water

MDPI

Article

Application of Machine Learning Models to Bridge Afflux Estimation

Reza Piraei [1], Majid Niazkar [2,*], Seied Hosein Afzali [1] and Andrea Menapace [2]

[1] Department of Civil Engineering, Shiraz University, Shiraz 71348511554, Iran; piraeir@gmail.com (R.P.); afzali@shirazu.ac.ir (S.H.A.)
[2] Faculty of Science and Technology, Free University of Bozen-Bolzano, Piazza Università 5, 39100 Bolzano, Italy; andrea.menapace@unibz.it
* Correspondence: majid.niazkar@unibz.it

Abstract: Bridges are essential structures that connect riverbanks and facilitate transportation. However, bridge piers and abutments can disrupt the natural flow of rivers, causing a rise in water levels upstream of the bridge. The rise in water levels, known as bridge backwater or afflux, can threaten the stability or service of bridges and riverbanks. It is postulated that applications of estimation models with more precise afflux predictions can enhance the safety of bridges in flood-prone areas. In this study, eight machine learning (ML) models were developed to estimate bridge afflux utilizing 202 laboratory and 66 field data. The ML models consist of Support Vector Regression (SVR), Decision Tree Regressor (DTR), Random Forest Regressor (RFR), AdaBoost Regressor (ABR), Gradient Boost Regressor (GBR), eXtreme Gradient Boosting (XGBoost) for Regression (XGBR), Gaussian Process Regression (GPR), and K-Nearest Neighbors (KNN). To the best of the authors' knowledge, this is the first time that these ML models have been applied to estimate bridge afflux. The performance of ML-based models was compared with those of artificial neural networks (ANN), genetic programming (GP), and explicit equations adopted from previous studies. The results show that most of the ML models utilized in this study can significantly enhance the accuracy of bridge afflux estimations. Nevertheless, a few ML models, like SVR and ABR, did not show a good overall performance, suggesting that the right choice of an ML model is important.

Keywords: bridge afflux; bridge backwater; machine learning; xgboost; river engineering

check for updates

Citation: Piraei, R.; Niazkar, M.; Afzali, S.H.; Menapace, A. Application of Machine Learning Models to Bridge Afflux Estimation. *Water* **2023**, *15*, 2187. https://doi.org/10.3390/w15122187

Academic Editor: Jianjun Ni

Received: 12 May 2023
Revised: 2 June 2023
Accepted: 7 June 2023
Published: 10 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Bridges connecting riverbanks are utilitarian structures for transportation purposes. Nevertheless, bridge piers and abutments may disrupt the natural flow of rivers. During floods, due to the spatial containment they provide, water levels upstream of bridges, in particular, can rise more than the actual river limit. Bridge backwater or afflux refers to the rise in water levels relative to the normal water depth [1]. Since bridge afflux can threaten the stability or service of bridges as well as riverbanks, its estimation is crucial for flood defense schemes and river management.

For many decades, researchers have sought to improve estimation methods for bridge backwater. Various approaches have been attempted in this regard and can be categorized based on different perspectives. According to bridge types, previous studies primarily focused on either horizontal soffit [2] or arch deck [3–5] bridges. In terms of methodology, numerical [6,7], experimental [8,9], and data mining [4,10,11] methods have been utilized to estimate bridge afflux, while most of them rely on laboratory or field data.

Numerical methods include the momentum method, the energy method, Yarnell's method, the United States Bureau of Public Roads' method (USBPR), Water-Surface PROfile (WSPRO), and the HR method. Generally, they have been implemented in numerical software, such as ISIS and Hydrologic Engineering Center's River Analysis System (HEC-RAS), for river or hydraulic structure simulations. Additionally, the limitations of the

numerical method are (i) an inevitable need for a calibration process, which requires a large dataset, and (ii) their inflexibility in estimating backwater for different types of bridges [12]. Furthermore, data mining methods also require large datasets, while they may not be applicable for bridges with different geometries or hydraulic conditions than those used in the training phase. Moreover, experimental methods are (i) expensive and (ii) relatively time-consuming. Lastly, this study focuses on developing data-driven models for estimating backwater in arch bridges. Therefore, previous contributions in the corresponding area are reviewed.

Biery and Delleur [3] proposed an empirical correlation based on the Froude number and opening ratio for single-span arch bridges. Furthermore, Brown [13] conducted experimental studies, collected field data, and employed the HR method to estimate arch bridge backwater. He considered the normal depth, Froude number, and blockage ratio as input variables. Mamak et al. [4] utilized regression analysis approaches to develop two explicit equations, i.e., Multiple Linear Regression (MLR) and Multiple Non-Linear Regression (MNLR), for bridge afflux estimations. Although explicit equations have been widely used, they are only applicable within their valid ranges and assumptions. Furthermore, these equations have limitations in terms of accuracy in professional river analysis and hydraulic structure design software. With the advancement of soft computational techniques, a few studies applied optimization algorithms and machine learning (ML) models to improve bridge afflux estimations [4,5,10–12,14]. For the latter models, they have employed Radial Basis Neural Network (RBNN), Multi-Layer Perceptron (MLP), Generalized Regression Neural Networks (GRNN), Adaptive Neuro-Fuzzy Inference System (ANFIS), and Genetic Programming (GP). For instance, Cobaner et al. [1] utilized MLR and ANN to forecast bridge backwater. The results showed that ANN models provided more accurate predictions than MLR. Additionally, Seckin et al. [11] developed bridge afflux estimation models using RBNN, MLP, and ANFIS and compared them with those obtained by MLR and MNLR. They reported the superior performance of ANFIS in predicting bridge afflux compared to other models. Furthermore, Niazkar et al. [12] utilized the Genetic Algorithm (GA) and hybrid MHBMO-GRG algorithm to develop two new explicit equations, whose coefficients were optimized by minimizing the Root Mean Square Error (RMSE) between the predicted and observed bridge afflux. They compared the performances of explicit formulas with those of two ML models (GP and ANN) and concluded the superiority of the ML models.

Although efforts have been made to improve the estimation of bridge afflux, there is still a need for more reliable and sustainable approaches developed by new and advanced powerful optimization algorithms, ML models, and other data mining techniques to achieve a higher accuracy [15]. Furthermore, despite the importance of bridge backwater in ensuring the safe design of piers and other hydraulic structures, few studies have focused on this issue in the literature, and the efficiency of ML models in addressing this issue has not been adequately assessed.

For this purpose, the present study aims to utilize eight ML models to develop models that can properly predict bridge afflux using 202 laboratory and 66 field data. The ML models consist of Support Vector Regression (SVR), Decision Tree Regressor (DTR), Random Forest Regressor (RFR), AdaBoost Regressor (ABR), Gradient Boost Regressor (GBR), eXtreme Gradient Boosting (XGBoost) for Regression (XGBR), Gaussian Process Regression (GPR), and K-Nearest Neighbors (KNN). To the best of the authors' knowledge, it is the first time that these ML models have been applied to estimate bridge afflux. Furthermore, the performance of the ML-based models was compared with those of ANN, GP, and explicit equations adopted from previous studies [4,12–14]. Finally, the findings indicate that most of the ML models utilized in this study can significantly enhance the accuracy of bridge backwater estimation.

2. Materials and Methods

2.1. Bridge Backwater Explicit Formulas

Under normal circumstances, water flows along the river and beneath the bridge at a normal depth. However, the water level rises during flood events due to bridge constriction [12]. Figure 1 depicts a schematic condition of bridge afflux to facilitate a better understanding of the phenomenon. According to Figure 1, the bridge backwater is represented by dh, while D_1 and D_3 denote the normal flow depth at sections 1 and 3, respectively. Previous studies [1,4,8,10,12,14] have identified four parameters that have the most significant impacts on the bridge afflux. These parameters can be utilized to evaluate dh for arched bridge construction in rivers. They include (i) the normal downstream depth (D_3), (ii) the Froude number at section 3 (Fr_3), (iii) the ratio of blockage area of the bridge to the flow area at section 1 (J_1), and (iv) the ratio of blockage area of the bridge to the flow area at section 3 (J_3). Therefore, by adopting a dimensionless analysis, the bridge backwater is determined as follows:

$$\frac{dh}{D_3} = F(J_1, J_3, Fr_3), \tag{1}$$

where F denotes a function.

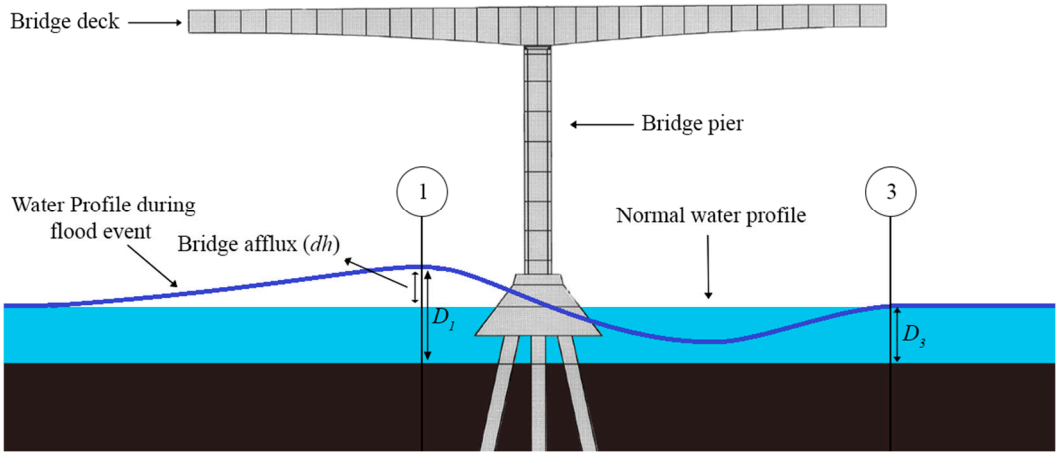


Figure 1. Schematic representation of the bridge afflux due to bridge constrictions. Section 1 is at the upstream of the bridge with the maximum bridge afflux during a flood event, while Section 3 is at the downstream of the bridge where the water level returns to a normal depth.

The literature presents five different empirical formulas for calculating the bridge backwater, namely, (i) Biery and Delleur Equation (2) [3], (ii) MLR Equation (3) [4], (iii) Multiple Non-Linear Regressions (MNLR) Equation (4) [4], (iv) GA Equation (5) [12], and (v) MHBMO-GRG Equation (6) [12],

$$\frac{dh}{D_3} = 0.47 \times \left[\frac{Fr_3}{1 - J_3}\right]^{2.26}, \tag{2}$$

$$\frac{dh}{D_3} = 1.62 \times J_1 - 1.54 \times J_3 + 0.429 \times Fr_3, \tag{3}$$

$$\frac{dh}{D_3} = 1.311 \times \left[\frac{J_1^{1.8} Fr_3^{1.23}}{J_3^{0.744}}\right], \tag{4}$$

$$\frac{dh}{D_3} = \begin{cases} 4.49 \times J_1^{1.390} \times J_3^{0.514} \times Fr_3^{1.421} \text{ for} Fr_3 < 1.179 \\ 4.946 \times J_1^{1.519} \times J_3^{-0.267} \times Fr_3^{-3.242} \text{ for} Fr_3 \geq 1.179 \end{cases}, \tag{5}$$

$$\frac{dh}{D_3} = \begin{cases} 2.274 \times J_1^{5.328} \times J_3^{-0.899} \times Fr_3^{0.596} \text{ for} Fr_3 < 0.2 \\ 5.243 \times J_1^{1.102} \times J_3^{0.822} \times Fr_3^{1.523} \text{ for} Fr_3 \geq 0.2 \end{cases}, \tag{6}$$

Biery and Delleur's Equation (2) is widely used as an empirical formula for calculating bridge backwater, while it does not include the effect of J_1. Based on Equation (2), an increase of Fr_3 leads to an increase in the bridge afflux, while decreasing J_3 causes a reduction in the bridge backwater. According to Equation (1), it is suggested that J_1 should also be included, which is the case in Equations (3)–(6). In other words, Equations (3)–(6) to incorporate all three independent parameters affecting the bridge afflux. Among the mentioned empirical equations, Equation (3) is the only linear one, whereas the rest have a nonlinear relationship.

According to Equations (5), when $Fr_3 < 1.179$, J_3 and Fr_3 have a positive correlation with dh/D_3. However, when $Fr_3 \geq 1.179$, J_3 and Fr_3 have a negative correlation with dh/D_3. In contrast, J_1 always has a positive correlation with dh/D_3. Furthermore, Equation (6) demonstrates that J_3 has two distinct impacts on the bridge afflux. Nevertheless, J_1 and Fr_3 always show a positive correlation.

2.2. Datasets

Since most bridge afflux estimation methods have been developed by utilizing laboratory and/or field data, this study utilizes a dataset that includes both laboratory and field data. The former were obtained from Hydraulic Research Wallingford experiments conducted on the bridge backwater estimation, while the latter consist of 66 observations between 1946 and 1983 [13]. Additionally, laboratory experiments were conducted on two rectangular flumes with different types of arched bridges, resulting in 202 data samples.

Figure 2 depicts the discrepancies of dh/D_3 with respect to J_1, J_3, and Fr_3. As shown, most data points have similar values for J_1 and J_3, while Fr_3 values are generally lower than 0.75. Additionally, most data points have dimensionless bridge afflux (i.e., dh/D_3) lower than 0.78. This database has been utilized in previous studies [4,12–14], indicating its technical reliability for the implementation of ML models in the proposed study.

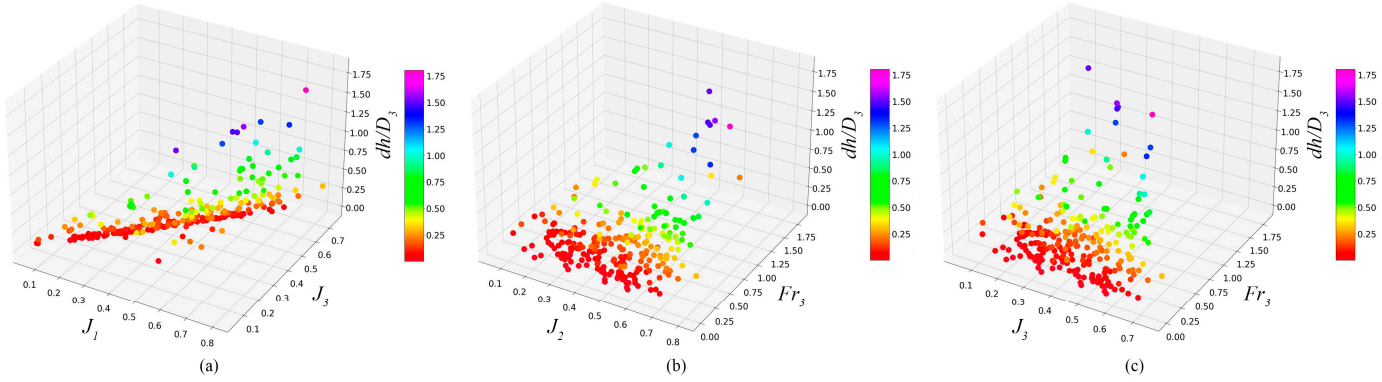


Figure 2. Discrepancy of dh/D_3 with respect to (**a**) J_1 and J_3, (**b**) J_1 and Fr_3, and (**c**) J_3 and Fr_3.

The database was divided into two categories by random selection, namely, training and testing datasets. From 268 data points that were collected, 80% (161 laboratory and 50 field data) were exploited to train ML models, while the remaining 20% (41 laboratory and 16 field data) were used for the comparison of results. Table 1 lists the maximum, minimum, average, and standard deviation of each variable for the training and testing datasets. It indicates that the minimum values of the training dataset are lower than those of the testing dataset, while the maximum values of the training dataset are higher than those of the testing dataset. This suggests that the data were well divided.

Table 1. Statistical characteristics of different variables in the training and testing datasets.

	Training Dataset				Testing Dataset			
Parameters	**Min**	**Mean**	**Max**	**Std Dev**	**Min**	**Mean**	**Max**	**Std Dev**
J_1	0.064	0.455	0.803	0.167	0.099	0.44	0.746	0.157
J_3	0.047	0.388	0.742	0.152	0.097	0.374	0.678	0.144
Fr_3	0.008	0.374	1.809	0.269	0.053	0.34	1.021	0.189
dh/D_3	0.002	0.261	1.805	0.324	0.008	0.223	0.685	0.190

2.3. ML Models

To develop an equation for estimating bridge afflux, it is required to derive a relationship between the hydraulic characteristics of the river and the bridge afflux. ML algorithms, on the other hand, utilize large datasets to train learning machines that can facilitate such tasks. Before training or fitting an ML model to a dataset, critical transformations, which can have a significant impact on a model's performance, should be performed [16]. For this purpose, the *MinMaxScaler* transformation from the *Scikit-learn* library was utilized to normalize the data in this study. This transformation entails subtracting the minimum value from each variable and dividing the result by the difference between the maximum and minimum values. Through this transformation, each variable rescales between 0 and 1. Moreover, if a model predicts bridge afflux as a negative value, the algorithm is constrained to replace it with zero.

In this study, seven ML models—including SVR, DTR, RFR, ABR, GBR, GPR, and KNN—were implemented in Python utilizing the *Scikit-learn* library, whereas the *xgboost* library was used for implementing an XGBR model. For applying these ML models, the ML primary hyperparameters, which are introduced in Table 2, were set using a trial-and-error process, while default values were selected for other ML hyperparameters.

Table 2. Brief descriptions of machine learning hyperparameters optimized in this study.

Hyperparameter	**Model**	**Description**
n_estimators	RFR, ABR, GBR, XGBR	Total number of trees.
criterion	DTR, RFR	Loss function, which can be one of *squared_error*, *absolute_error*, *poisson*, and *friedman_mse*.
max_depth	DTR, RFR, GBR, XGBR	Maximum depth allowed for each tree, positive integer or *None*.
min_samples_split	DTR, RFR, GBR	Minimum instances required to split data.
loss	ABR	Loss function, which can be one of *linear*, *square*, and *exponential*.
loss	GBR	Loss function, which can be one of *squared_error*, *absolute_error*, *huber*, and *quantile*.
p	KNN	The power of the distance function. If $p = 1$, the distance function is Manhattan, and if $p = 2$, it is Euclidean, while any other arbitrary value of p corresponds to Minkowski.
N_neighbors	KNN	Total number of neighbors.
Weights	KNN	The weight of each neighbor; includes *distance* for weighting based on the distance, *uniform* for equal weight, or any other user-defined functions.

Table 2. *Cont.*

Hyperparameter	Model	Description
Algorithm	KNN	The algorithm computing the nearest neighbor's parameter, which can be one of *auto*, *ball_tree*, *kd_tree*, and *brute*.
Kernel	SVR	The kernel function, which can be one of *linear*, *poly*, *rbf*, and *sigmoid*.
Degree	SVR	A non-negative parameter for *poly* kernel.
Gamma	SVR	A coefficient for *rbf*, *poly,* and *sigmoid* kernels, which can be *scale*, *auto*, or any non-negative value.
C	SVR	A positive regularization parameter.
kernel	GPR	The kernel function specifying the covariance function, which can be any user-defined function.
alpha	GPR	A value added to the diagonal of the kernel matrix during the fitting process.
learning_rate	ABR, GBR, XGBR	The weight assigned to each tree during each iteration. Increasing the learning rate increases the contribution of each tree; range: [0, 1].
min_split_loss	XGBR	Minimum loss reduction required to split a child node (gamma); range: [0, ∞].
reg_alpha	XGBR	L1 weight regularization term.
reg_lambda	XGBR	L2 weight regularization term.
min_child_weight	XGBR	Minimum summation of weights required in each child node. If the summation of instance weights is below this threshold, the algorithm will stop further partitioning.

2.3.1. Support Vector Regression

A Support Vector Machine (SVM) is an ML algorithm that can handle both linear and nonlinear regressions. The term SVM refers to both classification and regression tasks, whereas the term SVR is used specifically for regression tasks [17]. It utilizes a method, which is called the kernel trick, to handle nonlinearly separable data by transforming them into a higher dimensional space, where a linear separation is feasible. This method, along with kernel hyperparameters and a regularization term, highly influence the performance of an SVM model [18]. In this study, through a trial-and-error process, *kernel*, *gamma*, and *C* hyperparameters were set to *rbf*, *scale*, and 1, respectively.

In a multidimensional space, data points are represented by vectors and are consequently called support vectors [16]. By mapping an input vector to a higher dimensional space, SVM creates an optimal hyperplane as a decision boundary that separates the data into two classes. The creation of the optimal hyperplane is achieved by maximizing the margin between the hyperplane and the nearest support vectors. The weights of the SVM model determine the influence of each variable on the hyperplane, while each weight is decided by the hyperplane value at that point. Moreover, a small amount of bias is added to the model for adjusting the decision boundary to prevent overfitting.

2.3.2. Decision Tree Regressor

The Decision Tree algorithm is a common ML algorithm for both classification and regression (i.e., DTR) tasks. Decision Trees are generally prone to overfitting, which results in poor performance on testing datasets [16]. DTR partitions data based on feature values through branch nodes to produce a tree-like structure. Each data division represents the result of a splitting test on the training data. The branch nodes lead to leaf nodes, each representing an outcome of the model, which is obtained by averaging the data points from the training data placed in that node through the splitting process. In this study, optimum

values of the main hyperparameters of DTR were obtained by adopting a trial-and-error process. Therefore, *criterion*, *max_depth*, and *min_samples_split* hyperparameters were set to *absolute_error*, 5, and 3, respectively.

2.3.3. Random Forest Regressor

RFR is an ensemble method that combines multiple decision trees, normally 100 decision trees, not only to improve the DTR accuracy but also to address overfitting. Each tree in RFB is constructed utilizing a random subset of the training data and input features. The data bootstrapping procedure, which resamples the training dataset to create random subsets, ensures that each tree captures different patterns and relationships within the data. The final prediction of the model is the average of each tree prediction, which makes RFR less prone to overfitting [19]. This study, through a trial-and-error process, selects *n_estimators*, *criterion*, *max_depth*, and *min_samples_split* hyperparameters equal to 1500, *absolute_error*, 32, and 2, respectively.

2.3.4. AdaBoost Regressor

ABR is an ensemble ML method that combines weak learners to form a strong learner. Unlike RFR, which does not limit the structure of each tree, ABR constructs all the trees with a single node and two leaves (i.e., stumps), which are classified as weak learners with a limited predictive capability. ABR iteratively trains a series of weak learners, with each subsequent learner placing more emphasis on the misclassified data points from a previous learner. The sequence of creating weak learners in ABR is important because the mistake made by each weak learner affects the construction of subsequent weak learners. The algorithm assigns weights, i.e., importance, to each weak learner based on the classification error during the training process. The final prediction in ABR is the weighted summation of the weak learners [20]. A trial-and-error process conducted in this study results in *n_estimators*, *loss*, and *learning_rate* hyperparameters equal to 100, *square*, and 0.3, respectively.

2.3.5. Gradient Boost Regressor

GBR exploits gradient descent to optimize the loss function to enhance more accurate and flexible estimation models compared to ABR. It begins with a single leaf as a preliminary estimation, which is typically the average of the data when there is continuous data. Like ABR, it trains successive trees to correct the errors of their predecessors. However, in GBR, only users can limit the number of leaves in each tree, and each tree is assigned an equal weight. The process of constructing trees continues until either an additional tree no longer improves the model accuracy or a specified threshold is met. The final prediction obtained by GBR is a weighted summation of the trees. Compared to RFR and ABR, GBR is generally more accurate but more prone to overfitting [16]. In this study, *n_estimators*, *loss*, *learning_rate*, *max_depth*, and *min_samples_split* hyperparameters were set to 300, *huber*, 0.2, 2, and 3, respectively, using a trial-and-error process.

2.3.6. XGBoost for Regression

XGBoost is an advancement to the GBR algorithm that includes additional features, such as regularization and tree pruning to prevent overfitting [21]. XGBoost is a widely used ML model for both classification and regression (XGBR) tasks that provides flexibility in selecting loss functions for the evaluation of the model. It is well known for its swift and efficient processing of large datasets, block technology, and parallelism with CPU multithreading, and it continuously improves its algorithm for better accuracy [19]. Additionally, it has a unique objective function that consists of two main components: (i) The first component addresses overfitting by a model complexity reduction, and (ii) the second one utilizes a regularization term and loss function to determine residuals (i.e., the difference between observed and predicted values) [22]. The residuals are basically utilized to refine previous predictor errors during each iteration. Like other ensemble models, it iteratively

combines multiple weak learners to produce a strong learner. The final prediction in XGBR is a weighted summation of trees, while the weights are based on derivatives of the residuals [23]. Through a trial-and-error process conducted in this study, *n_estimators*, *reg_alpha*, *reg_lambda*, *learning_rate*, *max_depth*, *min_split_loss*, and *min_child_weight* hyperparameters were set to 300, 0, 1.7, 0.4, 15, 0, and 1, respectively.

2.3.7. K-Nearest Neighbors

KNN is an ML algorithm used for nonlinear regression. It predicts unseen data (i.e., testing data) by sorting and finding the data points closest to the training dataset, which are called nearest neighbors [24]. KNN principally finds the K nearest data points to a given testing data point and calculates the weighted average target values of these data points. The algorithm utilizes a distance function to measure the similarity between the data points in the training dataset and the testing data point. The most commonly used distance functions for continuous variables are Euclidean, Manhattan, and Minkowski [25]. Through a trial-and-error process, *n_neighbors*, *algorithm*, *weights*, and *p* hyperparameters were set to 2, *auto*, *distance*, and 2, respectively.

2.3.8. Gaussian Process Regression

GPR is a non-parametric kernel-based model that does not assume a fixed number of parameters, making it suitable for linear and non-linear problems [26]. Furthermore, GPR is a probabilistic ML model that assumes a Gaussian distribution generates the data and can be denoted by a covariance function and a mean function [27]. The former measures the similarity between input vectors of observed and desired data points, while the latter is used to control the complexity of the model. Additionally, the covariance function is typically more important than the mean function [28]. The Radial Basis Function (RBF) kernel is commonly used as the covariance function in GPR models, which maps data to a high-dimensional space. By computing the joint distribution of the training and testing data, GPR can predict new data points. This study used a trial-and-error process to set the '*alpha*' hyperparameter equal to 0.01. Moreover, Matern was utilized as the kernel function, which is an extension of the RBF kernel.

2.4. Feature Importance Analysis

XGBR has a built-in feature importance functionality that can be utilized as a sensitivity analysis to estimate the relative importance of each input feature. The feature importance values can be determined by two metrics: (i) *weight* that tracks how frequently each feature is used to divide data points across all trees in the model, or (ii) *gain* that calculates the average improvement in the accuracy achieved when each feature is used to split the data [29]. Higher importance values indicate greater influence on model predictions. Feature importance results can guide further analysis or feature selection. However, it is important to note that the feature importance analysis may not be generalizable to other datasets or applications as it is specific to the training process of the model and training dataset. Therefore, it should be interpreted in the context of the specific data, model, and hyperparameters in question. In this study, feature importance values are determined by utilizing the Python command *model.feature_importances_* and *gain* metric.

2.5. Performance Criteria

For evaluation of model performances, six criteria were utilized. They included (1) RMSE, (2) Mean Absolute Errors (MAE), (3) Mean Absolute Relative Error (MARE), (4) Maximum Absolute Relative Error (MXARE), (5) Nash–Sutcliffe efficiency (NSE), and (6) Determination coefficient (R^2) [30]. These metrics are presented in the following equations:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(O_i - P_i)^2}{n}}, \quad (7)$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|O_i - P_i|, \tag{8}$$

$$\mathrm{MARE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{O_i - P_i}{O_i}\right|, \tag{9}$$

$$\mathrm{MXARE} = \max\left(\left|\frac{O_i - P_i}{O_i}\right|\right) \text{ for } i = 1, \ldots, n, \tag{10}$$

$$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{\sum_{i=1}^{n}\left(O_i - \frac{\sum_{i=1}^{n} O_i}{n}\right)^2}, \tag{11}$$

$$R^2 = \left\{ \frac{\sum_{i=1}^{n}\left[\left(O_i - \frac{\sum_{i=1}^{n} O_i}{n}\right)\left(P_i - \frac{\sum_{i=1}^{n} P_i}{n}\right)\right]}{\sqrt{\sum_{i=1}^{n}\left(O_i - \frac{\sum_{i=1}^{n} O_i}{n}\right)^2 \sum_{i=1}^{n}\left(P_i - \frac{\sum_{i=1}^{n} P_i}{n}\right)^2}} \right\}^2, \tag{12}$$

where symbols n, O, and P refer to the number of data points, observed, and predicted bridge affluxes, respectively.

According to the definitions given for each metric, an increase in the precision of estimating bridge affluxes is related to higher values of R^2 and NSE, as well as lower values of RMSE, MAE, MARE, and MXARE.

2.6. Reliability Analysis

Reliability analysis can basically assess how well an estimation model performs in relation to a desired threshold. In this study, a reliability analysis was conducted to measure the reliability of the methods utilized to predict bridge afflux. The reliability of each method was determined by calculating the percentage of cases in which the relative error, Equation (13), was equal to or less than a threshold of 20%, which was suggested by previous studies [31]. The resulting percentage provides a measure of the model consistency and reliability in predicting the outcome:

$$\mathrm{RE} = \frac{|P_i - O_i|}{O_i}. \tag{13}$$

3. Results

In this study, eight ML models (i.e., SVR, DTR, RFR, ABR, GBR, XGBR, GPR, and KNN) have been utilized for the first time to estimate bridge afflux. Different models considered in this study were evaluated to determine their effectiveness in predicting bridge affluxes, and their performances were compared with those of other methods applied in previous studies. The results of the implementation of these models are described in the following subsections.

3.1. Results of Correlation

The correlations between the observed bridge affluxes and values predicted by different methods for the training and testing datasets are depicted using a logarithmic scale in Figures 3 and 4, respectively. The x-axis in these figures represents observed bridge affluxes, while the y-axis illustrates predicted bridge backwater depths.

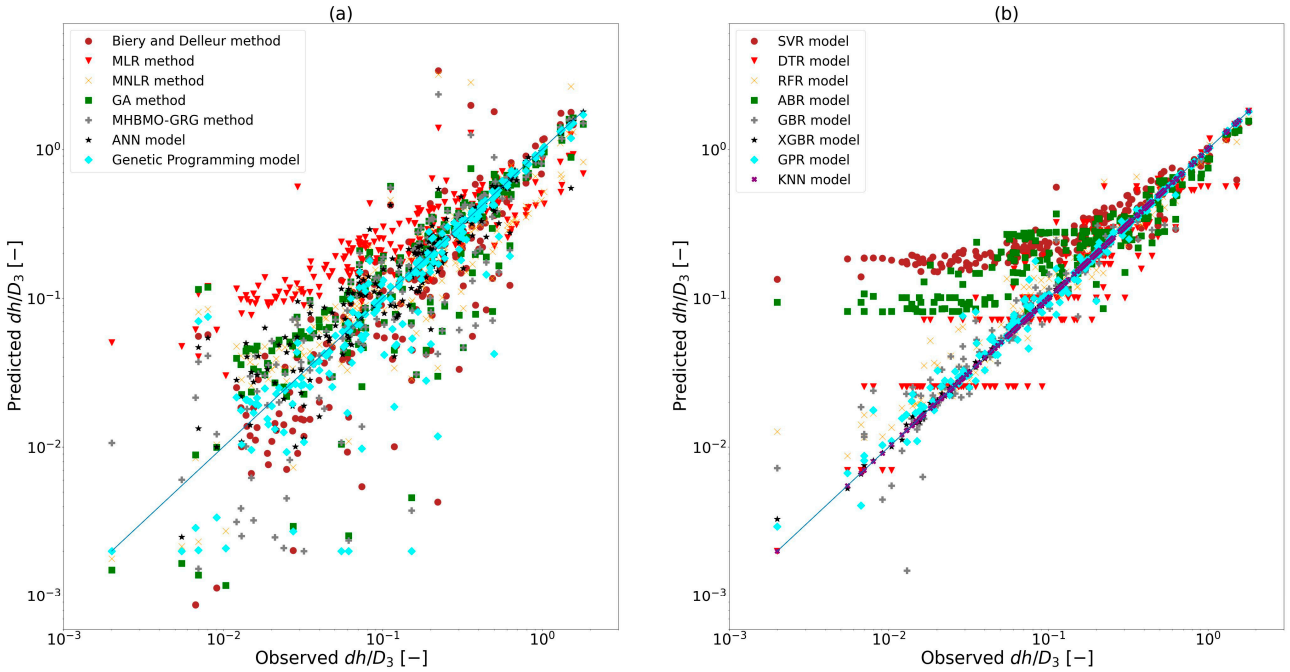


Figure 3. Correlation results for the training dataset obtained by methods used (**a**) in previous studies and (**b**) in this study.

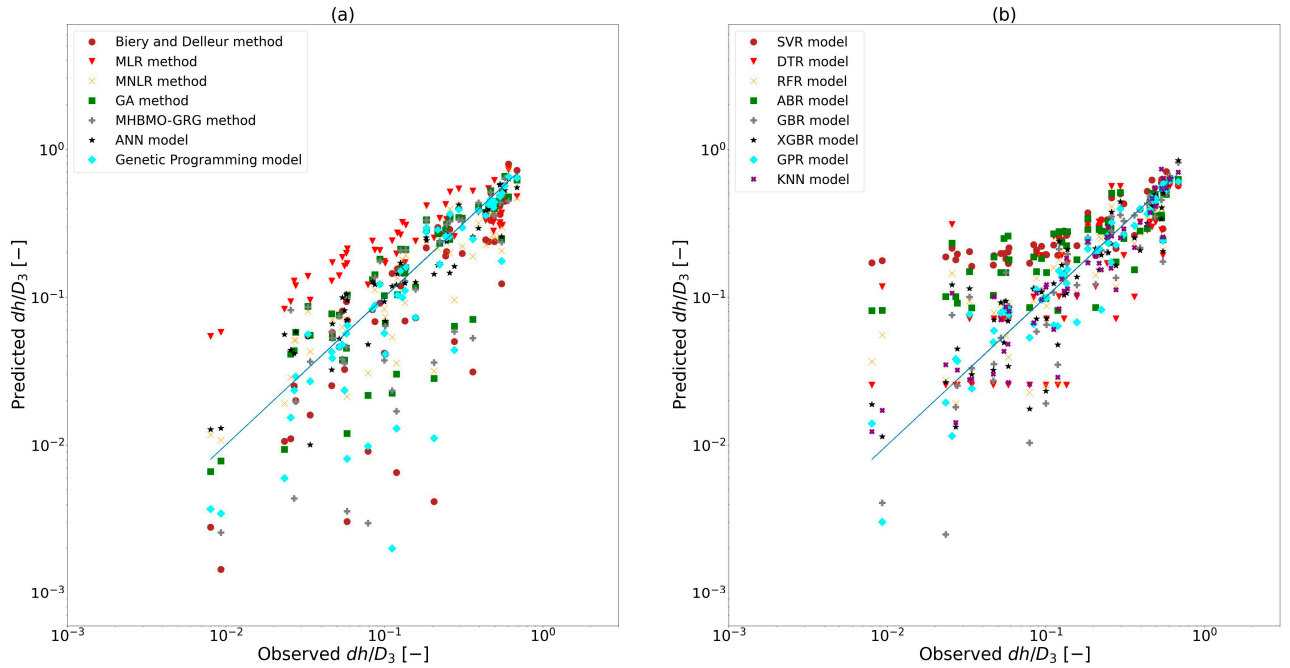


Figure 4. Correlation results for the testing dataset obtained by methods used (**a**) in previous studies and (**b**) in this study.

Based on Figures 3b and 4b, the SVR and ABR models strongly overestimate medium and small values of bridge afflux. Moreover, the DTR model predicts several afflux values as a constant value, leading to a stepwise correlation. The RFR and GBR models exhibit good correlations for the training data but display some scatters for the testing data, particularly for small and medium values. Lastly, the XGBR, GPR, and KNN models show excellent correlations with the training data and satisfactory correlations with the testing data.

3.2. Results of Performance Metrics

Table 3 provides the results of the performance evaluation metrics. For better clarification, Figure 5 presents a comparison of different methods through heat maps. Each heat map demonstrates the performances of estimation models using a range of colors, with blue representing a superior performance. As shown, there is a diversity of performances for predicting bridge afflux values. Regarding RMSE, the SVR, DTR, and ABR models exhibited poor performances with testing RMSE close to 0.12. The performance of the models used in this study in terms of RMSE was more robust for the training data, with KNN and XGBR models having RMSE of almost 0, and the GPR model RMSE equal to 0.02. Regarding the testing dataset, the GPR model demonstrated the best RMSE of 0.06. Additionally, regarding MAE and MARE, the KNN model obtained the best performance for the training data (MAE and MARE close to 0), and the GPR model achieved the best performance for the testing data (MAE = 0.04 and MARE = 0.26). Furthermore, regarding MXARE, the GPR model outperformed other methods with a testing MXARE of 1.32. Moreover, the SVR model achieved the worst performance regarding MARE (i.e., 2.04) and MXARE (i.e., 20.32) in the testing phase.

Table 3. Comparing performances of different methods for predicting bridge afflux.

Method	Dataset	RMSE	MAE	MARE	MXARE	NSE	R^2
Previous Studies [3,4,12]							
Biery and Delleur	Training	0.28	0.10	0.57	14.09	0.23	0.56
	Testing	0.12	0.08	0.40	**0.98**	0.61	0.68
MLR	Training	0.24	0.16	1.86	24.24	0.47	0.49
	Testing	0.13	0.12	1.23	5.81	0.51	0.61
MNLR	Training	0.32	0.12	0.67	13.24	0.02	0.37
	Testing	0.12	0.08	0.39	1.43	0.62	0.67
GA	Training	0.11	0.07	0.63	15.34	0.88	0.88
	Testing	0.09	0.06	0.41	1.63	0.79	0.79
MHBMO-GRG	Training	0.19	0.08	0.53	9.48	0.66	0.72
	Testing	0.09	0.06	0.46	2.20	0.77	0.78
ANN	Training	0.09	0.04	0.52	16.17	0.92	0.92
	Testing	0.07	0.05	0.33	1.39	0.88	0.89
GP	Training	0.07	0.03	0.31	9.01	0.95	0.96
	Testing	0.08	0.05	0.31	**0.98**	0.82	0.84
This study							
SVR	Training	0.15	0.13	3.05	66.10	0.78	0.89
	Testing	0.13	0.11	2.04	20.32	0.56	0.80
DTR	Training	0.12	0.06	0.39	3.71	0.86	0.86
	Testing	0.12	0.09	0.78	11.69	0.61	0.63
RFR	Training	0.06	0.03	0.24	5.37	0.97	0.97
	Testing	0.08	0.06	0.53	4.97	0.82	0.82
ABR	Training	0.11	0.09	1.70	49.38	0.89	0.92
	Testing	0.12	0.10	1.23	8.98	0.59	0.63
GBR	Training	0.04	0.01	0.17	2.62	0.99	0.99
	Testing	0.08	0.05	0.39	2.05	0.83	0.84
XGBR	Training	0.001	0.001	0.01	0.65	**1.00**	**1.00**
	Testing	0.08	0.06	0.42	3.74	0.81	0.82
GPR	Training	0.02	0.01	0.12	1.62	**1.00**	**1.00**
	Testing	**0.06**	**0.04**	**0.26**	1.32	**0.91**	**0.91**
KNN	Training	$\mathbf{8.3 \times 10^{-18}}$	$\mathbf{1.2 \times 10^{-18}}$	$\mathbf{1.1 \times 10^{-17}}$	$\mathbf{2.1 \times 10^{-16}}$	**1.00**	**1.00**
	Testing	0.07	0.05	0.33	3.16	0.86	0.88

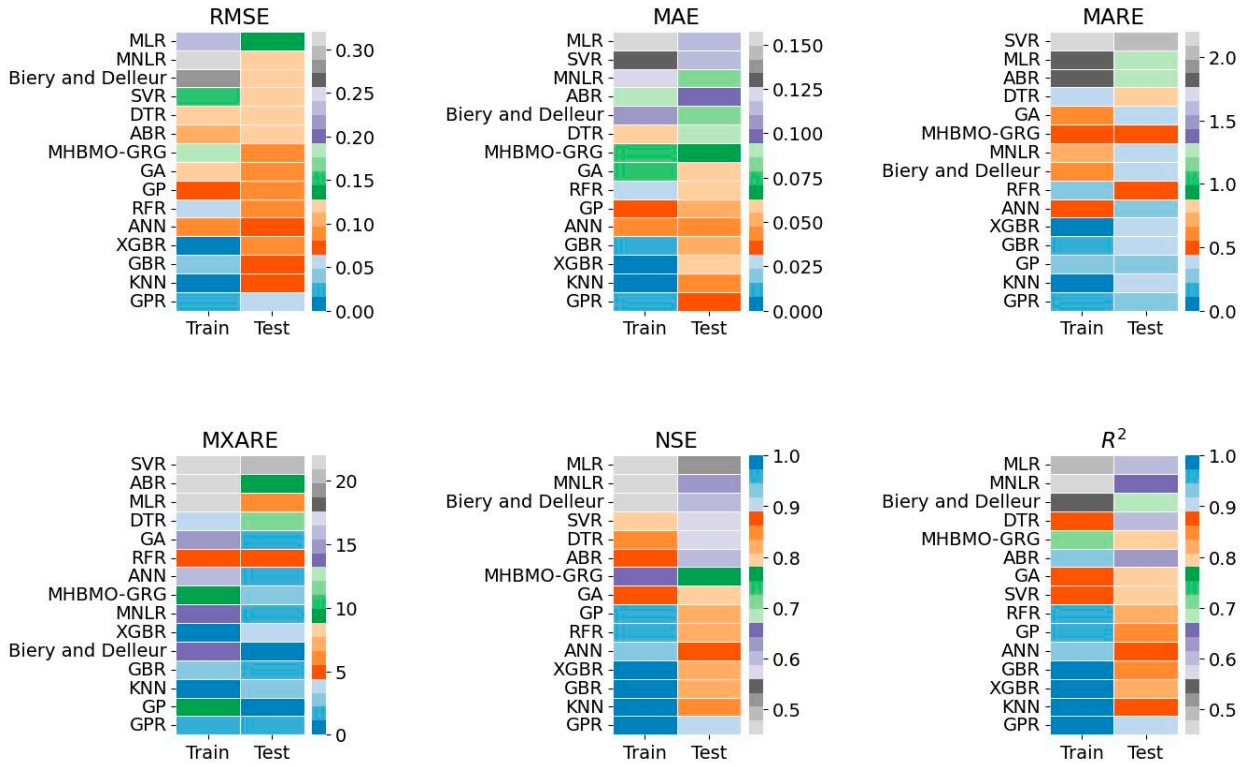


Figure 5. Heat map of the metrics results for both training and testing datasets.

Regarding R^2, a few ML models, such as DTR and ABR, regardless of their performance for the training data, did not perform well in estimating bridge afflux for the testing data (with testing R^2 values equal to 0.63). Other ML models, such as KNN, XGBR, and GPR, displayed a considerable difference between the metrics results for the training (i.e., $R^2 = 1$) and testing data (i.e., R^2 less than 0.91), indicating a dataset variance. Lastly, the NSE results are also similar to R^2, where the GPR model outperformed other methods with a testing NSE of 0.91.

3.3. Results of Ranking Analysis

To assess the performance of various estimation models based on multiple criteria, a ranking scheme was employed from the literature, which assigned an equal weight to each criterion [30]. The performance of different models for each metric (i.e., RMSE, MAE, MARE, MXARE, NSE, and R^2) was compared and ranked from the best to the worst using integers 1 through 15. After calculating the rank of each method for all metrics, the algebraic summation of the ranks was obtained for each dataset. The resulting values were then ranked again from the lowest to the highest, yielding a total rank for each method considering all metrics. Finally, Table 4 displays the ranking results obtained for each method.

According to Table 4, the GPR model ranks third in the training dataset and first in the testing dataset. As a result, it outperformed all other estimating methods based on the adopted ranking analysis. Although the KNN method outperformed others for the training dataset, it ranks fourth on the testing dataset, resulting in an overall second place. While the performance of the XGBR, GBR, GP, and ANN methods varies depending on the dataset, their overall suitable performance led to their joint third-place ranking. Biery and Delleur's equations and the RFR, GA, DTR, MHBMO-GRG, ABR, MNLR, SVR, and MLR methods are placed in subsequent places.

Table 4. Ranking results of different methods for predicting bridge affluxes.

Method	Dataset	RMSE	MAE	MARE	MXARE	NSE	R^2	Subset Rank	Total Rank
GPR (this study)	Training	3	3	3	3	3	3	3	1
	Testing	1	1	1	3	1	1	1	
KNN (this study)	Training	1	1	1	1	1	1	1	2
	Testing	3	3	4	9	3	3	4	
XGBR (this study)	Training	2	2	2	2	2	2	2	3
	Testing	7	6	9	10	7	7	7	
GBR (this study)	Training	4	4	4	4	4	4	4	3
	Testing	4	5	6	7	4	5	5	
GP [12]	Training	6	6	6	7	6	6	6	3
	Testing	6	4	2	2	6	4	3	
ANN [12]	Training	7	7	8	12	7	7	7	3
	Testing	2	2	3	4	2	2	2	
RFR (this study)	Training	5	5	5	6	5	5	5	7
	Testing	5	7	11	11	5	6	6	
GA [12]	Training	9	9	11	11	9	10	9	8
	Testing	8	8	8	6	8	9	8	
DTR (this study)	Training	10	8	7	5	10	11	8	9
	Testing	12	12	12	14	12	14	12	
Biery and Delleur [3]	Training	14	12	10	10	14	13	12	10
	Testing	11	10	7	1	11	11	9	
MHBMO-GRG [12]	Training	12	10	9	8	12	12	11	11
	Testing	9	9	10	8	9	10	11	
ABR (this study)	Training	8	11	13	14	8	8	10	12
	Testing	13	13	14	13	13	13	13	
MNLR [4]	Training	15	13	12	9	15	15	14	13
	Testing	10	11	5	5	10	12	10	
SVR (this study)	Training	11	14	15	15	11	9	13	14
	Testing	14	14	15	15	14	8	14	
MLR [4]	Training	13	15	14	13	13	14	15	15
	Testing	15	15	13	12	15	15	15	

3.4. Results of Reliability Analysis

Figure 6 presents the results of the reliability analysis conducted on both the training and testing datasets. The reliability percentage and name of each estimation model are shown on the x-axis and y-axis, respectively. Regarding the reliability for the training dataset, the KNN, XGBR, GPR, GBR, and RFR methods achieved superior performances, with percentages of 100, 99.53, 81.52, 78.67, and 70.14, respectively. On the other hand, ABR and SVR models indicated weaker performance with percentages of 30.81 and 18.48, respectively. Thus, the reliability of ML models for the training dataset is more robust than the testing dataset, while the GPR and KNN models outperformed other ML methods with testing reliabilities of 52.63% and 47.37%, respectively. Finally, the SVR and ABR models did not show good reliability overall.

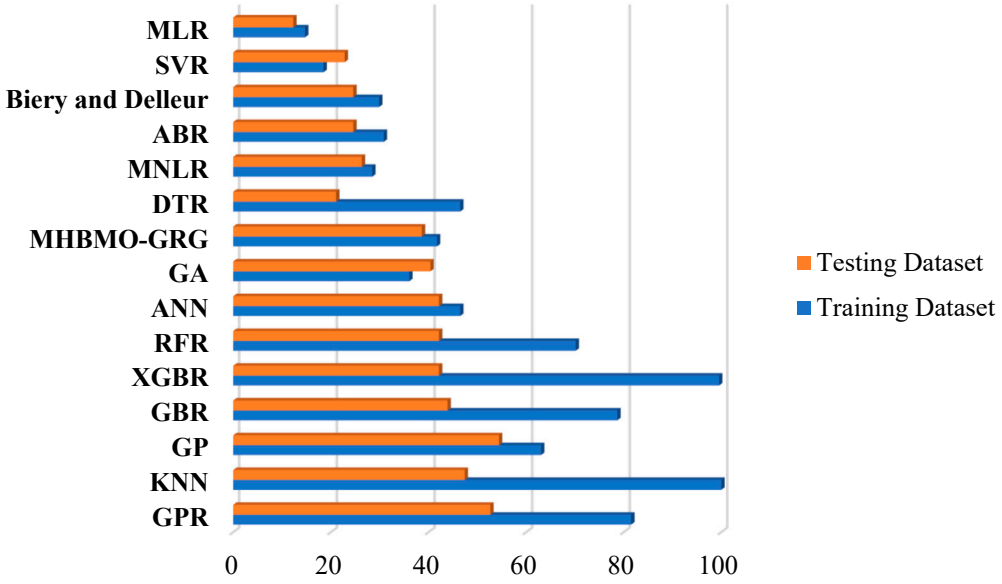


Figure 6. Reliability analysis results for the training and testing datasets.

3.5. Results of Feature Importance Analysis

Figure 7 depicts the results of the XGBR feature importance analysis conducted on the dataset. As shown, it reveals that Fr_3 is the most critical feature with an importance score of 0.61, followed by J_3 and J_1 with importance scores of 0.3 and 0.08, respectively.

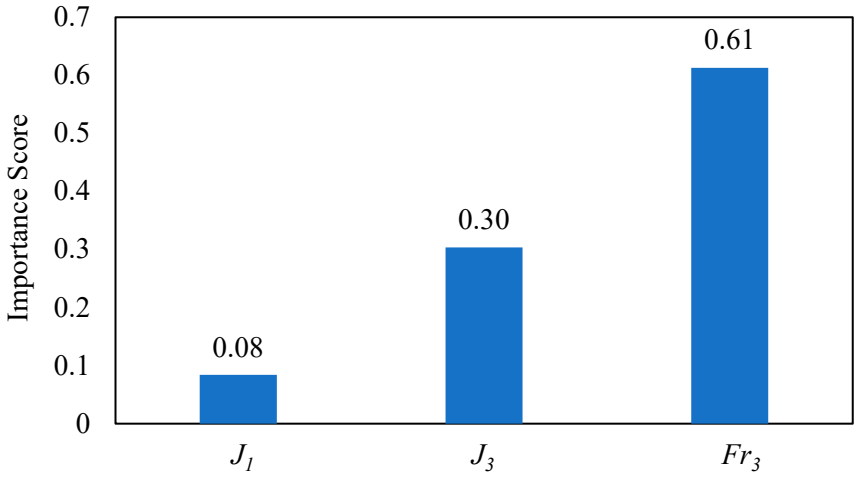


Figure 7. Results of the feature importance score.

4. Discussion

Modeling bridge afflux is essential for not only the safety of bridges but also the stability of riverbanks. Choosing an adequate estimation method is one of the important steps for modeling bridge afflux. Over the last few years, researchers have focused on applying ML models, and several studies reported the superiority of ML-based models over empirical formulas. For instance, Pinar et al. [5] utilized MLP, RBNN, GRNN, MLR, and MNLR methods to estimate bridge affluxes, and they found that MLP was more precise than others. Seckin et al. [10] examined several ANN models (i.e., Feed-Forward Back Propagation (FFBP), RBNN, and GRNN) to estimate bridge backwater in the Mississippi River basin and compared their performances with previous methods. They concluded that FFBP and RBNN outperformed other methods, with FFBP performing slightly better than RBNN. Furthermore, Seckin et al. [11] exploited RBNN, ANFIS, and MLP as alternatives to

the energy method to predict bridge afflux and compared the results with those of MLR and MNLR methods. They concluded that ANFIS outperformed other estimation methods.

In this study, eight ML models (i.e., SVR, DTR, RFR, ABR, GBR, XGBR, GPR, and KNN) were applied to predict bridge afflux using 202 laboratory and 66 field data. To the best of the authors' knowledge, this is the first time that these ML models have been applied to estimate bridge afflux. Previous studies have also implemented different empirical methods, such as MLR, MNLR, Biery and Delleur's equation, MHBMO-GRG, and GA on the same dataset [4,12–14]. In addition to the empirical equations, Niazkar et al. [12] employed two artificial intelligence techniques, i.e., ANN and GP, to estimate bridge backwater. This study compared the results of eight new ML models with those of the mentioned methods, which were suggested in the previous studies.

Generally, various factors, like the bridge type, may play a role in the bridge afflux phenomenon. This study specifically concentrates on arch bridges. Nevertheless, similar ML models can be used to estimate bridge backwater depth if a dataset for other bridge types is available. Furthermore, impacts of debris against bridges were not considered for estimating bridge backwater, which can be counted as one of the limitations of the present study. Since ML models require a large dataset, the lack of sufficient data in studies conducted prior to a bridge construction poses another limitation for ML implementations. Nonetheless, the proposed ML-based estimation models are not generally recommended, while they can be utilized for cases that have conditions similar to the one considered in this study as influencing parameters were derived through dimensional analysis.

4.1. Discussion of Correlation Results

Correlation diagrams were utilized to better compare the performance of each method in bridge afflux estimation. According to Figure 3a, Biery and Delleur's equation, GA, MHBMO-GRG, and ANN models demonstrated poor correlation with small afflux values in the training data, whereas they displayed acceptable correlations with medium and large values of the bridge afflux. In contrast, based on Figure 4a, Biery and Delleur's equation, GA, and MHBMO-GRG models underestimated bridge afflux for the testing data, whereas the ANN model exhibited an acceptable correlation performance for the testing data. Additionally, the MLR model generally overestimated afflux values for both datasets, while the MNLR model revealed acceptable correlation performances for both datasets (Figures 3a and 4a). Furthermore, the GP model underestimated small afflux values in the training data, while it displayed a good correlation for medium and large bridge afflux values. However, an underestimation of bridge backwater depths was obtained by the GP model for a few data points at different intervals of the testing data.

According to Figure 3, the bridge afflux estimation trend is not accurately modeled by the SVR, DTR, and ABR models, and their correlations are weaker compared to previous methods (Figure 3a). In contrast, the XGBR, GPR, and KNN models demonstrate significantly stronger correlations than other methods. In Figure 4, all methods demonstrate weaker correlations for the testing dataset compared to their performances for the training dataset. Based on the correlation results, the XGBR, GPR, and KNN models are more robust in predicting bridge affluxes.

4.2. Discussion of Perforamnce Metrics Results

According to Table 3 and Figure 5, the MLR, MNLR, and Biery and Delleur's equations demonstrated the weakest performance in terms of RMSE. Furthermore, the GP and ANN models utilized in previous studies had testing RMSE values of 0.08 and 0.07, respectively. None of the new models outperformed this threshold except for the GPR model (testing RMSE of 0.06). Compared to explicit equations, the GPR model demonstrated a better RMSE (i.e., 50%). Regarding MAE, MLR showed the worst performance (testing MAE = 0.12), while other methods from previous studies performed better than the SVR, ABR, and DTR models. After GPR with testing MAE of 0.04, the ANN, GP, KNN, and GBR methods performed adequately with MAE equal to 0.05. Moreover, after GPR with MARE of 0.26,

the GP model indicated the best results with MARE equal to 0.31. Furthermore, regarding MXARE, Biery and Delleur's equation and the GP model outperformed other methods with a testing MXARE value of 0.98. Finally, the SVR and ABR models performed poorly regarding MARE and MXARE.

Among methods previously used for estimating bridge backwater depths, the explicit equations have R^2 values less than 0.8, as shown in Table 3. Furthermore, the MNLR method demonstrated the weakest performance with a training R^2 value of 0.37, while the MHBMO-GRG and GA methods performed the best with R^2 values of 0.78 and 0.79 for the testing data, respectively. In addition, the GP and ANN models utilized in previous studies significantly outperformed other explicit equations with testing R^2 values of 0.84 and 0.89, respectively. Nonetheless, some of the new ML models considered in the present study enhanced their performances of previously suggested models. For example, the GPR model outperformed all methods with a testing R^2 value of 0.91 and a training R^2 value of almost 1, unlike that of the GP model, which was 0.92. On the other hand, while MLR demonstrated a testing R^2 value of 0.61, the DTR and ABR models performed poorly with R^2 values equal to 0.63. Compared to explicit equations, ML models, such as GPR, KNN, GBR, RFR, and XGBR, demonstrated better R^2 values by at least 15%, 11%, 6%, 4%, and 3% improvements, respectively.

Lastly, the NSE results showed results similar to R^2, where the MNLR and MLR methods had the weakest performances with a training NSE value of 0.02 and a testing NSE equal to 0.51, respectively. For the training data, the KNN, XGBR, and GPR models demonstrated the best performances with NSE values of 1, while the GPR model indicated an outperforming testing NSE of 0.91. Although the ML hyperparameters were tuned to prune the overfitting, there is still a significant variance between the training and testing dataset results in a few ML models, such as KNN. To be more specific, despite attempts to adjust values of ML hyperparameters and running the algorithms several times, KNN, XGBR, and GPR exhibited a tendency to fit the training data more, which may suggest overfitting. Nonetheless, the metrics results for the testing data are also satisfactory. For instance, the KNN performance is satisfactory regarding the testing dataset (KNN testing R^2 = 0.88 and NSE = 0.86). Therefore, the overfitting tendency to training data is a shortcoming of a few ML models as they are sensitive to their hyperparameters. Finally, when a large dataset is available, tuning hyperparameters becomes more effective on ML predictions.

4.3. Discussion of Ranking Analysis

The ranking scheme presented in Table 4 provides an overall evaluation of the performance of each method used in this study and previous studies. The results indicate that the GPR method had the best performance, followed by the KNN method, which performed well in the training dataset with lower performance for the testing dataset. The XGBR, GBR, GP, and ANN methods also performed adequately, resulting in a joint third-place ranking. Moreover, the GA-based explicit equation performed better than other empirical equations and a few ML models, such as DTR, ABR, and SVR. Although the MLR method showed the weakest performance and ranked last, the SVR and ABR methods did not demonstrate a better performance than explicit equations developed by previous studies. To be more specific, they were ranked 14th and 12th, respectively, with the MNLR equation ranked 13th. This implies that not all ML models perform better than available explicit equations for predicting bridge afflux.

4.4. Discussion of Reliability Analysis

Regarding the reliability of methods recommended by previous studies, the GP method demonstrated a superior performance of 63.03%. On the other hand, Biery and Delleur´s equation, and the MNLR and MLR methods indicated weaker performances with percentages of 29.86, 28.44, and 14.69, respectively. The reliability of the empirical equations was mostly similar for both datasets, while the reliability of ML models varied depending

on the dataset. Nevertheless, ML models also outperformed the explicit equations for reliability based on the testing data, where the GP and GPR models exhibited the best reliability values with 54.39% and 52.63%, respectively, and the MLR model demonstrated the weakest reliability (12.28%). Comparing the varying reliability percentages across different methods, it can be concluded that most ML models achieved higher reliability percentages compared to those obtained by the empirical equations. However, some ML models, such as SVR and ABR, did not demonstrate a better reliability than the empirical equations, suggesting that choosing the right ML model is essential in bridge afflux estimation.

4.5. Discussion of Feature Importance Analysis

The XGBR feature importance analysis can not only measure the relative importance of each feature in a predictive model but can also help identify the most effective features for more accurate estimations. The findings depicted in Figure 7 propose that Fr_3 has the most significant impact on the bridge afflux, indicating its potential significance in predicting the outcome. Nevertheless, a feature with a low importance value could still be crucial to the overall performance of an estimation model. Therefore, these results should not be interpreted in a way such that an essential feature, like J_1, is insignificant. Lastly, since the feature importance analysis showed the relative importance of Fr_3, it can be recommended that future studies explore the relationship between Fr_3 and afflux depth further to gain a deeper understanding of such influence.

5. Conclusions

Despite the importance of bridge afflux in the safe design of bridges and the stability of riverbanks, few studies have been carried out to develop accurate models for predicting bridge backwater. Based on previous studies, it is postulated that applying ML models can estimate affluxes with higher accuracy than the available empirical equations in the literature. Additionally, the application of ML models requires no prior knowledge of the problem in question. This study applied eight ML models for the first time to predict bridge afflux for 268 laboratory and field data and compared their performances with those of previous methods. All hyperparameters of the ML models considered in this study were tuned, and the ML models were run multiple times to reach the best results. The XGBR, GPR, and KNN models showed stronger correlations and were more robust in predicting bridge afflux than other methods. The GPR model had the best overall performance and ranked as the best method based on the ranking scheme. The KNN model demonstrated the highest accuracy for the training dataset. However, as it tends to overfit the training data, it ranked as the second-best method. The performance of other ML methods, such as XGBR, GBR, and RFR, is also satisfactory. The ML methods utilized in previous studies, i.e., GP and ANN models, demonstrated more accurate results than empirical equations, suggesting the superiority of the ML models. However, a few ML models, such as SVR, ABR, and DTR, did not show a good performance overall, to the extent that after MLR, which ranked the lowest, SVR showed the weakest performance. Moreover, DTR predicted a constant value for some data, which resulted in a stepwise correlation of the model. Furthermore, the SVR, ABR, and DTR models and empirical equations did not show good reliability either. In contrast, the GPR, KNN, GP, XGBR, GBR, and RFR models revealed satisfactory reliability. These findings suggest that not all ML models can improve the estimation accuracy of bridge afflux, and the appropriate selection of an estimation model is essential. The results of this study could be useful for engineers and decision-makers in other study fields since it compares the application of various ML models, providing insights into the performance and suitability of ML models for problems in other domains. Furthermore, future studies can implement ML models for estimating bridge afflux in the corresponding software and evaluate their efficiency, or they can utilize ensemble methods to improve the overall accuracy.

Author Contributions: Conceptualization, R.P. and M.N.; methodology, R.P. and M.N.; software, R.P. and M.N.; validation, R.P., M.N. and A.M.; formal analysis, R.P. and M.N.; investigation, R.P., M.N., S.H.A. and A.M.; resources, R.P., M.N., S.H.A. and A.M.; writing—original draft preparation, R.P. and M.N.; writing—review and editing, R.P., M.N. and A.M.; visualization, R.P. and M.N.; supervision, S.H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cobaner, M.; Seckin, G.; Kisi, O. Initial assessment of bridge backwater using an artificial neural network approach. *Can. J. Civ. Eng.* **2008**, *35*, 500–510. [CrossRef]
2. Hunt, J.; Brunner, G.W.; Larock, B.E. Flow Transitions in Bridge Backwater Analysis. *J. Hydraul. Eng.* **1999**, *125*, 981–983. [CrossRef]
3. Biery, P.F.; Delleur, J.W. Hydraulics of Single Span Arch Bridge Construction. *J. Hydraul. Div.* **1962**, *88*, 75–108. [CrossRef]
4. Mamak, M.; Seckin, G.; Cobaner, M.; Kisi, O. Bridge afflux analysis through arched bridge constrictions using artificial intelligence methods. *Civ. Eng. Environ. Syst.* **2009**, *26*, 279–293. [CrossRef]
5. Pinar, E.; Paydas, K.; Seckin, G.; Akilli, H.; Sahin, B.; Cobaner, M.; Kocaman, S.; Akar, M.A. Artificial neural network approaches for prediction of backwater through arched bridge constrictions. *Adv. Eng. Softw.* **2010**, *41*, 627–635. [CrossRef]
6. Biglari, B.; Sturm, T.W. Numerical Modeling of Flow around Bridge Abutments in Compound Channel. *J. Hydraul. Eng.* **1998**, *124*, 156–164. [CrossRef]
7. Seckin, G.; Haktanir, T.; Knight, D. A simple method for estimating flood flow around bridges. In *Proceedings of the Institution of Civil Engineers-Water Management*; Thomas Telford Ltd.: London, UK, 2007.
8. Seckin, G.; Atabay, S. Experimental backwater analysis around bridge waterways. *Can. J. Civ. Eng.* **2005**, *32*, 1015–1029. [CrossRef]
9. Seckin, G.; Yurtal, R.; Haktanir, T. Contraction and Expansion Losses through Bridge Constrictions. *J. Hydraul. Eng.* **1998**, *124*, 546–549. [CrossRef]
10. Seckin, G.; Akoz, M.S.; Cobaner, M.; Haktanir, T. Application of ANN techniques for estimating backwater through bridge constrictions in Mississippi River basin. *Adv. Eng. Softw.* **2009**, *40*, 1039–1046. [CrossRef]
11. Seckin, G.; Cobaner, M.; Ozmen-Cagatay, H.; Atabay, S.; Erduran, K.S. Bridge afflux estimation using artificial intelligence systems. In *Proceedings of the Institution of Civil Engineers-Water Management*; Thomas Telford Ltd.: London, UK, 2011.
12. Niazkar, M.; Talebbeydokhti, N.; Afzali, S.-H. Bridge backwater estimation: A Comparison between artificial intelligence models and explicit equations. *Sci. Iran.* **2020**, *28*, 573–585. [CrossRef]
13. Brown, P. "Afflux at Arch Bridges", Tech. Rep. Report SR 182, HR Wallingford (1988). Available online: https://eprints.hrwallingford.com/219/ (accessed on 6 June 2023).
14. Pinar, E.; Seckin, G.; Sahin, B.; Akilli, H.; Cobaner, M.; Canpolat, C.; Atabay, S.; Kocaman, S. ANN approaches for the prediction of bridge backwater using both field and experimental data. *Int. J. River Basin Manag.* **2011**, *9*, 53–62. [CrossRef]
15. Niazkar, M. Assessment of artificial intelligence models for calculating optimum properties of lined channels. *J. Hydroinformat.* **2020**, *22*, 1410–1423. [CrossRef]
16. Bisong, E. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*; Springer: Berlin/Heidelberg, Germany, 2019. [CrossRef]
17. Hou, W.; Yin, G.; Gu, J.; Ma, N. Estimation of Spring Maize Evapotranspiration in Semi-Arid Regions of Northeast China Using Machine Learning: An Improved SVR Model Based on PSO and RF Algorithms. *Water* **2023**, *15*, 1503. [CrossRef]
18. Leong, W.C.; Bahadori, A.; Zhang, J.; Ahmad, Z. Prediction of water quality index (WQI) using support vector machine (SVM) and least square-support vector machine (LS-SVM). *Int. J. River Basin Manag.* **2019**, *19*, 149–156. [CrossRef]
19. Lu, H.; Ma, X. Hybrid decision tree-based machine learning models for short-term water quality prediction. *Chemosphere* **2020**, *249*, 126169. [CrossRef]
20. Schapire, R.E. Explaining adaboost. In *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 37–52.
21. Bandara, A.; Hettiarachchi, Y.; Hettiarachchi, K.; Munasinghe, S.; Wijesinghe, I.; Thayasivam, U.A. A generalized ensemble machine learning approach for landslide susceptibility modeling. In *Data Management, Analytics and Innovation: Proceedings of the ICDMAI 2019*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 2, pp. 71–93. [CrossRef]
22. Katipoğlu, O.M.; Sarıgöl, M. Prediction of flood routing results in the Central Anatolian region of Türkiye with various machine learning models. *Stoch. Environ. Res. Risk Assess.* **2023**, *37*, 2205–2224. [CrossRef]
23. Han, Y.; Wu, J.; Zhai, B.; Pan, Y.; Huang, G.; Wu, L.; Zeng, W. Coupling a Bat Algorithm with XGBoost to Estimate Reference Evapotranspiration in the Arid and Semiarid Regions of China. *Adv. Meteorol.* **2019**, *2019*, 9575782. [CrossRef]

24. Müller, A.C.; Guido, S. *Introduction to Machine Learning with Python: A Guide for Data Scientists*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2016.
25. Nugrahaeni, R.A.; Mutijarsa, K. Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification. In Proceedings of the 2016 International Seminar on Application for Technology of Information and Communication (ISemantic), Semarang, Indonesia, 5–6 August 2016.
26. Schulz, E.; Speekenbrink, M.; Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **2018**, *85*, 1–16. [CrossRef]
27. Roushangar, K.; Shahnazi, S. Prediction of sediment transport rates in gravel-bed rivers using Gaussian process regression. *J. Hydroinformat.* **2019**, *22*, 249–262. [CrossRef]
28. Fathabadi, A.; Seyedian, S.M.; Malekian, A. Comparison of Bayesian, k-Nearest Neighbor and Gaussian process regression methods for quantifying uncertainty of suspended sediment concentration prediction. *Sci. Total. Environ.* **2021**, *818*, 151760. [CrossRef]
29. Zheng, H.; Yuan, J.; Chen, L. Short-Term Load Forecasting Using EMD-LSTM Neural Networks with a Xgboost Algorithm for Feature Importance Evaluation. *Energies* **2017**, *10*, 1168. [CrossRef]
30. Niazkar, M.; Zakwan, M. Developing ensemble models for estimating sediment loads for different times scales. *Environ. Dev. Sustain.* **2023**, 1–19. [CrossRef]
31. Zakwan, M.; Niazkar, M. A Comparative Analysis of Data-Driven Empirical and Artificial Intelligence Models for Estimating Infiltration Rates. *Complexity* **2021**, *2021*, 9945218. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.