

Machine Learning Based Prediction of Hydraulic Jump Lengths in Stilling Basins for Barrages and Low-Head Hydraulic Structures

Abdul Rehman^{a*}, Habib ur Rehman^{a,b}, Noor Muhammad Khan^a

^a Civil Engineering Department, University of Engineering and Technology, Lahore, Pakistan

^b Faculty of Engineering and Technology, The University of Lahore, Lahore, Pakistan

*Corresponding author. E-mail: abdulrehman@uet.edu.pk

Received: 17 July 2025, Revised: 09 December 2025, Accepted: 24 December 2025, Published: 01 January 2026

KEY WORDS

Appurtenances
Hydraulic jump
Machine learning
Stilling basins
Low Froude number

ABSTRACT

A hydraulic-jump-type stilling basin is a popular choice for energy dissipation, where the high-velocity, supercritical flow from a dam spillway transitions into a low-velocity, subcritical flow. Stilling basins are designed such that the hydraulic jump forms within the basin, ensuring the safety of the downstream natural streambed. Thus, prediction of length of hydraulic jump is critical in deciding the basins dimensions. Experimental data on hydraulic jump type stilling basins under low inflow Froude number were collected from the literature. Six machine learning (ML) models were trained and tested using experimental data related to low inflow Froude number stilling basins. For each model, the feature importance score was calculated using a permutation-based method, which showed that the inflow Froude number (F_1) and the sequent depth ratio (Y) were most significant features for prediction of the length of the hydraulic jump across all ML models. The evaluation of the models' performance indicated that the artificial neural network based model and CatBoost performed better, with higher coefficients of determination (from 0.85 to 0.93), lower error metrics, and a balanced distribution of errors during training, testing, and validation. Whereas Random Forest and AdaBoost based models showed least coefficient of determination (both having 0.84) at the model training stage. Support vector machine based model performed least at the testing stage with coefficient of determination of 0.54.

1. Introduction

The flow emerging from the spillway of a dam or barrage has significant kinetic energy that must be dissipated to a safe level for the natural streambed downstream of the dam. Stilling basins are typically used for energy dissipation downstream of dams. Peterka [1] made a significant contribution to the study of stilling basins, where several types of stilling basins were proposed based on factors such as the incoming flow Froude number and tailwater levels. The stilling basins proposed by Peterka [1] are popularly referred to as USBR stilling basins. Among the different types, hydraulic jump stilling basins remain a widespread choice, which dissipates the high energy of the incoming flow through turbulence created in the flow roller of a hydraulic jump.

The simplest form of hydraulic jump is the classic hydraulic jump (CHJ), which occurs on a horizontal, flat, and fixed bed. The CHJ has a specific length and requires a certain sequent depth for its formation. In hydraulic jump type stilling basins, the jumps are forced to form at a designated location using appurtenances which include chute blocks, deflector blocks, baffle blocks, and end sills etc. These structures help reduce the jump length and compensate for any shortages in tailwater levels, making the construction of the stilling basins more cost-effective.

Regarding low-head dams and barrages, which typically operate at inflow Froude numbers ranging between around 2.5 to 4.5, oscillating rollers form in hydraulic jumps, creating waves that may travel downstream of the stilling basin [2]. For low inflow

Froude numbers, Peterka [1] has recommended a USBR type IV stilling basin. This stilling basin has fewer appurtenances such as deflector blocks and an end sill. The recommended length of the stilling basin is equal to a CHJ as in USBR-IV stilling basin [1]. Research by George [3] and Bhowmik [4] suggest modifications with baffle blocks within the stilling basin, resulting in a shorter basin length.

Forecasting dimensions of a hydraulic jump is decisive in stilling basin designs. The dimensions which include the sequent depth ratio and the length of the hydraulic jump are estimated employing various methods. Traditional methods are based on equations derived by applying momentum balance before and after a hydraulic jump, and empirical formulations derived from laboratory and field observations [1,3], [4,5]. Furthermore, numerical modeling results have also been used in predicting hydraulic jump characteristics by providing insights into complicated flow behavior [6].

Recent decades have seen acceptance of using Machine learning (ML) based models in hydraulic engineering applications due to their ability to solve complex, non-linear, and data-intensive problems [7]. ML based models have been successfully applied in flow forecasting [8,9], sediment transport modeling [10,11,12], optimization of hydraulic structures [13], [14,15], improvement of the performance of water distribution networks and reservoir operation decisions [16,17]. Multilinear regression (MLR) is also a popular tool for establishing relationships among different dependent and independent variables. However, ML-based models have outperformed MLR at several instances [11,15].

Numerous studies demonstrated the success of ML models in predicting hydraulic jump characteristics. Roushangar et al. [18] found that Gene expression programming (GEP) performed better than empirical methods in predicting energy dissipation and sequent depth ratio in expanding channels with different sill and step arrangements. Sauda [19] compared the performance of artificial neural networks (ANN) and multilinear regression (MLR) in predicting the length of hydraulic jump under multiple gate openings. It was found that ANN performed better than MLR. Similarly, Ho [20] found that gradient boosting, ANN, and Extra Trees (ET) are better than empirical equations for predicting hydraulic jump length in rectangular channels.

Gul et al. [21] applied several ML models to estimate the sequent depth ratios and hydraulic jump lengths in abruptly expanding stilling basins. Azim et al. [22] and Karbasi et al. [23], assessed GEP to predict the hydraulic jump length on both smooth and gravel-roughened beds. Heydari [24] used self-adaptive Extreme Learning Machine (ELM) for similar predictions. Baharvand et al. [25] applied ML-based approaches to check accuracy of a non-linear algorithm to estimate sequent depth ratios on different channel bed conditions. Xu et al. [7] and Daneshfaraz [15] used neural network-based ML models to predict hydraulic jump length in smooth, gravel-roughened, and sudden expansion stilling basins.

The literature has shown promising results for application of ML models in predicting hydraulic jump characteristics. However, the prediction of hydraulic jump characteristics specifically for low inflow Froude numbers have not been extensively considered using ML models. Therefore, this study aimed to develop predictive models for hydraulic jump characteristics at low inflow Froude numbers with and without appurtenances. Six different ML models, including artificial neural networks (ANN), k-nearest neighbors (kNN), support vector machines (SVM), random forest (RF), CatBoost, and AdaBoost, were selected. Furthermore, this study tested the ML models on datasets from separate experimental campaigns, which has been rarely done in similar studies. The testing of ML models on datasets from separate experimental campaigns assessed the generalized ability of the ML models.

2. Methodology

2.1. Dimensional analysis

Figure 1 exemplifies a hydraulic jump of length L_j in a stilling basin, having baffle blocks and a sill of height h_1 and h_2 respectively. In figure 1 y_1 and y_2 denote the inflow depth and sequent depth of a hydraulic jump, X denotes the distance between the baffle blocks and end sill of a stilling basin. Thus, the height of sill and baffle blocks, the distance between them, inflow and sequent depths are the most influential geometric parameters affecting length of a hydraulic jump. Furthermore, flow parameters such as inflow velocity (V_1); and fluid properties such as density (ρ), viscosity (μ) and surface tension (σ) are also influencing hydraulic jump properties. These parameters can be shown as equation (1).

$$L_j = f(V_1, g, \rho, \mu, \sigma, y_1, y_2, h_1, h_2, x,) \quad (1)$$

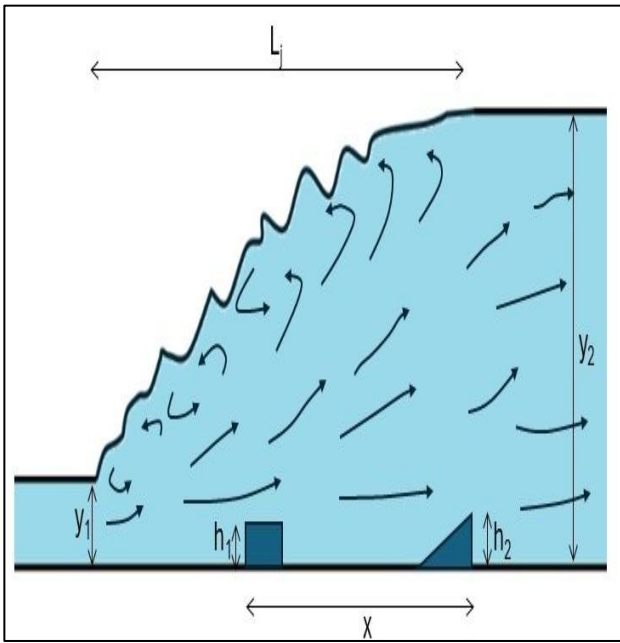


Fig.1. Physical dimensions of a hydraulic jump.

Fundamental dimensionless parameters influencing the length of hydraulic jump are derived by applying Buckingham Pi theorem. The following equation can be obtained after the analysis.

$$\frac{L_j}{y_1} = f \left[\frac{V_1}{\sqrt{gy_1}}, \frac{V_1 y_1}{\nu}, \frac{\rho V_1^2 y_1}{\sigma}, \frac{y_2}{y_1}, \frac{h_1}{y_1}, \frac{h_2}{y_1}, \frac{X}{y_1} \right] \quad (2)$$

Where L_j/y_1 can be said to be relative jump length (L), similarly terms on right side of equation are referred to as the inflow Froude number (F_1), Reynolds number (R), Weber number (W), sequent depth ratio (Y) of the hydraulic jumps, relative block height (H_1), relative sill height (H_2), and relative sill distance (X). Thus, Eq. 2 can be rewritten as.

$$L = f(F_1, R, W, Y, H_1, H_2, X) \quad (3)$$

In equation (3) the terms like Reynolds number and Weber number can be ignored because the experimental observations used in this study involve turbulent flows. Therefore, the following equation was derived.

$$L = f(F_1, Y, H_1, H_2, X) \quad (4)$$

2.2. Datasets Description

To test and train ML models, the data available in Bhowmik [4] and Peterka [1] was used in this study. Bhowmik [4] has performed comprehensive experiments focusing on hydraulic jumps at low inflow Froude numbers, wherein the influence of sills and baffle blocks on the length of the hydraulic jumps was investigated. A distinct feature of low inflow hydraulic jumps is the formation of oscillatory rollers generating surface waves that may travel downstream and may damage the natural streambed or the stilling

basin floor. Bhowmik [4] tested several configurations of baffle blocks and found a few arrangements that may eliminate the downstream wave problem while simultaneously shortening the required stilling basin length by 7-40%.

The experimental setup of Bhowmik [4] consisted of a 0.61 m wide channel where supercritical flow was created using a sluice gate. Initial tests were conducted without any appurtenance, and tailwater adjustments were performed to create a hydraulic jump. This was referred to as Basin A. Similarly, Basins B and C had end sills and baffle blocks, respectively, which were used to study the effects of these appurtenances without any tailwater adjustments. Furthermore, the testing program involving stilling basins was provided with different baffle-block and end-sill configurations. These basins were named D through L. Bhowmik [4] study aimed to achieve a shorter stilling basin length without waves travelling downstream of the stilling basin. Basin L showed satisfactory performance over a wider range of inflow Froude numbers (2.82 to 5.26), whereas the other configurations showed better performance for inflow Froude numbers ranging between 4 to 5. Inflow Froude numbers below 3 remained challenging for most configurations. Moreover, some appurtenance configurations did not produce stable hydraulic jumps for all the tested inflow Froude numbers. Further details can be found in [4]. For the present study, the observations collected from Basin A and the experimental cases that showed satisfactory hydraulic jumps were selected. The configurations for which data were selected are illustrated in Figure 2, where the side view and top view of the stilling basins for which the data were selected are shown. The arrow marks the flow direction. Multiple forms of baffle blocks and sill were tested. The dimensions, shape and distance between baffle blocks and sill are mentioned on this figure where the numbers are dimensions in mm units. Basin F in the experimental investigation of Bhowmik [4] was prepared with oblique arrangement of baffle blocks converging on a point along the center line (Figure 2 (a)). This was done anticipating concentration of disturbances near the centerline and intermixing of high velocity jets. Similarly, Basins G, H and L have baffle blocks aimed at concentrating flow near the centerline of flow. In the case of Basin K the arrangement of baffle blocks is different from the rest of the basins. The baffle blocks were placed with opposite orientation aiming to increase intermixing of high velocity jets.

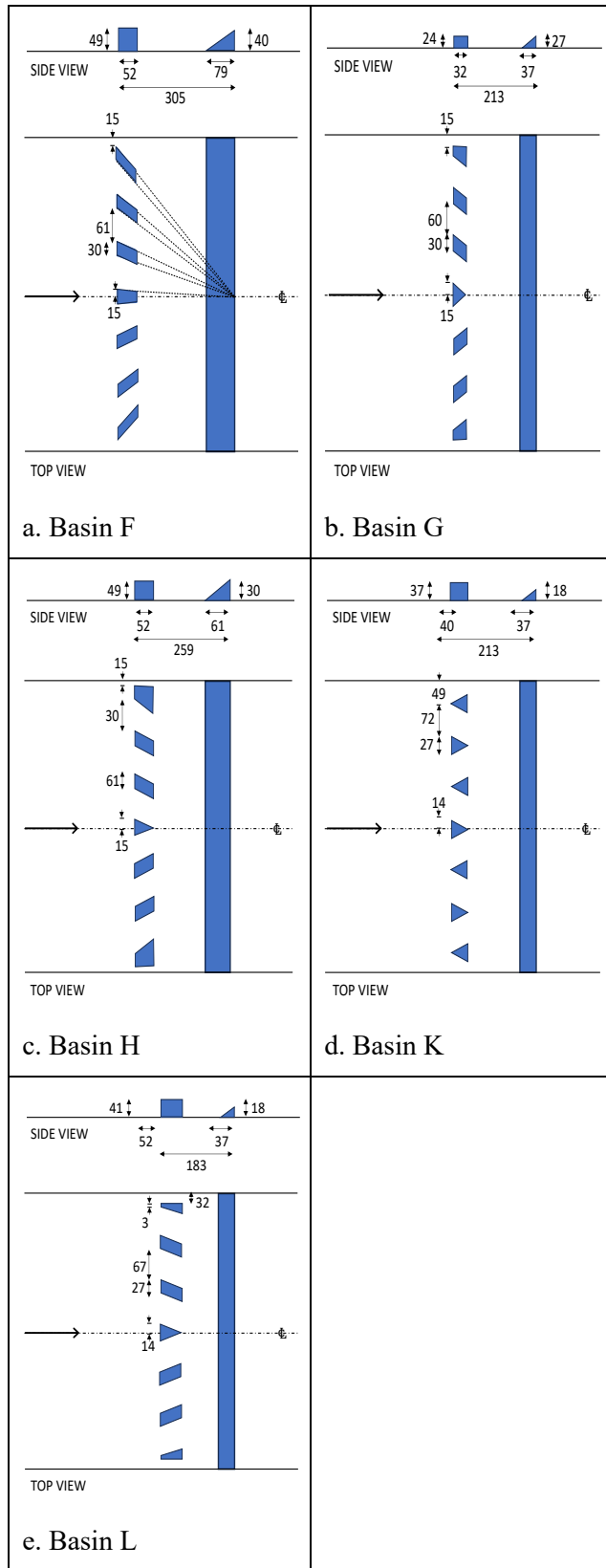


Fig. 2. Arrangements for basins F, G, H, K, and L used in the study by Bhowmik [4]. All dimensions are in mm. The single-headed arrow on all sketches marks the flow direction.

When the experimental observations of Bhowmik [4] were used in this study, the selected experimental data was converted into dimensionless form using the ratios derived in section 2.1 (Equation 4). This facilitates generalizing the findings of this study to

field scale prototype structures or other investigation carried out at a different scale. The refined dataset, having 104 data instances, was further segregated into two parts: 75% of the data served as a training dataset for model training and in-phase validation, and 25% was reserved for testing. This segregation of the data ensures that the trained models are evaluated using unseen data to assess their generalization performance. Furthermore, the ML models were again tested on 32 data instances pertinent to this study, which were taken from Peterka [1], so that the performance of the ML models again tested using entirely different data prepared in a separate experimental program. Hereafter in this article, the 25% data set aside from Bhowmik [4] is referred to as “Testing dataset-1,” and the dataset from Peterka [1] is referred to as “Testing dataset-2”. Table 1 summarizes characteristics of data used in this study.

Table 1.

Characteristics of the datasets used.

	F_1	Y	H_1	H_2	X	L
Training dataset	2.1-5.4	2.4-6.9	0-1.9	0-1.6	0-12	9.1-37
Testing dataset-1	2.3-5.3	2.8-6.7	0-1.8	0-1.6	0-12	10.9-30.7
Testing dataset-2	1.7-5	2.3-9.1	0	0	0	8.8-39

2.3. Implementation of ML models

In this study Orange [26], a freely available software for machine learning, was used to implement ML models. This software has been applied in numerous areas of study, such as [27], [28]. Orange [26] features a user-friendly graphical user interface (GUI) with a drag-and-drop workflow, providing a visual programming environment. Multiple ML models can be trained and tested in the same workflow, for which the software provides tools for data preprocessing, feature selection, result visualization, and performance metric comparison. However, regardless of its accessibility and visualization capabilities, the software has certain limitations. Like for example in this software most algorithms are configured using preset widgets, this allows relatively limited options for in-depth model customization and hyperparameter tuning.

Each ML model is controlled by a set of configuration variables, termed hyperparameters, which govern the model's structure and learning process. These hyperparameters are adjusted before the training process of any ML model and are

significant to the learning process of ML model. Among the various approaches applied for selecting optimum hyperparameters, this study used the random search method for all the ML models, where the hyperparameters were initially selected from some previous similar studies, and the performance during the validation of the testing dataset helped in choosing the best hyperparameters for each ML model. Different hyperparameters were tried near the selected initial settings and the settings which showed better performance of ML models during the validation of the training dataset were fixed.

2.4. Machine Learning Models

Different ML models, which include ANN, k-nearest neighbors (kNN), SVM, RF, CatBoost, and AdaBoost, were selected to predict the length of hydraulic jumps for low inflow Froude number. These models were selected considering their effective performance reported in the literature, and their ability to capture nonlinear and complex patterns in hydraulic engineering applications, and their widespread adoption in hydraulic modeling and regression tasks. The structure of these models, their functioning, and the hyperparameter tuning of these models for this study are outlined in this section.

2.4.1. Artificial neural networks

This is an extensively used ML model for regression and classification. The basic structure of this model is based on the assembly of the human brain having interconnected neurons. The model involves three distinct layers: an input layer, which in our study was comprised of the selected input parameters; a hidden layer, for which the number of hidden layers and of nodes referred to as neurons is decided as a part of hyperparameter tuning; and an output layer, which is comprised of the output features. Each node or neuron is interrelated through weighted connections and biases.

The model learns through the backpropagation technique, in which the weight assigned to each connection is adjusted to minimize the error. The L-BFGS-B (limited-memory Broyden–Fletcher–Goldfarb–Shanno with box constraints) solver was used as the optimization algorithm [29]. Furthermore, in neural network-based models, an activation function is employed whose purpose is to create nonlinearity, which increases the learning ability of the model. Among the various available activation functions, the rectified linear unit (ReLU) was applied to the model, which helped in fast learning by eliminating negative values [30]. The maximum number of iterations during the model training was set to 100.

2.4.2. k-Nearest Neighbors

The kNN model is also a popular ML tool. The core working principle behind the model is that the data points that have closely related features may yield similar results [31]. The model works by identifying k number of nearest points, which are termed as neighbors in the feature space for a given input. In regression tasks, the kNN model calculates the predicted result by averaging the numerical values of the k nearest data points, also termed as k nearest neighbors. The choice of k is of major significance, as a very low number of k may lead to overfitting of the model, whereas a large number of k values for a particular dataset may lead to underfitting of the model. In this study, considering the size of the dataset, the k value was set to 6, and the Euclidean method was used to calculate the distance between the neighbors.

2.4.3. Support vector machines

The SVM model works by determining a hyperplane that effectively separates the data points of one class from another. The separation between the data points and the hyperplane is achieved by maximizing the margin between the hyperplane and data points, where this margin is termed the support vector [32]. SVM applies kernel functions to facilitate the separation of the hyperplane, map data points in a high-dimensional space, and model nonlinear relationships [33]. Where commonly used kernel functions include linear, polynomial, and radial basis functions, in the present study, the polynomial kernel function was used due to its stability to capture nonlinear patterns in the hydraulic dataset. The regularization parameter C was set to 1, and the ϵ -insensitive loss parameter was set to 2.

2.4.4. Random forest

Random forest (RF) is an ensemble-based machine learning model that constructs several decision trees and combines their results to enhance predictive accuracy and reduce the risk of overfitting [34,35]. Ensemble learning is a methodology in ML models in which several models are trained and combined to improve the predictions of individual models. RF applies the bagging (bootstrap aggregating) technique, in which various versions of the model are trained on random subsets of the training dataset, and the prediction in the case of regression tasks is performed by averaging the outputs.

In this study, the RF model was configured with 12 trees in an ensemble. This number was selected considering that a small number of well-tuned trees can often achieve sufficient accuracy without a high computational time. The number of attributes

considered at each split (i.e., mtry parameter) was set to five. These hyperparameters were determined by examining the results of the validation of the training dataset to ensure an optimal trade-off between the model complexity and accuracy of the results.

2.4.5. Gradient Boosting

Gradient boosting is also a decision tree-based ML model that works by the boosting technique, where multiple decision trees are built sequentially. In gradient boosting, the decision trees are developed iteratively, with each new tree trained to correct the residual errors from the previous iterations. This is distinct from RF, where trees are built independently and in parallel.

Gradient Boosting is implemented by applying different forms, which include XGBoost, LightGBM, and CatBoost. In this study, the CatBoost algorithm was selected because of its superior handling of categorical variables and robust generalization capabilities, which are particularly beneficial for regression tasks involving structured data [36].

For model training, the CatBoost algorithm was configured with the following hyperparameters: the number of trees (iterations) was set to 100, the learning rate was set to 0.3, the maximum depth of individual trees was set to 6, and all available features were considered at each split (i.e., the feature fraction was set to 1). Studies conducted by [37], [38] were helpful in selecting these settings.

2.4.6. Adaptive Boosting

Adaptive Boosting (AdaBoost), originally proposed by Freund and Schapire [39], works by applying boosting techniques in which sequential models are built. But unlike gradient boosting, which

fits each new model to the residuals of the previous model and minimizes a specified loss function, the AdaBoost model emphasizes reweighting of misclassified samples. AdaBoost applies an exponential loss function, which also makes it sensitive to noisy data.

In the present study, AdaBoost was applied using 20 estimators (trees), a learning rate of 1.0, and a linear loss function for regression, balancing the model simplicity and predictive accuracy for the hydraulic dataset.

2.5. Performance evaluation of models

The predictive performance of the ML models during the training stage was assessed using a 20-fold cross-validation methodology [21], [40], in which the dataset was randomly divided into 20 subsets of equal size. For each of the 20 iterations, one subset was set as the testing dataset and the remaining 19 as training subsets, making sure that each data point is used once for validation and multiple times for training the ML model. After completing iterations, the performance metrics were averaged. The performance metrics included the coefficient of determination (R^2), mean absolute error (MAE), mean absolute percentage error (MAPE), mean square error (MSE), and root mean square error (RMSE). After training and validating the training dataset, the developed models were tested to predict L on two separate datasets applying the same performance metrics. Scatter plots and normal distributions of the errors between the measured and predicted values were plotted to visualize the difference between predicted and measured L. Figure 3 summarizes the process of data collection, chosen ML models, and model evaluations methods.

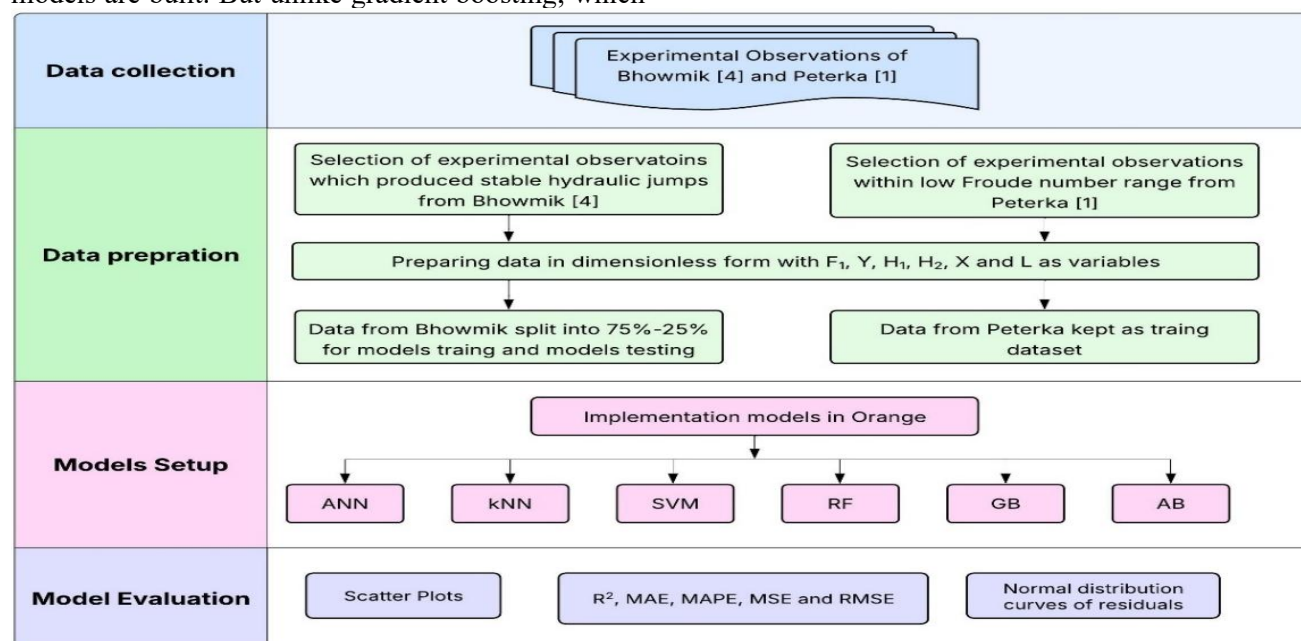


Fig. 3. Flow chart summarizing data preparation, chosen ML models and their performance evaluation.

3. Results and discussion

The present study applies several ML models to estimate the relative length of a hydraulic jump (L) in a stilling basin for low inflow Froude number. Input parameters include; the inflow Froude number (F_1), sequent depth ratio (Y), relative height (H_1) of baffle blocks, relative sill height (H_2), and relative distance between the baffle blocks and the sill (X). Six different ML models were used, i.e., ANN, kNN, SVM, RF, CatBoost, and AdaBoost. The performance of ML models was estimated using various statistical indices. The trained and validated models were tested on two separate datasets. The first dataset was separated from the same experimental campaign on which the models were trained; this dataset is referred to as “testing dataset-1”. Furthermore, the models were tested on a dataset prepared from the observations of Peterka [1], thereby testing models on observations from a separate campaign. In further discussion, the second dataset is referred to as “testing dataset-2”.

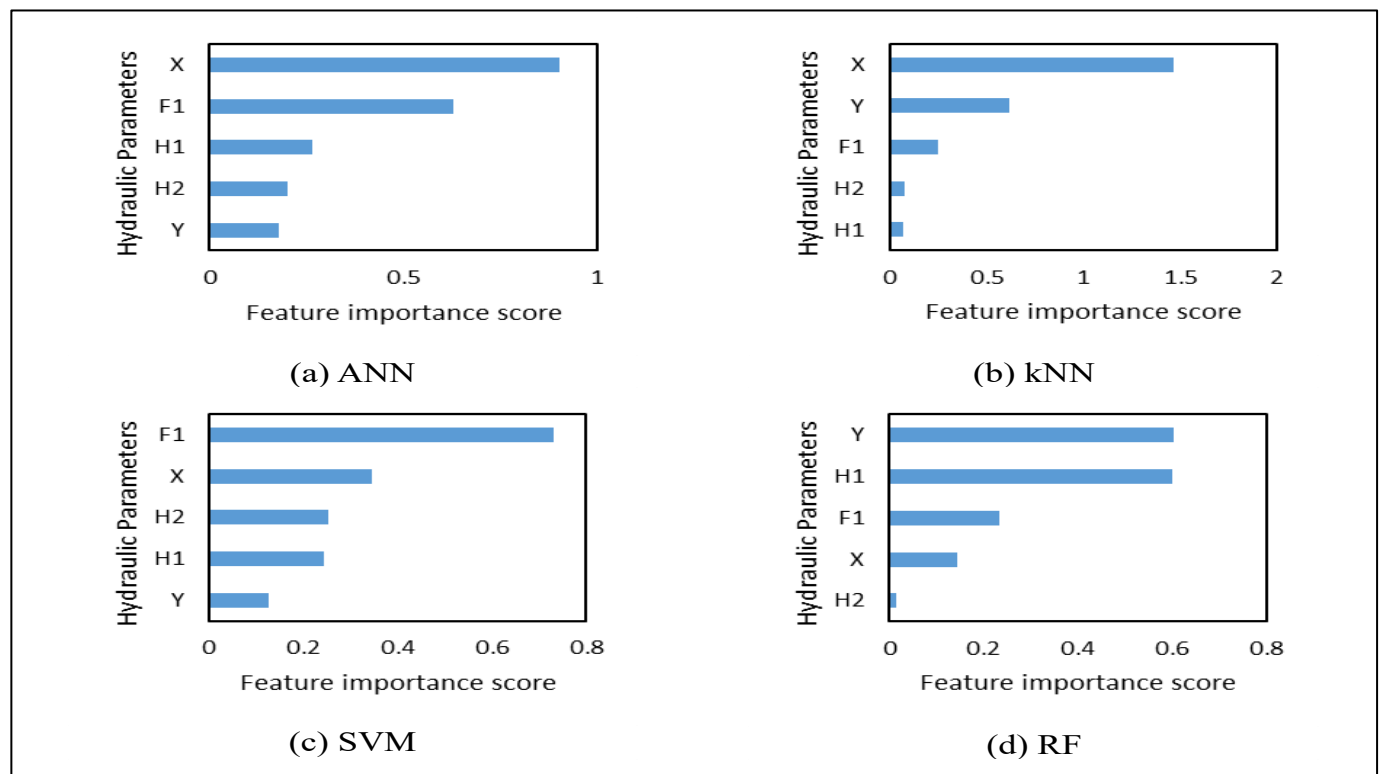
3.1. Feature importance scores of the study parameters

Feature importance scores suggest a deeper understanding of ML models by measuring contribution of each input parameter in making the necessary predictions [41]. This study utilized the permutation-based method for calculating feature importance scores, in which values of each feature are randomly shuffled across a dataset and then the resulting decline in the model’s performance is measured. R^2 was used as a performance metric, and five permutations were performed for each feature to reduce the effect of randomness. Features that, when

permuted, caused a significant drop in the R^2 score were considered more important, indicating the model’s reliance on those features and the output.

The feature importance score value shown in figure 4 represents the relative importance of each input hydraulic parameter in contributing to performance of each ML model. A relatively high score of a parameter indicates that the parameter has stronger influence on the model’s predictive performance. Conversely, lower score of any parameter indicates that this feature has relatively less effect on the model’s predictive performance. Considering the present study the feature importance scores provide insights into the relevance of each input hydraulic parameter being both hydraulically important and computationally effective for model training and prediction.

The results of feature importance revealed different results for each model (Figure 4). In the case of ANN, X , F_1 , and H_1 appeared as more significant features affecting the model prediction. In the case of SVM, F_1 , X , and H_2 were more important than H_1 and Y . For the kNN-based models, X , Y , and F_1 showed significance, and the other two features, H_1 and H_2 , had significantly lower feature importance scores. Similarities can be observed among tree-based models, specifically RF, CatBoost, and AdaBoost, as Y scores the highest feature importance for AdaBoost and RF, and H_1 is the second most important feature in all these ensemble models. Considering the importance scores of all ML models, F_1 and Y appear to be the most significant, as these two parameters consistently remain either the top scorer or the second-highest score across all models.



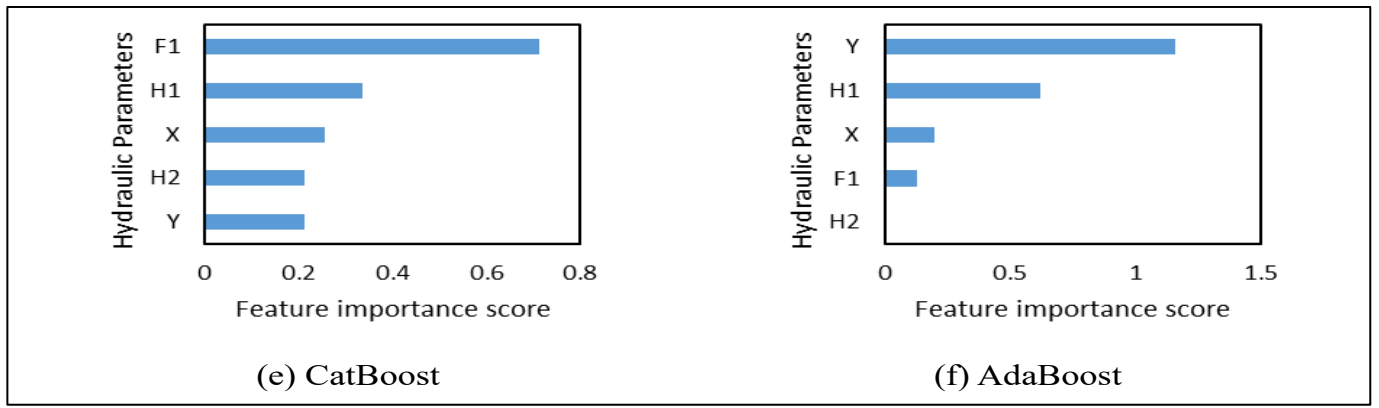


Fig. 4. Feature importance scores for the ML models used.

Furthermore, the Pearson correlation coefficient was calculated for all the study parameters used in training of ML models. The Pearson correlation coefficient evaluates the strength of the linear relationship among the parameters under a study. The coefficients value range between -1 and 1, indicating a strong positive and negative correlation respectively, and 0 for no correlation. Table 2 shows Pearson correlation matrix for prepared for the parameters considered in this study. The relative length of hydraulic jump (L) has shown a positive Pearson correlation coefficient with both F_1 and Y , indicating that an increase in these parameters increases the length of hydraulic jump. Whereas H_1 , H_2 , and X exhibited negative correlations with hydraulic jump length, indicating that higher values of these features tended to reduce the jump length. Based on the correlation coefficients, the parameters affecting hydraulic jump length can be ranked from most to least influential as follows: Y (0.48), F_1 (0.44), H_1 (-0.37), X (-0.27), and H_2 (-0.22).

Table 2.

Pearson Correlation Matrix prepared using training data.

	F_1	Y	H_1	H_2	X	L
F_1	1					
Y	0.98	1				
H_1	0.28	0.18	1			
H_2	0.33	0.24	0.86	1		
X	0.37	0.27	0.91	0.96	1	
L	0.44	0.48	-0.37	-0.22	-0.27	1

3.2 Performance evaluation

The validation of the training dataset of the ML models is presented in Figure 5 as scatter plots. The dotted lines in all these graphs mark the $\pm 15\%$ error lines. The solid line marks perfect agreement, where the predicted value equals the experimentally measured value in the input dataset. The scatter plots in Figure 5 show that all models exhibit similar trends. Except for a few instances for each ML model, all data points lie within the $\pm 15\%$ error lines.

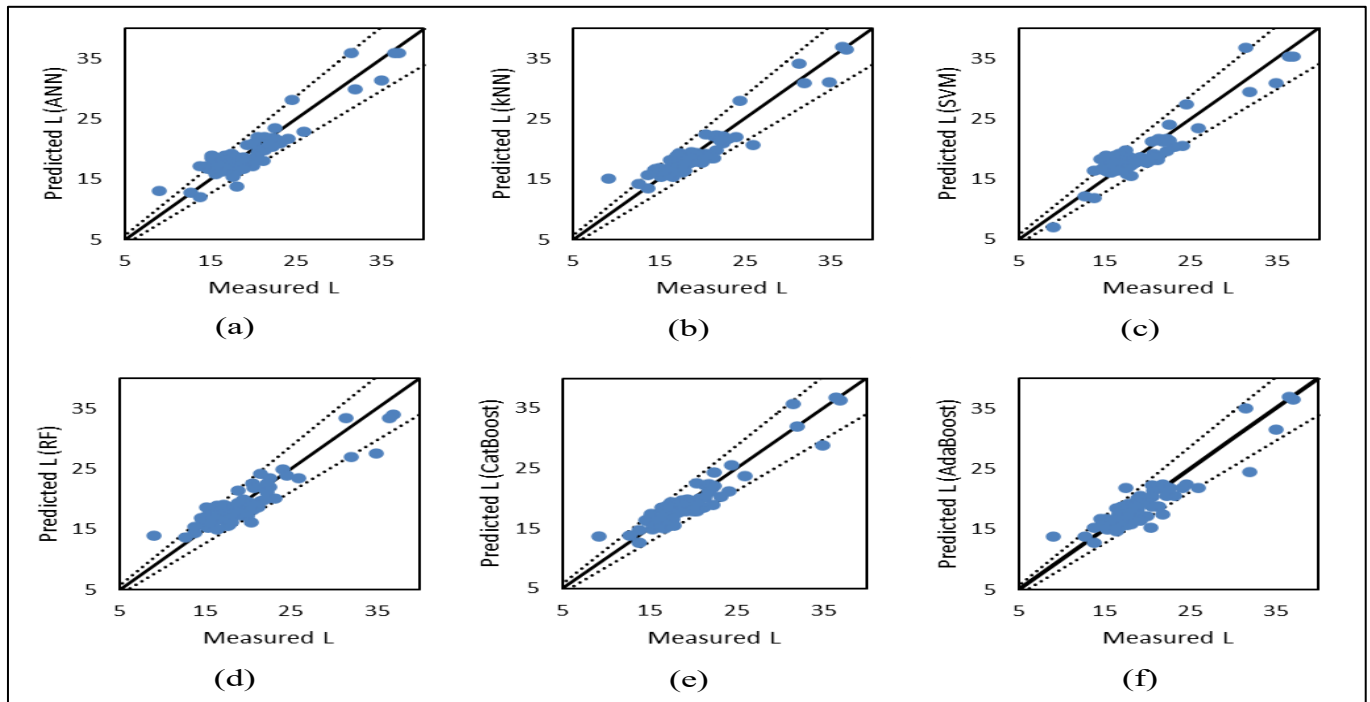


Fig. 5. Measured and predicted length of hydraulic jumps during validation of training dataset using (a) ANN, (b) kNN, (c) SVM, (d) RF, (e) CatBoost, and (f) AdaBoost.

The developed models were then applied to predict L using the features in the testing dataset-1 and then compared with the measured values. Following the same formatting as in Figure 5, Figure 6 shows the scatter plots for testing dataset-1. The predictions made by all ML models on the testing dataset show a similar behavior as that during the validation of the training dataset, as shown in Figure 6. Most of the predictions, except for some rare instances in the graphs in Figure 6, lie within the $\pm 15\%$ error lines for all the ML models.

Regarding the predictions which are out of the marked error range in Figure 6, kind of similar trend is observed in all ML models except ANN. In case of ANN all the measured and predicted L lie within the marked error range. Whereas, for rest all the ML models 1 or 2 datapoints are out of the marked error range. Furthermore, it can also be seen that in all these cases over Prediction of L is happening for the same instances, which is highlighted in Figure 6 with a cross on the markers showing these points. The same error in all these models may be attributed to some minor uncertainties in experimental observations which is common in laboratory based hydraulic studies.

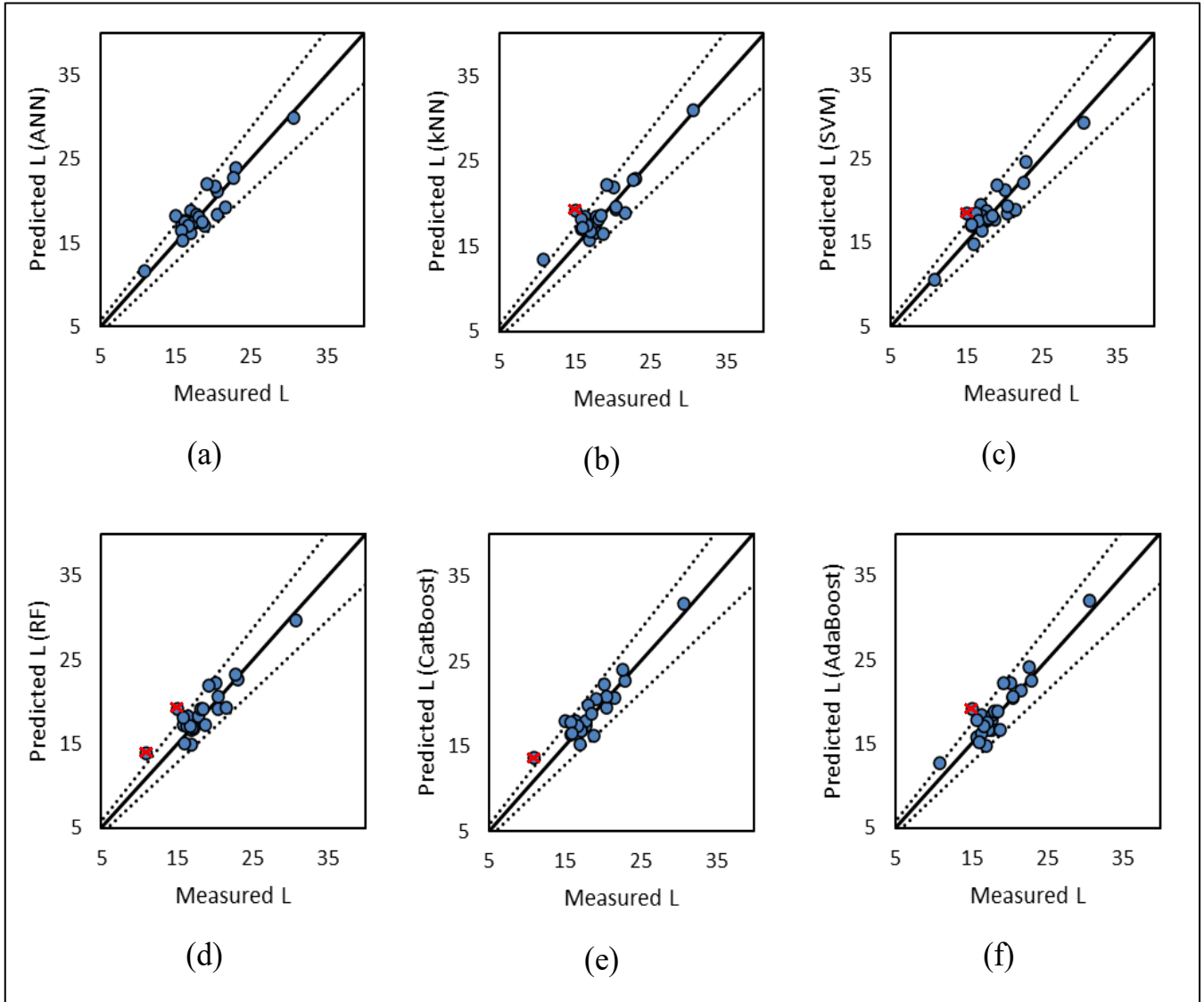


Fig. 6. Measured and predicted length of hydraulic jumps during testing with testing dataset-1 using (a) ANN, (b) kNN, (c) SVM, (d) RF, (e) CatBoost, and (f) AdaBoost.

Furthermore, experimental observations at Froude numbers of interest in the current study were collected from Peterka [1] to test the developed models again on an unseen dataset, and this time on a dataset collected from an entirely different experimental campaign. On the same formatting as in Figures 5 and 6, the scatter plots for the second testing phase are shown in Figure 7. For this testing dataset, relatively more reliable

performance is demonstrated by ANN as compared to other ML models considered in this study. In the case of kNN, RF, and CatBoost, most of the data points are within the $\pm 15\%$ error lines. The AdaBoost-based model shows more deviations from the line of perfect agreement. And the SVM-based model predictions mostly fall around the -15% error line.

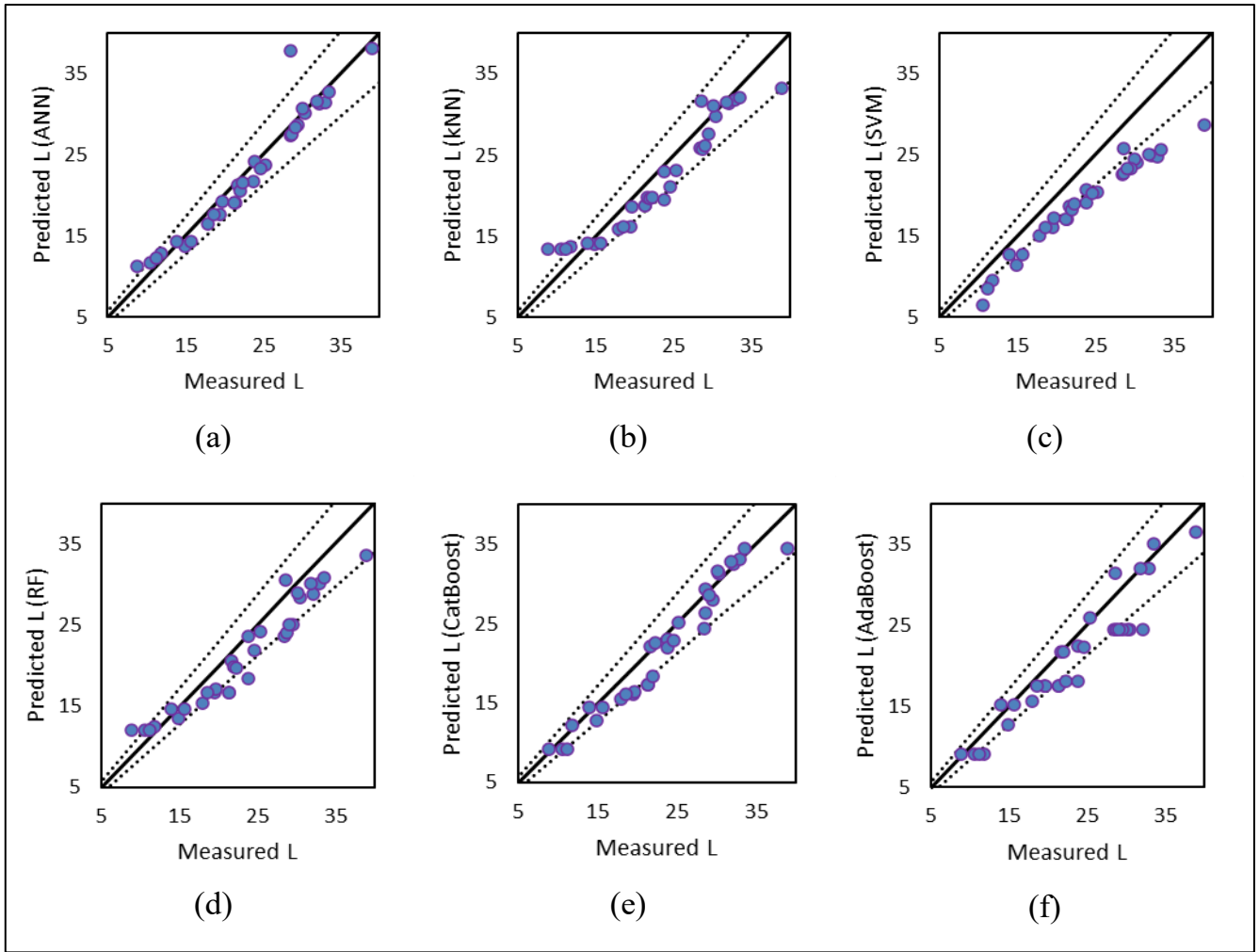


Fig. 7. Measured and predicted length of hydraulic jumps during testing with testing dataset-2 using (a) ANN, (b) kNN, (c) SVM, (d) RF, (e) CatBoost, and (f) AdaBoost.

Furthermore, the performance of the ML models developed to predict L was assessed using various statistical indices: R^2 , MAE, MAPE, MSE, and RMSE. Tables 3, 4, and 5 present the results of these indices for the model validation phase, testing phase, and external testing using testing dataset-2 respectively. These results are also represented in Figure 8.

During the validation of the training dataset, the CatBoost model outperformed all other models, achieving the highest R^2 value (0.89) and the lowest values for MSE (2.70), RMSE (1.64), MAE (1.23), and MAPE (0.07). The kNN model demonstrated the second-best performance, followed by the ANN, SVM, and RF. AdaBoost had the lowest R^2 and the highest error metrics.

Testing of the ML models with the testing dataset-1 showed similar trends to those observed during the model validation stage. Both the CatBoost and ANN models achieved the highest R^2 value (0.85) and low values for MAE, MAPE, MSE, and RMSE, thereby indicating a strong predictive performance. The

remaining ML models followed the same ranking as in the validation phase, however with contrasted performance metrics.

The ML models were again tested using dataset from Peterka [1], which includes hydraulic jumps on a horizontal plane bed without appurtenances. In this paper testing this dataset is referred to as testing data-2. Table 5 shows that the ANN model demonstrated the highest R^2 value (0.93), followed closely by the CatBoost model ($R^2 = 0.92$). Both models also reported comparable MAE (1.39 and 1.72, respectively) and MAPE (0.07 and 0.08, respectively), consistent with values observed during validation and initial testing. While the MAE and MAPE remained stable, the MSE and RMSE values increased slightly in this third phase, reflecting higher variability in the external dataset. However, ANN and CatBoost again achieved the lowest MSE and RMSE among all models. Other models showed acceptable performance, except for SVM, which exhibited a considerably lower R^2 value (0.55) and higher error metrics.

Table 3

Evaluation of the ML models during validation of the training dataset.

	R ²	MAE	MAPE	MSE	RMSE
ANN	0.86	1.43	0.08	3.26	1.80
kNN	0.88	1.26	0.07	2.77	1.66
SVM	0.86	1.51	0.08	3.46	1.86
RF	0.84	1.43	0.07	3.75	1.94
CatBoost	0.89	1.23	0.07	2.70	1.64
AdaBoost	0.84	1.46	0.08	3.92	1.98

Table 4

Evaluation of the ML models during testing of the training dataset-1.

	R ²	MAE	MAPE	MSE	RMSE
ANN	0.85	1.06	0.06	1.82	1.35
kNN	0.76	1.4	0.08	2.97	1.72
SVM	0.82	1.24	0.07	2.26	1.5
RF	0.79	1.24	0.07	2.6	1.61
CatBoost	0.85	1.07	0.06	1.85	1.36

AdaBoost	0.81	1.1	0.06	2.28	1.51
----------	------	-----	------	------	------

Table 5

Evaluation of the ML models during testing of the training dataset-2.

	R ²	MAE	MAPE	MSE	RMSE
ANN	0.93	1.39	0.07	4.24	2.06
kNN	0.97	2.07	0.11	5.89	2.43
SVM	0.55	4.66	0.21	25.73	5.07
RF	0.84	2.6	0.12	8.88	2.98
CatBoost	0.92	1.72	0.08	4.67	2.17
AdaBoost	0.81	2.67	0.12	10.82	3.29

A visual comparison of the variation in performance metrics for each model is shown in Figure 8. The CatBoost and ANN models consistently show superior performance, maintaining high R² values and low error metrics across all datasets. Whereas the SVM model shows low relative performance, with high error values and low R² values. Overall, Figure 8 shows that CatBoost and ANN remain most reliable under varying conditions, while SVM and AdaBoost show limitations in terms of generalization and accuracy.

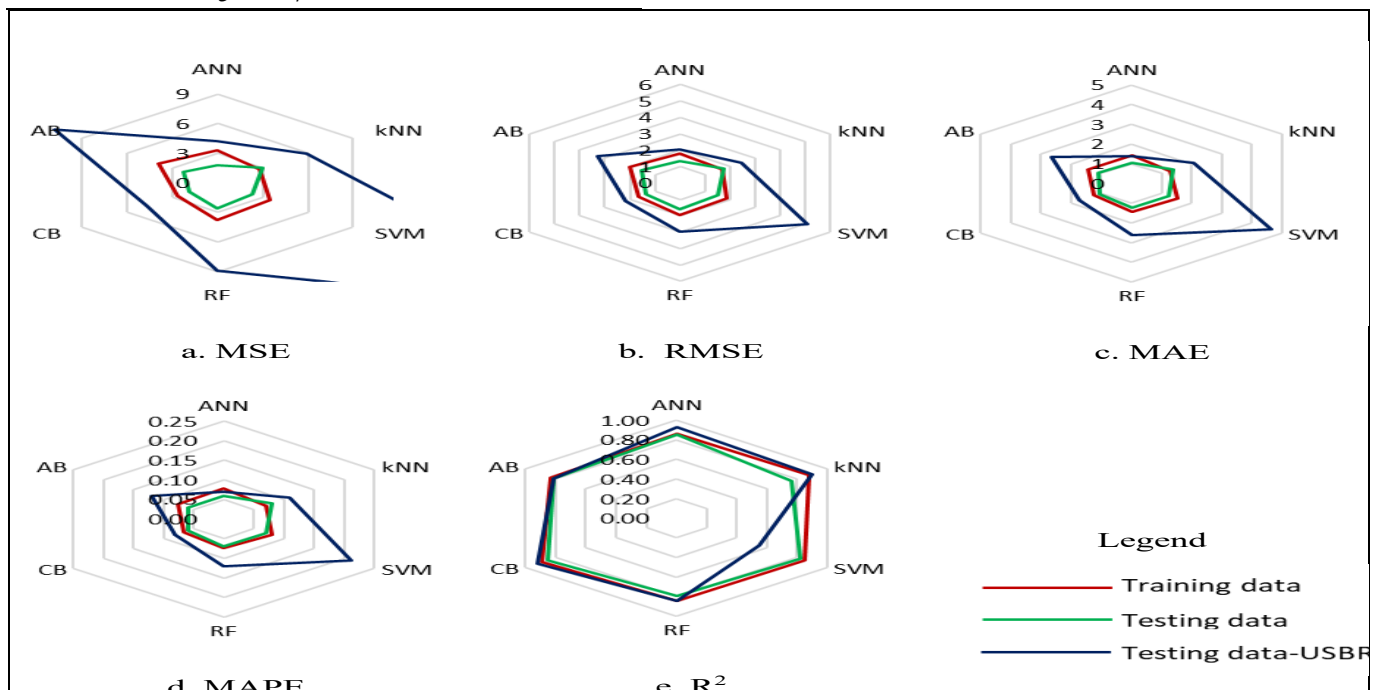
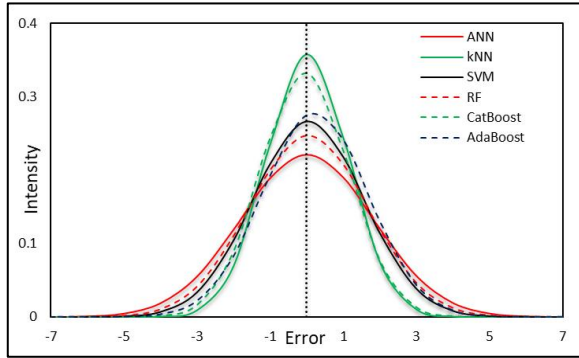


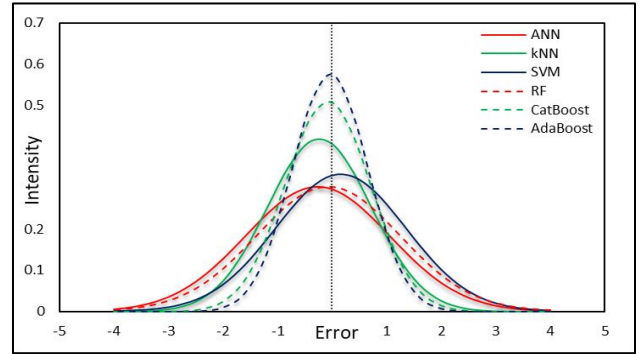
Fig. 8. Performance comparison of the ML models. (a) MSE, (b) RMSE, (c) MAE, (d) MAPE, and (e) R². CatBoost and AdaBoost are mentioned as CB and AB, respectively, in this figure only.

The normal distribution curve is a helpful tool for visualizing the spread of prediction errors. When the peak of the curve is centered on the zero-error line, it indicates balanced model performance, where there is an equal probability of over and under prediction. Whereas a shift of the peak to either side of this line suggests a bias either toward overprediction or underprediction. The normal distribution of prediction errors during the validation phase for all models is shown in Figure 9 (a). In this figure, the dotted vertical

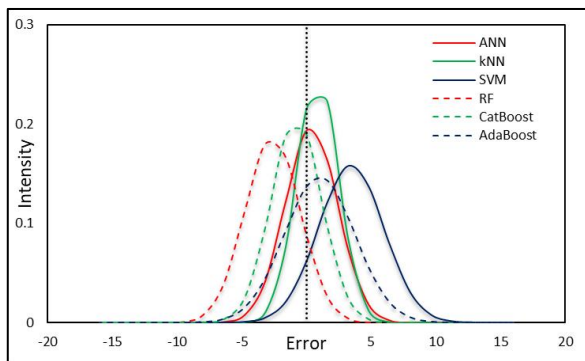
line represents the zero-error reference. The horizontal axis is the residual values (i.e., the differences between the predicted and actual values), and the vertical axis is the probability of occurrence. In Figure 9 (a), almost all the ML models show symmetrical error distributions, thereby indicating that all predictions are unbiased. A slight rightward shift in the error curve for the AdaBoost is visible in Figure 9 (a), suggesting a tendency toward underestimation.



(a)



(b)



(c)

Fig. 9: Error distribution curve between measured and predicted lengths of hydraulic jump using different ML models (a) Validation of training dataset, (b) testing of models with testing dataset-1, (c) testing of models with testing dataset-2

Similarly, Figure 9 (b) shows the error distribution curves for the first testing phase. The kNN and SVM models display a slight deviation from the center in Figure 9 (b). CatBoost and AdaBoost models show the most balanced performance, with their distribution peaks closely aligned with the zero-error line. Likewise, Figure 9 (c) shows the error distribution curves when the models are evaluated using the testing dataset-2 obtained from Peterka [1]. Among the models tested, ANN and CatBoost show the most balanced performance, with their peaks located near the centerline.

4. Conclusions

This study evaluated several ML models in predicting the length of hydraulic jump (which was used in a dimensionless form as relative length of the

jump as L) in stilling basins for inflow low Froude numbers. The ML models were trained and tested using experimental data, which was applied in a dimensionless form for better generalization and wider applicability.

An intercomparison of the tested ML models using statistical performance metrics revealed that the ANN and CatBoost models showed better performance than the others during validation and testing, achieving high R^2 values (0.85 to 0.93) and low error metrics, indicating a robust generalized performance. In the case of kNN model similar performance as compared to ANN and CatBoost is observed, with slightly lower R^2 and error metrics. RF and AdaBoost models, however, showed middling performance across validation and testing phases. The SVM model

showed poor performance when tested with the testing dataset-2, with the lowest R^2 (0.55).

Normal distribution of errors for the ML models indicated that all models exhibited a balanced distribution of errors during validation of the training dataset and testing using testing dataset-1. However, during testing of testing dataset-2, ANN and CatBoost showed balanced performance, demonstrating robustness across various hydraulic conditions.

Feature importance scores were calculated for the input parameters used in the study to determine the most influential parameters in making predictions. The results revealed that F_1 and Y were significant features for all models. A Pearson correlation matrix was prepared, which supported the results of the feature importance scores, with F_1 and Y showing higher coefficient values of 0.44 and 0.48, respectively, with L . H_2 remained the low scorer for feature importance scores across all ML models and had the smallest correlation coefficient (-0.22).

The application of ML models for estimating the length of hydraulic jumps at low inflow Froude numbers is explored in this paper. Furthermore, for any ML model, its general applicability to any unseen data is crucial, which was also addressed in this study by testing the ML models using experimental observations from two entirely different sources. Further studies can be done on extending the applicability of these models to provide optimal design solutions. As the present study was conducted on Orange in which the models are configured using preset widgets with relatively less options of advanced hyperparameter tuning and in-depth model customization as compared to programming-based environments such as Python or R, future studies may also benefit from using programming-based environments.

References

- [1] A. J. Peterka, "Hydraulic Design of Stilling Basins and Energy Dissipators," Denver, 1984.
- [2] R. Steinke et al., "Low Froude Number Stilling Basins-Hydrodynamic Characterization," *Journal of Hydraulic Engineering*, vol. 147, no. 4, Apr. 2021, doi: 10.1061/(asce)hy.1943-7900.0001867.
- [3] R. L. George, "Low Froude Number Stilling Basin Design," 1978, United States Department of the Interior, Bureau of Reclamation, Denver, Colorado.
- [4] N. G. Bhowmik, "Hydraulic Jump Type Stilling Basins for Froude Number 2.5 to 4.5," Urbana, 1971.
- [5] N. Hassanpour, A. H. Dalir, D. Farsadizadeh, and C. Gualtieri, "An experimental study of hydraulic jump in a gradually expanding rectangular stilling basin with roughened bed," *Water (Switzerland)*, vol. 9, no. 12, Dec. 2017, doi: 10.3390/w9120945.
- [6] M. W. Zaffar and I. Hassan, "Hydraulic investigation of stilling basins of the barrage before and after remodelling using FLOW-3D," *Water Supply*, vol. 23, no. 2, pp. 796–820, Feb. 2023, doi: 10.2166/ws.2023.032.
- [7] Z. Xu, Z. Liu, and Y. Peng, "Performance comparison of prediction of hydraulic jump length under Multiple neural network models," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3430075.
- [8] K. C. Shikhar, K. P. Bhattarai, T. De Shan, S. Mishra, I. Joshi, and A. K. Singh, "Comprehensive performance analysis of training functions in flow prediction mode using artificial neural network," *Water SA*, vol. 50, no. 2, pp. 190–200, Apr. 2024, doi: 10.17159/wsa/2024.v50.i2.4099.
- [9] P. Samui, S. N. Yesilyurt, H. Y. Dalkilic, Z. M. Yaseen, S. S. Roy, and S. Kumar, "Comparison of different optimized machine learning algorithms for daily river flow forecasting," *Earth Sci Inform*, vol. 16, no. 1, pp. 533–548, Mar. 2023, doi: 10.1007/s12145-022-00896-3.
- [10] Y. Uzun and S. Y. Kumcu, "Estimation of Equilibrium Scour Depth Around Abutments Using Artificial Intelligence," *Water (Basel)*, vol. 17, no. 7, p. 1010, Mar. 2025, doi: 10.3390/w17071010.
- [11] M. Nda, M. S. Adnan, M. A. B. M. Yusoff, and R. M. Nda, "An Overview of Machine Learning Techniques for Sediment Prediction †," *Engineering Proceedings*, vol. 56, no. 1, 2023, doi: 10.3390/ASEC2023-16599.
- [12] S. Q. Nguyen, L. C. Nguyen, T. Ngo-Duc, and S. Ouillon, "Applying a machine learning-based method for the prediction of suspended sediment concentration in the Red river basin," *Model Earth Syst Environ*, vol. 10, no. 2, pp. 2675–2692, Apr. 2024, doi: 10.1007/s40808-023-01915-y.
- [13] S. N. Mousavi, D. Farsadizadeh, F. Salmasi, and A. Hosseinzadeh Dalir, "Evaluation of

- pressure fluctuations coefficient along the USBR Type II stilling basin using experimental results and AI models,” *ISH Journal of Hydraulic Engineering*, vol. 28, no. S1, pp. 207–214, 2022, doi: 10.1080/09715010.2020.1743208.
- [14] P. Sihag, O. F. Dursun, S. S. Sammen, A. Malik, and A. Chauhan, “Prediction of aeration efficiency of Parshall and Modified Venturi flumes: application of soft computing versus regression models,” *Water Supply*, vol. 21, no. 8, pp. 4068–4085, Dec. 2021, doi: 10.2166/ws.2021.161.
- [15] R. Daneshfaraz et al., “Estimating the effect of sand-roughened bed on hydraulic jump characteristics using heuristic models,” *Results in Engineering*, vol. 23, Sep. 2024, doi: 10.1016/j.rineng.2024.102724.
- [16] M. Zarei, O. Bozorg-Haddad, S. Baghban, M. Delpasand, E. Goharian, and H. A. Loáiciga, “Machine-learning algorithms for forecast-informed reservoir operation (FIRO) to reduce flood damages,” *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-03699-6.
- [17] R. Gonzalez Perea, R. Ballesteros, J. F. Ortega, and M. Á. Moreno, “Water and energy demand forecasting in large-scale water distribution networks for irrigation using open data and machine learning algorithms,” *Comput Electron Agric*, vol. 188, Sep. 2021, doi: 10.1016/j.compag.2021.106327.
- [18] K. Roushangar and R. Ghasempour, “Explicit prediction of expanding channels hydraulic jump characteristics using gene expression programming approach,” *Hydrology Research*, vol. 49, no. 3, pp. 815–830, Jun. 2018, doi: 10.2166/nh.2017.262.
- [19] M. F. Sauda, “Prediction of hydraulic jump length downstream of multi-vent regulators using Artificial Neural Networks,” *Ain Shams Engineering Journal*, vol. 7, no. 2, pp. 819–826, Jun. 2016, doi: 10.1016/J.ASEJ.2015.12.005.
- [20] H. V. Ho, “Determination of the surface roller length of hydraulic jumps in horizontal rectangular channels using the machine learning method,” *Stochastic Environmental Research and Risk Assessment*, vol. 38, no. 7, pp. 2539–2562, Jul. 2024, doi: 10.1007/s00477-024-02697-0.
- [21] E. Gul, O. F. Dursun, and A. Mohammadian, “Experimental study and modeling of hydraulic jump for a suddenly expanding stilling basin using different hybrid algorithms,” *Water Supply*, vol. 21, no. 7, pp. 3752–3771, Nov. 2021, doi: 10.2166/ws.2021.139.
- [22] H. Azimi, H. Bonakdari, and I. Ebtehaj, “Gene expression programming-based approach for predicting the roller length of a hydraulic jump on a rough bed,” *ISH Journal of Hydraulic Engineering*, vol. 27, no. S1, pp. 77–87, 2021, doi: 10.1080/09715010.2019.1579058.
- [23] M. Karbasi and H. M. Azamathulla, “GEP to predict characteristics of a hydraulic jump over a rough bed,” *KSCE Journal of Civil Engineering*, vol. 20, no. 7, pp. 3006–3011, Nov. 2016, doi: 10.1007/s12205-016-0821-x.
- [24] M. Heydari, S. Shabanlou, and B. Sanahmadi, “Self-adaptive extreme learning machine-based prediction of roller length of hydraulic jump on rough bed,” *ISH Journal of Hydraulic Engineering*, vol. 28, no. 2, pp. 152–162, 2022, doi: 10.1080/09715010.2020.1852978.
- [25] S. Baharvand, A. Jozaghi, R. Fatahi-Alkouhi, S. Karimzadeh, R. Nasiri, and B. Lashkar-Ara, “Comparative Study on the Machine Learning and Regression-Based Approaches to Predict the Hydraulic Jump Sequent Depth Ratio,” *Iranian Journal of Science and Technology - Transactions of Civil Engineering*, vol. 45, no. 4, pp. 2719–2732, Dec. 2021, doi: 10.1007/s40996-020-00526-2.
- [26] J. Demšar et al., “Orange: Data Mining Toolbox in Python Tomaž Curk Matija Polajnar Laž Zagar,” 2013.
- [27] J. Saputra, C. Lawrencya, J. M. Saini, and S. Suharjito, “Hyperparameter optimization for cardiovascular disease data-driven prognostic system,” *Vis Comput Ind Biomed Art*, vol. 6, no. 1, Dec. 2023, doi: 10.1186/s42492-023-00143-6.
- [28] A. M. M. Abdalmajeed, M. Mahmoud, A. E. R. A. El-Fikky, H. A. Fayed, and M. H. Aly, “Improved indoor visible light positioning system using machine learning,” *Opt Quantum Electron*, vol. 55, no. 3, Mar. 2023, doi: 10.1007/s11082-022-04482-1.
- [29] M. S. Berkani, S. Giurgea, C. Espanet, J. L. Coulomb, and C. Kieffer, “Study on optimal

- design based on direct coupling between a FEM simulation model and L-BFGS-B algorithm,” *IEEE Trans Magn*, vol. 49, no. 5, pp. 2149–2152, 2013, doi: 10.1109/TMAG.2013.2245871.
- [30] R. Parhi and R. D. Nowak, “The Role of Neural Network Activation Functions,” *IEEE Signal Process Lett*, vol. 27, pp. 1779–1783, 2020, doi: 10.1109/LSP.2020.3027517.
- [31] A. R. Lubis, M. Lubis, and Al-Khowarizmi, “Optimization of distance formula in k-nearest neighbor method,” *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 1, pp. 326–338, Feb. 2020, doi: 10.11591/eei.v9i1.1464.
- [32] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” 1998.
- [33] V. Piccialli and M. Sciandrone, “Nonlinear optimization and support vector machines,” *Ann Oper Res*, vol. 314, no. 1, pp. 15–47, Jul. 2022, doi: 10.1007/s10479-022-04655-x.
- [34] J. C. M. Sánchez, H. G. A. Mesa, A. T. Espinosa, S. R. Castilla, and F. G. Lamont, “Improving wheat yield prediction through variable selection using Support Vector Regression, Random Forest, and Extreme Gradient Boosting,” *Smart Agricultural Technology*, vol. 10, p. 100791, Mar. 2025, doi: 10.1016/J.ATECH.2025.100791.
- [35] L. Breiman, “Random forests,” *Mach Learn*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324/METRICS.
- [36] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, “A comparative analysis of gradient boosting algorithms,” *Artif Intell Rev*, vol. 54, no. 3, pp. 1937–1967, Mar. 2021, doi: 10.1007/s10462-020-09896-5.
- [37] M. K. Elshaarawy, “Metaheuristic-driven CatBoost model for accurate seepage loss prediction in lined canals,” *Multiscale and Multidisciplinary Modeling, Experiments and Design*, vol. 8, no. 5, May 2025, doi: 10.1007/s41939-025-00800-8.
- [38] Y. Huang, Z. He, Y. Qin, Y. Lu, and K. Chen, “Optimizing office building performance in the HSWW region of China using simulation with Hyperopt CatBoost and SPEA2,” *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-85267-w.
- [39] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *J Comput Syst Sci*, vol. 55, no. 1, pp. 119–139, Aug. 1997, doi: 10.1006/JCSS.1997.1504.
- [40] T. T. Wong, “Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation,” *Pattern Recognit*, vol. 48, no. 9, pp. 2839–2846, Sep. 2015, doi: 10.1016/j.patcog.2015.03.009.
- [41] V. A. Huynh-Thu and P. Geurts, “Optimizing model-agnostic random subspace ensembles,” *Mach Learn*, vol. 113, no. 2, pp. 993–1042, Feb. 2024, doi: 10.1007/s10994-023-06427-5.