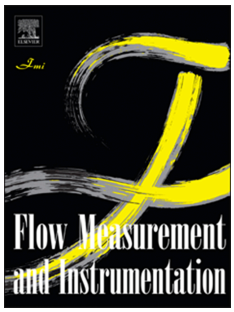


Enhancing Discharge Prediction over Type-A Piano Key Weirs: An Innovative Machine Learning Approach

Weiming Tian, Haytham F. Isleem, Abdelrahman Kamal Hamed, Mohamed Kamel Elshaarawy

PII: S0955-5986(24)00212-7

DOI: https://doi.org/10.1016/j.flowmeasinst.2024.102732

Reference: JFMI 102732

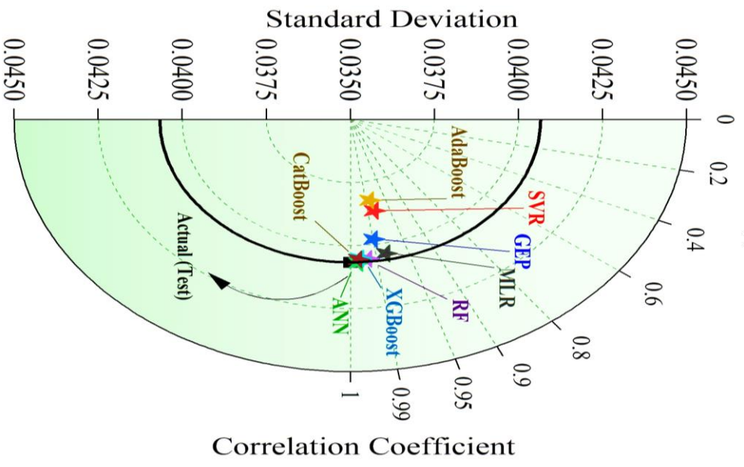
To appear in: *Flow Measurement and Instrumentation*

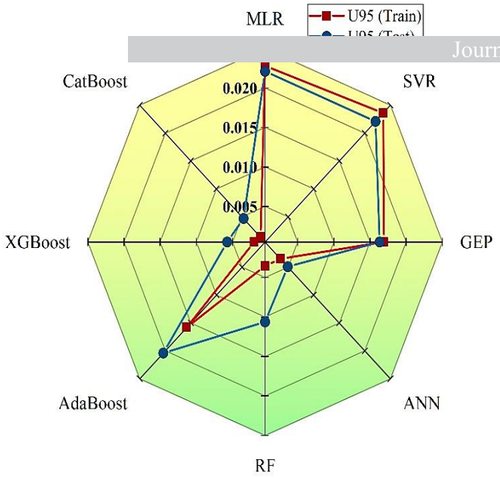
Received Date: 21 November 2023

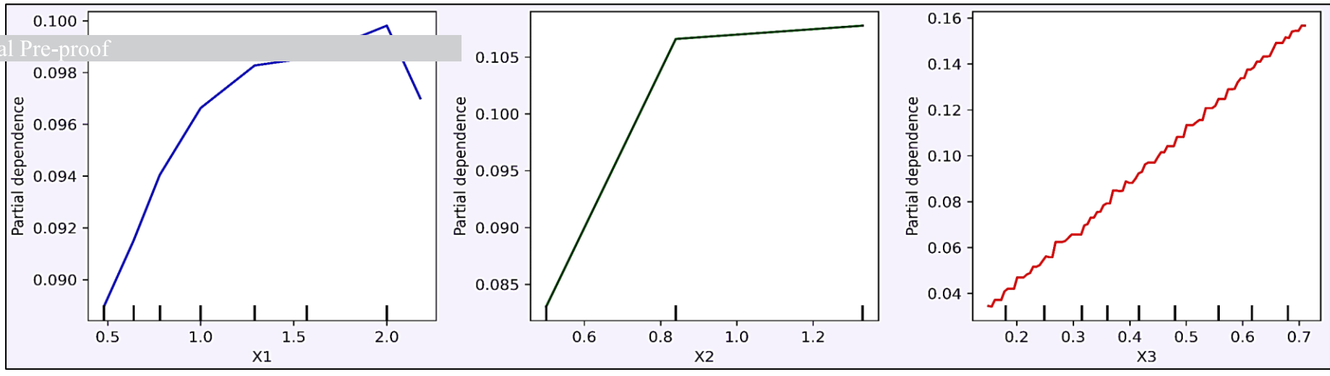
Revised Date: 16 September 2024

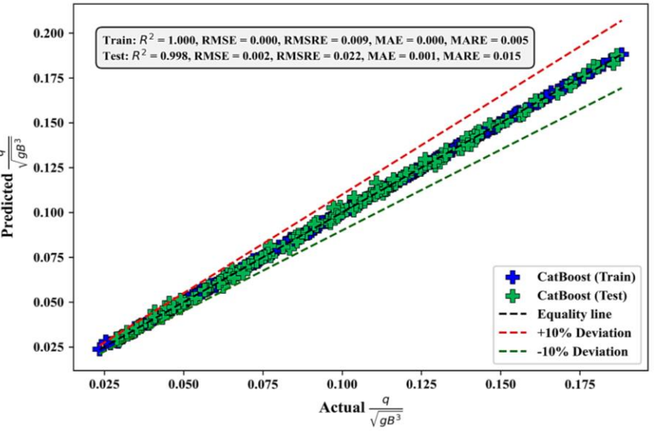
Accepted Date: 30 October 2024

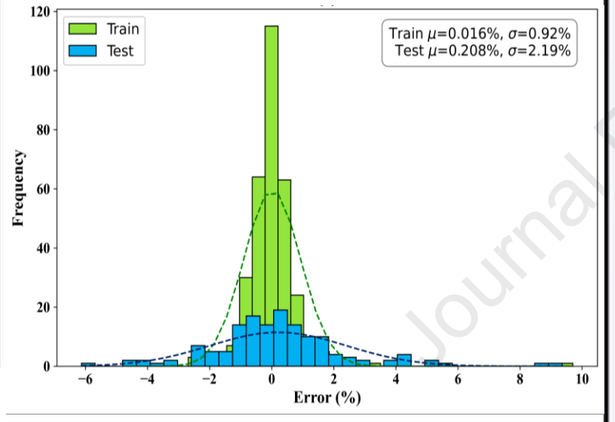
Please cite this article as: W. Tian, H.F. Isleem, A.K. Hamed, M.K. Elshaarawy, Enhancing Discharge Prediction over Type-A Piano Key Weirs: An Innovative Machine Learning Approach, *Flow Measurement and Instrumentation*, https://doi.org/10.1016/j.flowmeasinst.2024.102732.

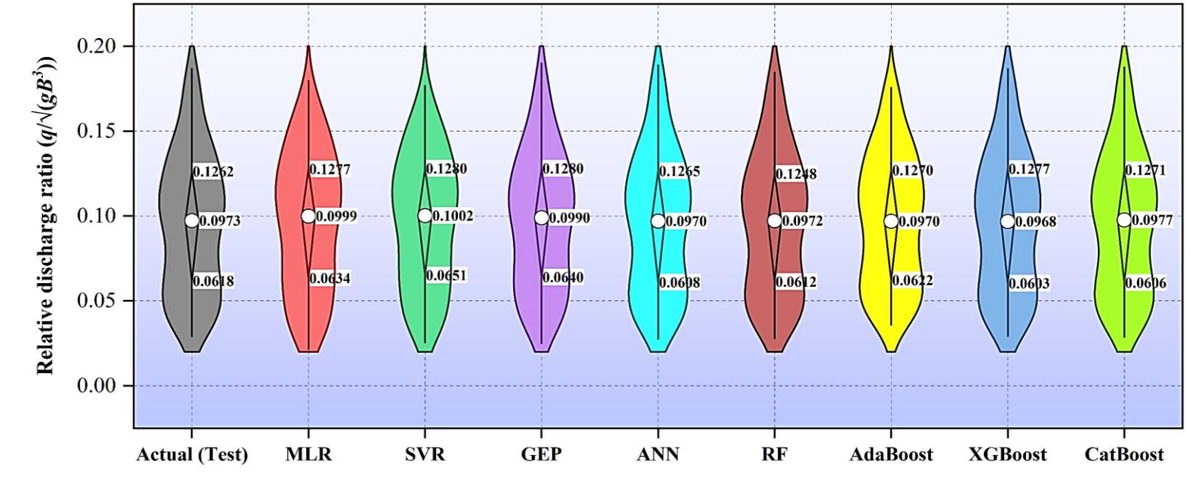


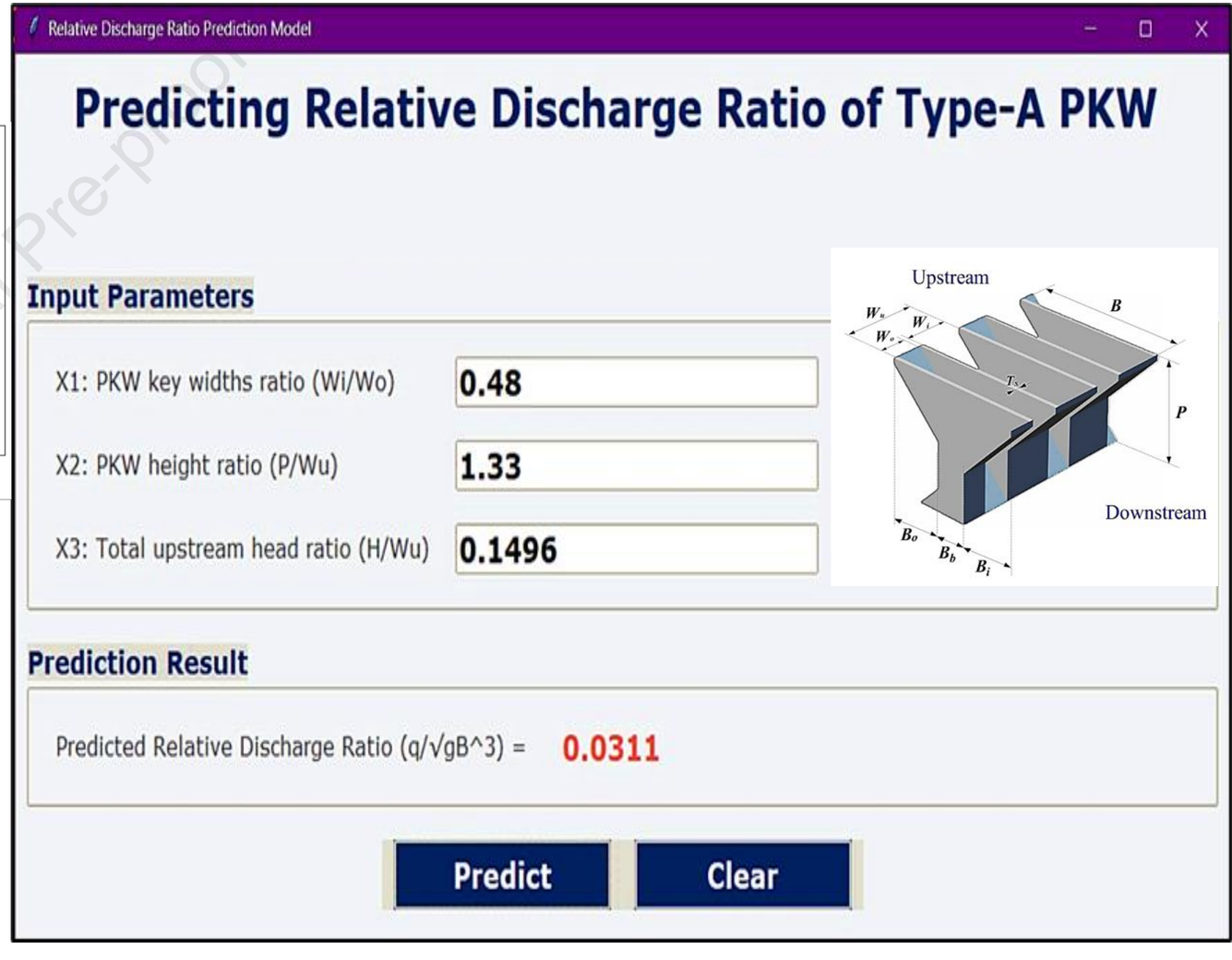












Enhancing Discharge Prediction over Type-A Piano Key Weirs: An Innovative Machine Learning Approach

Weiming Tian[a], Haytham F. Isleem[b], Abdelrahman Kamal Hamed[c] & Mohamed Kamel Elshaarawy[c*]

[a]College of Management, Hunan University of Information Technology, Changsha 410151, Hunan, China; tianweiming@hnuit.edu.cn

[b]Jadara Research Center, Jadara University, Irbid, Jordan; mps565@york.ac.uk

[c]Civil Engineering Department, Faculty of Engineering, Horus University-Egypt, New Damietta 34517, Egypt; akamal@horus.edu.eg; melshaarawy@horus.edu.eg

*Corresponding author

Abstract

Piano key weirs (PKWs) are an increasingly popular hydraulic structure due to their higher discharge capacity than linear weirs. Accurately predicting the discharge of PKWs is essential for appropriate design and operation. This study utilized eight Machine Learning algorithms, including non-ensemble and ensemble models to predict the discharge of type-A PKWs. Multiple-Linear-Regression (MLR), Support-Vector-Machine (SVM), Gene-Expression-Programming (GEP), and Artificial-Neural-Network (ANN) were adopted as non-ensemble models. While the ensemble models comprised Random-Forest (RF), Adaptive-Boosting (AdaBoost), Extreme-Gradient-Boosting (XGBoost), and Categorical-Boosting (CatBoost). A total of 476 experimental datasets were collected from previous research considering three critical dimensionless input parameters: PKW key widths, PKW height, and total upstream head. The models were trained on 70% of the dataset and tested on the remaining 30%. The hyperparameters of the models were optimized using the Bayesian Optimization technique, with 5-fold cross-validation ensuring high performance. Comprehensive analyses, including visual and quantitative methods, were employed to validate model effectiveness. CatBoost model consistently outperformed the other models, achieving the highest Determination-coefficient (R^2 = 0.998) and lowest Root-Mean-Squared-Error (RMSE = 0.002), highlighting its ability to handle complex data patterns and its superior optimization process. XGBoost follows closely behind, showing strong generalization, while ANN and RF perform well, but it's a slight increase in error metrics. The study also incorporated Shapley-Additive-exPlanations (SHAP) and Partial-Dependence-Plot (PDP) analyses, revealing that the total upstream head variable had the most significant impact on the discharge predictions. An interactive Graphical-User-Interface was developed to facilitate practical applications, enabling engineers to predict discharge quickly and economically.

Keywords

Piano Key Weir; Discharge; Machine Learning; Shapley-Additive-exPlanations; Prediction; Bayesian Optimization

1. Introduction

To ensure the safety of dams, it becomes essential to reevaluate the flood discharge and rehabilitate existing dam structures to accommodate the raised reservoir water level [1]. Free flow weirs are frequently used as hydraulic structures for measuring discharge in both newly constructed and existing dam structures, especially in situations involving flood release or open channel applications. Due to the non-linear geometry of PKW, they can maximize weir length and crest profile within a limited channel width, resulting in enhanced discharge capacity compared to

linear weirs.

Recently, a nonlinear weir type called a Piano Key weir (PKW), introduced by Lempérière and Ouamane [2], was widely employed to increase discharge capacity by stretching the crest of the overflow. PKW represents a modification of the labyrinth weir with a smaller base length to be used above the crest of existing or new concrete dams. The general weir head-discharge relationship is used to estimate the discharge capacity. It was expressed by Henderson [3] as:

$$\mathrm{Q} = \frac{2}{3} C_d \, W \sqrt{2g} \, H^{1.5} \quad \rightarrow \quad q = \frac{Q}{W} = \frac{2}{3} C_d \sqrt{2g} \, H^{1.5} \tag{1}$$

Where Q is the flow discharge (m^3/s), q is the relative discharge ($m^3/s/m$), C_d is the discharge coefficient, g is the gravitational acceleration ($m\ s^{-2}$), W is the transverse PKW width (m), and H is the total upstream head (m), which is equal to piezometric head (h) above the PKW crest plus velocity head ($V^2/2g$), where V is the average upstream velocity (m/s).

One of the difficulties related to the design of PKWs is the high number of geometric variables. The geometric parameters that are crucial for the design of PKW (most of which are depicted in Fig. 1a) include the PKW height (P), total crest length (L), inlet and outlet key widths (W_i and W_o, respectively), PKW cycle width (W_u) total PKW width (W), PKW base length (B_b), upstream (US) and downstream (DS) overhang lengths (B_o and B_i, respectively), total lateral crest length (B), number of cycles (N), and PKW wall thickness (T_s). On the other hand, PKWs have been classified into four types according to the presence of overhangs [4]. These types were type-A exhibits symmetrical overhangs in both the US and DS directions, type-B has only US overhangs, type-C has only DS overhangs, and type-D has no overhangs at all, as illustrated in Fig. 1b.

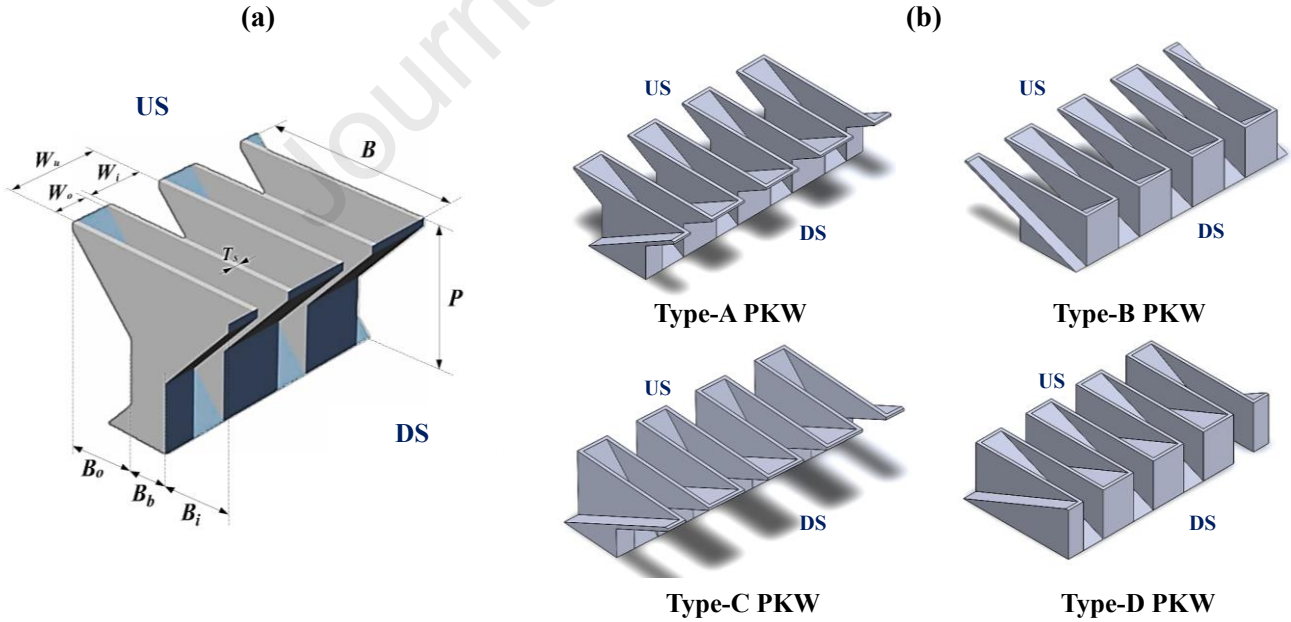


Fig. 1. 3D view of typical PKWs: (a) Geometric parameters, and (b) Different types.

Numerous experimental studies have been performed to evaluate the influence of geometric parameters of different PKW types to enhance the discharge capacity. Lempérière et al. [5] indicated that the (L/W) ratio is the most influential parameter on the discharge capacity of PKWs when ranging from 4 to 5. Leite Ribeiro et al. [6] suggested that PKWs were more discharge efficient when the PKW key widths ratio (W_i/W_o) ratio was 1.60. Anderson and

Tullis [7] explored the influence of five W_i/W_o ratios of (0.67, 0.80, 1.00, 1.25, 1.50) on the discharge capacity of type-A PKW. They found that the discharge efficiency appears to have reached a plateau since PKW discharge with W_i/W_o equal to 1.25 and 1.50 are relatively similar; further increases in W_i/W_o resulted in a decrease in discharge capacity. Anderson and Tullis [8] experimentally explore the effect of PKW overhang lengths on discharge capacity by comparing types A and D PKW. They found that at the same crest length, type-A PKW had higher discharge capacity than type-D PKW by 8.20% on average. Cicero [9] also investigated the impact of overhang lengths by comparing types A, B, and C PKW under free-flow conditions. The results indicated that the type-B PKW exhibited a higher level of efficiency compared to the type-A PKW by a range of 5% to 15% and showed up to a 15% greater efficiency than the type-C PKW.

Leite Ribeiro et al. [6] performed a laboratory investigation considering forty-nine different type-A PKW geometries with a half-circular crest. A general equation for the relationship between head and discharge was derived. Kabiri-Samani and Javaheri [10] conducted an experiment to derive empirical formulas for the discharge coefficient of different PKW types (A, B, C, and D) under free and submerged flow conditions. They developed two empirical equations based on the geometric ratios (L/W, W_i/W_o, B_i/B, B_o/B, and B/P) to predict the C_d-value under both conditions. They discovered submergence occurs when the submergence ratio (H_d/H) $\geq$ 0.60, where H_d is the total downstream head above the weir crest. Machiels et al. [10] examined the impact of different geometric designs of PKW with a flat crest on discharge capacity considering thirty-one geometries. They developed an analytical formula to calculate the PKW discharge as a function of their PKW geometries. Bekheet et al. [11] evaluated the influence of the PKW shapes (rectangular, trapezoidal, and triangular) on the discharge coefficient. They deduced that the trapezoidal shape had a higher discharge efficiency than the rectangular shape by a range of 3.5% to 7.0%, while the triangular shape had the lowest discharge efficiency. Recently, there has been an increasing use of Computational Fluid Dynamics (CFD) models, such as FLOW-3D and ANSYS, to enhance the discharge performance of PKWs [12–15].

Numerous publications have utilized Machine Learning (ML) techniques, such as Artificial Neural Network (ANN) and Gene Expression Programming (GEP). These methods are exceptional tools for forecasting that have been utilized in various civil engineering, hydraulic, and hydrological studies. Gholami et al. [16] introduced a robust GEP model to predict the stable bank profile shape of threshold channels in various hydraulic and geometric scenarios. The proposed GEP model has been shown to accurately predict the distinctive features of bank profiles, surpassing both existing empirical and theoretical models. Bilhan et al. [17] employed ANN, multiple linear regression (MLR), and multiple nonlinear regression (MNLR) to estimate the C_d-value of triangular labyrinth side weir in curved channels. It was found that the ANN model in the testing stage is superior in the estimation of C_d than the MNR and MNLR. Using Extreme Learning Machines (ELM), Azimi et al. [18] perform a sensitivity analysis to determine the parameters that influence the Cd-value in trapezoidal channels with side weirs. Zounemat-Kermani and Mahdavi-Meyma [19] estimated the discharge capacity of type-A PKWs using ten standard and hybrid Artificial Intelligence Data-Driven Models (AI-DDMs), including Multi-layer Perceptron (MLP) Neural Networks and Adaptive Neuro-Fuzzy Inference Systems (ANFIS). They found that the hybrid ANFIS model with a particle swarm optimization (PSO) algorithm (ANFIS-PSO) performed the best in determining the discharge capacity of PKWs. Also, all the AI-DDMs performed better than the empirical relations. Deng et al. [20] Utilizing a novel hybrid boosting ensemble ML model (BO-XGBoost), which integrates the benefits of the boosting model Extreme-

Gradient-Boosting (XGBoost) and Bayesian Optimization (BO), the C_d-value of circular and rectangular side orifices was predicted. An analysis reveals that the BO-XGBoost model surpasses other tree-based ML models. In line with this study, the Group Method of Data Handling (GMDH) is employed Ebtehaj et al. [21] to calculate the discharge coefficient of a rectangular side orifice. Parsaie [22] developed the MLP and Radial Basis Function (RBF) neural network model to predict the C_d-value of side-weir by collecting 477 datasets. The results indicated that the MLP model was more accurate than RBF. Ayaz and Mansoor [23] predicted the C_d-value of an oblique sharp-crested rectangular weir using the ANN model under free and submerged flow. They used experimental data from Borghei et al. [24] and showed a significant reduction in errors.

Haghbin and Sharafati [25] provide a comprehensive review of the application of Soft Computing (SC) models for estimating the C_d-value of different flow control structures such as PKWs. The findings reveal that SC models, particularly ANN and hybrid models incorporating evolutionary algorithms, have shown significant accuracy in predicting the C_d-value across various hydraulic structures. Parameters such as weir length, flow depth, weir height, and Froude number are crucial in these predictions. On the other hand, Singh and Kumar [26,27] predicted the energy dispersion rates of types A and B PKW using the GEP model using several dimensionless inputs such as H/P, L/W, W_i/W_o, and N. A greater percentage of researchers are employing the GEP model due to its simplicity, as explicit equations have been derived. For instance, the study by Salmasi [28] utilized the GEP model and MLR to determine new equations for predicting the C_d-values of the ogee weir. The GEP method was found to be more effective than regression equations. Elshaarawy and Hamed [29] utilized the ANN and GEP models to predict the C_d-value of a triangular side orifice (TSO) based on various geometric and hydraulic parameters. They concluded that both models predicted the C_d accurately, with the ANN model being the most reliable predictor.

Based on the abovementioned, predicting discharge over PKW using machine learning (ML) techniques has not attracted the researchers' attention. Thus, the present research presents an innovative approach by developing various ML models for predicting the discharge of type-A PKW, providing notable accuracy and practical relevance over conventional techniques. Type-A PKWs were chosen due to their widespread usage worldwide owing to their self-balanced configuration. The following are the authors' specific contributions:

1. Introduce a variety of ML models that have not been previously investigated and evaluate their efficacy in predicting the discharge.
2. The predictive accuracy is significantly improved by tuning hyperparameters through Bayesian Optimization (BO) with k-fold cross-validation.
3. Conduct a detailed performance analysis including visual (Scatter plots, Error histogram, Violin boxplot, and Taylor diagram) and quantitative (uncertainty analysis, Akaike Information Criterion (AIC), and regression metrics) to assess the adopted predictive capabilities.
4. Incorporate SHapley Additive Explanations (SHAP) and Partial Dependence Plot (PDP) analyses to identify the parameters that have the most significant impact on discharge prediction.
5. To address the discrepancy between complex computational predictions and practical real-world applications, the study will develop a user-friendly Graphical User Interface that will allow engineers to quickly predict the discharge of type-A PKW.

2. Methodology

Fig. 2 shows the adopted flowchart of the methodology used in this study. The methodological approach begins with collecting a comprehensive dataset of 476 experimental samples. The data was pre-processed through normalization and exploratory analysis using visual tools such as Histograms, Pair plots, Hex contour, and correlation heatmaps. Eight predictive models were employed, which split the dataset into 70% for the training stage and 30% for the testing stage. Bayesian Optimization with k-fold cross-validation was used for hyperparameter tuning, significantly enhancing model accuracy. The model's performance was evaluated using both visual methods and quantitative methods. Based on the best-performing model, SHapley Additive exPlanations (SHAP) and Partial Dependence Plots (PDPs) were then used to identify and interpret key features affecting discharge predictions. Finally, it was developed a user-friendly Graphical-User-Interface (GUI), ensuring accessibility for practical applications.

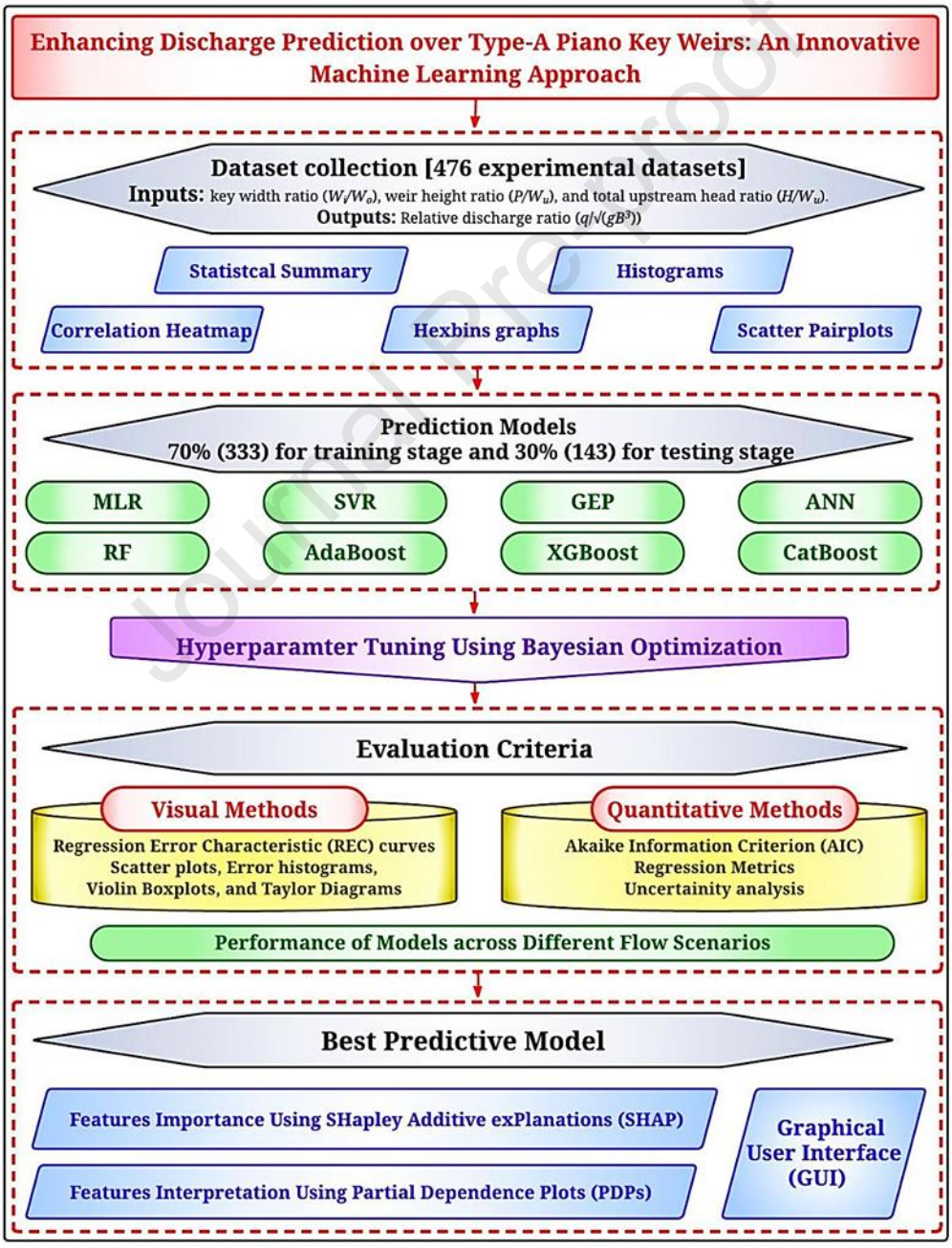


Fig. 2. Flowchart of the methodological approach.

2.1. Database collection

Researchers can develop models for predicting outcomes and statistically analyze the data using information from laboratory experiments or previously published studies. In this study, a significant dataset of 476 data points related to the discharge of type-A PKW was assembled from past research articles: Machiels et al. [10] and Ribeiro et al. [30]. The study's data analysis focused on three principal dimensionless variables, which were used as the input features. Input **X1** denotes the PKW key widths ratio (W_i/W_o), input **X2** denotes the PKW height ratio (P/W_u), and input **X3** denotes the total upstream head ratio (H/W_u). While the output variable, indicated by (**Y**), is the relative discharge ratio of type-A PKW ($q/\sqrt{gB^3}$).

2.1.1. Statistical summary

Table 1 presents a comprehensive summary of the collected data's characteristics, providing a succinct overview of the statistical description. The table provided presents descriptive statistics for the input and output variables. Each variable is characterized by its minimum, maximum, mean, median, standard deviation, kurtosis, and skewness, providing a comprehensive overview of their distributions. For X1, the minimum value is 0.480, and the maximum is 2.180, indicating a broad range of values for this variable. The mean of X1 is 1.122, which is slightly higher than the median of 1.000. This suggests a slight skew towards higher values, which is also supported by the positive skewness of 0.600. The standard deviation of 0.530 reflects a moderate spread around the mean. The kurtosis value of -0.771 indicates a flatter distribution than the normal distribution, meaning the data has lighter tails. For X2, the minimum value is 0.500, and the maximum is 1.330, giving a narrower range compared to X1. The mean is 0.800, and the median is 0.670, indicating that the distribution is skewed towards higher values, which is confirmed by the skewness of 0.630. The standard deviation of 0.346 shows less variability compared to X1. The kurtosis of -1.259 suggests that the distribution is even flatter than that of X1, with a more significant departure from normality.

The variable X3 has a minimum value of 0.149 and a maximum of 0.720, which is the smallest range among the three independent variables. The mean and median are nearly identical at 0.430 and 0.432, respectively, indicating a symmetric distribution, which is further supported by the very low skewness of 0.028. The standard deviation is 0.175, showing relatively low variability. The kurtosis of -1.220, like the other variables, indicates a flatter distribution with lighter tails compared to the normal distribution. Finally, the dependent variable Y ranges from a minimum of 0.024 to a maximum of 0.188. The mean is 0.095, and the median is 0.093, indicating that the data is fairly symmetric around the center. The skewness is 0.238, suggesting a slight positive skew, though not as pronounced as in X1 or X2. The standard deviation of 0.042 indicates that Y has the lowest variability among the variables listed. The kurtosis of -0.932 suggests that the distribution of Y is also flatter than normal, but less so than X2 and X3.

Table 1
Descriptive statistics for the collected data.

Variable	Symbol	Min	Max	Mean	Median	Std. Dev.	Kurtosis	Skewness
W_i/W_o	X1	0.480	2.180	1.122	1.000	0.530	-0.771	0.600
P/W_u	X2	0.500	1.330	0.800	0.670	0.346	-1.259	0.630
H/W_u	X3	0.149	0.720	0.430	0.432	0.175	-1.220	0.028
$\frac{q}{\sqrt{gB^3}}$	Y	0.024	0.188	0.095	0.093	0.042	-0.932	0.238

2.1.2. Histograms

Fig. 3 consists of four histograms, each illustrating the distribution of the variables X1, X2, X3, and Y. Each histogram provides a detailed view of how the data points are spread across different intervals or bins, helping to visualize the underlying distribution patterns of these variables. The first histogram represents the distribution of X1. The x-axis is divided into four bins, each representing a range of X1 values. The y-axis indicates the frequency, or the number of data points, that fall within each bin. The histogram reveals that the majority of data points are concentrated in the first bin, centered around the lower end of the X1 range, with 272 observations. This high frequency at lower values indicates a right-skewed distribution, where most data points are clustered towards the lower values of X1. As the X1 values increase, the frequency of data points sharply decreases, with the higher bins containing significantly fewer observations.

The second histogram depicts the distribution of X2. Similar to X1, the X2 values are grouped into four bins along the x-axis. The highest frequency of data points is observed in the first bin, centered around 0.500, with 238 observations. This suggests that a significant portion of the data is concentrated at the lower end of the X2 range, indicating a skewed distribution. The subsequent bins show a gradual decrease in frequency, with the third and fourth bins containing fewer data points, reflecting a distribution that tails off as X2 increases. This pattern reinforces the idea that X2 is also skewed towards lower values, though the distribution is slightly more spread out compared to X1. The third histogram shows the distribution of X3. The x-axis is divided into three bins, each representing a range of X3 values. Unlike X1 and (X, the histogram for X3 displays a more balanced distribution, with the frequencies of data points being relatively similar across the bins. The first bin, centered around 0.1490, contains 168 observations, while the second and third bins, around 0.3393 and 0.5297 respectively, each have 154 observations. This even distribution suggests that the data for X3 is more symmetrically spread across its range, with no single bin dominating the distribution. This symmetry indicates that X3 does not exhibit the same degree of skewness as X1 and X2.

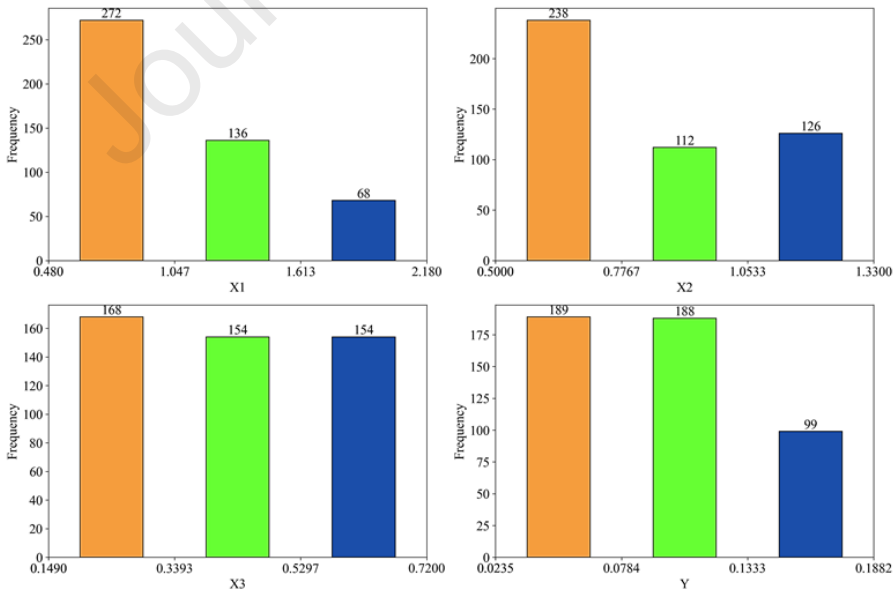


Fig. 3. Histograms of the input and output variables.

The fourth histogram illustrates the distribution of the dependent variable Y. The data is divided into three bins along the x-axis, representing different ranges of Y values. The first two bins, centered around 0.0235 and 0.0784, contain the highest frequencies, with 189 and 188 observations respectively. This indicates that the majority of Y values are concentrated in the lower to mid-range of its distribution. The third bin, around 0.1333, has a lower

frequency of 99 observations, suggesting a tapering off of the data as Y increases. The histogram for Y reflects a moderate skew towards lower values, with a noticeable drop in frequency as the values increase, but this skew is less pronounced compared to X1 and X2. These histograms collectively provide a comprehensive overview of the distribution patterns for X1, X2, X3, and Y. X1 and X2 exhibit strong right-skewed distributions, with most data points clustered at the lower end of their respective ranges. X3 shows a more symmetric distribution, with frequencies evenly distributed across its range. Y has a moderate skew towards lower values but is generally more balanced than X1 and X2.

2.1.3. Hex contour

Fig. 4 contains three hex contour graphs, each showing the relationship between the dependent variable Y and the three inputs (X1-X3). The hex contour graphs provide a visual representation of the data distribution and the association between these variables. The first graph on the left examines the relationship between Y and X1. The graph shows that as X1 increases, Y also tends to increase, suggesting a positive relationship between these two variables. The data points are color-coded according to the frequency of occurrences at each data point, as indicated by the gradient in the colour bar on the right. Darker shades represent higher counts, meaning there are more data points concentrated around those values. The graph shows that the values of Y increase gradually with X1, though there is a noticeable spread, indicating variability in the data.

In the second graph, the relationship between Y and X2 is shown. The graph illustrates that Y also has a positive association with X2. However, unlike X1, the data points in this graph appear to be more clustered, with distinct vertical alignments at specific values of X2. This clustering suggests that certain discrete values of X2 are more common, around which the values of Y are more concentrated. The colour gradient again indicates the count of data points, with darker shades highlighting higher concentrations of points. The third graph displays the relationship between Y and X3. The trend is similar to the first two plots, where Y increases as X3 increases, indicating a positive correlation between these variables. The graph is more uniformly distributed across the range of X3 compared to the previous graphs. The colour coding reveals that higher frequencies of data points occur at certain intervals, shown by darker red shades, indicating that certain ranges of X3 are more frequently observed in the data.

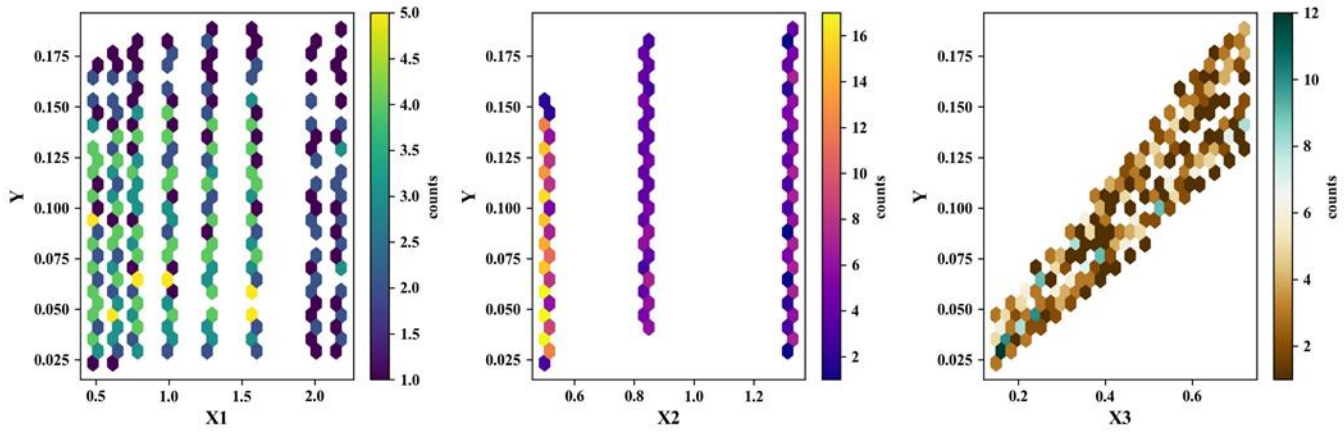


Fig. 4. Hex contour graphs showing the relationship between the input and output variables.

2.1.4. Correlation analysis

To find the most effective prediction model, it is essential to analyze the correlation between variables in order to understand the connections between dependent features and the objective strength factor. Fig. 5 presents a correlation

heatmap that demonstrates the influence of each variable on all other variables. Each cell in the map shows the correlation coefficient between the pair of variables corresponding to that cell's row and column. For the diagonals, each variable correlates perfectly with itself, as expected, resulting in a correlation coefficient of 1.0.

The non-diagonal values represent the correlation between different variables. Values nearing 1.0 confirm a strong positive correlation. Values close to 0 indicate a lack of correlation. Values approaching -1 (although none appear in Fig. 5) would suggest a solid negative correlation. The input **X1** has very weak correlations with all other variables (including Y), indicating that X1 may not be a strong predictor of Y or related to other variables. Additionally, the input **X2** shows a weak correlation with Y (0.266) and is not strongly correlated with X1 (0.0782) or X3 (0.00637). It could have a slight predictive power for Y but is not particularly strong on its own. In contrast, the input **X3** stands out with a very strong correlation with **Y** (0.942). This suggests that X3 is likely a very significant predictor of Y in this dataset.

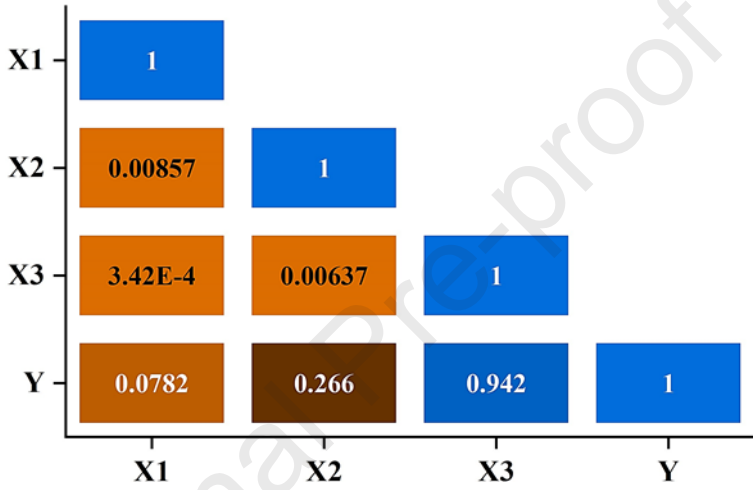


Fig. 5. Heatmap correlation matrix.

2.1.5. Scatter pair plots

Fig. 6 displays a scatterplot matrix that offers a thorough visual examination of the three input variables and their correlation with the output variable (Y). The matrix has histograms along the diagonal, which depict the distribution of each variable separately. The non-diagonal cells in the matrix display scatter plots illustrating the pairwise correlations between variables. Every scatter plot offers a graphical depiction of the relationship between two variables. For input **X1**, the distribution is multimodal, indicating that the data clusters around several different values. The input **X2** also shows a multimodal distribution, but with more pronounced peaks and valleys, suggesting distinct groupings within the data. The input **X3** exhibits a more uniform distribution, with a broad, flat peak, indicating that its values are spread relatively evenly across a certain range. While the output (**Y)** has a nearly symmetric distribution, centered around its mean, with values tapering off towards the tails.

Moving to the scatter plots, the off-diagonal cells depict the relationships between pairs of variables. The scatter plot between X1 and X2 (the second cell in the first row) shows little to no apparent linear relationship, as the points are scattered with no clear trend. The scatter plot between X1 and X3 (the third cell in the first row) also shows a similar lack of correlation, with the points spread across the plot without any discernible pattern. The relationship between X1 and Y (fourth cell in the first row) shows a slight positive trend, as indicated by the line fit, suggesting a weak positive correlation where Y tends to increase with X1. However, the scatter of points is quite broad, indicating a lot of variability around this trend. The scatter plot between X2 and X3 (third cell in the second row)

demonstrates a near-vertical clustering of points, indicating that changes in X2 do not significantly affect X3, which aligns with the low correlation observed in the previous correlation matrix. The plot between X2 and Y (fourth cell in the second row) shows a slightly stronger positive correlation, with Y increasing as X2 increases, as suggested by the slight upward slope of the fit line. The scatter plot between X3 and Y (the fourth cell in the third row) stands out, showing a strong positive correlation. The points form a clear upward trend, and the fit line highlights a significant linear relationship, indicating that as X3 increases, Y increases substantially as well. This visual correlation is consistent with the high correlation coefficient seen in the previous correlation matrix.

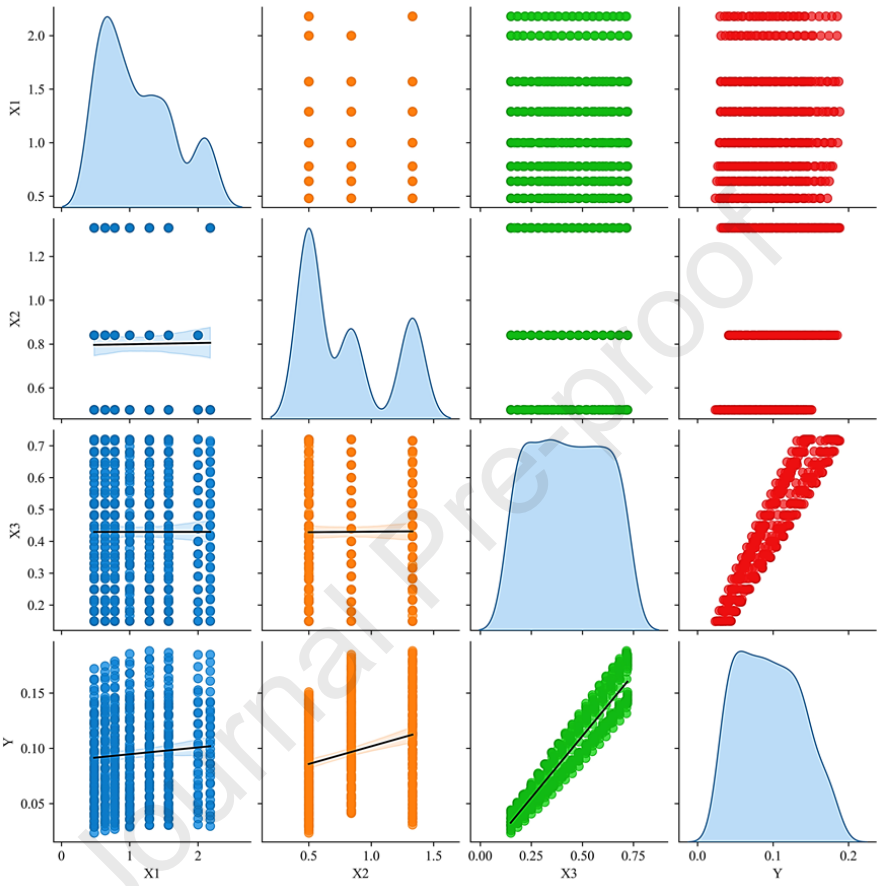


Fig. 6. Scatter pair plots matrix for input and output variables.

2.2. Overview of adopted models

ML models, inspired by the human brain's structure, are designed to be trained and adapted from datasets to make predictions. These models consist of input and output layers connected by one or more hidden layers that compute and uncover patterns in the input data. In the current study, individual ML models (non-ensemble) and ensemble models with weak learners were adopted to forecast the discharge of type-A PKW, which is briefly discussed in subsequent sections. Four non-ensemble ML models have been developed: Artificial Neural Network (ANN), Support Vector Machine (SVM), Gene Expression Programming (GEP), and Multiple Linear Regression (MLR). Additionally, four ensemble ML models were developed: Random Forest (RF), Adaptive Boosting (AdaBoost), Extreme Gradient Boosting (XGBoost), and Categorical Boosting (CatBoost). It is essential to note that this research utilizes the Python programming environment within the ANACONDA software, MATLAB, and SPSS programs.

2.2.1. MLR model

Regression models estimate the degree of correlation and determine the relationships between input and output variables. Most MLR models are fitted using least squares methods. MLR assesses the correlation between a

dependent variable and various independent variables, yielding a linear relationship [31]. The common formulation of MLR is depicted as follows:

$$Y = a_0 + \sum_{i=1}^{m} a_i X_i \quad (2)$$

where Y is the output of an MLR model, Xi represent the input variables, a_0 = equation constant, and a_1, a_2, … and a_m are partial regression coefficients.

2.2.2. SVM model

The Support Vector Regression (SVR) model, introduced by Cortes and Vapnik [32] as a variant of Support Vector Machines (SVM) tailored for regression problems, is designed to predict continuous outcomes. It achieves this by constructing a function that ensures deviations from the actual observed values remain within a predefined margin across all training data points. What sets SVR apart is its approach to determining an optimal hyperplane within a transformed, higher-dimensional feature space, which is facilitated by a kernel function. The primary objective of SVR is to strike a balance between reducing the complexity of the model, represented by the flatness of the hyperplane, and minimizing the deviations from the true data points, provided these deviations are within the specified tolerance [33]. This equilibrium between reducing error and maintaining model simplicity is essential for the SVR's ability to generalize well, making it a versatile algorithm for regression tasks in various applications.

2.2.3. GEP model

The GEP approach was introduced by Ferreira [34] as a research technique that involves computer programs, which is an extension of genetic programming (GP) [35]. Expression trees (ETs) result from translating the linear chromosomes that contain the code for all GEP computer programs. Random population chromosomes started the procedure. Expression of the chromosomes determined each individual's fitness. Individuals were selected for their ability to reproduce after genetic modification, producing offspring with novel properties. Consecutively, the individuals of the present generation had a similar developmental process, which involved the expression of their genomes, encounter with the selective environment, and reproduction with an alteration [34]. The genome was replicated and passed to the next generation. Only the remaining operators' actions added genetic diversity. These operators randomly select the chromosomes to be changed. GEP allows several operators to alter or leave a chromosome unchanged.

2.2.4. ANN model

The concept of Artificial Neural Networks (ANNs), inspired by the neural networks in the human brain, was first developed by McCulloch and Pitts in 1943 [36]. ANNs are adept at modeling linear and nonlinear systems without the constraints often associated with traditional statistical methods. Typically, an ANN comprises three layers: an input layer, one or more hidden layers, and an output layer. The training process of an ANN involves adjusting connection weights according to a learning rule derived from an input-output dataset, with the goal of minimizing the difference between predicted outputs and actual values, as highlighted by Shayya and Sablani [37]. This training process includes two main phases: feed-forward and back-propagation. During the feed-forward phase, data flows from the input layer through the hidden layers to the output layer. In the back-propagation phase, weights are adjusted from the output layer back to the input layer based on the overall error in the network [38]. To further reduce this error, an appropriate back-propagation algorithm is employed, with the Levenberg-Marquardt technique being the

most efficient and commonly used method for this purpose.

2.2.5. RF model

RF model employs bootstrap sampling to generate multiple training sets from the original dataset [39]. Each of these sets includes approximately two-thirds of the original data, leaving one-third out as out-of-bag data. For each training set, a decision tree is constructed, and these trees collectively form the forest. During the growth of each tree, the best attributes for branching are randomly selected from a predefined maximum depth. This approach ensures diversity among the classification models, enhancing the overall predictive power of the combined model. After training the model multiple times, the algorithm determines the class of a new sample by taking the majority vote from all the trees. This final step is based on an equation that aggregates these votes.

2.2.6. AdaBoost model

AdaBoost is a type of machine learning algorithm under the category of ensemble methods and is recognized as the new boosting algorithm for both classification and regression tasks [40,41]. Specifically, AdaBoost regression targets regression problems. Like other ensemble approaches, this method merges several simple models to form a more accurate and robust classifier. The core concept behind AdaBoost regression involves constructing a robust model by amalgamating several weak ones, where each new model focuses more on the data points that its predecessors inaccurately predicted. Through iterative training, AdaBoost regression fine-tunes weak models on a weighted dataset version, emphasizing previously mispredicted data points with each round. This process continues until several models are trained or a desired level of accuracy is reached.

2.2.7. XGBoost model

XGBoost is a highly effective and versatile ML library that excels in handling large datasets and complex predictive modeling tasks [42]. Its ability to handle missing values, regularization techniques, and parallel processing capabilities make it a reliable choice for various applications. With its high accuracy, speed, and scalability, XGBoost is a popular choice for many industries, including finance, computer vision, and natural language processing [33]. Overall, XGBoost is a powerful tool for data scientists and machine learning engineers, offering a robust and efficient way to build predictive models that can drive business decisions and improve outcomes.

2.2.8. CatBoost model

CatBoost is a powerful machine learning algorithm designed to handle categorical features and produce accurate predictions. It is a variant of gradient boosting that can handle both categorical and numerical features without requiring preprocessing like one-hot encoding or label encoding. CatBoost employs its built-in encoding system called "ordered boosting" to process categorical data directly, resulting in faster training and better model performance [43]. It is particularly useful for regression tasks where the goal is to predict a continuous target variable. CatBoost is known for its speed, accuracy, and ease of use, especially in situations involving structured data with many categorical features [44]. It also offers feature relevance rankings that help with feature selection and understanding model choices.

2.3. Hyperparameters tuning

Selecting the right hyperparameters is key to enhancing model performance [45]. Techniques like Grid Search (GS), Random Search (RS), and Bayesian Optimization (BO) are typically employed to adjust these parameters in ML models [46]. GS and RS may not always find the best solution, as they often involve high variance and can be

time-intensive due to numerous trials. On the other hand, BO utilizes previous evaluations to search for optimal solutions more effectively, often requiring less time to identify the most suitable parameters compared to GS and RS. Consequently, this study adopts the BO method for determining the predictive models' ideal parameters. To ensure model robustness on new data and to prevent overfitting, cross-validation (CV) is also applied. Specifically, a 5-fold CV integrated with BO (BO+5CV) is utilized for hyperparameter optimization of the predictive models [20,40].

The use of 5-fold cross-validation (CV) in this research is justified for several reasons, focusing on efficiency, bias-variance trade-off, and robustness. 5-fold CV offers a practical balance between computational cost and quality of model evaluation. While higher k-values in CV can reduce bias and provide more accurate performance estimates, they also increase computational demands. A 5-fold CV reduces the number of iterations compared to a higher number of folds like 10, making it less time-consuming while still providing a reliable evaluation. It also strikes a balance in the bias-variance trade-off, offering sufficient training and validation sets to ensure stable performance metrics without overfitting. Additionally, a 5-fold CV is widely recognized as a standard practice, making it a robust choice for various applications. When integrated with BO, it effectively narrows down optimal hyperparameters, enhancing search efficiency and reducing computational costs. Overall, 5-fold CV is a pragmatic and effective method for hyperparameter optimization and model evaluation in this study.

2.4. Evaluation criteria

To ensure the practicality and scientific reliability of the outcomes, assessing the effectiveness of each model is essential [47,48]. While training datasets help construct models, they only reveal how well they fit the given data. Therefore, testing datasets are crucial for validating the models. Testing datasets serve as a benchmark to evaluate how well a model generalizes to unseen data. Without a testing dataset, it is impossible to determine if a model is simply memorizing the training data or if it can truly make accurate predictions on new inputs. This step is critical to ensure the model's reliability and applicability in real-world scenarios .

In this study, the evaluation and comparison of models typically involve two main methods: visual and quantitative assessments. Visual assessments involve graphical representations such as scatter plots, error histograms, violin boxplots, Taylor diagrams, and Regression Error Characteristic (REC) curves [29], which provide intuitive insights into the model's performance. For instance, scatter plots are commonly used for regression models to show the relationship between predicted and actual values. These visual tools help identify patterns, trends, and potential anomalies that might not be evident through numerical measures alone [49]. Additionally, AIC (Akaike Information Criterion) and uncertainty analyses are performed to further evaluate the models' robustness and reliability. AIC is particularly useful for model selection, as it considers both the goodness of fit and the complexity of the model, allowing for a more balanced evaluation. Quantitative assessments, on the other hand, provide numerical metrics that offer an objective evaluation of the model's performance. These quantitative measures are essential for comparing different models and selecting the best one for a given task.

2.4.1. Akaike Information Criterion (AIC)

AIC was developed by Hirotsugu Akaike in the early 1970s. It is a statistical tool used for model selection. It evaluates the relative quality of different models based on their fit to a dataset, balancing model complexity and accuracy [50]. AIC helps estimate the information lost when using a model to approximate the data-generating process, addressing overfitting and underfitting risks. AIC is calculated using the formula (Eq. 3) as follows [51]:

$$\mathrm{AIC} = 2K + n\log\left(\frac{\mathrm{RSS}}{n}\right) \quad (32)$$

where *k* is the number of parameters in the model, *n* is the number of observations and *RSS* is the residual sum of squares. The model with the lowest AIC value is considered the best, as it indicates a good fit with fewer parameters.

2.4.2. Regression metrics

Understanding various metrics is crucial for evaluating the performance of regression models. These metrics provide different perspectives on how well a model predicts the target variable. In various research articles [52–54], several metrics were used to thoroughly evaluate the constructed models and to compare their performances overall. To achieve this aim, five distinct metrics, namely R^2, RMSE, RMSRE, MAE, and MAPE, as listed in Table 2.

Table 2
Regression metrics used to assess the performance of developed models.

Metric	Description	Formula	Ideal value
R^2	Determination coefficient	$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$	1
RMSE	Root Mean Squared Error	$\sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$	0
RMSRE	Root Mean Squared Relative Error	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2}$	0
MAE	Mean Absolute Error	$\frac{\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert}{n}$	0
MAPE	Mean Absolute Percentage	$\frac{\sum\left\lvert\frac{y_i - \hat{y}_i}{y_i}\right\rvert}{n} \times 100$	0

Where *n* is the dataset number; y_i and $\hat{y}_i$ are actual and predicted i^{th} values, respectively; $\bar{y}$ is the mean of actual values; $\bar{\hat{y}}$ is the mean of predicted values.

2.4.3. Uncertainty analysis

Uncertainty analysis is an effective technique used to evaluate and quantify the uncertainty associated with the prediction accuracy of hybrid machine learning models. In the context of forecasting critical systems, it is crucial to identify and address the uncertainties present in prediction algorithms, which include uncertainties related to experimental conditions, input predictors, and model outcomes [55,56]. This methodology allows for a reliable interpretation of the results and enables well-informed decision-making based on the degree of uncertainty in the algorithm. The uncertainty measure U_{95} is calculated using the following formula:

$$U_{95} = 1.96 \times \sqrt{\mathrm{SD}^2 + \mathrm{RMSE}^2} \quad (43)$$

Where SD represents the standard deviation of the model prediction performance error, while 1.96 is the z-value corresponding to a 95% confidence interval in a standard normal distribution.

2.5. Features sensitivity analysis

Understanding and interpreting machine learning models are crucial for ensuring their effectiveness, trustworthiness, and reliability. Two advanced methods often used for feature sensitivity analysis are the SHAP approach and PDPs. These techniques help researchers and practitioners gain insights into the model's decision-

making process by identifying the influence of various features on predictions. Below, these methods are described in detail.

2.5.1. SHapley Additive exPlanations (SHAP)

To evaluate the sensitivity and interpret ML models at both broad and detailed levels, researchers utilize the SHAP method, which is based on cooperative game theory principles [57]. The SHAP method was applied to assess the relative impact of input variables on the prediction process. SHAP elucidates the complex interactions between input variables and model predictions as a sophisticated technique within eXplainable Artificial Intelligence (XAI). It provides critical insights by highlighting which features influence the predictions most and how they affect the predicted outcomes [58]. The Shapley value ϕ_i for feature *i* is determined using Eq. (5), which calculates the average marginal contribution of that feature across all possible permutations of features. In this equation, *N* represents the set of all features, *S* denotes a subset of features excluding feature *i*, $|S|$ indicates the cardinality of set (*S*), $v(S)$ represents the model's prediction based only on features in set (*S*), and $v(S \cup \{i\})$ denotes the model's prediction when feature *i* is added to set *S* [59].

$$\phi_i = \sum_{S \subseteq N\{i\}} \frac{|s|!(|N|-|S|-1)!}{|N|!} [v(S \cup \{i\}) - v(S)] \quad (54)$$

2.5.2. Partial Dependence Plot (PDP)

A PDP demonstrates the functional relationship between a limited set of input parameters and the model's predictions. It illustrates the extent to which the predictions are influenced by the values of these specific input parameters [60]. Moreover, PDPs highlight the impact of each parameter on the predicted outcomes generated by a machine learning algorithm. As a global method, PDPs consider all instances within the dataset, thereby revealing the overall relationship between a feature and the forecasted outcome [61]. These plots offer valuable insights into the relative contribution of each input variable to the predicted outcome. Furthermore, one-dimensional PDPs (PDPs-1D) are utilized to depict the relationship between the predicted outcome and a single input parameter.

2.6. Graphical User Interface (GUI)

To make the predictive model for estimating the relative discharge ratio accessible and user-friendly, a GUI is developed using the Tkinter library in Python [62]. Tkinter, a standard GUI library, offers a straightforward approach to creating interactive applications [63]. The development process starts with setting up the Python environment with Tkinkter, which is included in the standard Python distribution. The interface design focuses on simplicity and user guidance, incorporating input fields for users to enter required features, a button to trigger predictions, and labels or instructions for ease of use. The core functionality of the GUI involves creating input fields with TextEntry widgets, a Button widget to initiate the prediction process, and a Label or Text widget to display the predicted seepage loss ratio. The trained predictive model is integrated into the GUI using a library-like pickle to load the model and make predictions based on user inputs. To ensure accessibility and collaboration, the GUI application is hosted on GitHub (https://github.com), which provides version control, collaboration features, and easy distribution.

3. Results and Discussion

3.1. Hyperparametric configuration of models

Table 3 presents the optimal hyperparameters obtained across the BO+5CV tuning process for the developed non-ensemble and ensemble models. These hyperparameters have been fine-tuned to achieve the best possible

performance for each model. For the non-ensemble models, the MLR model is configured with four coefficients: a_0 = -0.033, a_1 = 0.007, a_2 = 0.031, and a_3 = 0.224, which define the linear relationships between the input variables and the target variable (see Eq A.1 in the appendix A). The SVR model uses the Radial basis kernel function. The optimal values for the hyperparameters include a Box-Constraint (C) of 0.048 and a Gamma of 0.0416, which control the model's margin and the influence of individual data points. For the GEP model, the configuration includes 30 chromosomes, 3 genes, and a fitness function based on RMSE. The linking function used is multiplication, and the function set incorporates basic arithmetic operations and various root and power functions, along with inputs involving addition, subtraction, and multiplication. The equation derived from the GEP model (Eq. B.1) and its sub-expression trees (Fig. B.1) are provided in Appendix B. The ANN model is optimized with a 1.0 hidden layer and a hidden layer size of 20, providing a relatively simple architecture for learning complex patterns.

For ensemble models, on the other hand, the RF model used 816 estimators, ensuring a substantial number of trees to improve accuracy. The maximum depth is set to 18, with minimum samples split of 2 and a minimum samples leaf of 1.0, allowing the model to capture fine details in the data. The AdaBoost model is configured with 1000 estimators, focusing on iterative boosting to enhance the performance of weak learners. A learning rate of 2.67 is applied, resulting in a more aggressive weight adjustment process, and the loss function is set to "linear." For the XGBoost model, 431 estimators are utilized, and the maximum depth is set to 25 to allow deeper trees and better capture intricate interactions. The learning rate is set to 0.094, and the minimum child weight is configured at 5.0 to prevent overfitting by regulating the minimum sum of instance weights in the child nodes. Lastly, the CatBoost model is optimized with a learning rate of 0.182 and a depth of 4, emphasizing simplicity in the trees for better generalization. The model also includes an l2_leaf_reg parameter set to 1.87, which provides regularization by penalizing large leaf values and reducing model complexity. The open access link for all codes utilized in the development of ML models is provided in Appendix C.

Table 3
Obtained optimal hyperparameters from the BO+5CV tuning process for implemented models.

Category	Model	Hyperparameter
Non-ensemble	**MLR**	a_0 = -0.033; a_1= 0.007; a_2 = 0.031; a_3 = 0.224
	SVR	Kernel function = Radial basis; C = 0.048; Gamma = 0.0416
	GEP	No. of chromosomes = 30; No. of genes = 3; Fitness function = RMSE; Linking function = Multiplication; Function set (+, -, *, /, Sqrt, 3Rt, 4Rt, 5Rt, Pow of 2, 3, 4, 5, Addition with 3, 4 inputs, Subtraction with 3, 4 inputs, Multiplication with 3, 4 inputs)
	ANN	n_hidden_layers = 1; hidden_layer_size = 20
Ensemble	**RF**	n_estimators = 816; max_ depth =18; min_samples_split =2; min_samples_leaf = 1
	AdaBoost	n_estimators = 1000; learning_rate = 2.67; loss = linear
	XGBoost	n_estimators = 431; max_depth =25; learning_rate = 0.094; min_child_weight = 5
	CatBoost	learning_rate = 0.182; depth = 4; l2_leaf_reg = 1.87

3.2. Evaluating model Performance: A Visual-Based Approach

3.2.1. REC curves

Fig. 7 presents Regression Error Characteristic (REC) curves for both training and testing predictions across the eight adopted models. The REC curve plots the cumulative distribution of residual errors on the y-axis against the residual error on the x-axis. A steeper curve that reaches a higher cumulative distribution with a lower residual error indicates better model performance. Fig. 7a displays the REC curves for the training stage. In this plot, the CatBoost

model stands out as the top performer with the steepest REC curve, indicating minimal residual errors and a strong ability to capture patterns in the data. Following closely behind is the XGBoost model, which also demonstrates efficient learning with low residual errors, although its curve is slightly more gradual compared to CatBoost. The RF and ANN models perform reasonably well, but their curves indicate slightly higher residual errors, suggesting a moderate fit to the training data. GEP, AdaBoost**,** and SVR exhibit more gradual curves, reflecting higher errors and weaker learning capabilities. Lastly, the MLR model struggles the most during training, with a notably flat curve indicating the highest residual errors among all models.

In the testing stage (Fig. 7b), the CatBoost model continues to excel, with the steepest curve and the lowest residual errors, confirming its strong generalization abilities to unseen data. XGBoost also performs very well, with a curve that is slightly more gradual than the CatBoost model's but still indicative of low error rates. ANN shows a moderately steep curve but flattens more than the top-performing models. The RF model shows moderate performance, with higher residual errors than CatBoost, XGBoost, and ANN models, suggesting weaker generalization. GEP's curve in testing is more gradual, similar to its performance in the training stage. It exhibits higher errors than the top-performing models. SVR's curve shows a slow rise, indicating that it struggles with generalization, having higher residual errors in the test stage. While the AdaBoost and MLR models once again perform the worst, with much flatter curves, signifying higher residual errors. Overall, these REC curves underscore the strong performance of ensemble models, particularly CatBoost and XGBoost in both training and testing stages, with CatBoost having a slight edge. In contrast, the most effective models among the non-ensemble models are ANN and GEP. Although their performance does not match that of the top ensemble models.

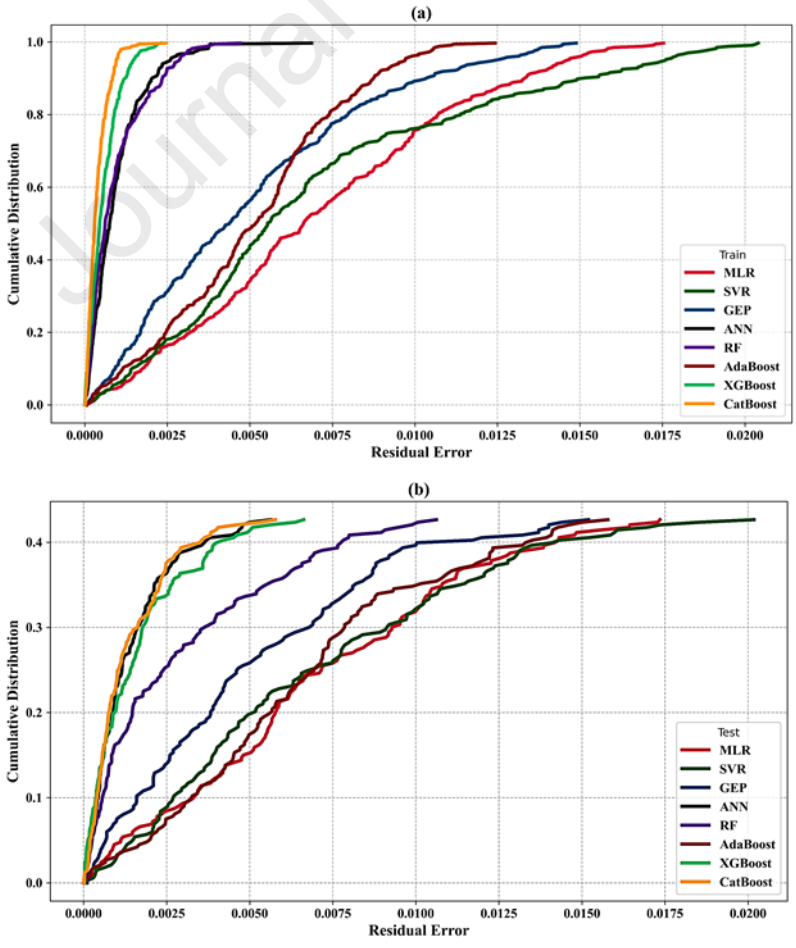
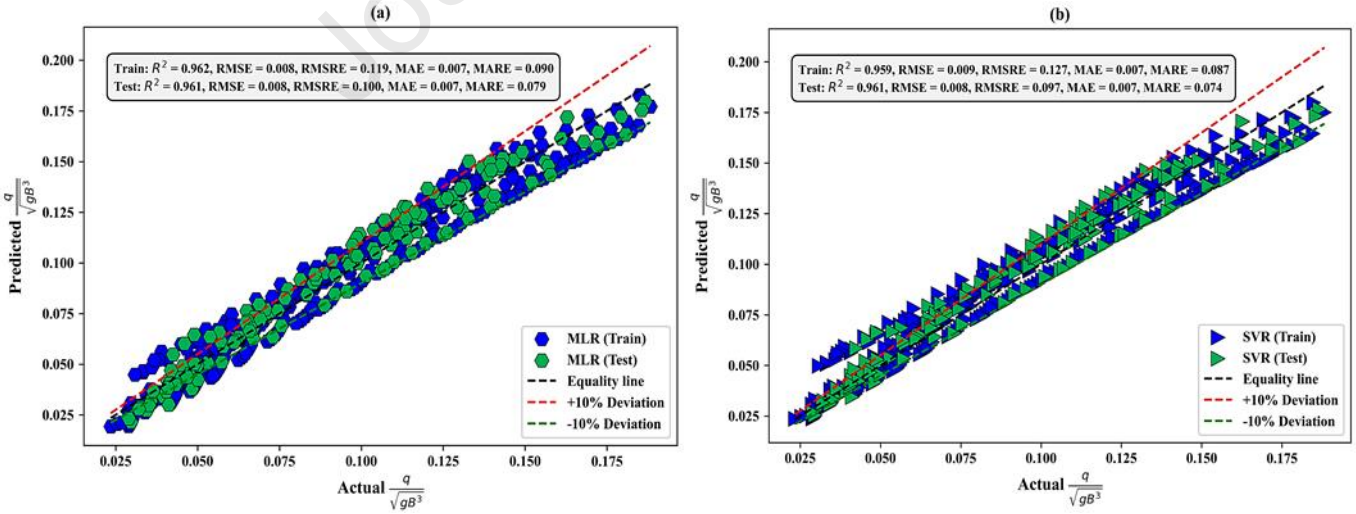


Fig. 7. REC curves illustrate the performance of the developed models during the (a) training and (b) testing stages.

3.2.2. Scatter plots

Fig. 8 displays the predicted versus actual values for the developed models across both training and testing datasets. Each plot includes the equality line (black dashed line), along with lines indicating ±10% deviation (red and green dashed lines), providing a visual benchmark for model accuracy. The predicted points of XGBoost, CatBoost, and ANN are tightly clustered along the equality line and well within the ±10% deviation lines, indicating that they outperform the other models in the training stage. This tight clustering suggests that these models have minimal prediction errors and fit the training data exceptionally well. In comparison to the XGBoost and CatBoost models, the RF model also exhibits strong performance; however, its points exhibit a slight increase in scatter, although the majority of them remain within the ±10% range. The distributions of the GEP, AdaBoost, and SVR models are more scattered, with points beginning to deviate from the equality line, indicating higher errors. Among all models, MLR exhibits the most dispersion, with several points falling further from the equality line, suggesting that it experiences greater difficulty fitting the training data.

Although there are a few outliers, the CatBoost, XGBoost, and ANN models continue to exhibit excellent results during the testing stage. The majority of points remain within the ±10% deviation lines and in close proximity to the equality line, suggesting that they are capable of generalizing to unseen data. RF maintains strong performance, albeit with a slight increase in scatter compared to its training phase. However, the majority of points remain within the acceptable error range. The AdaBoost and GEP models exhibit moderate scatter, with numerous points falling outside the ±10% deviation lines, suggesting a weaker generalization. The SVR and MLR models exhibit the most substantial scatter during testing, with a lot of outliers and a large dispersion beyond the ±10% borders. This indicates that their accuracy is reduced and their errors are higher when generalizing to new data. Regarding the overall dispersion of data in both stages, ensemble models like XGBoost and CatBoost, as well as non-ensemble models like ANN, consistently achieve the best-performing models. In the upcoming subsections, the qualitative comparison will be examined using regression metrics.



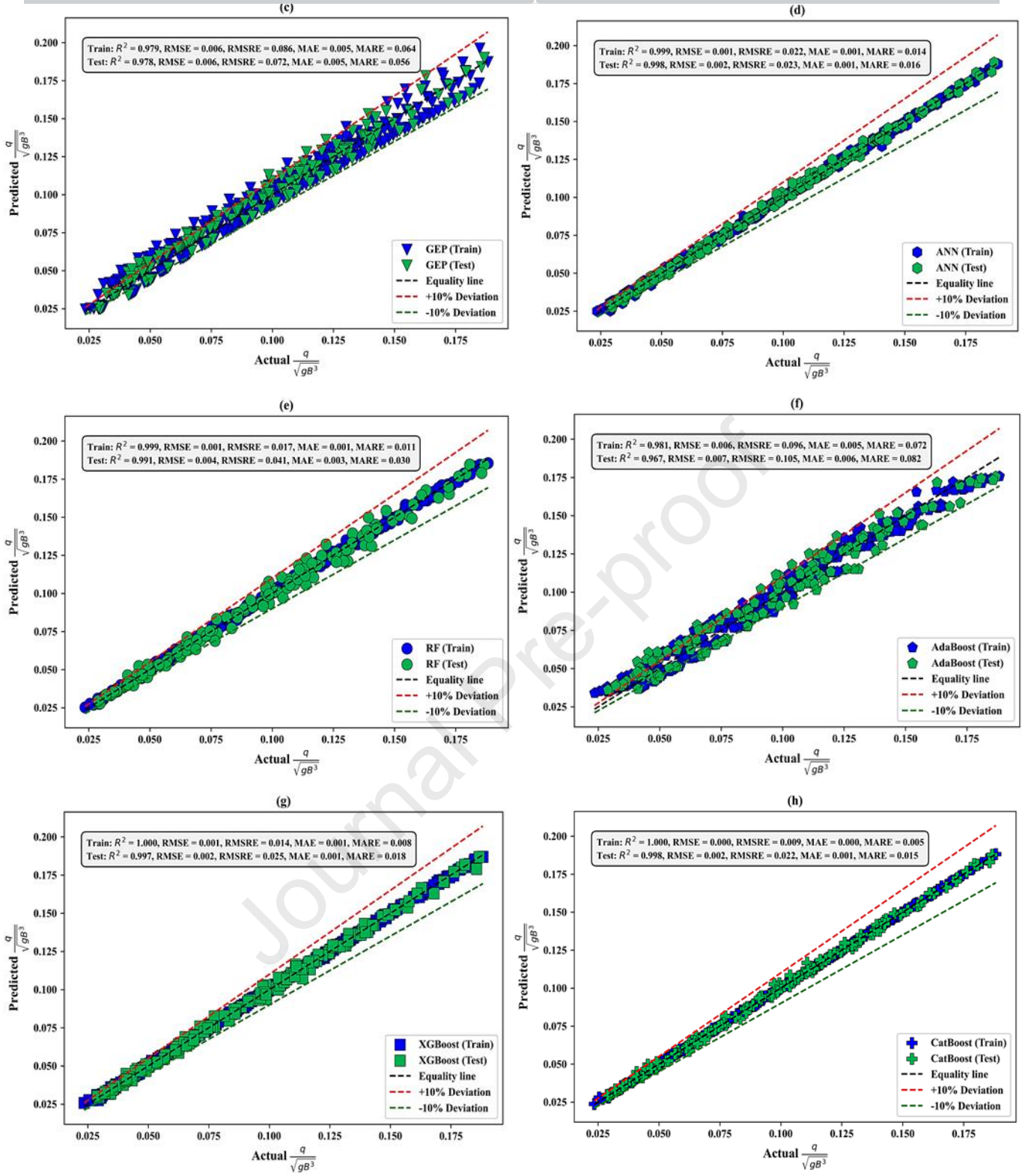


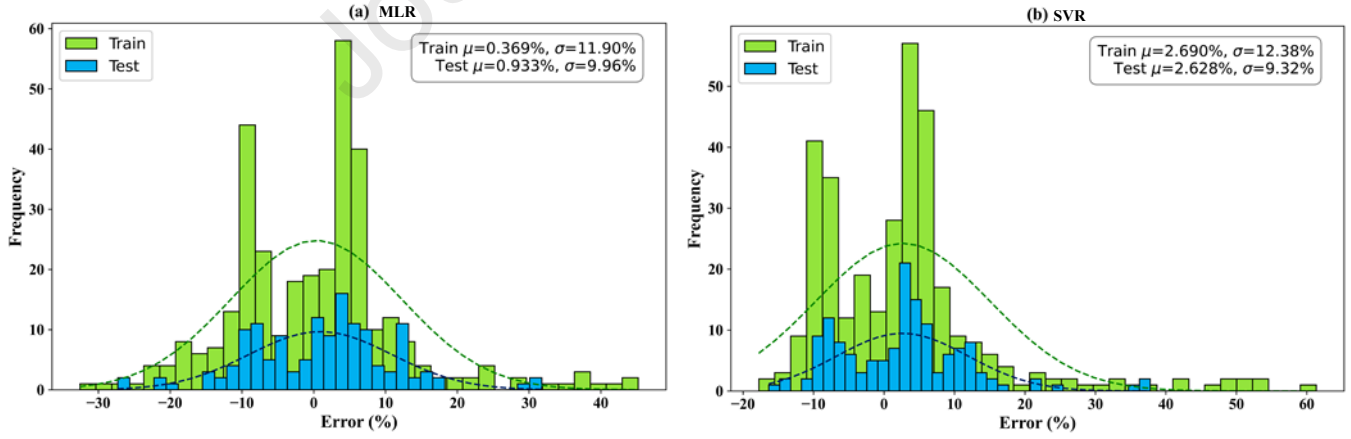
Fig. 8. Scatter plots between actual and predicted values in the training and testing stages based on (a) MLR, (b) SVR, (c) GEP, (d) ANN, (e) RF, (f) AdaBoost, (g) XGBoost, and (h) CatBoost models.

3.2.3. Error histograms

Fig. 9 illustrates the distribution of error histograms across the training and testing stages for the developed models. Error histograms are crucial in visual evaluation as they provide a clear overview of the distribution and spread of prediction errors, allowing for quick assessment of model accuracy, bias, and variance. By visualizing how errors are concentrated around zero, they help identify overfitting, underfitting, and the model's generalization ability. In the training stage, non-ensemble models, display varying degrees of error. MLR and SVR have higher mean errors (μ) and standard deviations (σ), with MLR showing a mean value of 0.369% and an SD of 11.90%, while SVR has a

mean of 2.690% and an SD of 12.38%. These values indicate a considerable spread in their error distributions, suggesting a less accurate fit to the training data. GEP model improves upon this with a lower mean error (μ = 1.238% and σ = 8.52%), while ANN further enhances the performance with a mean error close to zero (μ = -0.052% and σ = 2.18%), indicating a well-fitted model. Ensemble models, particularly RF, XGBoost, and CatBoost outperform the non-ensemble models, with XGBoost (μ = 0.058% and σ = 1.38%) and CatBoost (μ = 0.016% and σ = 0.92%) showing exceptionally tight and concentrated error distributions around zero. This suggests that these ensemble models are highly effective in capturing the complexities of the training data with minimal variance.

On the other hand, during the testing stage, the performance of the non-ensemble models generally declines. MLR shows a mean error of 0.933% and an SD of 9.96%, reflecting limited generalization capabilities. SVR's error increases significantly with a mean of 2.628% and an SD of 9.32%, indicating potential overfitting. GEP shows moderate performance with a mean error of 2.124% and an SD of 6.83%, while ANN performs better with a near-zero mean (μ = 0.010%) and a low SD (σ = 2.28%), indicating a relatively strong generalization ability. Among the ensemble models, XGBoost and CatBoost stand out for their robust generalization. XGBoost maintains a near-zero mean error (μ = 0.013%) with a low SD (σ = 2.52%), while CatBoost similarly exhibits a mean error (μ = 0.208%) and SD (σ = 2.19%), demonstrating highly accurate predictions on unseen data with minimal variance. These results highlight the superior generalization capabilities of ensemble methods, especially XGBoost and CatBoost. In summary, the ensemble models outperform the non-ensemble models in both the training and testing stages, with CatBoost and XGBoost standing out due to their consistently low error rates and tight error distributions. Non-ensemble models like MLR, SVR, and GEP show higher variability and less precise predictions. Among these, ANN performs the best, showing a concentrated error distribution with a near-zero mean during both the training and testing stages. However, the superior performance of ensemble models suggests they are more effective in capturing complex patterns in the data and generalizing them to unseen instances. Therefore, for tasks requiring high predictive accuracy, ensemble models, especially CatBoost and XGBoost, are recommended.



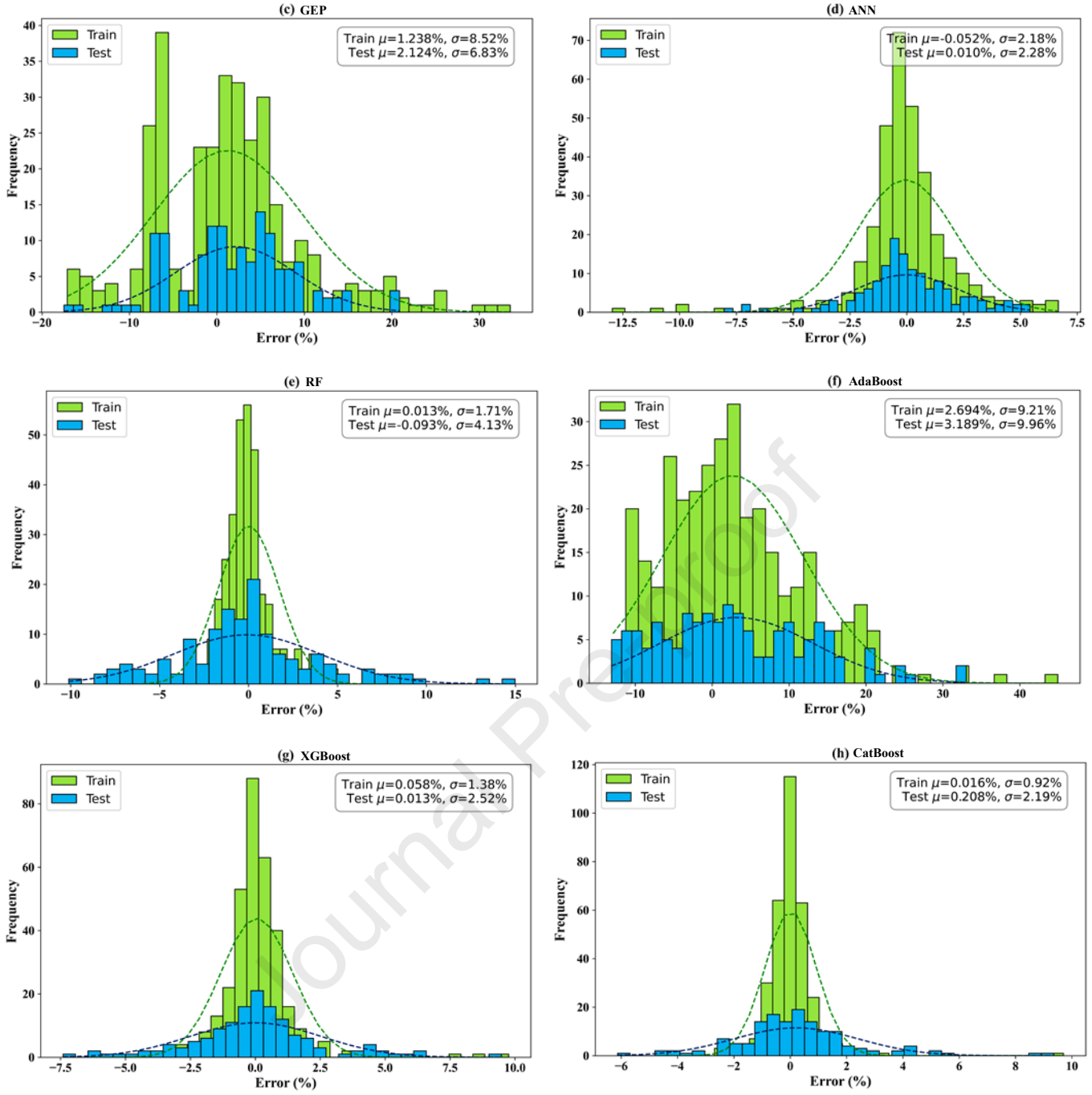


Fig. 9. Distribution of error histograms during the training and testing stages based on (a) MLR, (b) SVR, (c) GEP, (d) ANN, (e) RF, (f) AdaBoost, (g) XGBoost, and (h) CatBoost models.

3.2.4. Violin boxplots

Fig. 10 presents violin plots for the relative discharge ratio across the training and testing datasets for each adopted model. Violin plots are used to visualize the distribution of the data and include a marker for the median value, offering a detailed view of the distribution's shape, spread, and skewness. In Fig. 10, each model's distribution is compared against the actual distribution of the relative discharge ratio. The median values are displayed within the plots, providing a quick reference for central tendencies. During the training stage (Fig. 10a), the actual mean discharge ratio was 0.0905, serving as the baseline for evaluating the model performances. Among the various models applied, XGBoost achieved the closest mean discharge ratio to the actual value, recording 0.0903, with a very small difference of only 0.0002. This minimal deviation highlights the model's high accuracy in predicting the discharge ratio during training. RF model was another top performer, with a mean of 0.0908, differing by just 0.0003 from the

actual value, making it a reliable predictive model. CatBoost also demonstrated strong accuracy, producing a mean discharge ratio of 0.0899, which resulted in a difference of 0.0006 from the actual data. On the other hand, models like MLR and SVR showed larger discrepancies, with differences of 0.0035 and 0.0036, respectively, indicating that these models were less accurate in the training stage.

In the testing stage (Fig. 10b), the actual mean discharge ratio was 0.0973, and this value was used to evaluate the model performances under unseen data conditions. RF model outperformed all other models during this stage, with a mean discharge ratio of 0.0972, producing an almost negligible difference of 0.0001 from the actual value, indicating excellent generalization ability. Both ANN and AdaBoost also performed well, each with a mean discharge ratio of 0.0970, resulting in a small difference of 0.0003 from the actual value. These models demonstrated reliable accuracy in testing, closely approximating the actual discharge ratio. XGBoost and CatBoost also maintained strong performances, with differences of 0.0005 and 0.0004, respectively, though slightly less accurate compared to RF. This consistency in predictive ability across various models reflects their robustness, with RF standing out as the best model for the testing phase. In conclusion, combining outstanding generalization during testing with great predictive power in training, XGBoost, RF, and CatBoost are the recommended choices for this dataset.

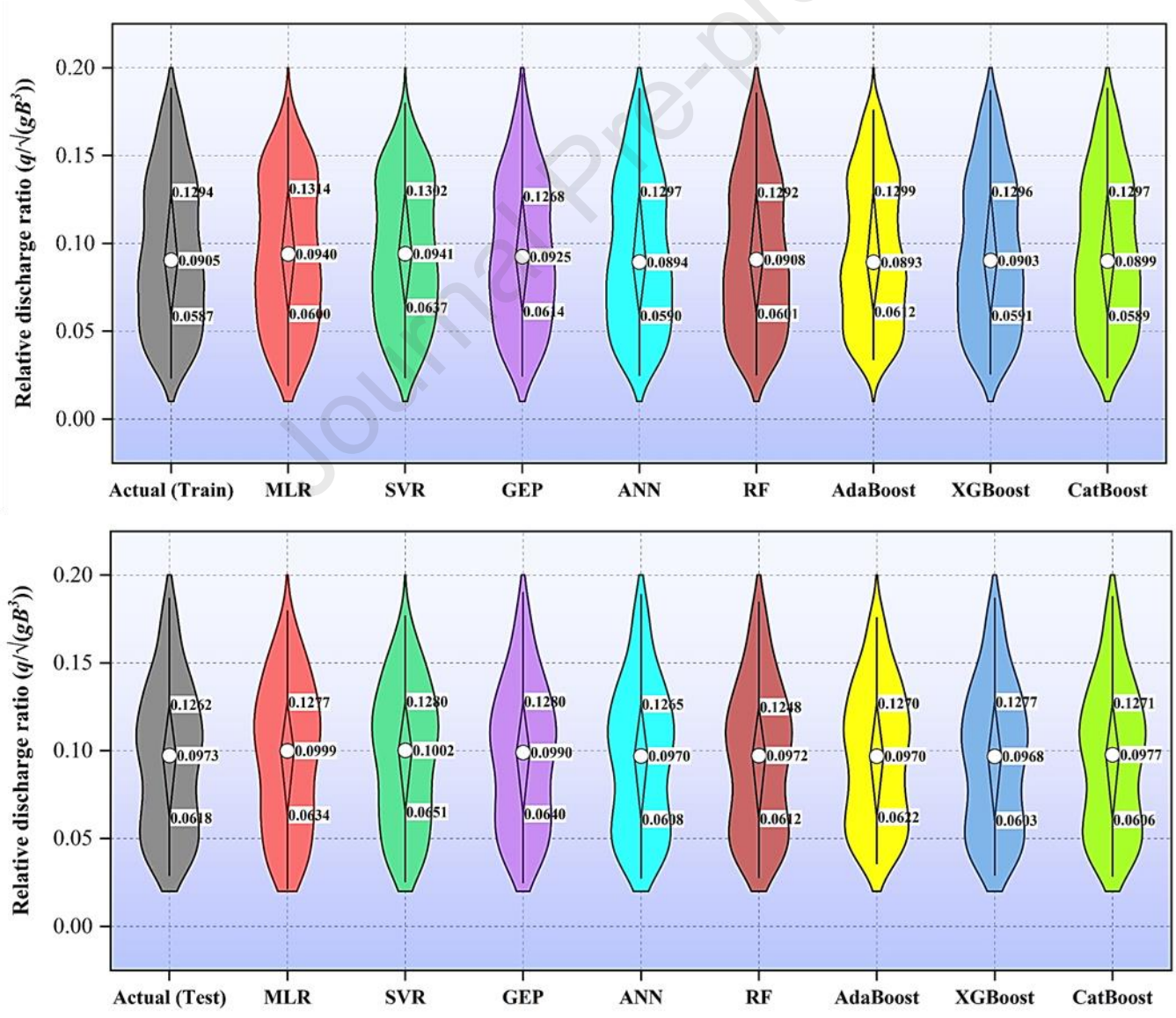


Fig. 10. Violin box plots for the adopted models during (a) training and (b) testing stages.

3.2.5. Taylor diagrams

The Taylor diagrams depicted in Fig. 11 offer a specialized visual comparison of the performance of different models using three key statistical metrics: Standard Deviation (SD), centered Root-Mean-Square-Difference (RMSD), and correlation coefficient (CC). These diagrams provide a concise overview of how accurately different models replicate the behavior of a reference dataset, specifically the actual training data. The radial axis on the diagram represents the SD, showing how the variability of the models compares to that of the actual training data. The angular axis represents the CC, which indicates the strength of the linear relationship between the model predictions and the actual data. The distance from the actual data point along the arc illustrates the centered RMSD, where shorter distances imply better model performance.

In the training stage (Fig. 11a), the actual dataset had an SD of 0.04204 and a variance of 0.00177. Across the models evaluated, CatBoost achieved the highest accuracy, with a CC of 0.99993, indicating an almost perfect linear relationship with the actual training data. Furthermore, it had the lowest centered RMSD of 4.96×10^{-4}, signifying that its predictions were closest to the actual values with minimal error. XGBoost followed closely, demonstrating a strong performance with a CC of 0.99985 and a centered RMSD of 7.26×10^{-4}, indicating its high predictive capability. RF model, with a CC of 0.99954 and an RMSD of 0.00128, performed well but remained slightly behind CatBoost and XGBoost in terms of overall accuracy. ANN also showed excellent accuracy, with a CC of 0.99955 and a centered RMSD of 0.00127, making it one of the top-performing models. By contrast, MLR and SVR showed lower accuracy, with CC values of 0.98064 and 0.98039, and larger RMSD values of 0.00822 and 0.00848, respectively. This suggests these models were less effective at capturing the actual data's behavior.

During the testing stage (Fig. 11b), the actual dataset had an SD of 0.04066 and a variance of 0.00165. Once again, CatBoost demonstrated the best performance, achieving the highest CC of 0.99913 and the lowest centered RMSD of 1.69×10^{-3}, indicating its superior generalization ability when applied to unseen data. XGBoost performed similarly well, with a CC of 0.9987 and a centered RMSD of 2.07×10^{-3}, confirming its robustness across different datasets. ANN also maintained its strong performance, achieving a CC of 0.99904 and a centered RMSD of 0.00177, proving its effectiveness in the testing phase. RF, with a CC of 0.99553 and an RMSD of 0.00383, ranked among the more accurate models but still trailed behind CatBoost, XGBoost, and ANN in terms of overall accuracy. In contrast, MLR and SVR once again showed weaker performance compared to the top models, with CCs of 0.98073 and 0.98134, and higher centered RMSD values of 0.00793 and 0.00789, respectively, indicating that they were less capable of replicating the actual test data. Overall, CatBoost stands out as the best model, with XGBoost and ANN also delivering highly accurate results.

Based upon all the aforementioned implemented visual techniques, the superior performance of ensemble models such as XGBoost, CatBoost, and RF compared to non-ensemble models like MLR and SVR can be attributed to their ability to iteratively improve predictions by combining multiple weak learners. This process enhances the overall accuracy, robustness, and generalization of the models, leading to better predictive performance in both the training and testing phases. In contrast, non-ensemble models lack this error-correction mechanism, resulting in lower predictive accuracy and an inability to handle complex data patterns as effectively.

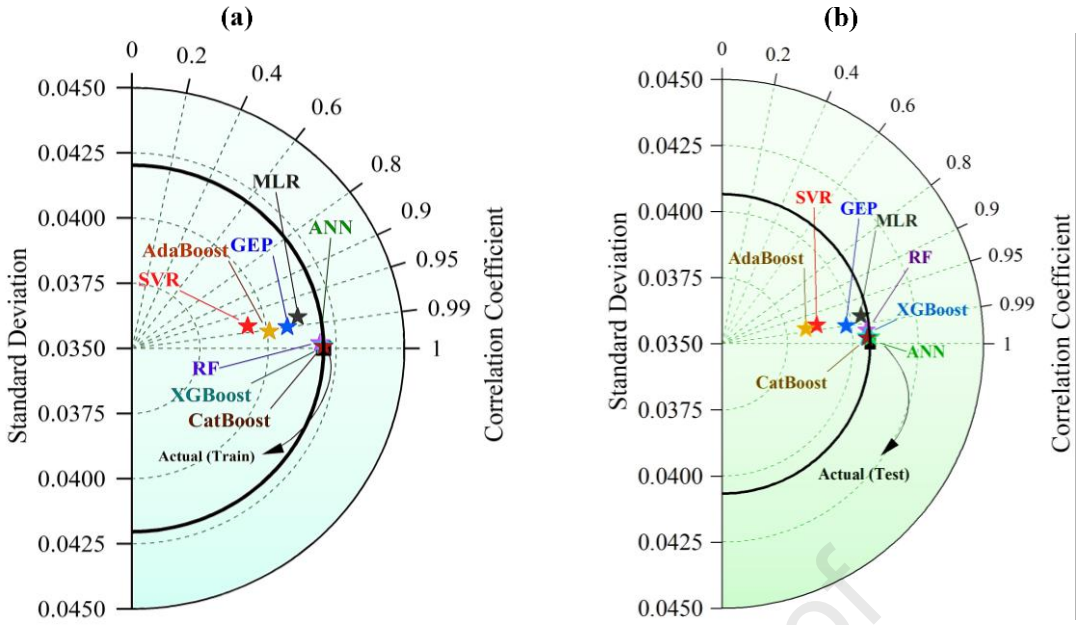


Fig. 11. Taylor diagrams for the adopted models during the (a) training and (b) testing stages.

3.3. Evaluating model Performance: A Quantitative-Based Approach

3.3.1. AIC analysis

Table 4 provides the AIC analysis results including the applied criteria for estimating the number of parameters (k) for each model. The AIC is a statistical measure used to compare different models and assess their relative quality based on their fit to the data and complexity. It balances the goodness of fit with the model's complexity, where a lower AIC value indicates a better model [64]. Negative AIC values generally signify that the model provides a better fit to the data, whereas positive values indicate a poorer fit [65]. The AIC calculation results in Table 4 indicate that the MLR model has 4 parameters and an AIC value of -3189.62. Despite its simplicity and a relatively small number of parameters, the AIC value is higher compared to some other models, suggesting that while MLR may be straightforward, it does not fit the data as effectively as other models. The SVR model has 186 parameters and an AIC value of -5975.15. This model offers a better fit than the MLR model, with its larger number of parameters allowing for a more complex representation of the data, although it does not achieve the best fit among all models. ANN model, with 81 parameters, has an AIC value of -6769.62. This model provides a strong fit to the data, outperforming both MLR and SVR, indicating a good balance between complexity and fit. The GEP model has 67 parameters and an AIC value of -6655.62. With a slightly smaller parameter count than ANN, the GEP model demonstrates a robust fit to the data, reflecting a well-balanced approach in modeling.

In contrast, for non-ensemble models, the RF model has a significantly higher parameter count of 15,504, with an AIC value of 22132.75. This model's extremely high parameter count results in a very poor AIC value, suggesting that its complexity leads to a less effective fit to the data. The AdaBoost model, with 1000 parameters, results in an AIC value of -4877.16. Although it manages a decent fit, it is not as strong as ANN or GEP, indicating that while it handles complexity, it does not fit the data as effectively as some other models. The XGBoost model, with 11,206 parameters, yields an AIC value of 12785.15. Despite its considerable parameter count, the model does not provide as strong a fit to the data as the best-performing models. The CatBoost model, with only 12 parameters, achieves the lowest AIC value of -10110.28. This model stands out as the best fit among those listed, as indicated by its very low

AIC value. The combination of a minimal number of parameters and an exceptionally low AIC value suggests that the CatBoost model is the most effective in balancing fit and complexity. In summary, the CatBoost model is identified as the best model based on its AIC analysis, signifying the most effective balance between model complexity and fit to the data.

Table 4
AIC analysis results for the developed models.

Category	Model	Equations for estimating *k*	No. of (*k*)	AIC value
Non-ensemble	MLR	n_inputs + intercept (constant)	4	-3189.62
	SVR	n_support_vectors + n_hyperparameters	186	-5975.15
	GEP	program_size + n_function_set_parameters + n_literals_and_inputs	67	-6655.62
	ANN	(input_size × hidden_layer_size) + (hidden_layer_size × output_size) + hidden_layer_size + output_size	81	-6769.62
Ensemble	RF	n_estimators × (max_depth + 1)	15,504	22132.75
	AdaBoost	n_estimators	1000	-4877.16
	XGBoost	n_estimators × (max_depth + 1)	11,206	12785.15
	CatBoost	n_inputs × depth	12	-10110.28

3.3.2. Statistical analysis

Expanding upon the earlier visual examination of scatter plots, a rank analysis was performed to assess the performance of the generated models based on their regression metrics in both the training and testing stages. For each metric in each stage, a model with a rank value of 1 indicates the most optimal performance, while a model value of 8 indicates the least. The overall ranking of each model is determined by summing up the individual metrics in both stages. The model with the highest total rank is considered the least effective, whereas the model with the lowest is considered the most effective. Table 5 presents the rank analysis results for both the training and testing stages. It is important to note that the rank numbers are displayed within parentheses. During the training stage, the CatBoost model excels as the top performer, attaining a total score of 5 by consistently leading all regression metrics. This demonstrates its superior precision and minimizes errors during training. The XGBoost model is closely behind with a total score of 10, placing it in second place overall and demonstrating robust performance. RF and ANN models achieve the third and fourth spots, respectively, with scores of 15 and 20, indicating their strong predictive abilities. The GEP and AdaBoost techniques are ranked fifth and sixth, respectively, with scores of 27 and 28, indicating a moderate level of effectiveness. MLR and SVR secure the seventh and eighth positions, respectively, with scores of 36 and 39, indicating the lowest level of effectiveness among the models during the training process.

Throughout the testing stage, CatBoost consistently exhibits outstanding performance, obtaining the lowest overall score of 5. The consistency observed throughout the stages of training and testing underscores the robustness and reliability of the CatBoost model. With a score of 12, the ANN model achieves a strong second and immediately third position in the training stage, demonstrating its reliable performance in real-world testing situations. Based on a total score of 13, the XGBoost model ranked third in the testing stage, compared to second in the training stage. The RF model maintains its fourth-place position in the testing stage, mirroring its performance in the training stage, obtaining a score of 20. Once again, the GEP and AdaBoost models achieved rankings of fifth and sixth, respectively, with scores of 25 and 34. Notably, the performance of the AdaBoost model saw a substantial decline. Furthermore, the MLR and SVR models continue to exhibit the lowest effectiveness, scoring 35 and 36, respectively. These scores align with their training performance, so underscoring the notable difficulties in their predictive accuracy.

In summary, ensemble models, particularly CatBoost and XGBoost, demonstrate the best overall performance,

achieving the lowest overall ranks of 10 and 23, respectively, indicating strong predictive power and generalization. ANN also performed exceptionally well with an overall rank of 32, being the best among non-ensemble models. Meanwhile, traditional models like MLR and SVR show comparatively weaker performance, with higher overall ranks, which indicates their limitations in capturing the underlying patterns in the dataset.

Table 5

Rank analysis for the developed models.

Category	Model	Stage	R^2	RMSE	RMSRE	MAE	MAPE (%)	Total Score	Overall Rank
Non-ensemble	**MLR**	**Train**	0.962 (7)	0.008 (7)	0.119 (7)	0.007 (7)	9.0 (8)	36	71
		Test	0.961 (7)	0.008 (7)	0.100 (7)	0.007 (7)	7.9 (7)	35	
	SVR	**Train**	0.959 (8)	0.009 (8)	0.127 (8)	0.007 (8)	8.7 (7)	39	75
		Test	0.961 (8)	0.008 (8)	0.097 (6)	0.007 (8)	7.4 (6)	36	
	GEP	**Train**	0.979 (6)	0.006 (6)	0.086 (5)	0.005 (5)	6.4 (5)	27	52
		Test	0.978 (5)	0.006 (5)	0.072 (5)	0.005 (5)	5.6 (5)	25	
	ANN	**Train**	0.999 (4)	0.001 (4)	0.022 (4)	0.001 (4)	1.4 (4)	20	32
		Test	0.998 (2)	0.002 (3)	0.023 (2)	0.001 (3)	1.6 (2)	12	
Ensemble	**RF**	**Train**	0.999 (3)	0.001 (3)	0.017 (3)	0.001 (3)	1.1 (3)	15	35
		Test	0.991 (4)	0.004 (4)	0.041 (4)	0.003 (4)	3.0 (4)	20	
	AdaBoost	**Train**	0.981 (5)	0.006 (5)	0.096 (6)	0.005 (6)	7.2 (6)	28	62
		Test	0.967 (6)	0.007 (6)	0.105 (8)	0.006 (6)	8.2 (8)	34	
	XGBoost	**Train**	1.000 (2)	0.001 (2)	0.014 (2)	0.001 (2)	0.8 (2)	10	23
		Test	0.997 (3)	0.002 (2)	0.025 (3)	0.001 (2)	1.8 (3)	13	
	CatBoost	**Train**	1.000 (1)	0.001 (1)	0.009 (1)	0.001 (1)	0.5 (1)	5	10
		Test	0.998 (1)	0.002 (1)	0.022 (1)	0.001 (1)	1.5 (1)	5	

3.3.3. Uncertainty analysis

Fig. 12 presents the uncertainty analysis for various developed models during training and testing stages. In the training stage, CatBoost exhibits the lowest uncertainty (0.00138), followed by XGBoost (0.00201), indicating that these models perform with high reliability on the training data. ANN and RF also show low uncertainty during training, but RF experiences a significant increase in uncertainty during the test stage (0.01063), suggesting overfitting. In the testing stage, CatBoost remains the most consistent with a low uncertainty of 0.00471, making it the most reliable model overall. XGBoost also performs well with slightly higher test uncertainty (0.00574), while ANN shows a small increase in uncertainty during testing (0.00493) but remains competitive. Models like GEP, MLR, SVR, and AdaBoost exhibit higher uncertainties in both stages, indicating less reliable performance compared to CatBoost and XGBoost. In general, CatBoost is the best-performing model with the lowest uncertainty across both stages, making it the most reliable. XGBoost follows closely behind, showing strong generalization, while ANN and

RF perform well, but its slight increase in test uncertainty suggests room for improvement in generalization. Other models exhibit moderate performance with higher uncertainty throughout.

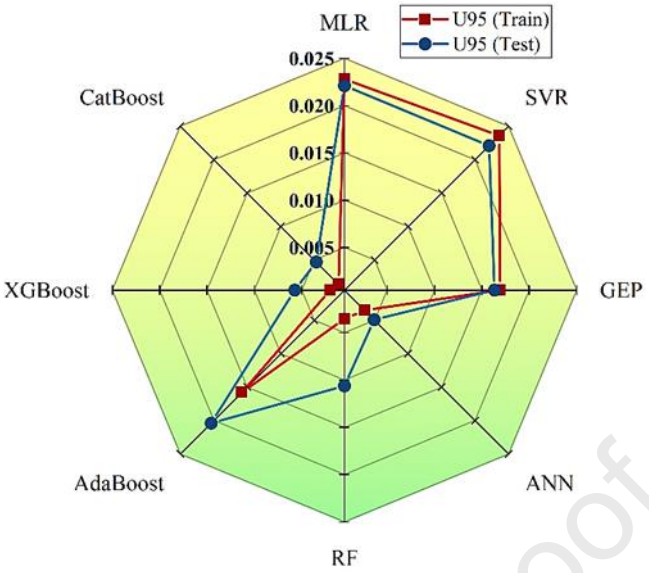


Fig. 12. Performance of the adopted models based on U95 value for uncertainty analysis.

3.4. Performance of Models comparison across different flow scenarios

The design of PKWs is primarily focused on maintaining high discharge efficiency across various upstream headwater conditions, making the understanding of the headwater ratio essential for their effective implementation. In this context, the total upstream head ratio (H/W_u) is defined as the relationship between the upstream water level height and the cycle width of the PKW. Typically, this ratio is categorized into lower and higher heads to analyze how varying upstream water levels influence the hydraulic performance of the PKW. At lower heads, the flow over the PKW is more stable and efficient, enabling effective water discharge with minimal energy loss. Conversely, at higher headwater ratios, the flow becomes increasingly turbulent, which can reduce the discharge efficiency and challenge the weir's capacity to manage extreme flow conditions safely. Therefore, this classification is crucial for optimizing the design and functionality of PKWs across different flow scenarios. Table 6 presents a comparative analysis of the models employed, considering two data ranges for the H/W_u: lower heads ($0.15 \leq H/W_u \leq 0.43$) and higher heads ($0.43 < H/W_u \leq 0.72$). Based on the Min, Max, and Mean values of the current dataset shown in Table 1, this classification was established.

In the lower head scenario, ensemble models clearly outperform non-ensemble models in terms of both R^2 and MAPE. Among the non-ensemble models, the ANN model stands out with an R^2 value of 0.996 and a MAPE of 1.9%, showcasing its superior predictive power compared to the other non-ensemble methods. The GEP model follows with an R^2 value of 0.939 and a MAPE of 8.4%, while the SVR and MLR models show lower performance, with R^2 values of 0.889 and 0.894 and higher MAPE values of 10.5% and 11.1%, respectively. On the other hand, the ensemble models demonstrate remarkable accuracy, with CatBoost achieving the highest R^2 value of 0.998 and the lowest MAPE of 1.1%, indicating its ability to provide precise predictions. XGBoost and RF also show strong performance with R^2 values of 0.997 and 0.992, and low MAPE values of 1.5% and 2.0%, respectively. Although the AdaBoost model does not perform as well as the other ensemble methods, with an R^2 value of 0.912 and a MAPE of 11.3%, it still surpasses most of the non-ensemble models except ANN. Overall, CatBoost emerges as the best-performing model for the lower head scenario, followed closely by XGBoost and RF, while ANN is the top non-

ensemble model.

In the higher head scenario, the ensemble models again outperform the non-ensemble models, although the performance gap is narrower compared to the lower head scenario. Among the non-ensemble models, ANN remains the best performer with an R^2 value of 0.996 and a MAPE of 0.9%, indicating its robustness across different flow conditions. GEP shows improved performance in this scenario with an R^2 value of 0.939 and a much lower MAPE of 3.9%, compared to its performance in the lower head scenario. The SVR and MLR models continue to lag behind, with R^2 values of 0.899 and 0.894, and MAPE values of 6.0% and 6.3%, respectively, showing limited predictive capability under higher flow conditions. Among the ensemble models, CatBoost again leads with an R^2 value of 0.998 and an impressively low MAPE of 0.6%, confirming its consistency and accuracy. XGBoost also maintains high performance with an R^2 value of 0.997 and a MAPE of 0.7%. RF shows a slight decrease in performance compared to the lower head scenario, with an R^2 value of 0.989 and a MAPE of 1.4%, but it still outperforms the non-ensemble models. AdaBoost improves in this scenario, achieving an R^2 value of 0.957 and a MAPE of 3.8%, making it more competitive. In summary, for the higher head scenario, CatBoost and XGBoost continue to be the top models, demonstrating exceptional predictive accuracy, while ANN remains the best among non-ensemble models.

Table 6

Estimated performance of the implemented models at different flow scenarios.

Category	Model	Lower head ($0.15 \leq H/W_u \leq 0.43$)		Higher head ($0.43 < H/W_u \leq 0.72$)	
		R^2	MAPE (%)	R^2	MAPE (%)
Non-ensemble	MLR	0.894	11.1	0.894	6.3
	SVR	0.889	10.5	0.899	6.0
	GEP	0.939	8.4	0.939	3.9
	ANN	0.996	1.9	0.996	0.9
Ensemble	RF	0.992	2.0	0.989	1.4
	AdaBoost	0.912	11.3	0.957	3.8
	XGBoost	0.997	1.5	0.997	0.7
	CatBoost*	**0.998**	**1.1**	**0.998**	**0.6**

* The bold values indicated the best predictive model across different flow scenarios.

3.5. Feature Importance and Interpretability

3.5.1. SHAP analysis

The purpose of sensitivity analysis is to conduct a more thorough examination of data types and determine the importance of each input parameter to the output [66]. In light of the aforementioned visual and quantitative evaluations, the CatBoost model emerged as the most efficient predictive model. It will be utilized to produce the SHAP feature importance results in the testing stage. The findings derived from the model are illustrated in Fig. 13, which includes a variety of dependency charts. Fig. 13a shows the summary dot plot that provides an extensive visualization of feature importance across all instances. Each dot represents a SHAP value for a specific instance, with the color indicating the feature value. The inputs are ranked vertically based on their average impact on the relative discharge ratio of type-A PKW, revealing that the X3 (H/W_u) is the most significant feature. This is evident from the widest range spread of SHAP values ranging from approximately -0.06 to 0.06 and the clear separation of colors, indicating a substantial and consistent influence on the predictions. The inputs X2 (P/W_u) and X1 (W_i/W_o) follow in importance, with X2 showing a more considerable impact than X1, as indicated by a larger spread and more distinct color variation. Fig. 13b shows the summary bar plot that simplifies the information from the dot plot by ranking the features based on their mean absolute SHAP values. This plot confirms the ranking observed in the dot

plot, with the X3 at the top, followed by X2 and X1. The length of the bars in this plot directly correlates with the importance of each feature, solidifying the total upstream head ratio as the most influential, with PKW height ratio and PKW key widths ratio being less so, but still significant. Fig. 13c illustrates the waterfall plot of individual features based on the CatBoost model. A Waterfall Chart is often used in data analytics, in which each step represents how different variables contribute to an overall outcome, denoted here as f (x). The red bars and the blue bars represent positive (additive) and negative (subtractive) contributions to the overall f (x), respectively. X3 is the most influential feature, with a significant positive SHAP contribution of +0.02. X2 also plays a key role, adding +0.01 to the predictions. Meanwhile, the input X1 has a very small negative contribution.

SHAP force plot (Fig. 13d) provides a detailed breakdown of how individual features contribute to a specific prediction, with a base value starting at 0. In this instance, X2 has the most significant positive impact, with a SHAP value of 1.33, indicating that it strongly pushes the model's output higher. X3 also contributes positively, with a SHAP value of 0.516, further increasing the prediction. On the other hand, X1 has a smaller negative impact with a SHAP value of 0.78, slightly decreasing the final prediction. The cumulative effect of these SHAP values leads to the final model output of 0.03 for this specific instance. This plot clearly illustrates how the inputs P/W_u and H/W_u are the dominant forces driving the prediction upwards, while the input W_i/W_o acts as a minor counteracting influence, providing a transparent view of the model's reasoning for this prediction. In addition, the SHAP heatmap in Fig. 13e offers a more detailed view of how SHAP values vary across different instances for each feature. Each row represents a feature, and each column corresponds to an instance, with the color intensity indicating the magnitude of the SHAP values. Red shades signify a positive contribution, pushing the model's output higher, while blue shades indicate a negative contribution. The SHAP heatmap reveals that X3 consistently exerts a strong influence on the model's predictions, as indicated by the intense red and blue colors across most instances. In contrast, X1 appears to have a more subdued and balanced influence, with frequent white areas indicating minimal impact on the model's predictions. The mixed color patterns for X2 suggest its impact fluctuates across different instances, hinting at complex interactions or non-linear effects. Moving to Fig. 13f, the river flow plot visualizes the cumulative effect of features on the model's prediction for a specific instance. This plot highlights how P/W_u and H/W_u predominantly contribute to the final prediction, with H/W_u often having a larger segment, showing its higher impact. While the W_i/W_o present typically exhibits a smaller segment, signifying a less significant yet still crucial contribution. The flow of the segments illustrates the net positive or negative contribution of each feature, reinforcing the ranking seen in previous plots.

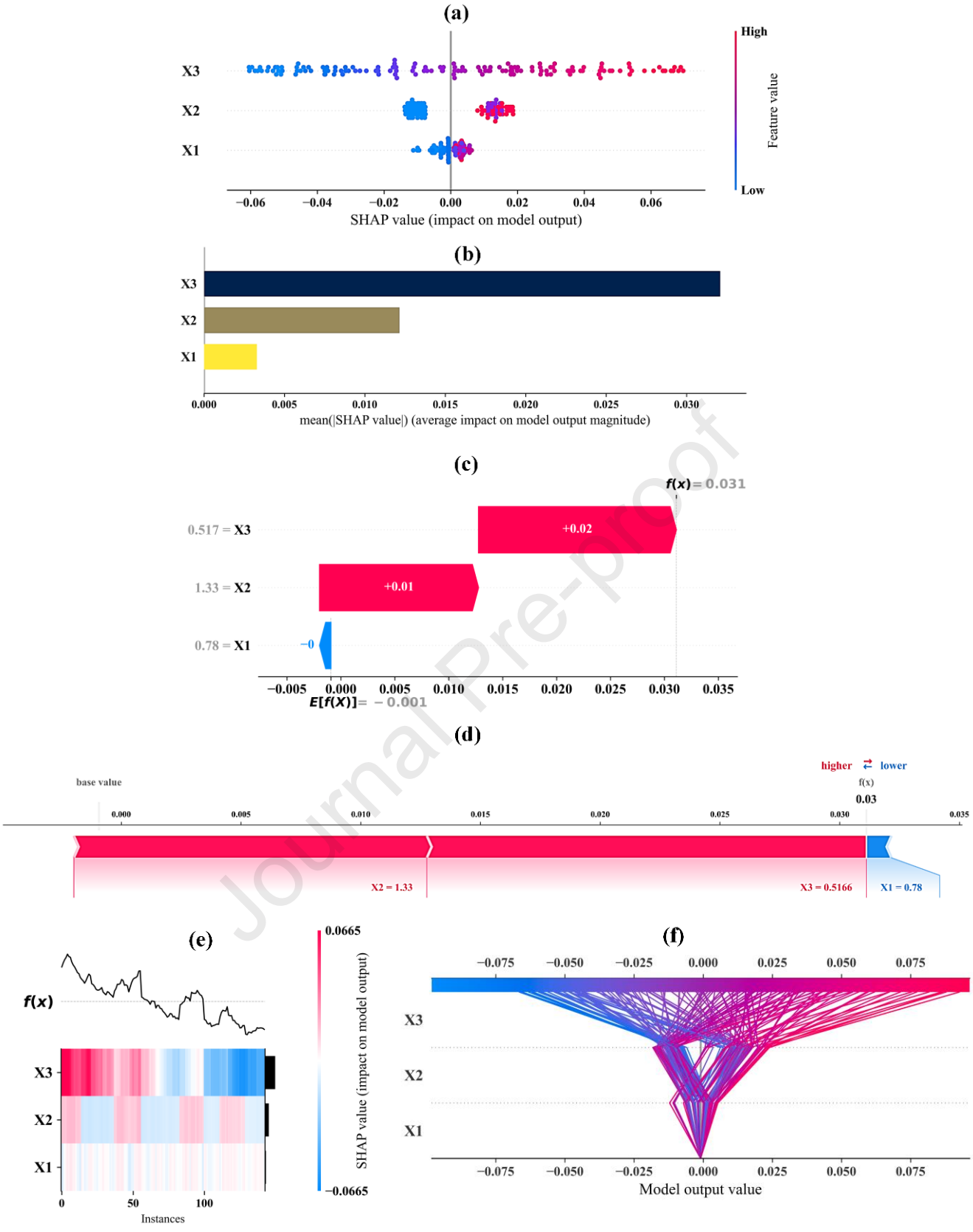


Fig. 13. SHAP feature importance analysis based on the CatBoost model in the testing stage: (a) summary dot plot, (b) summary bar plot, (c) waterfall plot (d) force plot, (e) heatmap, and (f) river flow.

3.5.2. PDP analysis

Fig. 14 presents PDPs for the studied inputs based on the CatBoost model. These plots help to understand the individual impact of each input on the model's predictions by depicting the average change in the predicted relative

discharge ratio when varying the input value while keeping all other inputs constant. Fig. 14a shows the partial dependence of the feature X1. As X1 increases from approximately 0.5 to 2.0, the model's predicted outcome also increases, suggesting a positive relationship between W_i/W_o and the relative discharge. However, after W_i/W_o reaches around 2.0, there is a slight decrease in the predicted outcome, indicating a potential threshold or diminishing returns in how W_i/W_o influences the discharge. This implies that as the W_i/W_o ratio increases, the inlet cross-section increases, increasing PKW discharge. However, when W_i/W_o increases, the outlet key becomes too narrow because $W_i + W_o$ = constant. It causes significant interference between descending lateral nappes from adjacent side crests in the outlet key. This significantly submerges the PKW crest and reduces discharge efficiency [67]. Fig. 14b depicts the partial dependence of X2. The plot shows at the beginning of the curve (X2 ≈ 0.6 to 0.9), there is a steep rise in partial dependence, indicating that an increase in P/W_u significantly enhances the discharge capacity. This is because increasing the height of the PKW (P) allows it to handle a greater volume of water flow [68]. The additional height increases the potential energy of the water, resulting in higher flow discharge. Beyond a certain height ratio (X2 ≈ 0.9 to 1.2), the curve flattens, showing that further increases in PKW height have a diminishing effect on discharge. This plateau indicates that after a certain point, the discharge is less sensitive to changes in weir height. This can occur due to several factors, such as the weir reaching a submergence condition where increasing the height no longer significantly increases the head difference or due to flow constraints like turbulence and energy dissipation that limit the discharge capacity [69].

Fig. 14c shows the partial dependence of X3. The plot reveals a clear positive relationship between H/W_u and the relative discharge of type-A PKW. The increase is gradual, with no indication of diminishing returns, highlighting the significance of H/W_u as a key feature in driving the model's predictions. This trend is expected because of an increase in upstream head, the potential energy of the water body above the weir increases. This added energy contributes to a greater flow velocity over the weir crest, thereby enhancing the discharge of PKW [70]. Overall, the PDP analysis in Fig. 14 indicates that all studied features positively influence the discharge's predictions, but to varying degrees and with different patterns. W_i/W_o shows a generally positive relationship with a slight decline at higher values (i.e., $W_i/W_o > 2.0$), P/W_u exhibits a linear positive relationship, and H/W_u has a strong and consistent positive impact throughout its range.

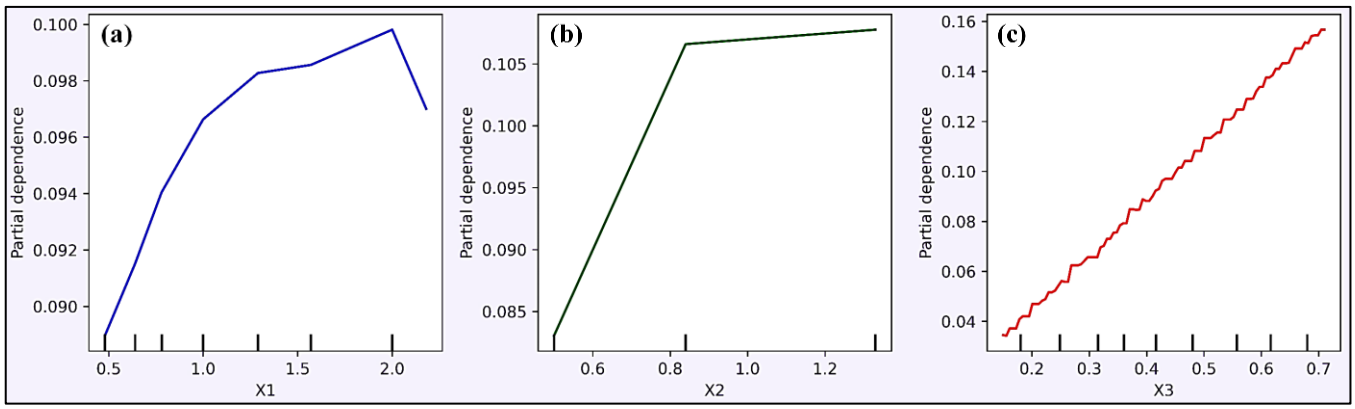


Fig. 14. Dependence PDP analysis results for inputs: (a) W_i/W_o, (b) P/W_u, and (c) H/W_u.

3.6. Interactive GUI for prediction

This section introduces a substantial development that addresses the real-world needs of engineers and designers in the efficient utilization of ML models. Despite the traditional impediment of the seamless integration of ML into

everyday design tasks from the intricate processes of database compilation, model training, and testing, an innovative solution has been developed. A Python web application has been created featuring a model with optimized hyperparameters accessible through a GUI tool. This GUI is specifically designed to predict the relative discharge of type-A PKW, as shown in Fig. 15. The GUI presents a streamlined layout where users can enter values for the input variables. The calculated output (relative discharge value) is dynamically displayed upon inserting these inputs. To facilitate wider access and foster collaborative improvements, the GUI was and has been hosted on GitHub, making it readily available for use and further development by the community. This not only democratizes the use of advanced predictive models but also invites contributions to refine the tool and adapt it to various specific needs within the civil and hydraulic engineering field. Finally, the GUI can be accessed via the URL link provided in Appendix C.

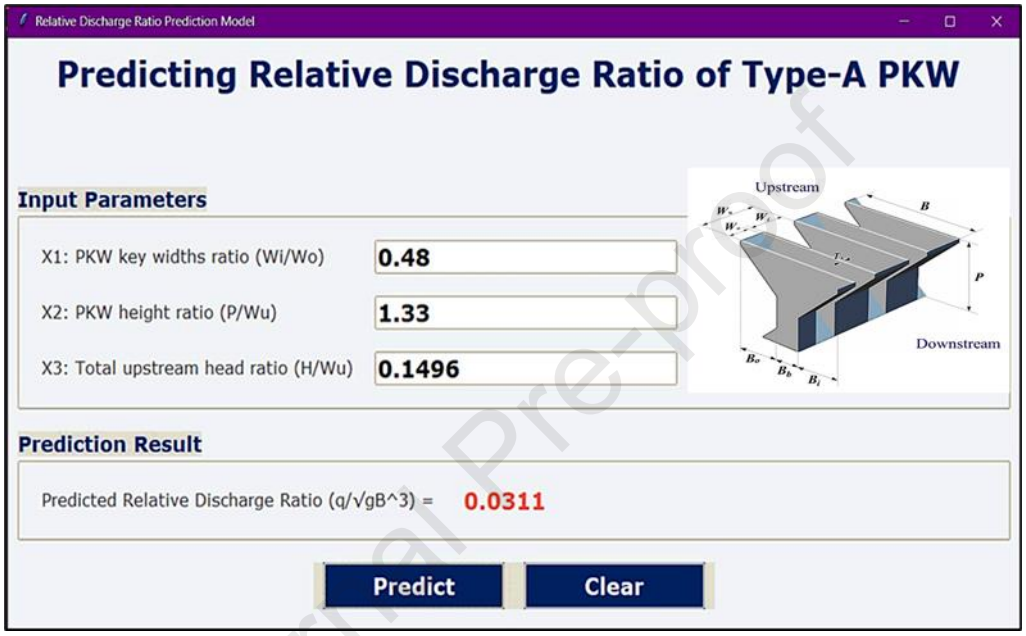


Fig. 15. GUI screenshot for predicting the relative discharge ratio of type-A PKW.

4. Conclusions

This study involved a thorough examination of 476 experimental datasets related to discharge over type-A PKWs. The data was obtained from prior research published in journals. The primary goal of the research was to assess and compare the effectiveness of eight machine learning models in predicting the relative discharge of type-A PKW. These models consist of four non-ensemble models (MLR, SVR, GEP, and ANN) and four ensemble models (RF, AdaBoost, XGBoost, and CatBoost). Three dimensionless input variables were implemented: the total upstream head, the width of the PKW key, and the height of the PKW. The results obtained from the models were compared visually and quantitatively methods to assess their predictive capabilities. The following key findings emerged from this study:

1) All the models that have been developed achieve R^2 values that exceed 0.96 during the training and testing stages, which provides a compelling argument in Favor of the use of these models in the prediction of the relative discharge of type-A PKW.
2) The CatBoost model demonstrated the highest accuracy in prediction with an R^2-value of 0.998, RMSE of 0.002, and MAPE of 1.50 %, outperforming other models during the testing stage.
3) The XGBoost model was ranked second, and the ANN model was ranked behind it. Moderate performance is demonstrated by the RF and GEP models. However, the MLR and SVR models exhibit significantly lower

performance than the other models, as demonstrated by the highest error metrics.

4) Across low and high-flow scenarios, ensemble models, particularly CatBoost and XGBoost, demonstrate a clear advantage over non-ensemble models. While ANN was the most effective model in both scenarios among the non-ensemble models.
5) Based on the SHAP and PDP analyses, the most important factor was the total upstream head, followed by PKW height. The least important were the PKW key widths.
6) To bridge the gap between complex computational predictions and real-world applications, the CatBoost model was integrated into a user-friendly GUI and hosted on GitHub, allowing the designers and engineers to predict the relative discharge of type-A PKW quickly and easily.

It is advised that future research be conducted to enhance the dataset by incorporating a broader range of parameters, which can enhance the performance of ML models. Also, to improve the generalizability of the ML models, it is recommended that the discharge of other PKW types (i.e., B, C, and D) be predicted, as this study exclusively concentrates on predicting discharge for type-A PKWs.

Appendices

Appendix A

The optimal form of the equation developed by the MLR model was the result of extensive testing. The derived linear equation from the MLR model can be written as follows:

$$\frac{q}{\sqrt{gB^3}} = -0.033 + 0.007\, W_i/W_o + 0.031\, P/W_u + 0.224\, H/W_u \tag{A.1}$$

Appendix B

It is important to note that through the trial and error process used in the GEP model, an algebraic equation was derived that establishes a relationship between the output variable ($q/\sqrt{gB^3}$) and the input variables (W_i/W_o, P/W_u, and H/W_u). The developed equation based on the GEP model for predicting the relative discharge ratio of the type-A PKW was expressed as:

$$\frac{q}{\sqrt{gB^3}} = 101\left(9157\,(P/W_u) + H/W_u - \frac{16.8}{(H/W_u)^2} + 132\right)^{\frac{1}{3}} * \left(9.98^{\sqrt[3]{H/W_u}} * 6.8\text{x}10^{-5}\,\sqrt[3]{H/W_u} * (W_i/W_o - P/W_u + 4.3)^{\frac{1}{4}}\right) \tag{B.1}$$

The relevant expression tree (ET), including the three sub-expression trees (Sub-ETs), were represented in Fig. B.1, where d0, d1, and d2 indicated W_i/W_o, P/W_u, and H/W_u, respectively. The numerical constants in the first gene (Fig. B.1a), c0, c1, c3, c4, c5, and c8, were -4.1, 10.5, 8.5, -12.6, 6.0, and 4.6, respectively. Similarly, c3, c6, and c7 in the second gene (Fig. B.1b) were 1.5, -4.6, and 2.4, respectively. Likewise, c0, c1, c2, and c5 in the third gene (Fig. B.1c) were -7.9, 9.98, 9.0, and -7.9, respectively.

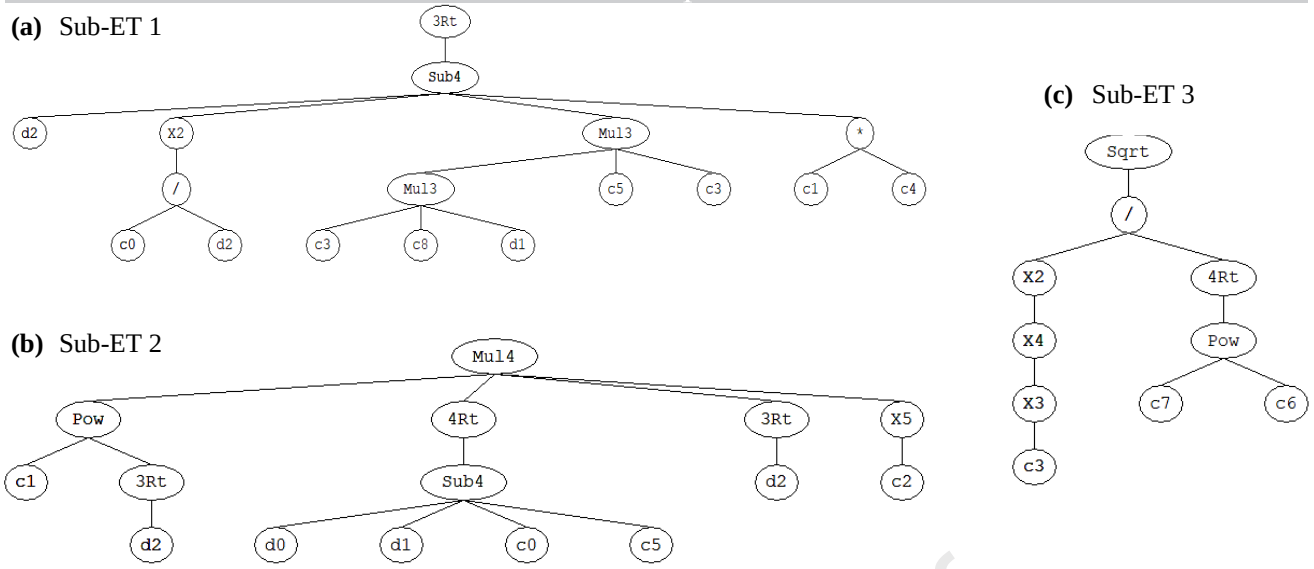


Fig. B.1 GEP model trees: (a) Sub-ET 1, (b) Sub-ET 2, and (c) Sub-ET 3.

Appendix C

The interactive GUI and the codes for the machine learning models used in this research are available for open access at the following link: https://github.com/mkamel24/FM_PKW.

Statements and Declarations

Funding

This research did not receive funding from specific agencies in the public, commercial, or not-for-profit divisions.

Competing Interests

The authors declare no conflicts of interest.

Data Availability

Data is available upon reasonable request from the corresponding author.

Acknowledgments

The authors would like to thank the journal editor and the anonymous reviewers for their editing and comments.

References

[1] T. Selim, A.K. Hamed, M. Elkiki, M.G. Eltarabily, Numerical investigation of flow characteristics and energy dissipation over piano key and trapezoidal labyrinth weirs under free-flow conditions, Modeling Earth Systems and Environment. (2023). https://doi.org/10.1007/s40808-023-01844-w.

[2] F. Lempérière, A. Ouamane, The Piano Keys weir: a new cost-effective solution for spillways, International Journal on Hydropower & Dams. 10 (2003) 144–149. https://edisciplinas.usp.br/pluginfile.php/7678719/mod_resource/content/0/The Piano Keys weir.pdf.

[3] F.M. Henderson, Open channel flow, (1996). https://www.scribd.com/doc/136483721/Open-Channel-Flow-Henderson.

[4] R.K. Bhukya, M. Pandey, M. Valyrakis, P. Michalis, Discharge Estimation over Piano Key Weirs: A Review of Recent Developments, Water. 14 (2022) 3029. https://doi.org/10.3390/w14193029.

[5] F. Lempérière, J. Vigny, A. Ouamane, General comments on Labyrinths and Piano Key Weirs, in: Labyrinth and Piano Key Weirs, CRC Press, 2011: pp. 17–24. https://doi.org/10.1201/b12349-4.

[6] M. Leite Ribeiro, M. Bieri, J.-L. Boillat, A.J. Schleiss, G. Singhal, N. Sharma, Discharge Capacity of Piano Key Weirs, Journal of Hydraulic Engineering. 138 (2012) 199–203. https://doi.org/10.1061/(asce)hy.1943-7900.0000490.

[7] R.M. Anderson, B.P. Tullis, Piano Key Weir Hydraulics and Labyrinth Weir Comparison, Journal of Irrigation and Drainage Engineering. 139 (2013) 246–253. https://doi.org/10.1061/(asce)ir.1943-4774.0000530.

[8] R.M. Anderson, B.P. Tullis, Comparison of Piano Key and Rectangular Labyrinth Weir Hydraulics, Journal of Hydraulic Engineering. 138 (2012) 358–361. https://doi.org/10.1061/(asce)hy.1943-7900.0000509.

[9] G.M. Cicero, Influence of some geometrical parameters on piano key weir discharge efficiency, (2016). https://doi.org/https://doi.org/10.15142/T3320628160853.

[10] O. Machiels, M. Pirotton, A. Pierre, B. Dewals, S. Erpicum, Experimental parametric study and design of Piano Key Weirs, Journal of Hydraulic Research. 52 (2014) 326–335. https://doi.org/10.1080/00221686.2013.875070.

[11] A.A. Bekheet, N.M. AboulAtta, N.Y. Saad, D.A. El-Molla, Effect of the shape and type of piano key weirs on the flow efficiency, Ain Shams Engineering Journal. 13 (2022). https://doi.org/10.1016/j.asej.2021.10.015.

[12] S. Li, G. Li, D. Jiang, Physical and numerical modeling of the hydraulic characteristics of type-A piano key weirs, J Hydraul Eng. 146 (2020) 1–11. https://doi.org/10.1061/(ASCE)HY.1943-7900.0001716.

[13] E. Khanahmadi, A.A. Dehghani, S.N. Alenabi, N. Dehghani, E. Barry, Hydraulic of curved type-B piano key weirs characteristics under free flow conditions, Model Earth Syst Environ. (2023) 1–18. https://doi.org/10.1007/s40808-023-01790-7.

[14] B.M. Savage, B.M. Crookston, G.S. Paxson, Physical and numerical modeling of large headwater ratios for a 15 labyrinth spillway, J Hydraul Eng. 142 (2016). https://doi.org/10.1061/(ASCE)HY.1943-7900.0001186.

[15] R. Ghanbari, M. Heidarnejad, Experimental and numerical analysis of flow hydraulics in triangular and rectangular piano key weirs, Water Sci. 34 (2020) 32–38. https://doi.org/10.1080/11104929.2020.1724649.

[16] A. Gholami, H. Bonakdari, M. Zeynoddin, I. Ebtehaj, B. Gharabaghi, S.R. Khodashenas, Reliable method of determining stable threshold channel shape using experimental and gene expression programming techniques, Neural Computing and Applications. 31 (2019) 5799–5817. https://doi.org/10.1007/s00521-018-3411-7.

[17] O. Bilhan, M.E. Emiroglu, O. Kisi, Use of artificial neural networks for prediction of discharge coefficient of triangular labyrinth side weir in curved channels, Advances in Engineering Software. 42 (2011) 208–214. https://doi.org/10.1016/j.advengsoft.2011.02.006.

[18] H. Azimi, H. Bonakdari, I. Ebtehaj, Sensitivity analysis of the factors affecting the discharge capacity of side weirs in trapezoidal channels using extreme learning machines, Flow Measurement and Instrumentation. 54 (2017) 216–223. https://doi.org/10.1016/j.flowmeasinst.2017.02.005.

[19] M. Zounemat-Kermani, A. Mahdavi-Meymand, Hybrid meta-heuristics artificial intelligence models in simulating discharge passing the piano key weirs, Journal of Hydrology. 569 (2019) 12–21. https://doi.org/10.1016/j.jhydrol.2018.11.052.

[20] Y. Deng, D. Zhang, D. Zhang, J. Wu, Y. Liu, A hybrid ensemble machine learning model for discharge coefficient prediction of side orifices with different shapes, Flow Measurement and Instrumentation. 91 (2023) 102372. https://doi.org/10.1016/j.flowmeasinst.2023.102372.

[21] I. Ebtehaj, H. Bonakdari, F. Khoshbin, H. Azimi, Pareto genetic design of group method of data handling type neural network for prediction discharge coefficient in rectangular side orifices, Flow Measurement and Instrumentation. 41 (2015) 67–74. https://doi.org/10.1016/j.flowmeasinst.2014.10.016.

[22] A. Parsaie, Predictive modeling the side weir discharge coefficient using neural network, Modeling Earth Systems and Environment. 2 (2016) 63. https://doi.org/10.1007/s40808-016-0123-9.

[23] M. Ayaz, T. Mansoor, Discharge coefficient of oblique sharp crested weir for free and submerged flow using trained ANN model, Water Science. 32 (2018) 192–212. https://doi.org/10.1016/j.wsj.2018.10.002.

[24] S.M. Borghei, Z. Vatannia, M. Ghodsian, M.R. Jalili, Oblique rectangular sharp-crested weir, Thomas Telford Ltd, 2003. https://doi.org/10.1680/wame.2003.156.2.185.

[25] M. Haghbin, A. Sharafati, A review of studies on estimating the discharge coefficient of flow control structures based on the soft computing models, Flow Measurement and Instrumentation. 83 (2022) 102119. https://doi.org/10.1016/j.flowmeasinst.2021.102119.

[26] D. Singh, M. Kumar, Gene expression programming for computing energy dissipation over type-B piano key weir, Renewable Energy Focus. 41 (2022) 230–235. https://doi.org/10.1016/j.ref.2022.03.005.

[27] D. Singh, M. Kumar, Computation of energy dissipation across the type-A piano key weir by using gene

expression programming technique, Water Supply. 22 (2022) 6715–6727. https://doi.org/10.2166/ws.2022.255.

[28] F. Salmasi, Effect of downstream apron elevation and downstream submergence in discharge coefficient of ogee weir, ISH Journal of Hydraulic Engineering. 27 (2021) 375–384. https://doi.org/10.1080/09715010.2018.1556125.

[29] M.K. Elshaarawy, A.K. Hamed, Predicting discharge coefficient of triangular side orifice using ANN and GEP models, Water Science. 38 (2024) 1–20. https://doi.org/10.1080/23570008.2023.2290301.

[30] M.L. Ribeiro, M. Pfister, A.J. Schleiss, J.-L. Boillat, Hydraulic design of A-type Piano Key Weirs, Journal of Hydraulic Research. 50 (2012) 400–408. https://doi.org/10.1080/00221686.2012.695041.

[31] M. Elshaarawy, A.K. Hamed, S. Hamed, Regression-Based Models for Predicting Discharge Coefficient of Triangular Side Orifice, Journal of Engineering Research. 7 (2023) 224–231. https://doi.org/digitalcommons.aaru.edu.jo/erjeng/vol7/iss5/31.

[32] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning. 20 (1995) 273–297. https://ise.ncsu.edu/wp-content/uploads/sites/9/2022/08/Cortes-Vapnik1995_Article_Support-vectorNetworks.pdf.

[33] M.G. Eltarabily, H.F. Abd-Elhamid, M. Zeleňáková, M.K. Elshaarawy, M. Elkiki, T. Selim, Predicting seepage losses from lined irrigation canals using machine learning models, Frontiers in Water. 5 (2023) 37–76. https://doi.org/10.3389/frwa.2023.1287357.

[34] C. Ferreira, What is gep? from genexprotools tutorials-a gepsoft web resource, (2010). https://www.gepsoft.com/tutorial002.htm.

[35] J.R. Koza, Genetic programming III: Darwinian invention and problem solving, Morgan Kaufmann, 1999. https://books.google.com.eg/books?id=pIQEJHeevEcC&lpg=PR2&ots=JhfLBIMrMx&dq=%2C Genetic programming III%3A Darwinian invention and problem solving%2C Morgan Kauf&lr&pg=PR2#v=onepage&q=, Genetic programming III: Darwinian invention and problem solving, Mor.

[36] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, The Bulletin of Mathematical Biophysics. 5 (1943) 115–133. https://doi.org/10.1007/BF02478259.

[37] W.H. Shayya, S.S. Sablani, An artificial neural network for non-iterative calculation of the friction factor in pipeline flow, Computers and Electronics in Agriculture. 21 (1998) 219–228. https://doi.org/10.1016/S0168-1699(98)00032-5.

[38] T. Selim, M.K. Elshaarawy, M. Elkiki, M.G. Eltarabily, Estimating seepage losses from lined irrigation canals using nonlinear regression and artificial neural network models, Applied Water Science. 14 (2024) 90. https://doi.org/10.1007/s13201-024-02142-1.

[39] M.K. Elshaarawy, A.K. Hamed, Stacked Ensemble Model for Optimized Prediction of Triangular Side Orifice Discharge Coefficient, Engineering Optimization. (2024). https://doi.org/10.1080/0305215X.2024.2397431.

[40] M.K. Elshaarawy, M.M. Alsaadawi, A.K. Hamed, Machine learning and interactive GUI for concrete compressive strength prediction, Scientific Reports. 14 (2024) 16694. https://doi.org/10.1038/s41598-024-66957-3.

[41] S. Paudel, A. Pudasaini, R.K. Shrestha, E. Kharel, Compressive strength of concrete material using machine learning techniques, Cleaner Engineering and Technology. 15 (2023) 100661.

[42] T. Chen, C. Guestrin, XGBoost, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2016: pp. 785–794. https://doi.org/10.1145/2939672.2939785.

[43] A.V. Dorogush, V. Ershov, A. Gulin, CatBoost: gradient boosting with categorical features support, ArXiv Preprint ArXiv:1810.11363. (2018). https://doi.org/https://doi.org/10.48550/arXiv.1810.11363.

[44] J.T. Hancock, T.M. Khoshgoftaar, CatBoost for big data: an interdisciplinary review, Journal of Big Data. 7 (2020) 94. https://doi.org/10.1186/s40537-020-00369-8.

[45] N.-V. Luat, S.W. Han, K. Lee, Genetic algorithm hybridized with eXtreme gradient boosting to predict axial compressive capacity of CCFST columns, Composite Structures. 278 (2021) 114733. https://doi.org/https://doi.org/10.1016/j.compstruct.2021.114733.

[46] W. Zhang, C. Wu, H. Zhong, Y. Li, L. Wang, Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization, Geoscience Frontiers. 12 (2021) 469–477. https://doi.org/https://doi.org/10.1016/j.gsf.2020.03.007.

[47] M.G. Eltarabily, M.K. Elshaarawy, M. Elkiki, T. Selim, Modeling Surface Water and Groundwater Interactions for Seepage Losses Estimation from Unlined and Lined Canals, Water Science. 37 (2023) 315–328. https://doi.org/10.1080/23570008.2023.2248734.

[48] M.G. Eltarabily, M.K. Elshaarawy, M. Elkiki, T. Selim, Computational fluid dynamics and artificial neural networks for modelling lined irrigation canals with low-density polyethylene and cement concrete liners, Irrigation and Drainage. 73 (2024) 910–927. https://doi.org/10.1002/ird.2911.

[49] M.K. Elshaarawy, M. Elkiki, T. Selim, M.G. Eltarabily, Hydraulic Comparison of Different Types of Lining for Irrigation Canals Using Computational Fluid Dynamic Models, M.Sc. Thesis, Civil Engineering Department, Faculty of Engineering, Port Said University, 2024. https://doi.org/10.13140/RG.2.2.21927.97441.

[50] I. Ebtehaj, H. Bonakdari, A reliable hybrid outlier robust non-tuned rapid machine learning model for multi-step ahead flood forecasting in Quebec, Canada, Journal of Hydrology. 614 (2022) 128592. https://doi.org/10.1016/j.jhydrol.2022.128592.

[51] I. Ebtehaj, H. Bonakdari, M. Zeynoddin, B. Gharabaghi, A. Azari, Evaluation of preprocessing techniques for improving the accuracy of stochastic rainfall forecast models, International Journal of Environmental Science and Technology. 17 (2020) 505–524. https://doi.org/10.1007/s13762-019-02361-z.

[52] M.A. Elazab, A.E. Kabeel, E.M.S. El-Said, H.A. Dahab, A.K. Hamed, M.M. Alsaadawi, A. Elbrashy, Exergoeconomic assessment of a multi-section solar distiller coupled with solar air heater: Optimization and economic viability, Desalination and Water Treatment. 319 (2024) 100535. https://doi.org/10.1016/j.dwt.2024.100535.

[53] A.E. Kabeel, M.A. Elazab, M.E.H. Attia, M.K. Elshaarawy, A.K. Hamed, M.M. Alsaadawi, M.A. Enasr, M. Bady, Exploring the Potential of Conical Solar Stills: Design Optimization and Enhanced Performance Overview, Desalination and Water Treatment. (2024). https://doi.org/10.1016/j.dwt.2024.100642.

[54] M.G. Eltarabily, M.K. Elshaarawy, Risk Assessment of Potential Groundwater Contamination by Agricultural Drainage Water in the Central Valley Watershed, California, USA, in: 2023: pp. 37–76. https://doi.org/10.1007/698_2023_1051.

[55] A. Kashem, R. Karim, S.C. Malo, P. Das, S.D. Datta, M. Alharthai, Hybrid data-driven approaches to predicting the compressive strength of ultra-high-performance concrete using SHAP and PDP analyses, Case Studies in Construction Materials. 20 (2024) e02991. https://doi.org/10.1016/j.cscm.2024.e02991.

[56] M.G. Eltarabily, T. Selim, M.K. Elshaarawy, M.H. Mourad, Numerical and experimental modeling of geotextile soil reinforcement for optimizing settlement and stability of loaded slopes of irrigation canals, Environmental Earth Sciences. 83 (2024) 246. https://doi.org/10.1007/s12665-024-11560-y.

[57] P. Das, A. Kashem, Hybrid machine learning approach to prediction of the compressive and flexural strengths of UHPC and parametric analysis with shapley additive explanations, Case Studies in Construction Materials. 20 (2024) e02723. https://doi.org/10.1016/j.cscm.2023.e02723.

[58] M.K. Elshaarawy, M.G. Eltarabily, Machine Learning Models for Predicting Water Quality Index: Optimization and Performance Analysis for El Moghra, Egypt, Water Supply. (2024). https://doi.org/10.2166/ws.2024.189.

[59] M. Sireesha, V. Mahammood, K. Rao, Prediction of soil salinity in the Upputeru river estuary catchment, India, using machine learning techniques, Environmental Monitoring and Assessment. 195 (2023). https://doi.org/10.1007/s10661-023-11613-y.

[60] J. Zhang, D. Li, Y. Wang, Toward intelligent construction: Prediction of mechanical properties of manufactured-sand concrete using tree-based models, Journal of Cleaner Production. 258 (2020) 120665. https://doi.org/10.1016/j.jclepro.2020.120665.

[61] V. Quan Tran, V. Quoc Dang, L. Si Ho, Evaluating compressive strength of concrete made with recycled concrete aggregates using machine learning approach, Construction and Building Materials. 323 (2022) 126578. https://doi.org/10.1016/j.conbuildmat.2022.126578.

[62] H.F. Isleem, Q. Tang, M.M. Alsaadawi, M.K. Elshaarawy, D.M. Mansour, F. Abdullah, N.A.H. Sor, A. Jahami, Numerical and Machine Learning Modeling of GFRP Confined Concrete-Steel Hollow Elliptical Columns,

Scientific Reports. (2024). https://doi.org/10.1038/s41598-024-68360-4.

[63] F. Lundh, An introduction to tkinter, 539 (1999) 540. www.pythonware.com/library/tkinter/introduction/index.htm.

[64] R. Walton, A. Binns, H. Bonakdari, I. Ebtehaj, B. Gharabaghi, Estimating 2-year flood flows using the generalized structure of the Group Method of Data Handling, Journal of Hydrology. 575 (2019) 671–689. https://doi.org/10.1016/j.jhydrol.2019.05.068.

[65] M.K. Elshaarawy, N.H. Elmasry, T. Selim, M. Elkiki, M.G. Eltarabily, Determining Seepage Loss Predictions in Lined Canals Through Optimizing Advanced Gradient Boosting Techniques, Water Conservation Science and Engineering. (2024). https://doi.org/10.1007/s41101-024-00306-3.

[66] K. Elbaz, S.-L. Shen, A. Zhou, Z.-Y. Yin, H.-M. Lyu, Prediction of Disc Cutter Life During Shield Tunneling with AI via the Incorporation of a Genetic Algorithm into a GMDH-Type Neural Network, Engineering. 7 (2021) 238–251. https://doi.org/10.1016/j.eng.2020.02.016.

[67] S.K. Mero, D.A.J. Haleem, A.A. Yousif, The influence of inlet to outlet width ratio on The hydraulic performance of piano key weir (PKW-type A), Water Practice and Technology. 17 (2022) 1273–1283. https://doi.org/10.2166/wpt.2022.055.

[68] M.G. Eltarabily, A.K. Hamed, M. Elkiki, T. Selim, Hydraulic assessment of different types of piano key weirs, ISH Journal of Hydraulic Engineering. (2024) 1–24. https://doi.org/10.1080/09715010.2024.2415938.

[69] H. F. Isleem, M. K. Elshaarawy, A. K. Hamed, Analysis of Flow Dynamics and Energy Dissipation in Piano Key and Labyrinth Weirs Using Computational Fluid Dynamics, in: Computational Fluid Dynamics - Analysis, Simulations, and Applications, IntechOpen, 2024. https://doi.org/10.5772/intechopen.1006332.

[70] A.K. Hamed, M. Eltarabily, M. El-Kiki, T. Selim, Optimum Hydraulic Design of Weirs for Downstream Zone Energy Dissipation, M.Sc. Thesis, Civil Engineering Department, Faculty of Engineering, Port Said University, 2024. https://doi.org/10.13140/RG.2.2.18564.54409.

Highlights

1. CatBoost achieves high accuracy in predicting PKW discharge with $R^2 = 0.998$ and RMSE = 0.002.
2. Total upstream head is the most critical factor, followed by PKW height, in predicting discharge.
3. Ensemble models like CatBoost and XGBoost outperform non-ensemble models across different flow conditions.
4. An interactive GUI integrates the CatBoost model, offering quick and user-friendly discharge predictions.
5. The study links machine learning predictions to practical hydraulic engineering, boosting efficiency and accessibility.

Declaration of competing interest

The authors declare that they do not have any known competing financial interests or personal relationships that could appear to have influenced the work reported in this paper.