Development of new computational machine learning models for longitudinal dispersion coefficient determination: case study of natural streams, United States

Article *in* Environmental Science and Pollution Research · May 2022

DOI: 10.1007/s11356-022-18554-y

CITATIONS
33

READS
4,850

10 authors, including:

Hai Tao
Universiti Malaysia Pahang Al-Sultan Abdullah
123 PUBLICATIONS **4,207** CITATIONS
SEE PROFILE

Sinan Salih
Al-Bayan University
96 PUBLICATIONS **4,568** CITATIONS
SEE PROFILE

Atheer Oudah
Al-Ayen University
32 PUBLICATIONS **291** CITATIONS
SEE PROFILE

Sani Isah Abba
Prince Mohammad bin Fahd University
328 PUBLICATIONS **6,916** CITATIONS
SEE PROFILE

Environmental Science and Pollution Research
https://doi.org/10.1007/s11356-022-18554-y

RESEARCH ARTICLE

Development of new computational machine learning models for longitudinal dispersion coefficient determination: case study of natural streams, United States

Hai Tao[1,2,3] · Sinan Salih[4,5] · Atheer Y. Oudah[6,7] · S. I. Abba[8,9] · Ameen Mohammed Salih Ameen[10] · Salih Muhammad Awadh[11] · Omer A. Alawi[12] · Reham R. Mostafa[13] · Udayar Pillai Surendran[14] · Zaher Mundher Yaseen[15,16,17]

Received: 25 July 2021 / Accepted: 4 January 2022

Abstract

Natural streams longitudinal dispersion coefficient (*Kx*) is an essential indicator for pollutants transport and its determination is very important. Kx is influenced by several parameters, including river hydraulic geometry, sediment properties, and other morphological characteristics, and thus its calculation is a highly complex engineering problem. In this research, three relatively explored machine learning (ML) models, including Random Forest (RF), Gradient Boosting Decision Tree (GTB), and XGboost-Grid, were proposed for the *Kx* determination. The modeling scheme on building the prediction matrix was adopted from the well-established literature. Several input combinations were tested for better predictability performance for the *Kx*. The modeling performance was tested based on the data division for the training and testing (70–30% and 80–20%). Based on the attained modeling results, XGboost-Grid reported the best prediction results over the training and testing phase compared to RF and GTB models. The development of the newly established machine learning model revealed an excellent computed-aided technology for the *Kx* simulation.

Keywords Longitudinal dispersion coefficient · Data division · Machine learning · Input variability

Introduction

One of the key factors considered during modeling rivers pollution transport is the longitudinal dispersion coefficient (*Kx*) (Seo and Baek 2004; Kargar et al. 2020; Shihab and Ahmad 2020; Goliatt et al. 2021). This coefficient is usually determined upon completion of the cross-sectional mixing (Deng et al. 2001; Kashefipour and Falconer 2002). However, *Kx* prediction is a tedious task due to the disparity in riverbed and banks, secondary flow development, velocity irregularities, dead zone, etc. Over the past literature, several researchers have focused on studying *Kx*. Hence, many techniques have been developed to the estimation of *Kx* (Elder 1959; Seo and Cheong 1998; Kashefipour and Falconer 2002; Azamathulla and Wu 2011; Sahay 2011; Fischer et al. 1979; Tutmez and Yuceer 2013; Disley et al. 2015; Najafzadeh and Tafarojnoruz 2016; Shareef 2019). Several empirical equations have been developed in the area of the field and experimental measurements; these expressions consider shear velocity, channel width (W), water depth (H), and flow velocity as the major factors affecting *Kx*. However, there are apparent differences in the *Kx* predictions, as summarized in Table 1 (Etemad-Shahidi and Taghipour 2012). Literature studies confirmed the limitation of the introduced empirical formulas for *Kx* estimation with respect to accuracy (Tutmez and Yuceer 2013; Sahin 2014). These formulas have exhibited a significant level of prediction error in *Kx* prediction compared to the natural stream measurements (Noori et al. 2011). Hence, recommendations have been made to develop simpler and more reliable methods of *Kx* prediction in urban streams (Disley et al. 2015).

Over the past, a few decades, machine learning (ML) models have received much attention in diverse engineering and sciences domains (Zhai et al. 2020; Araba et al. 2021; Zahrawi and Mohammad 2021; Zhong et al. 2021). ML models can provide an efficient alternative

Responsible Editor: Marcus Schulz

✉ Zaher Mundher Yaseen
zaheryaseen88@gmail.com

Extended author information available on the last page of the article

Published online: 21 January 2022

Table 1 The reported literature review on the empirical formulations on the *Kx* determination

Nos.	References	Equations
1	(Taylor 1953, 1954)	$\left(\frac{\partial C}{\partial t}\right)+U\left(\frac{\partial C}{\partial x}\right)=Kx\left(\frac{\partial^2 C}{\partial x^2}\right)$
2	(Elder 1959)	$Kx=5.93HU_*$
3	(Fischer 1975)	$\frac{Kx}{HU_*}=0.011\left(\frac{U}{U_*}\right)^2\left(\frac{W}{H}\right)^2$
4	(Liu 1977)	$\frac{Kx}{HU_*}=0.018\left(\frac{U}{U_*}\right)^{0.5}\left(\frac{W}{H}\right)^2$
5	(Iwasa 1991)	$\frac{Kx}{HU_*}=2\left(\frac{W}{H}\right)^{1.5}$
6	(Koussis and Rodríguez-Mirasol 1998)	$\frac{Kx}{HU_*}=0.6\left(\frac{W}{H}\right)^2$
7	(Seo and Cheong 1998)	$\frac{Kx}{HU_*}=5.915\left(\frac{U}{U_*}\right)^{1.428}\left(\frac{W}{H}\right)^{0.62}$
8	(Deng et al. 2001)	$\frac{Kx}{HU_*}=0.15\left(\frac{1}{8\left(0.145+\left(\frac{1}{3520}\right)\left(\frac{W}{H}\right)^{1.38}\right)\left(\frac{U}{U_*}\right)}\right)\left(\frac{W}{H}\right)^{1.667}\left(\frac{U}{U_*}\right)^2$
9	(Kashefipour and Falconer 2002)	$\frac{Kx}{U_*H}=[7.428+1.775\left(\frac{W}{H}\right)^{0.62}\left(\frac{U}{U_*}\right)^2;\text{if}\ \frac{W}{H}>50$
10	(Kashefipour and Falconer 2002)	$\frac{Kx}{U_*H}=10.612\left(\frac{U}{U_*}\right)^2;\text{if}\ \frac{W}{H}>50$
11	(Sahay and Dutta 2009)	$\frac{Kx}{HU_*}=2\left(\frac{U}{U_*}\right)^{1.25}\left(\frac{W}{H}\right)^{0.96}$
12	(Azamathulla and Wu 2011)	$\frac{Kx}{HU_*}=e^{e^{\cos(U/U_*)+((U/U_*)^2/((W/H)+3.956))}}$
13	(Etemad-Shahidi and Taghipour 2012)	$\frac{Kx}{HU_*}=15.49\left(\frac{W}{H}\right)^{0.78}\left(\frac{U}{U_*}\right)^{0.11};\text{if}\ \frac{W}{H}>30.6$ $\frac{Kx}{HU_*}=8.36\left(\frac{W}{H}\right)^{0.61}\left(\frac{U}{U_*}\right)^{0.85};\text{if}\ \frac{W}{H}<30.6$
14	(Li et al. 2013)	$\frac{Kx}{U_*H}=2.282\left(\frac{W}{H}\right)^{0.7613}\left(\frac{U}{U_*}\right)^{1.4713}$
15	(Sahay 2013)	$\frac{Kx}{U_*H}=2\left(\frac{W}{H}\right)^{0.72}\left(\frac{U}{U_*}\right)^{1.37}S_i^{1.52}$
16	(Zeng and Huai 2014)	$\frac{Kx}{HU_*}=5.4\left(\frac{W}{H}\right)^{0.7}\left(\frac{U}{U_*}\right)^{1.13}$
17	(Disley et al. 2015)	$\frac{Kx}{HU_*}=3.563F_r^{-0.4117}\left(\frac{W}{H}\right)^{0.6776}\left(\frac{U}{U_*}\right)^{1.0132}$
18	(Wang et al. 2017)	$\frac{Kx}{HU_*}=17.648\left(\frac{W}{H}\right)^{0.3619}\left(\frac{U}{U_*}\right)^{1.16}$
19	(Alizadeh et al. 2017a)	$\frac{Kx}{HU_*}=5.319\left(\frac{W}{H}\right)^{1.206}\left(\frac{U}{U_*}\right)^{0.075}\text{if}\ \frac{W}{H}\le 28$ $\frac{Kx}{HU_*}=9.931\left(\frac{W}{H}\right)^{0.187}\left(\frac{U}{U_*}\right)^{1.802}\text{if}\ \frac{W}{H}\ge 28$

Kx, longitudinal dispersion coefficient (m^2/s); *C*, cross-sectional average concentration (kg/m^3); *x*, direction of the mean flow; *t*: time (s); *U*, average velocity (m/s); *H*, depth of flow; *W*, channel width; U_*, bed shear capacity; *S*, longitudinal slope of the stream reach (m/m); *i*, counter indices; Fr, Froude number

methodology for addressing the nonlinearity and nonstationary associated problems (Tao et al. 2021b). There have been a remarkable success in the implementation of ML models in hydrology (Tao et al. 2021a), climate (Ghorbani et al. 2017; Tao et al. 2021c), geo-science (Bayatvarkeshi et al. 2021), hydraulic (Bykov et al. 2019), and several others. Regarding the focus of the current research, Tayfur and Singh (2005) presented one of the earliest ML models for *Kx* modeling by developing an artificial neural network (ANN) model for *Kx* prediction in natural streams. They studied reported strong performance of the ANN model in *Kx* prediction compared to the empirical regression-based predictive models. Another study by Noori et al. (2009) investigated two ML models (support vector machine (SVM) and adaptive neuro-fuzzy inference system (ANFIS)) for *Kx* prediction and reported that both models exhibited almost a similar level of *Kx* prediction accuracy. Sahay (2011) presented a backpropagation

ANN model for *Kx* prediction. The performance of the model was evaluated and found to perform better than the practical equations. The proposed ANN model also achieved a better *Kx* prediction accuracy level ($K < 100$ m^2/s and $\frac{W}{H} < 50$). An SVM model for *Kx* prediction in rivers has been developed by Azamathulla and Wu (2011). The observed performance of the model suggested that it can be reliably used for *Kx* estimation. The development of an M5 model tree for *Kx* prediction has been reported by Etemad-Shahidi and Taghipour (2012). The performance of the developed model was better than that of the existing formulas; hence, it can be used as a vital tool for *Kx* prediction. A new gene expression model for *Kx* prediction was developed by Sattar and Gharabaghi (2015) using 150 sets of published geometric and hydraulic parameters data in natural streams in the USA, New Zealand, Canada, and Europe. The outcome of the study showed that the proposed relations can be used for effective and accurate *Kx* prediction in natural streams.

Alizadeh et al. (2017a, b, c) reported using the ANN model for *Kx* prediction in rivers. The ANN models were trained using four metaheuristics—genetic algorithm (GA), imperialist competitive algorithm (ICA), bee algorithm (BA), and cuckoo search (CS) algorithm. The study's outcome showed that the training of the ANN with the metaheuristics resulted in a higher correlation between the predicted and the measured *Kx* values. Hence, the metaheuristics can be used as necessary tools for improving the performance of the traditional ANN models. The study by Mohamad Javad Alizadeh et al. (2017b) reported the development of a cluster-based Bayesian network for *Kx* prediction in natural streams and rivers. The performance of the proposed model was compared with that of the ANN model and other well-known equations and found to be better than that of the existing models and equations. The performance of ML models (i.e., SVM, Gaussian process regression, M5 model tree (M5P), and RF), as well as that of multiple linear regression in *Kx* prediction has been investigated by Kargar et al. (2020) in natural streams. The study found the M5P model with simple formulations to achieve better performance than the other ML models and empirical models and recommended it for efficient *Kx* prediction in rivers. A study by Saberi-Movahed et al. (2020) presented an improved group method of data handling (GMDH) for *Kx* prediction using extreme learning machine (ELM), gravitational search algorithm (GSA), and particle swarm optimization (PSO) as the base models. The performance evaluation showed that the GMDH-ELM model recorded the best performance compared to the other soft computing tools and traditional predictive models.

Although some of the referenced studies were promising in some aspects, some shortcomings were also noted; for instance, the developed equation's *Kx* prediction accuracy depends on the training and testing subsets. Furthermore, most environmental studies rely on the trial-and-error method of subset selection, but this can lead to a situation where the high/low values of two subsets are not equally spread in the training and testing subsets. The other shortcomings of these referenced methods include insufficient feature selection capability, long convergence time, and prolonged-time of reaching optimality. Generally, the accurate estimation of *Kx* estimation remains paramount in environmental studies; hence, there is a need to assess the accuracy level of most of the newly developed ML models to ensure better *Kx* prediction in natural rivers. Therefore, the current research was initiated on these forgoing limitations. The current study investigated the feasibility of three newly explored ML models: Random Forest (RF), Gradient Boosting Decision Tree (GTB), and XGboost-Grid for the *Kx* prediction. The proposed predictive models were tested based on the collected dataset gathered from the open-source literature review. Two different data division were examined the predictability performance of the models. Seven based input combinations were assessed on the development of the applied models were inspired by the adopted literature.

The structure of the article is presented as follows: "Introduction" presents the introductory of the topic. "Material and methods" exhibited the explanation of the dataset and the applied ML models. "Application results and discussion" reported the statistical and graphical results analysis of the adopted predictive models. Finally, "Conclusion" revealed the conclusions of the research findings.

Material and methods

Dataset description

In this study, the dataset was collected from the open-source literature, including geometrical, morphological, hydraulic, and *Kx* parameters with the total number of observations 71 for natural streams from the USA (Tayfur and Vijay 2005). The statistical description of the parameters were reported in the published article (Tayfur and Vijay 2005). As the variance of the data division between training and testing modeling phases has substantial influence on the predictive models performance, 80–20% and 70–30% were tested for data division investigation. The relative shear velocity can be associated with the river bed's roughness and hydrodynamic characteristics (Etemad-Shahidi and Taghipour 2012). The channel shape parameter, given by β=ln (B/H), reflects the vertical irregularities of the river bed (Deng et al. 2001). The channel sinuosity is defined as the channel length ratio to the valley length (Sahay 2013). For more explanation

about the investigated dataset used in the current research, interested readers are recommended to refer to (Tayfur and Vijay 2005).

Random Forest

In ensemble learning, decision trees are widespread base models (Kotsiantis 2013). Despite decision trees' advantages of flexibility, ease of use, and interpretability (Duda et al. 2001), it has many drawbacks, such as its suboptimal efficiency and lack of robustness. The strong learners made up of many trees are called "forests." RF model developed by (Breiman 2001) is an ensemble learning algorithm by the bagging process in which deep trees are coupled to generate a lower-variance output (Fig. 1). RF operates by constructing many trees (ntree) on a bootstrap sample of the training data with replacements drawn from the original observations and outputting the results by taking the majority votes in the classification task or taking the average in regression task (Ali et al. 2021). The overall training process of RF algorithm is illustrated in algorithm 1.

Algorithm#1 Random Forest (RF) algorithm

Input ← Dataset $\boldsymbol{D} = \{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$, Number of Trees $(\boldsymbol{T})$

Output ← RF final ensemble model $(\boldsymbol{f(x)})$

1. **Begin**
2. **For** $\boldsymbol{i = 1, 2, \ldots, T}$ **do**
3. Create a bootstrap sample $(\boldsymbol{D_i})$ with replacement based on dataset
4. Build a fully regression tree $(\boldsymbol{T_i})$ for each bootstrap sample
5. Prune tree to minimize out-of-bag error
6. **End for**
7. Final prediction output $\boldsymbol{f(x) = \frac{1}{n}\sum_{i=1}^{n} T_i(x)}$
8. **End**

Gradient Boosting Decision Tree

The gradient boosting decision tree algorithm, an ensemble learning algorithm based on the boosting method, was introduced by (Friedman 2001). The fundamental principle of boosting is to combine a series of weak base learners into a strong one. A gradient boosting approach depends on sequential training that progressively, additively, and sequentially trains multiple models to optimize the loss function (Tao et al. 2021d). This simple concept approaches to generate new base learners that can be correlated to the loss function's negative gradient. Unlike the conventional boosting algorithm, the previous model's loss function along the path of the gradient has been cut by any current gradient boosting model.

GTB is a gradient boosting algorithm in which decision trees are the base learners. It creates an ensemble of weak decision tree learners by boosting iteratively (Fig. 2). It is conspicuous that each iterative stage of the GTB model generates a new regression tree learned to optimize the negative gradient of the previous weak tree. GTB will better prevent the risk of overfitting, which the decision tree still does. Specifically, GTB has superior robustness since the size of training sets is less likely to impact it, and outliers and insignificant features will not easily alter its efficiency (Lu et al. 2018).

GTB, like any other ML algorithm in training time, uses a dataset $\{x_i, y_i\}_1^n$ of n predictor variables $x = (x_1, \ldots, x_n)$ and y the response variable. $f(x)$ is the approximation function of the response variable $y = f(x)$. The main aim of GTB is to obtain approximation function $\hat{f}(x)$ to a function $f(x)$ that minimize the value of certain loss function L. The loss function that indicates a measure of prediction performance is defined as follows:

$$L = \sum_{i=1}^{n} L\left(y_i, f\left(x_i\right)\right)$$

In boosting process, GTB iteratively construct M different individuals decision trees $h_1(x), \ldots, h_M(x)$ as follows:

$$h_t(x) = \operatorname{argmin} \sum_{i=1}^{n} L\left(y_i, f_{t-1}\left(x_i\right) + h\left(x_i\right)\right)$$

Therefore, the overall ensemble function (strong learner) $\hat{f}(x)$ is expressed as an additive function as follows:

$$\hat{f}(x) = \sum_{t=0}^{M} \hat{f}_t(x) = \sum_{t=0}^{M} \beta_m h_m(x)$$

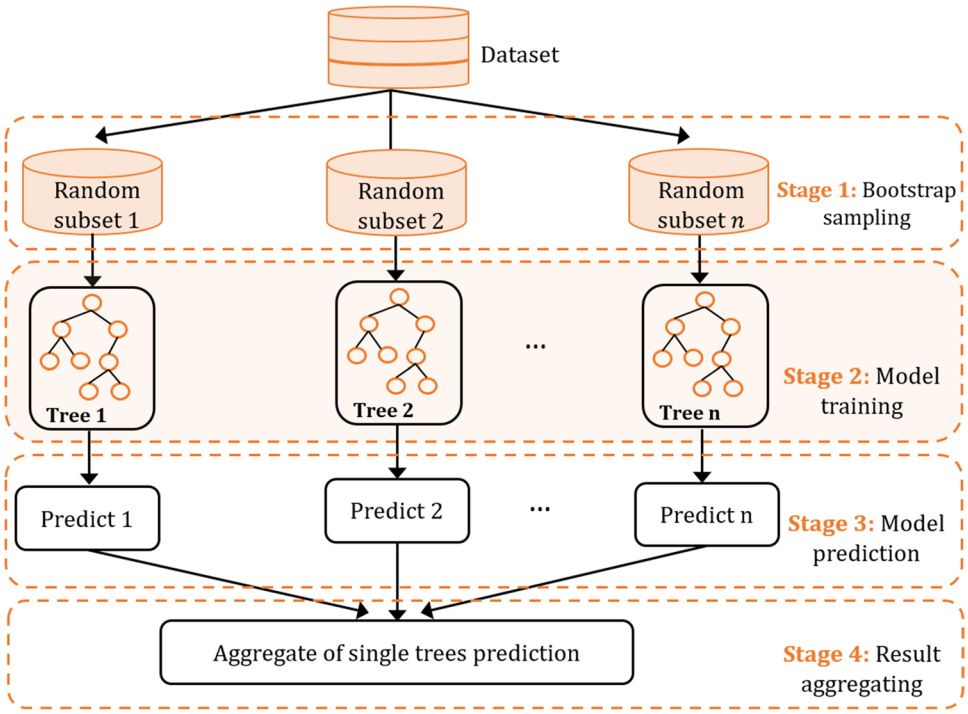

Fig. 1 Random Forest (RF) algorithm example

where $f_0(x)$ is the base learner in which GTB adding DT sequentially, and β_m is the weight of each weak learners in the strong learner.

Algorithm 2 presents the GTB algorithm pseudocode demonstrating how it operates. Assuming that there is a dataset $\{(x_i, y_i)\}_{i=1}^{n}$ and loss function $L(y_i, f(x_i))$, the first step in fitting GTB to the training data is to create the base learner $f_0(x)$ that minimize the loss function. In step 2, the negative gradient (residuals) r_{im} is calculated for each

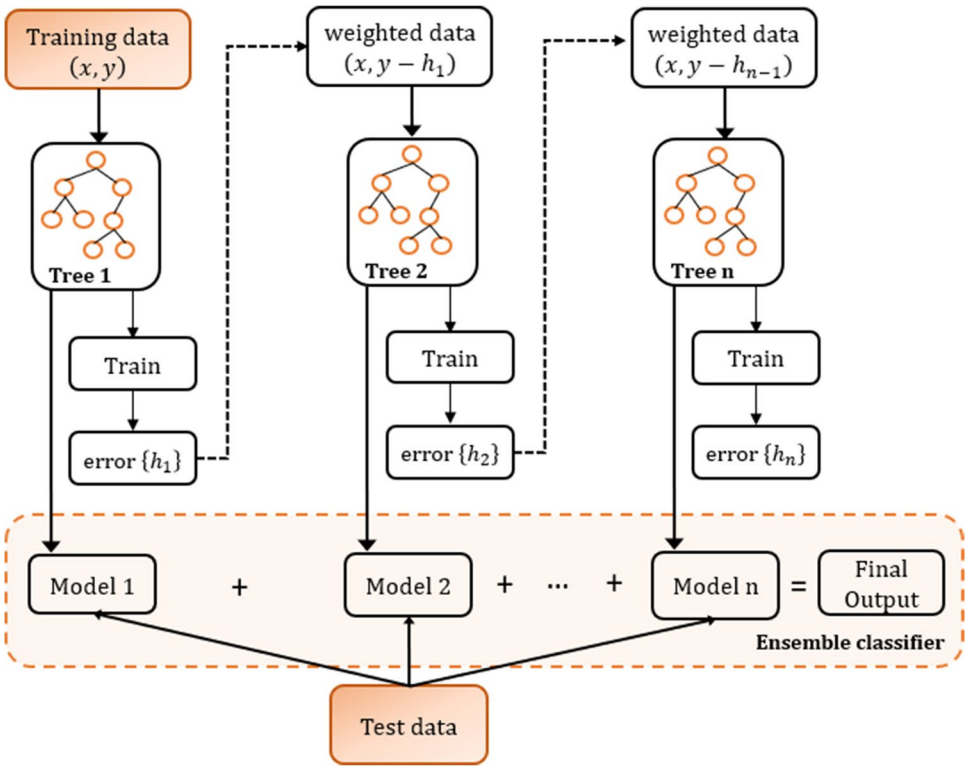

Fig. 2 Gradient boosting decision tree (GBDT)

instance in the dataset (residuals show how good the initial base learner is). In step 3, the regression tree is fitted to the residuals to reduce these errors and create a terminal region R_{jm}. Next, for each defined region, gradient descent step size γ_m is calculated. Finally, the process continues, and the trees are added sequentially to obtain the final prediction $\widehat{f}(x)$.

XGboost-Grid

Though GTB has an excellent record, weaknesses remain, such as slow training pace and reliance on poor learners (Naganna et al. 2020). Chen and Guestrin (2016) proposed an effective implementation of GTB, referred to as XGboost (Extreme Gradient Boosting), to overcome the limitation of GTB. XGboost, like an unextreme gradient boost, uses "residuals error" as a learning approach, in which it fits regression trees to the residuals.

$$\widehat{y}_i = \varnothing(x_i) = \sum_{i=1}^{m} f_m(x_i), f_m \in Z$$

where x_i and $\widehat{y}_i$ are the inputs and observed value, respectively. Z is the functional space of m decision trees. The function f_m are learned by minimizing the following objective function:

$$L(\varnothing) = \sum_{i=1}^{n} l(\widehat{y}_i, y_i) + \sum_{k=1}^{m} \Omega(f_m)$$

where n is the number of predictions, t is the number of iteration, l is the differential loss function. Here, $\widehat{y}_i$ can be given as follows:

$$\widehat{y}_i^t = \widehat{y}_i^{t-1} + f_t(x_i)$$

The regularization term $\Omega(f_m)$ is defined as follows:

Algorithm#2 Gradient Boosting Decision Tree (GTB)

Input ← Dataset $\boldsymbol{D} = \{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$, Base model $\boldsymbol{f(x)}$, Number of iteration ($\boldsymbol{M}$), Loss function $\boldsymbol{L(y_i, f(x_i))}$, hyperparameters.

Output ← **GTB** final ensemble model $\boldsymbol{(\widehat{f}(x))}$

1. **Begin**
2. Create the initial base learner $\boldsymbol{f_0(x) = \min \sum_{i=1}^{n} L(y_i, \beta)}$
3. **For** the number of iterations $\boldsymbol{m = 1, 2, \ldots, M}$ **do**
4. Compute negative gradient $\boldsymbol{r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]}$
5. Fit a new regression tree $\boldsymbol{h_m(x)}$ to the target $\boldsymbol{r_{im}}$ and create terminal region $\boldsymbol{R_{jm}}$, j=1, 2, …, $\boldsymbol{j_m}$
6. Compute the gradient descent step size as $\boldsymbol{\gamma_{jm} = min \sum_{i=1}^{n} L(y_i, f_{m-1}(x) + \gamma_m h_m(x))}$
7. Update the model $\boldsymbol{f_m(x) = f_{m-1}(x) + \gamma_m h_m(x)}$
8. **End for**
9. Output $\boldsymbol{\widehat{f}(x) = \sum_{i=1}^{M} f_m(x)}$
10. **End**

The XGBoost increases the optimization by two main enhancements compared with the GTB. Initially, the XGBoost applies a regularization term to the objective function to increase generalization and avert model overfitting. Next, the XGBoost implements Taylor's second-order expansion on the objective function, compared with the GTB that uses the first derivative in optimization, which allows the XGBoost more efficient in determining the loss function.

The details of the XGboost model are illustrated as follows. Generally, XGboost for the m-th decision tree can be represented as follows:

$$\Omega(f_m) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_i^2$$

where γ is the penalty coefficient, T is the number of leaves, λ is the regularization term, and w is the vector of scores on the leaves. In order to optimize the objective function, XGboost updates it iteratively as follows:

$$L(\varnothing)^m = \sum_{i=1}^{n} l(y_i, \widehat{y}_i^{m-1} + f_m(x_i)) + \Omega(f_m)$$

Using the Taylor expansion, the objective function can be finally derived as follows:

$$L(\varnothing)^m = \sum_{i=1}^{n}\left[l\left(y_i,\hat{y}_i^{m-1}\right) + g_i f_m(x_i) + \frac{1}{2}h_i f_m^2(x_i)\right] + \Omega(f_m)$$

where g_i is the first derivative of loss function and h_i is the second derivative of loss function. g_i and h_i are defined as follows:

$$g_i = \frac{\partial}{\partial \hat{y}^{(t-1)}}\left(l\left(y_i,\hat{y}^{(t-1)}\right)\right)$$

$$h_i = \frac{\partial^2}{\partial^2 \hat{y}^{(t-1)}}\left(l\left(y_i,\hat{y}^{(t-1)}\right)\right)$$

The overall training process of XGboost model is illustrated in Algorithm 3.

The performance of XGBoost models is largely dependent on the appropriate choice of its hyperparameters. In this section, the parameters for XGboost were determined and optimized by grid search algorithm. Grid search is merely an exhaustive search method that configures a grid of possible combinations of hyperparameter values and for each combination trains a model. This approach trials and tests all potential data combinations using k-fold cross-validation methodology.

The hybrid XGboost-grid model is illustrated in Fig. 3, and it comprises the following steps:

Step 1: Define the XGboost searching hyperparameters range.
Step 2: Construct the grid search space of all possible combinations of hyperparameters values.
Step 3: Build an initial XGboost model, and training is performed on training data using k-fold cross-validation.
Step 4: Evaluate the performance of the XGboost model through fitness function (RMSE). If RMSE assures the performance in terms of accuracy, then select the sets of optimal parameters; otherwise, go to the next step.
Step 5: Explore and search multiple parameters with the minimum RMSE value.
Step 6: The range of searching is redefined near the optimal parameters and decreases the search step, then goes back to step 2.
Step 7: Repeat the algorithm from step 2 to step 6 until the value of the hyperparameter is obtained that satisfies the stop condition.
Step 8: Construct the XGboost model with the optimal parameters values, and estimate the output value of test data.

Algorithm#3 XGBoost algorithm

Input ← Dataset $\boldsymbol{D} = \{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$, Base model $\boldsymbol{f(x)}$, Number of iteration ($\boldsymbol{M}$), Loss function ($\boldsymbol{l}$), hyperparameters.

Output ← XGBoost final ensemble model ($\boldsymbol{f(x)}$)

Begin
  Initialize base learner $\boldsymbol{f_0(x)}$
  For $\boldsymbol{m = 1, 2, \ldots, M}$ **do**
    For $\boldsymbol{i = 1, 2, \ldots, n}$ **do**
      Calculate $\boldsymbol{g_i = \partial / \partial \hat{y}^{(t-1)}}\left(\boldsymbol{l}\left(\boldsymbol{y_i}, \hat{\boldsymbol{y}}^{(\boldsymbol{t-1})}\right)\right)$
      Calculate $\boldsymbol{h_i = \partial^2 / \partial^2 \hat{y}^{(t-1)}}\left(\boldsymbol{l}\left(\boldsymbol{y_i}, \hat{\boldsymbol{y}}^{(\boldsymbol{t-1})}\right)\right)$
    End for
    Compute objective function $\boldsymbol{L(\emptyset)^m = \sum_{i=1}^{n}\left[l\left(y_i, \hat{y}_i^{m-1}\right) + g_i f_m(x_i) + \frac{1}{2}h_i f_m^2(x_i)\right] + \Omega(f_m)}$
    Generate a tree $\boldsymbol{\hat{b}(x)}$
    Add trees $\boldsymbol{f_m(x) = f_{m-1}(x) + \varepsilon \hat{b}(x)}$
  End for
  Output $\boldsymbol{f(x) = \sum_{i=1}^{M} f_m(x)}$
End

Performance metrics

The applied predictive models were tested based on several statistical performance metrics and graphical presentations. The performance metrics include root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), Nash-Sutcliffe efficiency (NSE), Willmott's index (WI), and determination coefficient (R^2) (Tung and Yaseen 2020; Yaseen 2021).

Application results and discussion

The assumptions of any computing intelligent models over existing empirical and theoretical equations are that the latter should be superior to the former one, this has been

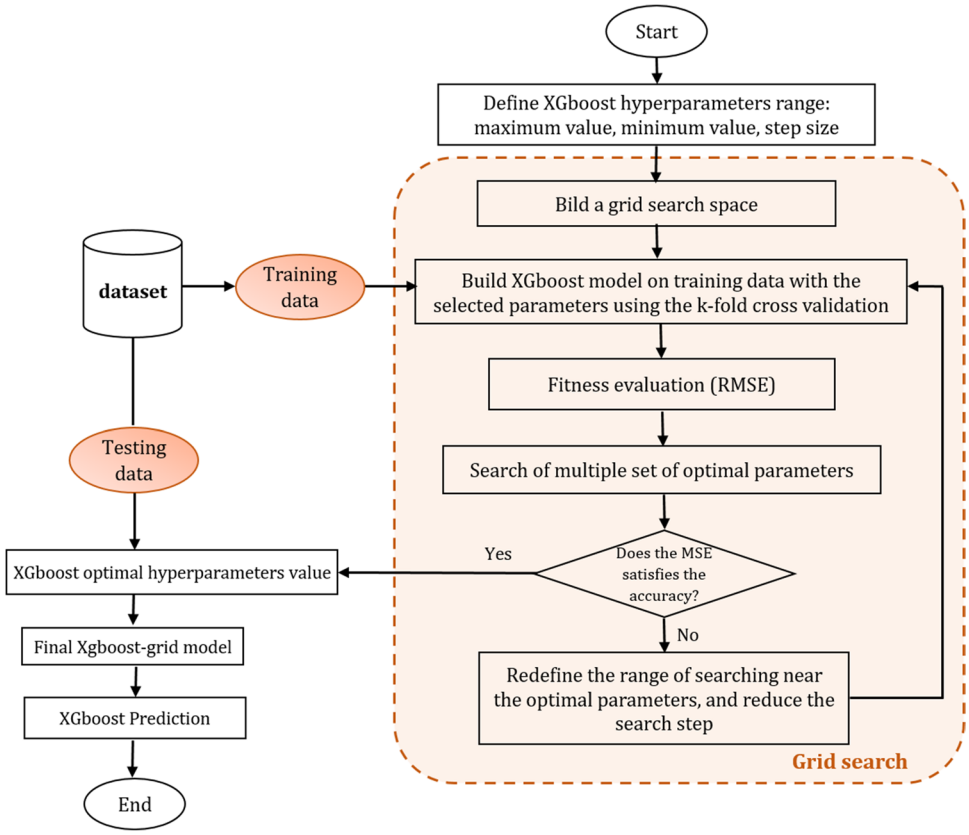

Fig. 3 Flowchart of XGboost-grid model

justified in several engineering applications (Mehr and Akdegirmen 2021; Tao et al. 2021d; Tur and Yontem 2021). In this study, three different emerging data-driven models, including XGboost-Grid, GTB, and RF, were developed to estimate longitudinal dispersion coefficient. Prior to the models development, the dominant input variables selections approach was used which served as significant factor affecting the accuracy of any intelligent computational models. Several input selection approaches such as correlation, autocorrelation, and principal components analysis have been reported in different technical literature but associated with linearity problems. This study employed nonlinear sensitivity analysis to overcome the criticized linear correlation input variable method. The following model is evaluated based on the percentage accuracy: model I (U, H, B), model II (Q), model III (U), model IV (U,β), model V (U,β,σ), model VI ($U/u*$), model VII ($U/u*$, β, σ) as presented in Table 2. The calibrated models were assessed using five performance indictors (RMSE, MAE, MAPE, NSE, WI, R^2); these indices are efficient for assessing the performance of any ML-based models. As reported by Legates and McCabe Jr. (1999) and Elkiran et al. (2019) for satisfactory analysis of any data-driven models, the efficiency criteria should include at least one absolute error (MAE and RMSE), and one goodness-of-fit (R^2) measure.

However, the conventional approach of data division can have a substantial impact on the performance of ML-based models. Despite rapid development in intelligent models, there is a very limited technical systematic approach for accurate data partitioning (Bowden et al. 2002). Generally, data partitioning was carried out using the common and classical approach of splitting the data using a simple random selection method. The distribution of data regarding homogeneous pattern can be understood by considering the statistical characteristic of both training and testing data so that the generalization can be guaranteed. On the other hand, validation such as cross-validation can be carried out to overcome the problems of data arrangement and partition

Table 2 The input combinations for the predictive models construction

Input combinations	Input parameters
Model I	U, H, B
Model II	Q
Model III	U
Model IV	U,β
Model V	U, β,σ
Model VI	$U/u*$
Model VII	$U/u*$, β,σ

Table 3 The modeling performance accuracy for the applied predictive models using 70:30 (training-testing) datasets

	RMSE	MAE	MAPE	NSE	WI	R^2
Model I	**Training phase**					
Grid-XGBoost	6.79	3.07	0.05	0.99	0.99	0.99
GTB	77.5	42.58	0.53	0.83	0.94	0.86
RF	75.72	35.12	0.31	0.84	0.95	0.87
	Testing phase					
Grid-XGBoost	22.80	11.26	0.36	0.99	0.95	0.95
GTB	44.39	34.95	0.99	0.99	0.79	0.81
RF	34.30	23.83	0.79	0.99	0.89	0.90
Model II	**Training phase**					
Grid-XGBoost	4.95	2.23	0.08	0.99	0.99	0.99
GTB	84.24	45.08	0.56	0.80	0.94	0.80
RF	68.32	32.41	0.47	0.86	0.96	0.88
	Testing phase					
Grid-XGBoost	26.77	17.13	0.91	0.99	0.93	0.93
GTB	56.71	33.61	0.70	0.98	0.69	0.71
RF	54.58	31.95	0.92	0.97	0.84	0.88
Model III	**Training phase**					
Grid-XGBoost	24.44	11.26	0.49	0.98	0.99	0.98
GTB	67.12	44.19	0.96	0.87	0.95	0.93
RF	89.13	45.51	0.94	0.77	0.91	0.85
	Testing phase					
Grid-XGBoost	35.26	22.10	0.79	0.99	0.87	0.89
GTB	51.26	34.47	0.93	0.99	0.38	0.86
RF	47.99	32.19	0.77	0.99	0.54	0.85
Model IV	**Training phase**					
Grid-XGBoost	75.34	40.57	0.51	0.84	0.94	0.93
GTB	118.12	56.62	0.72	0.60	0.82	0.74
RF	83.93	50.95	0.79	0.80	0.93	0.83
	Testing phase					
Grid-XGBoost	52.87	35.40	0.87	0.99	0.46	0.82
GTB	51.77	38.39	0.96	0.99	0.50	0.76
RF	53.02	37.65	0.89	0.99	0.47	0.74
Model V	**Training phase**					
Grid-XGBoost	29.60	18.22	0.28	0.97	0.99	0.98
GTB	82.43	46.48	0.94	0.80	0.93	0.84
RF	78.99	47.30	0.94	0.82	0.94	0.88
	Testing phase					
Grid-XGBoost	28.01	20.68842	0.89	0.99	0.90	0.93
GTB	36.69	30.13219	0.91	0.99	0.88	0.90
RF	52.85	36.39961	0.99	0.99	0.64	0.74
Model VI	**Training phase**					
Grid-XGBoost	10.25	4.90	0.16	0.99	0.99	0.99
GTB	70.87	46.36	0.86	0.85	0.95	0.89
RF	99.09	53.63	0.95	0.72	0.89	0.78
	Testing phase					
Grid-XGBoost	58.82	61.12	0.94	0.99	0.68	0.88
GTB	47.21	34.76	0.96	0.99	0.85	0.87
RF	45.95	35.12	0.97	0.99	0.79	0.81
Model VII	**Training phase**					
Grid-XGBoost	53.91	12.10	0.09	0.91	0.97	0.91
GTB	98.59	57.87	0.97	0.72	0.89	0.79

Table 3 (continued)

	RMSE	MAE	MAPE	NSE	WI	R^2
RF	87.77	52.75	0.99	0.78	0.91	0.85
	Testing phase					
Grid-XGBoost	41.50	25.76	0.52	0.99	0.72	0.85
GTB	51.94	39.22	0.99	0.99	0.53	0.76
RF	49.62	35.23	0.97	0.99	0.68	0.80

phases (Abba et al. 2019). In this paper, the modeling was carried out using two different data splitting methods including, 70/30% and 80/20% for both training/testing phases. Table 3 presents the direct comparison of computational models (XGboost-Grid, GTB, and RF) using 70% training and 30% testing approach. From Table 3, it can be seen that the prediction accuracy of all the seven models' combinations (model I to model VII) are relatively high, which indicated that the output trend pattern of the models matched the original observational data of *Kx*, this can be observed by considering the values of goodness-of-fit ($R^2 > 0.7$) in both training and testing phase.

However, the direct comparison of the results depicted that XGboost-Grid with the combination of model I (*U, H, B)* outperformed the other models with regards to MAE (model I (MAE=11.26)) in the testing phase, hence emerged the best prediction model. Furthermore, the hierarchical assessment of the models shows that for model I, model II, model III, model IV, and model VII follows the model superiority patterns of XGboost-Grid > RF> GTB, model V(XGboost-Grid > GTB > RF), and model VI (GTB > RF > XGboost-Grid). These hierarchical results obviously indicated that no exceptional model behaves uniformly and superiority to all the combinations. Hence, employing different types of models could lead to a logical conclusion of any nonlinear system, especially predicting *Kx* in natural streams from flow variables and geometric channel characteristics. Box plot was generated to present the models results distribution compared to the observed dataset (Fig. 4). Thus, the variation of XGboost-Grid (model I) is similar to the observational data spread but different from the quantitative accuracy of the models. Moreover, box plots are used to illustrate overall trends of response for a group. They offer a useful means for the range and other characteristics of responses for a large group to be visualized. They serve as a graphical representation of knowledge that illustrates statistical explanation and dataset (Abba et al. 2020a, b). The box plot of all the models clearly indicated that the spread of data indicated minimum, first quartile, median, third quartile, and maximum are associated with XGboost-Grid model.

Figure 5 shows the box plot modeling results for the applied predictive models using 20% of the dataset for the testing phase. According to the extent of spread of XGboost-Grid (model I) precisely resembles the actual *Kx* values; this indicated that XGboost-Grid (model I) proved merit over other models' combinations. It is worth mentioning that the superiority of XGboost-Grid in comparison to standalone ML-based models have been displayed in various studies (Abba et al. 2020a, b; Chen et al. 2015; Chen et al. 2019; Hadi et al. 2019). Similarly, the hierarchical comparison of the models in terms of MAE shows the dominance trend of XGboost-Grid > GTB > RF belonging to all model combinations except model I with XGboost-Grid > RF> GTB in the testing phase. This statement could be justified by considering the numerical statistical indices presented in Table 4. The goodness-of-fit for all the modeling results attained satisfactory accuracy with the range of (R^2 = 0.99–0.72) in the testing stage. The box plot modeling results for the applied predictive models using 20% of the dataset for the testing phase (Fig. 5).

The Taylor diagrams offer a means to graphically illustrate how exactly a pattern (or group of patterns) fits observations (Figs. 6 and 7). The diagram highlighted different statistical indices, including the standard deviation between the observed and predicted values, RMSE, and correlation (R) (Taylor 2001). In addition, Taylor diagrams offer a means to graphically illustrate how exactly a pattern (or group of patterns) fits observations. Their correlation, centered RMSE difference and standard deviations, can quantify the similarity between two patterns. These graphs are beneficial in analyzing various facets of complex models or in calculating the relative ability of several different models (Taylor 2001). Figures 6 and 7 visualizations use three performance metrics for the applied predictive models using 30 and 20% of the dataset for the testing phase, respectively. Among the multiple performances summarized by Taylor diagram, RMSE plays a vital quantifier and could be served as a good numerical comparison indicator. Therefore, quantitative analysis of the model with regards to error depicted that RMSE ranges from 22.84 to 58.82, and 7.24 to 37.14, in the testing phase for 70/30, and 80/20 data division, respectively. The prediction results also indicated that for the prediction of *Kx* the lower RMSE = 7.24 was achieved by XGboost-Grid model using 80/20 division, in another words, the prediction error of 80/20 was reduced by 15% in comparison with 70/30 scenario. This can be concluded that 80/20 partitioning of the data is recommended for *Kx* modeling. The overall average prediction accuracy of this

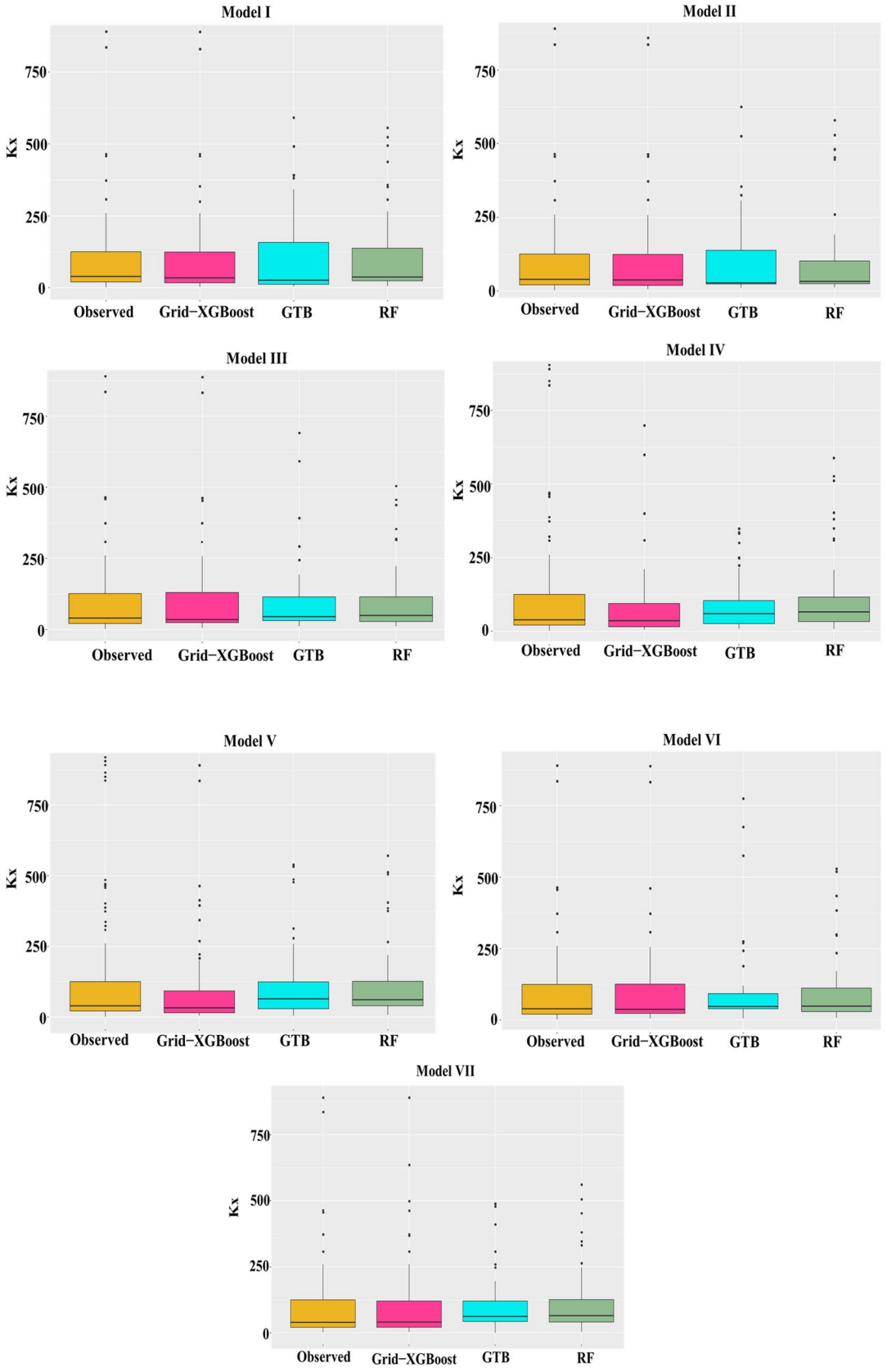

Fig. 4 The box plot modeling results for the applied predictive models using 30 percentage of the dataset for the testing phase

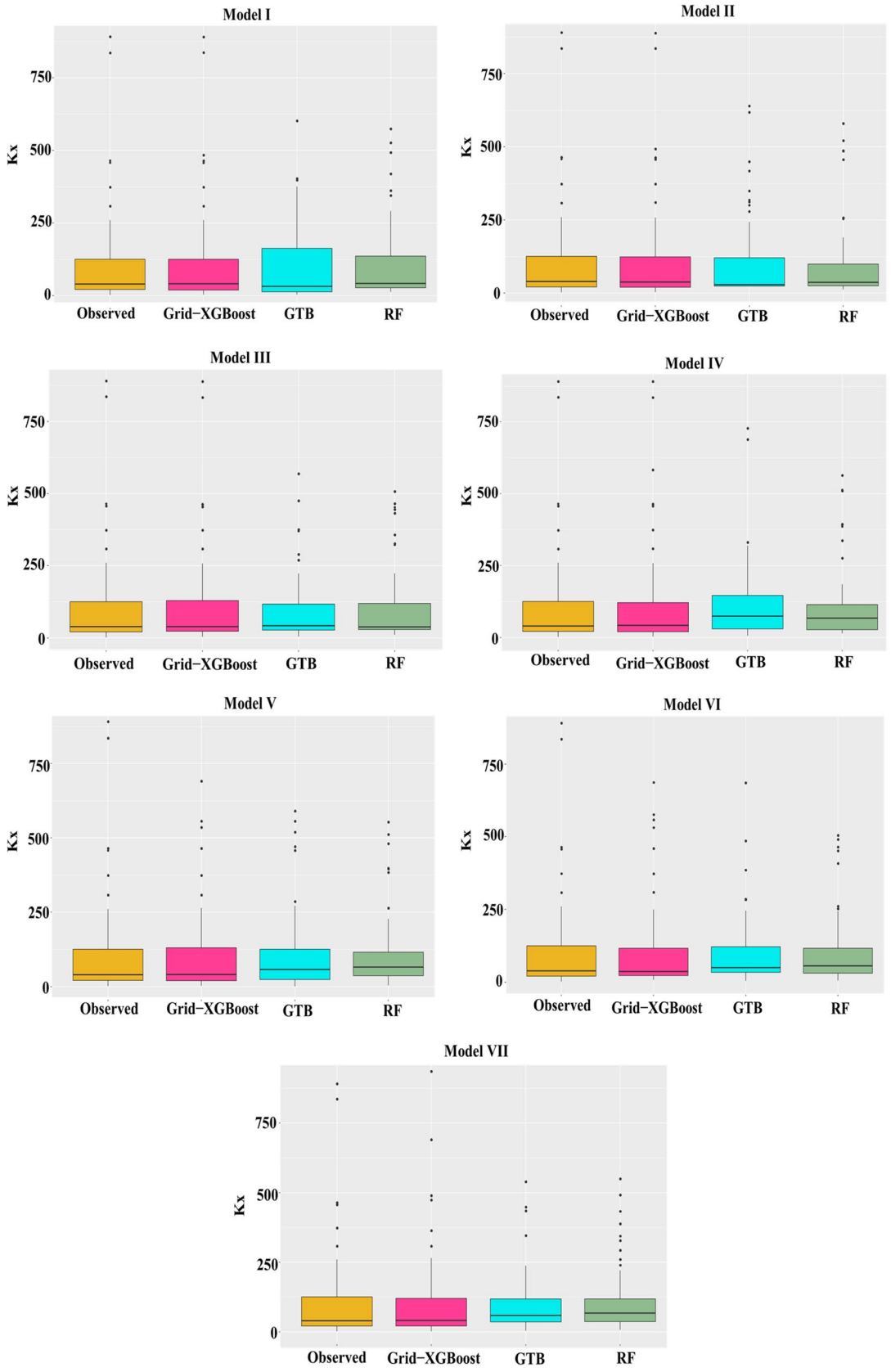

Fig. 5 The box plot modeling results for the applied predictive models using 20 percentage of the dataset for the testing phase

Table 4 The modeling performance accuracy for the applied predictive models using 80:20 (training-testing) datasets

	RMSE	MAE	MAPE	NSE	WI		R^2
Model I	**Training phase**						
Grid-XGBoost	3.07	0.92	0.03	0.99	0.99		0.99
GTB	86.42	43.21	0.70	0.78	0.97		0.79
RF	70.65	34.66	0.61	0.85	0.98		0.89
	Testing phase						
Grid-XGBoost	7.24	3.10	0.10	0.99	0.99		0.98
GTB	23.30	19.74	0.48	0.97	0.96		0.87
RF	20.9	15.85	0.39	0.97	0.97		0.90
Model II	**Training phase**						
Grid-XGBoost	4.27	1.96	0.08	0.99	0.99		0.99
GTB	63.24	36.08	0.85	0.88	0.98		0.89
RF	64.92	30.81	0.59	0.87	0.98		0.88
	Testing phase						
Grid-XGBoost	10.99	7.715	0.33	0.99	0.99		0.96
GTB	22.33	16.84	0.37	0.98	0.96		0.87
RF	21.70	17.59	0.43	0.98	0.97		0.89
Model III	**Training phase**						
Grid-XGBoost	25.23	12.178	0.58	0.98	0.99		0.98
GTB	87.86	51.37	0.89	0.77	0.97		0.84
RF	93.84	49.47	0.99	0.74	0.96		0.76
	Testing phase						
Grid-XGBoost	33.41	18.95	0.33	0.99	0.91		0.72
GTB	35.77	25.11	0.54	0.99	0.87		0.67
RF	37.14	26.74	0.53	0.99	0.87		0.62
Model IV	**Training phase**						
Grid-XGBoost	15.85	3.16	0.07	0.99	0.99		0.99
GTB	72.10	44.01	0.89	0.85	0.96		0.87
RF	79.16	46.64	0.96	0.81	0.97		0.85
	Testing phase						
Grid-XGBoost	25.26	12.57	0.24	0.99	0.94		0.87
GTB	30.30	20.42	0.43	0.99	0.91		0.79
RF	33.60	25.92	0.53	0.99	0.87		0.76
Model V	**Training phase**						
Grid-XGBoost	56.28	14.52	0.04	0.91	0.97		0.93
GTB	77.38	47.33	0.99	0.82	0.97		0.85
RF	75.39	44.13	0.98	0.83	0.98		0.89
	Testing phase						
Grid-XGBoost	31.27	15.47	0.33	0.99	0.93	0.82	
GTB	28.46	18.79	0.37	0.99	0.93	0.78	
RF	32.26	24.64	0.56	0.99	0.89	0.75	
Model VI	**Training phase**						
Grid-XGBoost	53.61	18.68	0.30	0.91	0.98		0.92
GTB	75.22	44.13	0.80	0.83	0.97		0.88
RF	93.86	51.84	0.90	0.74	0.96		0.78
	Testing phase						
Grid-XGBoost	25.54	12.30	0.46	0.99	0.95		0.83
GTB	34.07	23.18	0.80	0.99	0.92		0.79
RF	35.92	24.32	0.77	0.99	0.91		0.73
Model VII	**Training phase**						
Grid-XGBoost	44.20	13.02303	0.03	0.94	0.98		0.94
GTB	86.13	50.1142	0.93	0.78	0.96		0.84

Table 4 (continued)

	RMSE	MAE	MAPE	NSE	WI	R^2
RF	85.63	49.32019	0.99	0.78	0.967	0.85
Testing phase						
Grid-XGBoost	26.04	14.40	0.35	0.99	0.95	0.82
GTB	28.92	20.27	0.64	0.99	0.93	0.79
RF	30.68	26.07	0.74	0.99	0.92	0.77

study attained satisfactory accuracy regarding robustness and estimation accuracy compared to the research scenario conducted by Tayfur and Singh (2005) that obtained the range of RMSE (21.2–183.0) using ANN model. However, both 70/30 and 80/20 proved promising and reliable results for XGboost-Grid with the combination of model I (*U, H, B)*. This conclusion is in line with the one attained by Tayfur and Singh (2005).

Besides RMSE, and MAE indicators, MAPE is another essential index that offers a better approach in understanding the reliability of the predicted model with regards to the percentage error. As for other relative errors, the smaller the MAPE, the more satisfactory the prediction model. It can be observed from Tables 3 and 4 that the range of MAPE values is 0.36–0.99, and 0.10–0.77 in the testing phase for 70:30 and 80:20, respectively. The forgoing MAPE range values proved that all the models for both data divisions attained the fitted prediction accuracy. It is also stated by Gaya et al. (2013), (2020) that the appropriate range of MAPE is 0–1 which proved the above results. The results also show the capability of all three models in mimicking the pattern trend of *Kx*, for all the input variables. To continue with the numerical evaluation of the predicting models, according to the range of MAPE values it can be depicted that 80/20 data division reduced the average prediction percentage error by 24% compared to 70/30. Further visualization of the results is presented using scatter plots to provide a deeper examination (Figs. 8 and 9).

This assessment is essential to obtain a logical goodness-of-fit trend, as reflected in Fig. 8 using 30 percentage of dataset for the testing phase. Figure 8 visualized variations of goodness-of-fit follows the range of XGboost-Grid (0.82–0.95), GTB (0.71–0.9), and RF (0.74–0.9); hence, XGboost-Grid attained the highest estimation skills and averagely increased the prediction accuracy by 8% over GTB, and RF for both lower and higher values of R^2. Similarly, for 80:20, the R^2 follows the range of XGboost-Grid (0.76–0.98), GTB (0.67–0.87), and RF (0.62–0.9) with an average accuracy increase of 10% by XGboost-Grid over GTB, and RF for both lower and higher values of R^2. The overall comparison depicted that high prediction skills are associated with XGboost-Grid (model I) with 80:20 partitioning set. The enhancement in term of discrepancy and mutual agreements of NSE, and WI shows the proposed XGboost-Grid has the highest computing goodness-of-fit and lowest absolute error than GTB, and RF models, as such served as an evident in the generalization ability by displaying optimum consistency in predicting the *Kx* values. It is indeed paramount to validate and compare the present work with other established technical studies in a similar manner. For example, Soltani-gerdefaramarzi et al. (2015) proposed ANFIS model to predict *Kx* in which the best results was found to have RMSE = 72.21 and R^2 = 0.87. Noori et al. (2016) proposed different ML-based model to predict *Kx* the obtained results depicted that the R^2 (ANN = 0.86, ANFIS = 0.87, and SVM = 0.94). Alizadeh et al. (2017a, b, c) achieved 70–83% accuracy using Bayesian cluster network. More recent state-of-art comparison was done by considering the studies of Kargar et al. (2020) that employed SVM, GPR, M5P, and RF and achieved the best results using M5P with RMSE = 454.9, and R = 0.823. Similarly, Madvar et al. (2020) used hybrid models to predict the *Kx* and the outcomes show that RMSE= 81.47, and 137.39 for ANN-POS, and ANN-CSO, respectively. The aforementioned comparison on the longitudinal dispersion declared that ML-based models and optimization algorithms are capable of capturing chaotic system, and the outcomes of the current study depicted that by proposing the emerging boosting algorithm for example XGboost-Grid the prediction accuracy improved with lowest absolute error attained.

Conclusion

The motivation of the current research was to develop a robust and reliable neurocomputing model for natural streams longitudinal dispersion coefficient prediction. Three different types of ML models were developed for this purpose called Random Forest, Gradient Boosting Decision Tree, and XGboost-Grid. The prediction of the *Kx* is very significant for multiple river engineering sustainability and management. The models were constructed based on several river hydraulic geometry, sediment properties, and other morphological characteristics. The input combinations were built following the adopted scheme by the literature. The modeling performance was tested based on the data

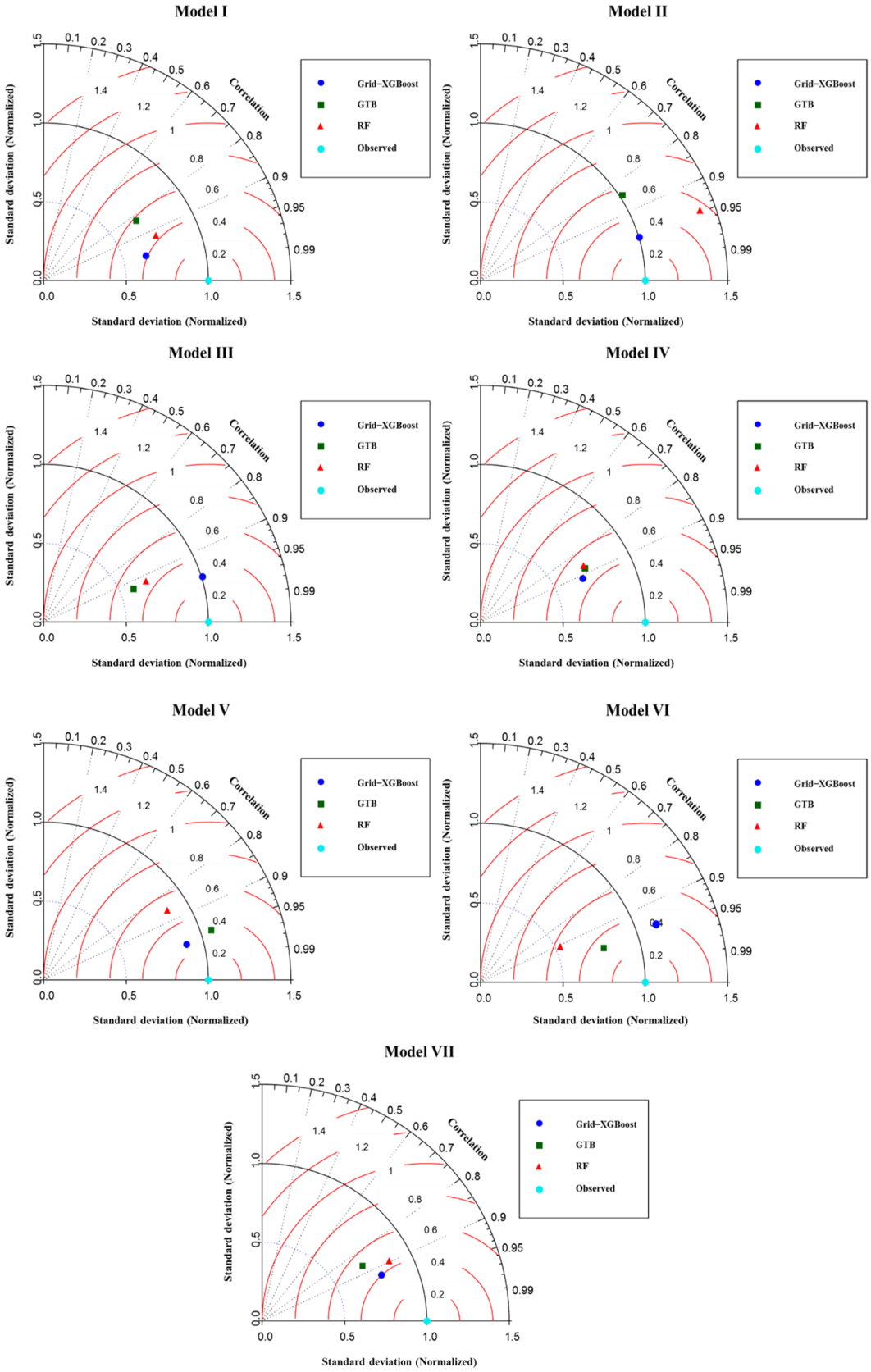

Fig. 6 Taylor diagram visualization using three performance metrics for the applied predictive models using 30 percentage of the dataset for the testing phase

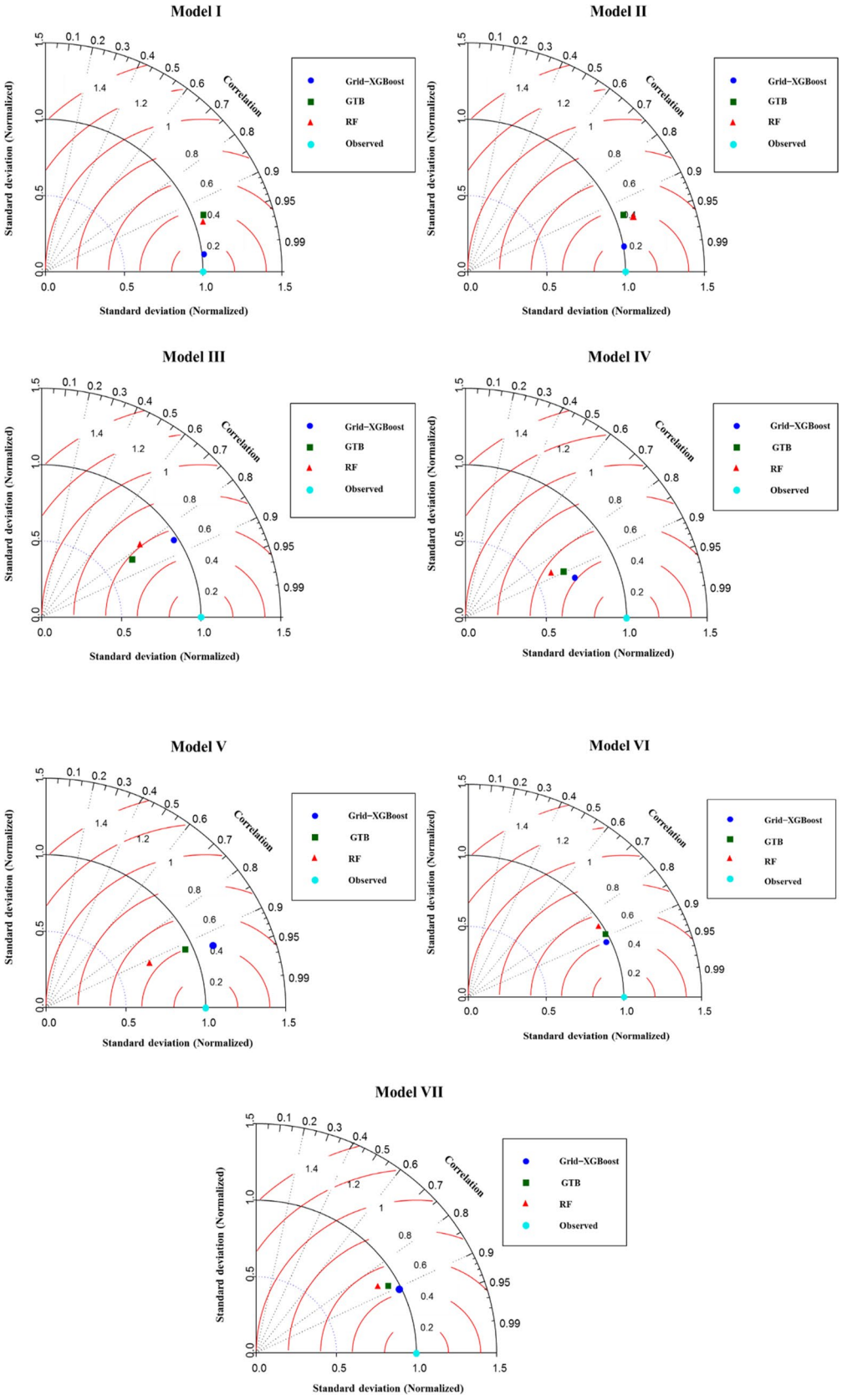

Fig. 7 Taylor diagram visualization using three performance metrics for the applied predictive models using 20 percentage of the dataset for the testing phase

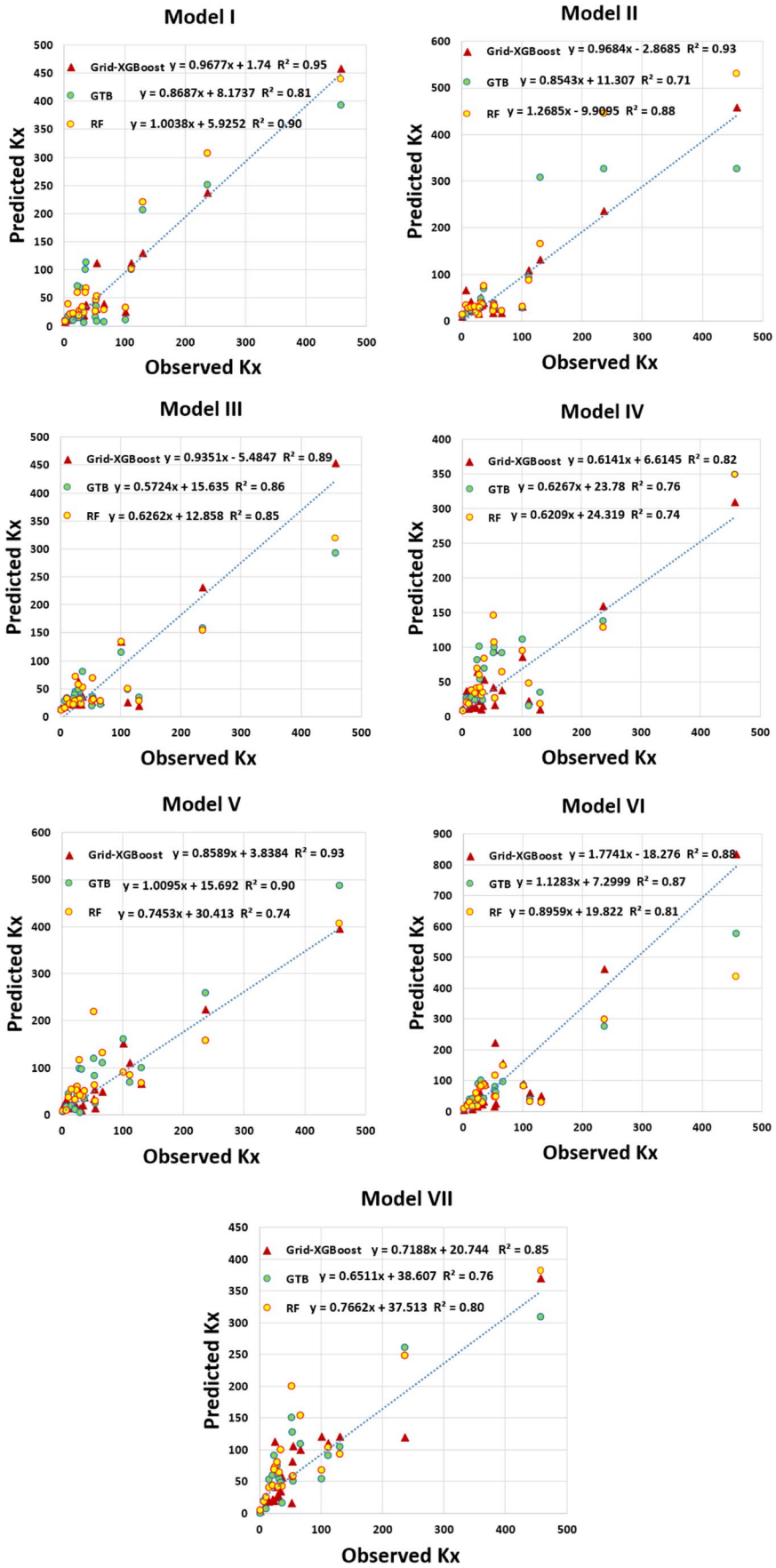

Fig. 8 Scatter plots between the observed and predicted *Kx* using the applied predictive models for the 30 percentage of dataset for testing phase

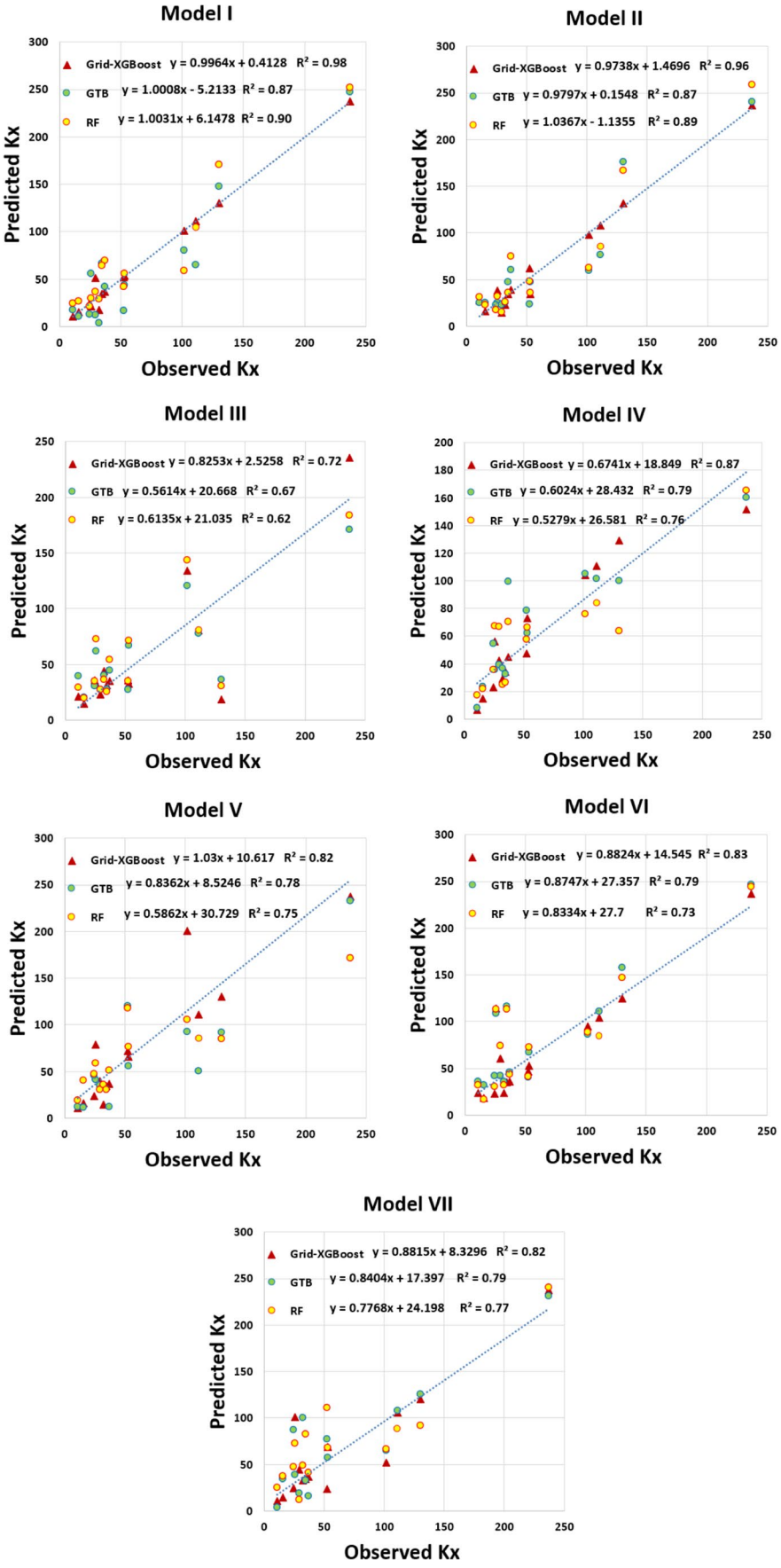

Fig. 9 Scatter plots between the observed and predicted *Kx* using the applied predictive models for the 30 percentage of dataset for testing phase

division for the training-testing modeling phases (70–30% and 80–20%). The modeling results indicated that the developed XGboost-Grid reported the best prediction results over the training and testing phase compared to RF and GTB models. The development of the new established machine learning model revealed an excellent computed-aided technology for the Kx simulation.

Abbreviations Kx: longitudinal dispersion coefficient; ML: machine learning; RF: Random Forest; GTB: Gradient Boosting Decision Tree; W: channel width; H: water depth; C: cross-sectional average concentration; *x*: direction of the mean flow; *t*: time; *U*: average velocity; *W*: channel width; U_*: bed shear capacity; *S*: longitudinal slope of the stream reach; *i*: counter indices; Fr: Froude number; ANN: artificial neural network; SVM: support vector machine; ANFIS: adaptive neuro-fuzzy inference system; GA: genetic algorithm; ICA: imperialist competitive algorithm; BA: bee algorithm; CS: cuckoo search; M5P: M5 model tree; GMDH: group method of data handling; ELM: extreme learning machine; GSA: gravitational search algorithm; PSO: particle swarm optimization; RMSE: root mean square error; MAE: mean absolute error; MAPE: mean absolute percentage error; NSE: Nash-Sutcliffe efficiency; WI: Willmott's Index; R^2: determination coefficient; R: correlation; XGboost: extreme gradient boosting

Author contribution Hai Tao: conceptualization, supervision, writing up, validation. Sinan Salih: data collection, writing up, research investigation. Atheer Oudah: writing up, validation, discussion and analysis. Sani Abba: writing up, research investigation, assessment. Ameen Mohammed Salih Ameen: writing up, discussion and analysis. Salih Muhammad Awadh: writing up, discussion and analysis. Omer A. Alawi: writing up, discussion and analysis. Reham Mostafa: modeling, methodology, writing up, analysis. Udayar Surendran: supervision, conceptualization, writing up, analysis. Zaher Mundher Yaseen: project leader, supervision, assessment, validation, writing and revision, conceptualization.

Data availability Data were obtained from the open source of the literature and can be shared upon request.

Declarations

Ethical approval The manuscript is conducted within the ethical manner advised by the Environmental Science and Pollution Research.

Consent to publish The research is scientifically consented to be published.

Conflict of interest The authors declare no competing interests.

References

Abba SI, Hadi SJ, Sammen SS et al (2020) Evolutionary computational intelligence algorithm coupled with self-tuning predictive model for water quality index determination. J Hydrol 587:124974. https://doi.org/10.1016/j.jhydrol.2020.124974

Abba SI, Linh NTT, Abdullahi J et al (2020) Hybrid machine learning ensemble techniques for modeling dissolved oxygen concentration. IEEE Access 8:157218–157237. https://doi.org/10.1109/ACCESS.2020.3017743

Ali M, Prasad R, Xiang Y et al (2021) Variational mode decomposition based random forest model for solar radiation forecasting: new emerging machine learning technology. Energy Rep 7:6700–6717

Alizadeh MJ, Ahmadyar D, Afghantoloee A (2017) Improvement on the Existing equations for predicting longitudinal dispersion coefficient. Water Resour Manag 31:1777–1794. https://doi.org/10.1007/s11269-017-1611-z

Alizadeh MJ, Shabani A, Kavianpour MR (2017) Predicting longitudinal dispersion coefficient using ANN with metaheuristic training algorithms. Int J Environ Sci Technol 14:2399–2410. https://doi.org/10.1007/s13762-017-1307-1

Alizadeh MJ, Shahheydari H, Kavianpour MR et al (2017) Prediction of longitudinal dispersion coefficient in natural rivers using a cluster-based Bayesian network. Environ Earth Sci 76:86

Araba AM, Memon ZA, Alhawat M et al (2021) Estimation at completion in civil engineering projects: review of regression and soft computing models. Knowledge-Based Eng Sci 2:1–12

Azamathulla HM, Wu F-C (2011) Support vector machine approach for longitudinal dispersion coefficients in natural streams. Appl Soft Comput 11:2902–2905

Bayatvarkeshi M, Bhagat SK, Mohammadi K et al (2021) Modeling soil temperature using air temperature features in diverse climatic conditions with complementary machine learning models. Comput Electron Agric 185:106158. https://doi.org/10.1016/j.compag.2021.106158

Bowden GJ, Maier HR, Dandy GC (2002) Optimal division of data for neural network models in water resources applications. Water Resour Res 38:2-1-2–11. https://doi.org/10.1029/2001wr000266

Breiman L (2001) Random forests. Mach Learn 45:5–32. https://doi.org/10.1023/A:1010933404324

Bykov AD, Voronov VI, Voronova LI (2019) Machine learning methods applying for hydraulic system states classification. In: 2019 Systems of Signals Generating and Processing in the Field of on Board Communications. IEEE, pp 1–4

Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, pp 785–794

Chen T, He T, Benesty M et al (2015) Xgboost: extreme gradient boosting. R Packag version 04–2:1–4

Chen Z-Y, Zhang T-H, Zhang R et al (2019) Extreme gradient boosting model to estimate PM2. 5 concentrations with missing-filled satellite data in China. Atmos Environ 202:180–189

Deng Z-Q, Singh VP, Bengtsson L (2001) Longitudinal dispersion coefficient in straight rivers. J Hydraul Eng 127:919–927. https://doi.org/10.1061/(ASCE)0733-9429(2001)127:11(919)

Disley T, Gharabaghi B, Mahboubi AA, Mcbean EA (2015) Predictive equation for longitudinal dispersion coefficient. Hydrol Process. https://doi.org/10.1002/hyp.10139

Duda RO, Hart PE, Stork DG (2001) Pattern classification. New York John Wiley, Sect

Elder JW (1959) The dispersion of marked fluid in turbulent shear flow. J Fluid Mech 5:544–560. https://doi.org/10.1017/S0022112059000374

Elkiran G, Nourani V, Abba SI (2019) Multi-step ahead modelling of river water quality parameters using ensemble artificial intelligence-based approach. J Hydrol 577:123962. https://doi.org/10.1016/j.jhydrol.2019.123962

Etemad-Shahidi A, Taghipour M (2012) Predicting longitudinal dispersion coefficient in natural streams using M5´ model tree. J Hydraul Eng 138:542–554. https://doi.org/10.1152/ajpcell.00303.2005

Fischer HB (1975) Discussion of 'simple method for predicting dispersion in streams' by RS McQuivey and TN Keefer. J Environ Eng 504:3

Fischer HB, List EJ, Koh RCY, Imberger J, Brooks (1979) Mixing in inland and coastal waters. Academic, New York, pp 104–138
Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232. https://doi.org/10.1214/aos/1013203451
Gaya MS, Abba SI, Abdu AM, Tukur AI (2020) Estimation of water quality index using artificial intelligence approaches and multilinear regression. 9:126-134. doi: https://doi.org/10.11591/ijai.v9.i1.pp126-134
Gaya MS, Wahab NA, Sam YM et al (2013) ANFIS modelling of carbon removal in domestic wastewater treatment plant. Appl Mech Mater 372:597–601. https://doi.org/10.4028/www.scientific.net/AMM.372.597
Ghorbani MA, Deo RC, Yaseen ZM et al (2017) Pan evaporation prediction using a hybrid multilayer perceptron-firefly algorithm (MLP-FFA) model: case study in North Iran. Theor Appl Climatol. https://doi.org/10.1007/s00704-017-2244-0
Goliatt L, Sulaiman SO, Khedher KM et al (2021) Estimation of natural streams longitudinal dispersion coefficient using hybrid evolutionary machine learning model. Eng Appl Comput Fluid Mech 15:1298–1320
Hadi SJ, Abba SI, Sammen SS, Salih SQ, Al-Ansari N, Yaseen ZM (2019) Non-linear input variable selection approach integrated with non-tuned data intelligence model for streamflow pattern simulation. IEEE Access 7:141533–141548
Iwasa Y (1991) Predicting longitudinal disperdsion coefficient in open-channel flows. Environ Hydraul I:505–510
Kargar K, Samadianfard S, Parsa J et al (2020) Estimating longitudinal dispersion coefficient in natural streams using empirical models and machine learning algorithms. Eng Appl Comput Fluid Mech 14:311–322
Kashefipour SM, Falconer RA (2002) Longitudinal dispersion coefficients in natural channels. Water Res 36:1596–1608. https://doi.org/10.1016/S0043-1354(01)00351-7
Kotsiantis SB (2013) Decision trees: a recent overview. Artif Intell Rev 39:261–283
Koussis AD, Rodríguez-Mirasol J (1998) Hydraulic estimation of dispersion coefficient for streams. J Hydraul Eng 124:317–320
Legates DR, McCabe GJ Jr (1999) Evaluating the use of "goodness of fit" measures in hydrologic and hydroclimatic model validation. Water Resour Res 35:233–241. https://doi.org/10.1029/1998WR900018
Li X, Liu H, Yin M (2013) Differential Evolution for prediction of longitudinal dispersion coefficients in natural streams. Water Resour Manag. https://doi.org/10.1007/s11269-013-0465-2
Liu H (1977) Predicting dispersion coefficient of streams. J Environ Eng Div 103:59–69
Lu X, Ju Y, Wu L et al (2018) Daily pan evaporation modeling from local and cross-station data using three tree-based machine learning models. J Hydrol. https://doi.org/10.1016/j.jhydrol.2018.09.055
Madvar HR, Dehghani M, Memarzadeh R et al (2020) Derivation of optimized equations for estimation of dispersion coefficient in natural streams using hybridized ANN with PSO and CSO Algorithms. IEEE Access 8:156582–156599. https://doi.org/10.1109/ACCESS.2020.3019362
Mehr AD, Akdegirmen O (2021) Estimation of urban imperviousness and its impacts on flashfloods in Gazipaşa, Turkey. Knowledge-Based Eng Sci 2:9–17
Naganna SR, Beyaztas BH, Bokde N, Armanuos AM (2020) ON THE EVALUATION OF THE GRADIENT TREE BOOSTING MODEL FOR GROUNDWATER LEVEL FORECASTING. Knowledge-Based Eng Sci 1:48–57
Najafzadeh M, Tafarojnoruz A (2016) Evaluation of neuro-fuzzy GMDH-based particle swarm optimization to predict longitudinal dispersion coefficient in rivers. Environ Earth Sci 75:157
Noori R, Deng Z, Kiaghadi A, Kachoosangi FT (2016) How reliable Are ANN, ANFIS, and SVM techniques for predicting longitudinal dispersion coefficient in natural rivers? J Hydraul Eng 142:4015039. https://doi.org/10.1061/(asce)hy.1943-7900.0001062
Noori R, Karbassi A, Farokhnia A, Dehghani M (2009) Predicting the longitudinal dispersion coefficient using support vector machine and adaptive neuro-fuzzy inference system techniques. Environ Eng Sci 26:1503–1510
Noori R, Karbassi AR, Mehdizadeh H et al (2011) A framework development for predicting the longitudinal dispersion coefficient in natural streams using an artificial neural network. Environ Prog Sustain Energy 30:439–449
Abba SI, , (2019) Multi-parametric modeling of water treatment plant using AI-based non-linear ensemble 2:1–15. https://doi.org/10.2166/wst.2011.079
Saberi-Movahed F, Najafzadeh M, Mehrpooya A (2020) Receiving more accurate predictions for longitudinal dispersion coefficients in water pipelines: training group method of data handling using extreme learning machine conceptions. Water Resour Manag 34:529–561
Sahay RR (2011) Prediction of longitudinal dispersion coefficients in natural rivers using artificial neural network. Environ Fluid Mech 11:247–261
Sahay RR (2013) Predicting longitudinal dispersion coefficients in sinuous rivers by genetic algorithm. J Hydrol Hydromechanics 61:214–221. https://doi.org/10.2478/johh-2013-0028
Sahay RR, Dutta S (2009) Prediction of longitudinal dispersion coefficients in natural rivers using genetic algorithm. Hydrol Res 40:544–552. https://doi.org/10.2166/nh.2009.014
Sahin S (2014) An Empirical approach for determining longitudinal dispersion coefficients in rivers. Environ Process 1:277–285. https://doi.org/10.1007/s40710-014-0018-6
Sattar AMA, Gharabaghi B (2015) Gene expression models for prediction of longitudinal dispersion coefficient in streams. J Hydrol 524:587–596. https://doi.org/10.1016/j.jhydrol.2015.03.016
Seo IW, Baek KO (2004) Estimation of the longitudinal dispersion coefficient using the velocity profile in natural streams. J Hydraul Eng 130:227–236
Seo W, Cheong Tae S (1998) Predicting longitudinal dispersion coefficient in natural streams. J Hydraul Eng 124:25–32. https://doi.org/10.1061/(Asce)0733-9429(2005)131:11(991)
Shareef MA (2019) Assessment of Tigris River water quality using multivariate statistical techniques. Tikrit J Eng Sci 26:26–31
Shihab AS, Ahmad AM (2020) Performance study of tube settlers in removing low turbidity from the Tigris River water using a bench scale model. Tikrit J Eng Sci 27(4):1–7
Soltani-gerdefaramarzi S, Taghizadeh-mehrjerdi R, Ghasemi M (2015) Prediction of Longitudinal Dispersion Coefficient in Natural Streams using Soft Computing Techniques. Iran J Soil Water Res 46(3):385–394
Tao H, Al-Bedyry NK, Khedher KM et al (2021a) River water level prediction in coastal catchment using hybridized relevance vector machine model with improved grasshopper optimization. J Hydrol 598:126477
Tao H, Al-khafaji ZS, Qi C et al (2021) Artificial intelligence models for suspended river sediment prediction: state-of-the art, modeling framework appraisal, and proposed future research directions. Eng Appl Comput Fluid Mech 15(1):1585–1612
Tao H, Awadh SM, Salih SQ et al (2021c) Integration of extreme gradient boosting feature selection approach with machine learning models: application of weather relative humidity prediction. Neural Comput Appl. https://doi.org/10.1007/s00521-021-06362-3
Tao H, Habib M, Aljarah I, et al (2021d) An intelligent evolutionary extreme gradient boosting algorithm development for modeling scour depths under submerged weir. Inf Sci (Ny)

Tayfur G, Singh VP (2005) Predicting longitudinal dispersion coefficient in natural streams by artificial neural network. J Hydraul Eng 131:991–1000. https://doi.org/10.1061/(asce)0733-9429(2005)131:11(991)

Tayfur G, Vijay S (2005) Predicting longitudinal dispersion coefficient in natural streams by artificial neural network. J Hydraul Eng 131:991–1000. https://doi.org/10.1061/(Asce)0733-9429(2005)131:11(991)

Taylor G (1954) The dispersion of matter in turbulent flow through a pipe. Proc R Soc A Math Phys Eng Sci 223:446–468. https://doi.org/10.1098/rspa.1954.0130

Taylor GI (1953) Dispersion of soluble matter in solvent flowing slowly through a tube. Proc R Soc London Ser A Math Phys Sci 219:186–203

Taylor KE (2001) Summarizing multiple aspects of model performance in a single diagram. J Geophys Res Atmos 106:7183–7192. https://doi.org/10.1029/2000JD900719

Tung TM, Yaseen ZM (2020) A survey on river water quality modelling using artificial intelligence models: 2000–2020. J Hydrol 585:124670

Tur R, Yontem S (2021) A comparison of soft computing methods for the prediction of wave height parameters. Knowledge-Based Eng Sci 2:31–46

Tutmez B, Yuceer M (2013) Regression kriging analysis for longitudinal dispersion coefficient. Water Resour Manag 27:3307–3318

Wang Y-F, Huai W-X, Wang W-J (2017) Physically sound formula for longitudinal dispersion coefficients of natural rivers. J Hydrol 544:511–523. https://doi.org/10.1016/j.jhydrol.2016.11.058

Yaseen ZM (2021) An insight into machine learning models era in simulating soil, water bodies and adsorption heavy metals: review, challenges and solutions. Chemosphere 277:130126. https://doi.org/10.1016/j.chemosphere.2021.130126

Zahrawi M, Mohammad A (2021) Implementing recommender systems using machine learning and knowledge discovery tools. Knowledge-Based Eng Sci 2:44–53

Zeng YH, Huai WX (2014) Estimation of longitudinal dispersion coefficient in rivers. J Hydro-Environment Res 8:2–8. https://doi.org/10.1016/j.jher.2013.02.005

Zhai X, Yin Y, Pellegrino JW et al (2020) Applying machine learning in science assessment: a systematic review. Stud Sci Educ 56:111–151

Zhong S, Zhang K, Bagheri M et al (2021) Machine learning: new ideas and tools in environmental science and engineering. Environ Sci Technol 55(19):12741–12754

Authors and Affiliations

Hai Tao[1,2,3] · Sinan Salih[4,5] · Atheer Y. Oudah[6,7] · S. I. Abba[8,9] · Ameen Mohammed Salih Ameen[10] · Salih Muhammad Awadh[11] · Omer A. Alawi[12] · Reham R. Mostafa[13] · Udayar Pillai Surendran[14] · Zaher Mundher Yaseen[15,16,17]

1 School of Electronics and Information Engineering, Ankang University, Ankang, China

2 School of Computer Sciences, Baoji University of Arts and Sciences, Shaanxi, China

3 Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia

4 Computer Science Department, Dijlah University College, Al-Dora, Baghdad, Iraq

5 Artificial Intelligence Research Unit (AIRU), Dijlah University College, Al-Dora, Baghdad, Iraq

6 Department of Computer Sciences, College of Education for Pure Science, University of Thi-Qar, Thi-Qar, Iraq

7 Scientific Research Center, Al-Ayen University, Thi-Qar 64001, Iraq

8 Interdisciplinary Research Center for Membrane and Water Security, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

9 Faculty of Engineering, Department of Civil Engineering, Baze University, Abuja, Nigeria

10 Department of Water Resources, University of Baghdad, Baghdad, Iraq

11 Department of Geology, College of Science, University of Baghdad, Baghdad, Iraq

12 Department of Thermofluids, School of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Bahru, Malaysia

13 Information Systems Department, Faculty of Computers and Information Sciences, Mansoura University, Mansoura 35516, Egypt

14 Land and Water Management Research Group, Centre for Water Resources Development and Management (CWRDM), Kozhikode, Kerala, India

15 Department of Urban Planning, Engineering Networks and Systems, Institute of Architecture and Construction, South Ural State University, 76, Lenin Prospect, 454080 Chelyabinsk, Russia

16 New era and development in civil engineering research group, Scientific Research Center, Al-Ayen University, Thi-Qar, Nasiriyah 64001, Iraq

17 College of Creative Design, Asia University, Taichung City, Taiwan