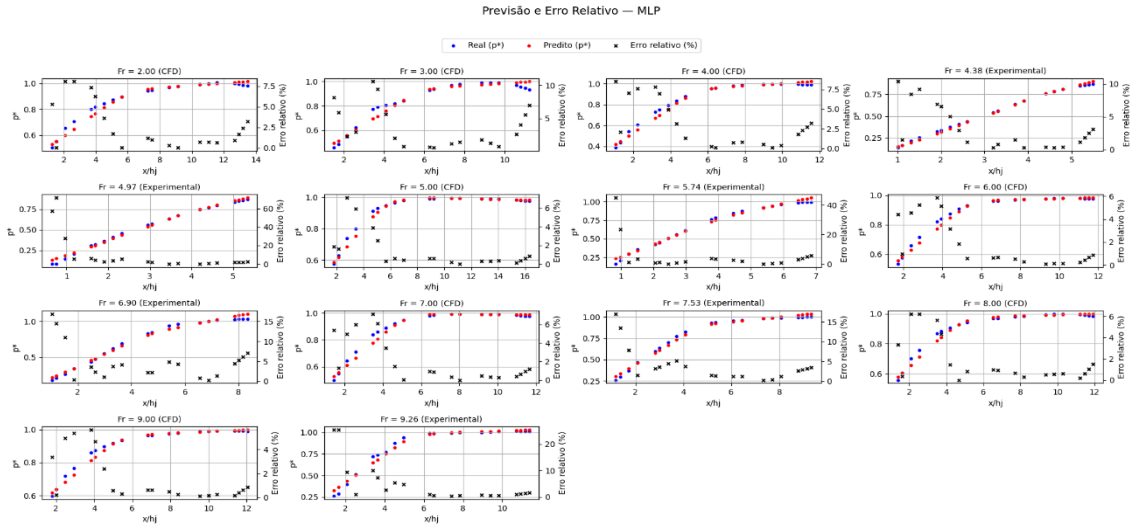


Figura 5: Comparações com o modelo MLP para cada número de Froude

Fonte: OS AUTORES, 2025

Os resultados obtidos com o XGBoost, sintetizados na Figura 6, indicam desempenho competitivo com o Random Forest, principalmente na região central das curvas. As previsões tendem a acompanhar bem o crescimento de p^* em função de x/h_j , inclusive com boa sobreposição entre pontos reais e preditos em muitos casos. Contudo, observa-se que, para baixos valores de x/h_j , há uma tendência maior de subestimativa, especialmente visível para os Fr mais baixos. Em comparação com o MLP, o XGBoost demonstrou menor dispersão e menor ocorrência de discrepâncias.

Em relação aos dados experimentais, o XGBoost apresentou resposta visualmente adequada, como indicado na Figura 6, com uma modelagem que respeita o padrão de crescimento observado empiricamente. A precisão é notável para Fr intermediários e altos, como 6.90 e 7.53, com discretas divergências em pontos mais distantes. Em síntese, embora alguns desvios pontuais ainda sejam visíveis, o XGBoost se mostra mais robusto que o MLP, embora menos estável que o Random Forest em alguns cenários.

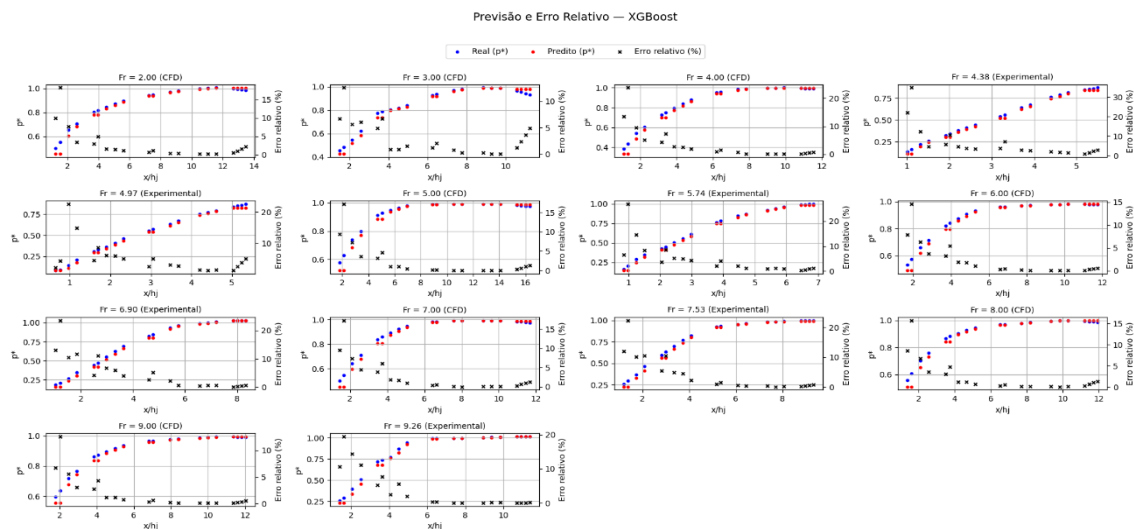


Figura 6: Comparações com o modelo XGBoost para cada número de Froude.

Fonte: OS AUTORES, 2025

A análise dos gráficos do modelo Stacking (Figura 7) evidencia uma excelente capacidade de previsão tanto para os dados de CFD quanto para os experimentais, com elevada sobreposição entre valores reais e previstos. Nas simulações, as curvas seguem de forma precisa a evolução de p^* em função de x/h_j , com pequenas discrepâncias apenas em $Fr = 2.00, 3.00$ e 8.00 , especialmente em discrepâncias na porção final. Para os dados experimentais, observa-se bom desempenho em todos os casos. Ainda que haja leve superestimação pontual em $Fr = 5.74$ e 9.26 , o modelo se manteve consistente, reproduzindo adequadamente o comportamento hidráulico ao longo de todo o domínio.

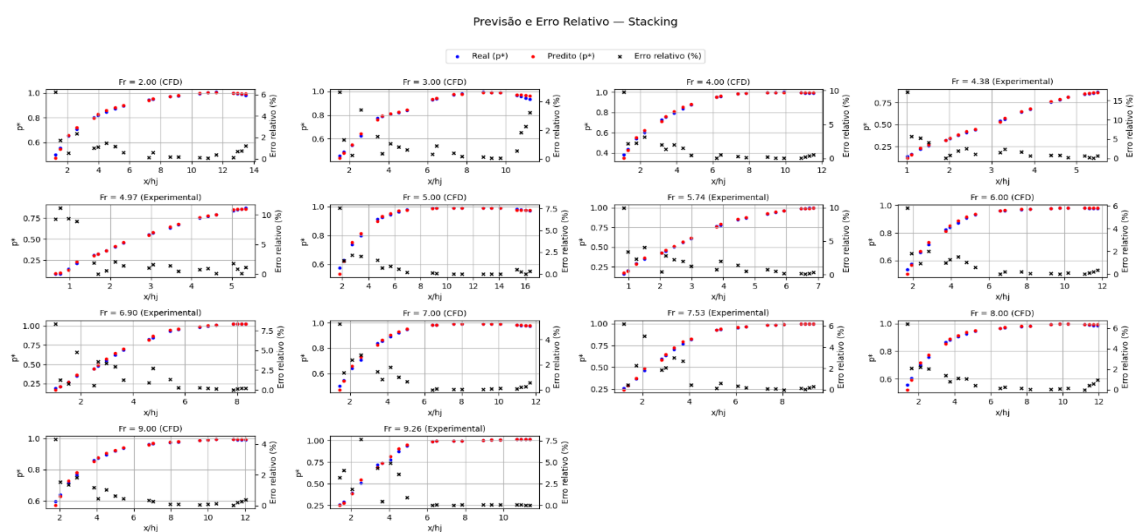


Figura 7: Comparações com o modelo Stacking para cada número de Froude.

Fonte: OS AUTORES, 2025

Comparação quantitativa entre os modelos

A análise quantitativa dos modelos aplicados à base de dados completa revela que todos os algoritmos de aprendizado de máquina testados apresentaram desempenho satisfatório na previsão de p^* em função de x/h_j , com coeficientes de determinação (R^2) superiores a 0,94, como indicado na Figura 8. Entre os modelos avaliados, o XGBoost obteve o melhor desempenho global, com R^2 próximo de 0,99, erro quadrático médio (MSE) inferior a 0,01 e erro absoluto médio (MAE) inferior a 0,06, indicando elevada capacidade de generalização e ajuste aos dados. O modelo Stacking, que combina previsões de múltiplos algoritmos, apresentou desempenho muito próximo ao do XGBoost, superando os demais em termos de R^2 e mantendo MAE e MSE em níveis baixos. Por outro lado, a rede neural MLP obteve os piores indicadores entre os quatro modelos avaliados, com o menor valor de R^2 (em torno de 0,94) e os maiores valores de MSE e MAE, sugerindo maior sensibilidade a flutuações locais e possível sobreajuste. O Random Forest apresentou desempenho intermediário, com boa estabilidade e métricas superiores às da MLP, embora ligeiramente inferiores às dos métodos baseados em boosting e combinação.

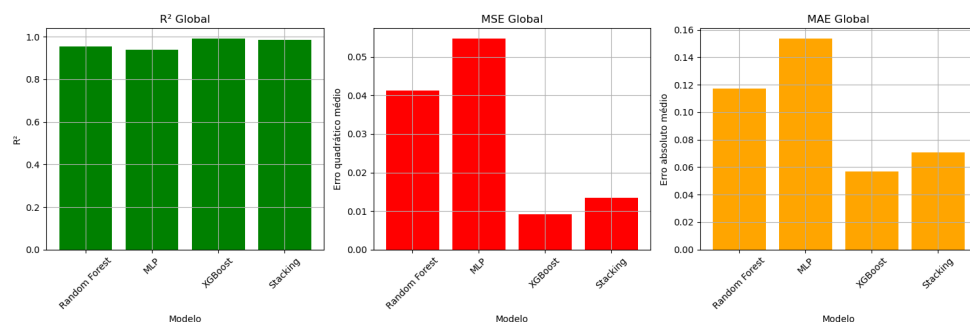


Figura 8: Métricas para a modelagem de todos os dados em conjunto.

Fonte: OS AUTORES, 2025

A avaliação por Fr apresentada na Figura 9 revela aspectos importantes sobre a capacidade preditiva dos modelos em diferentes regimes do ressalto hidráulico. A Figura 9a, que apresenta os valores de R^2 , mostra que o modelo Stacking obteve desempenho consistentemente elevado, superando os demais em quase todos os valores de Fr analisados, com exceção de $Fr = 2$, onde o Random Forest apresentou desempenho ligeiramente superior. O MLP apresentou os menores valores de R^2 , especialmente nos menores Fr, o que sugere uma relativa limitação do modelo em representar a não linearidade mais acentuada nessas condições, embora os valores de R^2 tenham sido altos.

A Figura 9b mostra que os menores valores de erro quadrático médio (MSE) também são associados ao Stacking, com destaque para os Fr experimentais, nos quais o modelo obteve os melhores resultados. O XGBoost, por outro lado, apresentou maiores variações no MSE, com picos notáveis em $Fr = 5, 8$ e 9 . Essa tendência se repete na Figura 9c, que mostra o erro absoluto médio (MAE), em que o modelo Stacking mantém os menores valores, reforçando sua robustez. O Random Forest apresentou desempenho competitivo,

especialmente nos Fr extremos (2 e 9.26), enquanto o MLP registrou os maiores erros em quase todas as faixas.

Figura 9: Comparação entre os modelos preditivos para valores individuais de Fr: (a) Coeficiente de determinação (R^2); (b) Erro quadrático médio (MSE); (c) Erro absoluto médio (MAE).



Fonte: OS AUTORES, 2025

Apesar da convergência nas métricas R^2 , MSE e MAE indicarem o Stacking como melhor preditor, a análise qualitativa revelou que, em alguns casos, esse modelo gerou discrepâncias fisicamente inconsistentes nos perfis de p^* , o que não ocorreu com o Random Forest. Isso sugere que métricas estatísticas podem não capturar desvios locais relevantes, reforçando a necessidade de avaliação visual e coerência hidráulica na aplicação de modelos de aprendizado de máquina a escoamentos.

Previsão a partir de um dos modelos testados

Com o objetivo de ampliar a aplicabilidade prática dos modelos treinados, foi implementada uma funcionalidade interativa (ver código na Figura 10) que permite ao usuário fornecer um valor do número de Froude (Fr) de interesse e, com base nesse valor, obter a predição do perfil da variável adimensional p^* ao longo de x/h_j . Essa predição é possível mesmo para valores de Fr não pertencentes ao conjunto original de dados, uma vez que os modelos foram treinados de forma global, utilizando como variáveis de entrada tanto Fr quanto x/h_j .

Dessa forma, a interpolação entre diferentes curvas de Fr não é realizada por métodos tradicionais, como interpolação linear entre curvas conhecidas, mas sim por inferência estatística realizada diretamente pelos algoritmos de aprendizado de máquina. Os modelos aprendem a relação multivariada entre Fr , x/h_j e p^* , permitindo estimar o perfil completo de p^* para qualquer valor de Fr dentro do intervalo de treinamento. A entrada fornecida ao modelo é uma malha de pontos de x/h_j associada ao valor constante de Fr especificado, e a predição do perfil é feita ponto a ponto por meio do modelo treinado.

O usuário pode escolher entre quatro modelos: Random Forest (100 árvores), MLP (2000 iterações), XGBoost (erro quadrático) e Stacking Regressor, que combina os três anteriores com Random Forest como meta-estimador. A predição envolve padronização das variáveis, aplicação do modelo e retorno aos valores reais de p^* . Essa abordagem permite a geração de perfis contínuos e fisicamente coerentes de p^* em função de x/h_j , mesmo para valores intermediários de Fr , ampliando as possibilidades de aplicação prática em contextos computacionais e experimentais. Destaca-se que para valores de Fr fora do intervalo dos dados de treinamento, será realizada extrapolação, cujos resultados devem ser utilizados com cautela, devendo ser evitados.

Conclusão

O presente artigo avaliou a capacidade de algoritmos de aprendizado de máquina em prever o perfil da pressão média adimensionalizada (p^*) ao longo do escoamento em ressaltos hidráulicos, em função da razão x/h_j . A partir da combinação de dados experimentais e resultados de simulações numéricas por CFD, foram desenvolvidos modelos regressivos com redes neurais artificiais (MLP), Random Forest, XGBoost e Stacking, utilizando abordagem global e também por número de Froude (Fr) individualizado. Os dados foram previamente interpolados por LOWESS, assegurando suavidade nas curvas e densidade adequada de pontos para o treinamento. As predições demonstraram que o modelo Random Forest apresentou o melhor desempenho global em termos de acurácia e consistência física, com excelente capacidade de seguir a tendência dos dados para diferentes valores de Fr , inclusive nos extremos. O modelo Stacking também obteve bom desempenho, mas apresentou discrepâncias localizadas em valores elevados de x/h_j , que não se alinham

com o comportamento esperado para o final do ressalto, o que motivou sua não adoção como preditor para aplicações práticas. O modelo MLP mostrou-se sensível a variações locais, enquanto o XGBoost teve desempenho intermediário. Os resultados quantitativos confirmaram a superioridade do Random Forest em praticamente todas as métricas e valores de Fr testados, validando sua escolha como algoritmo principal para a geração de perfis de p^* interpolados em novos valores de Fr não experimentais. A estratégia adotada permitiu criar uma base contínua de predição com fundamentação física, potencialmente útil em aplicações de projeto, análise e instrumentação de ressaltos hidráulicos. Para trabalhos futuros, sugere-se a aplicação de novos bancos de dados que contenham outras variáveis de entrada relacionadas ao ressalto, como o comprimento do ressalto e o comprimento do rolo do ressalto, assim como os momentos centrais curtose e assimetria das pressões instantâneas junto ao fundo do ressalto. Também seria útil desenvolver abordagens que combinem métricas quantitativas e qualitativas para avaliação preditiva, assegurando maior coerência física nos resultados.

Referências

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. In 12th, Symposium on Operating Systems Design and Implementation, pp. 265-283.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794.
- Dai Prá, M. (2011). Uma abordagem para determinação das pressões junto ao fundo de dissipadores de energia por ressalto hidráulico. Tese (Doutorado), Universidade Federal do Rio Grande do Sul.
- Endres, L.A.M. (1990). Contribuição ao desenvolvimento de um sistema para aquisição e tratamento de dados de pressões instantâneas em laboratório. Dissertação (Mestrado), Universidade Federal do Rio Grande do Sul.
- Ikinciogullari E. Emiroglu ME. Estimation of triangular labyrinth side weir discharge capacity using schmidt approach. *Sigma J Eng Nat Sci* 2019;37:289–303.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Marques, M.G.; Drapeau, J.; Verrette, J-L. (1997). “Flutuação de pressão em um ressalto hidráulico”. *Revista Brasileira de Recursos Hídricos*, Vol. 2, No. 2, Jul/Dez 1997, pp. 45-52.
- Pakgar F, Daneshfaraz R, Joudi AR Numerical simulation of flow on a siphon spillway and investigation of the effect of a bottom/outlet angle on hydraulic parameters. *Sigma*

J Eng Nat Sci 2016;34:279–290

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

PYTHON SOFTWARE FOUNDATION. *Python Language*. Disponível em: <<https://www.python.org/doc/>>. Acesso em: 06 mar. 2025.

Santos, I. N. R. R. dos; Simões, A. L. A.; Minho, T. R. P.; Jesus, Y. S. de; Queiroz, L. M.; Schulz, H. E.; Porto, R. M. Definição para o comprimento do ressalto hidráulico em canais retangulares empregando a dinâmica dos fluidos computacional. In: Congresso Latinoamericano de Hidráulica – CLH, 30., 2022, Foz do Iguaçu. Anais eletrônicos... Foz do Iguaçu: IAHR, 2022.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(2), 241-259.

AGRADECIMENTOS

Este estudo foi apoiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de financiamento 001 e número do processo 88881.708215/2022-01 – PDPG Emergencial para Consolidação Estratégica de Programas de Pós-Graduação (PPGs) stricto sensu com conceitos 3 e 4; 88881.691452/2022-01 - PDPGPOSDOC/Programa de Pós-Doutorado Estratégico do Programa de Pós-Graduação em Desenvolvimento (PDPG), ambos do Mestrado em Meio Ambiente, Água e Saneamento – MAASA – UFBA.

Anexo-A

```
# -*- coding: utf-8 -*-
"""
Predição da curva p* em função de x/hj para um valor de Fr fornecido pelo usuário.
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor, StackingRegressor
from sklearn.neural_network import MLPRegressor
import xgboost as xgb

def executar_predicao_curva(df_interp, n_x=200):
    # Modelos disponíveis
    modelos_disponiveis = {
        'Random Forest': RandomForestRegressor(n_estimators=100, random_state=42),
        'MLP': MLPRegressor(random_state=42, max_iter=2000),
        'XGBoost': xgb.XGBRegressor(objective='reg:squarederror', random_state=42),
        'Stacking': StackingRegressor(
            estimators=[
                ('rf', RandomForestRegressor(n_estimators=100, random_state=42)),
                ('mlp', MLPRegressor(random_state=42, max_iter=2000)),
                ('xgb', xgb.XGBRegressor(objective='reg:squarederror', random_state=42))
            ],
            final_estimator=RandomForestRegressor(random_state=42)
        )
    }

# 1. Entrada do usuário
try:
    fr_usuario = float(input("Digite o valor de Fr para predizer o perfil de p*: "))
except ValueError:
    print("Valor de Fr inválido. Encerrando.")
    return

print("\nModelos disponíveis:")
for i, nome in enumerate(modelos_disponiveis.keys()):
    print(f"{i+1}. {nome}")

try:
    idx_modelo = int(input("Escolha o número do modelo que deseja usar: ")) - 1
    nome_modelo = list(modelos_disponiveis.keys())[idx_modelo]
except:
    print("Modelo inválido. Encerrando.")
    return

modelo = modelos_disponiveis[nome_modelo]
print(f"\n Modelo selecionado: {nome_modelo}")
```

```
# 2. Treinamento com todos os dados
X_full = df_interp[['Fr', 'x_hj']].values
y_full = df_interp['p_star'].values
scaler_X = StandardScaler().fit(X_full)
scaler_y = StandardScaler().fit(y_full.reshape(-1, 1))
X_scaled = scaler_X.transform(X_full)
y_scaled = scaler_y.transform(y_full.reshape(-1, 1)).flatten()
modelo.fit(X_scaled, y_scaled)

# 3. Predição do perfil
xhj_grid = np.linspace(df_interp['x_hj'].min(), df_interp['x_hj'].max(), n_x)
X_pred = np.column_stack([np.full_like(xhj_grid, fr_usuario), xhj_grid])
X_pred_scaled = scaler_X.transform(X_pred)
y_pred_scaled = modelo.predict(X_pred_scaled)
y_pred = scaler_y.inverse_transform(y_pred_scaled.reshape(-1, 1)).flatten()

# 4. Exibir gráfico
plt.figure(figsize=(8, 5))
plt.plot(xhj_grid, y_pred, label=f'Fr = {fr_usuario:.2f}', color='darkred')
plt.xlabel('x/hj')
plt.ylabel('p*')
plt.title(f'Curva prevista de p* para Fr = {fr_usuario:.2f} - {nome_modelo}')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()

# 5. Exportar CSV
df_saida = pd.DataFrame({
    'Fr': fr_usuario,
    'x_hj': xhj_grid,
    'p_star_pred': y_pred
})
nome_arquivo = f'perfil_predito_Fr_{str(fr_usuario).replace(".", "p")}_{nome_modelo.replace(" ",
"_")}.csv'
df_saida.to_csv(nome_arquivo, index=False)
print(f"\n Arquivo salvo: {nome_arquivo}")

# === Execução ===
if __name__ == '__main__':
    df_interp = pd.read_csv('dados_ressalto_interpolados.csv')
    executar_predicao_curva(df_interp)
```

Figura 10: Código em Python para execução do modelo.

Fonte: OS AUTORES, 2025