Arabian Journal of Geosciences

On the simulation of streamflow using hybrid tree-based machine learning models: A case study of Kurkursar basin, Iran

--Manuscript Draft--

Manuscript Number:	AJGS-D-22-00641
Full Title:	On the simulation of streamflow using hybrid tree-based machine learning models: A case study of Kurkursar basin, Iran
Article Type:	Original Paper
Abstract:	The most important concern of hydrologists in the analysis of water resources for planning and managing sustainable water resources is the reliable and accurate prediction of streamflow. In this regard, the application of data mining algorithms has grown significantly in recent decades. In the present study, seven standalone decision tree models, namely Random Forest (RF), Random Tree (RT), REP Tree (REPT), instance-based learning (IBK), KStar, M5P, and Bagging, and six hybrid models, namely Random Sub Space-REP Tree (RS-REPT), Random Sub Space-Random Tree (RS-RT), Random SubSpace-M5P (RS-M5P), Random Committee-REP Tree (RC-REPT), and Random Committee-Random Tree (RC-RT), were used to predict streamflow in the Kurkursar River, Iran. A time series from 1989 to 2019 was used for training (8%) and testing (20%) phases. The RMSE, PSR, MAE, PBAIAS, and NSE statistical criteria were employed to evaluate the performance and accuracy of the models, and dimensional diagrams were drawn to assess them visually. The results showed that among all the standalone and hybrid models, BAGGING had the best performance (RMSE=0.51615 [[EQUATION]] , MAE=0.09201 [[EQUATION]] , NSE=0.7467, PBIAS=-10.511%, and PSR=0.50319) and the REP Tree (REPT) model the weakest performance (RMSE= 1.36649, MAE= 0.2451607, NSE= 0.3327, PBIAS -11.274, and PSR= 0.81685). The research results generally show that standalone and hybrid models performed very well. It can also be deduced that the performance of standalone and hybrid models are very close to each other, and there is no significant superiority between single and hybrid models.

On the simulation of streamflow using hybrid tree-based machine learning models: A case study of Kurkursar basin, Iran

Edris Merufinia[1], Ahmad Sharafati[1*], Hirad Abghari[2], Youssef Hassanzadeh[3]

[1] Department of Civil Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

[2]Department of Range and Watershed Management, Urmia University, Urmia, Iran

[3]Department of Water Engineering, Center of Excellence in Hydroinformatics, Faculty of Civil Engineering, University of Tabriz, Tabriz, Iran

Corresponding authors: Ahmad Sharafati

Corresponding emails: asharafati@gmail.com, asharafati@srbiau.ac.ir

Abstract

The most important concern of hydrologists in the analysis of water resources for planning and managing sustainable water resources is the reliable and accurate prediction of streamflow. In this regard, the application of data mining algorithms has grown significantly in recent decades. In the present study, seven standalone decision tree models, namely Random Forest (RF), Random Tree (RT), REP Tree (REPT), instance-based learning (IBK), KStar, M5P, and Bagging, and six hybrid models, namely Random Sub Space-REP Tree (RS-REPT), Random Sub Space-Random Tree (RS-RT), Random SubSpace-M5P (RS-M5P), Random Committee-REP Tree (RC-REPT), and Random Committee-Random Tree (RC-RT), were used to predict streamflow in the Kurkursar River, Iran. A time series from 1989 to 2019 was used for training (8%) and testing (20%) phases. The RMSE, PSR, MAE, PBAIAS, and NSE statistical criteria were employed to evaluate the performance and accuracy of the models, and dimensional diagrams were drawn to assess them visually. The results showed that among all the standalone and hybrid models, BAGGING had the best performance (RMSE=0.51615 $\frac{\boldsymbol{m^3}}{\boldsymbol{s}}$, MAE=0.09201 $\frac{\boldsymbol{m^3}}{\boldsymbol{s}}$, NSE=0.7467, PBIAS=-10.511%, and PSR=0.50319) and the REP Tree (REPT) model the weakest performance (RMSE= 1.36649, MAE= 0.2451607, NSE= 0.3327, PBIAS -11.274, and PSR= 0.81685). The research results generally show that standalone and hybrid models performed very well. It can also be deduced that the performance of standalone and hybrid models are very close to each other, and there is no significant superiority between single and hybrid models.

Keywords: Data mining, Hybrid model, Kurkursar river, Streamflow prediction

1. Introduction

Streamflow prediction for watershed planning and management, drought risk assessment, and water resource development has become vital and challenging for engineers and hydrologists (Steinfeld et al., 2015). However, the complex nature of processes such as streamflow makes their accurate prediction difficult. At the moment, there are several hydrological methods for simulating the streamflow process, including physical, conceptual, experimental, Artificial Intelligence (AI), and Data Mining (DM) models (Jothiprakash & Magar, 2009; C. L. Wu & Chau, 2011).

Today, the growth of technology has increased so much that it has changed the direction of our lives, and, in this regard, AI, which is a relatively new trend in science, has brought about fundamental changes. As a part of artificial intelligence, machine learning and data mining models are structured to employ fundamental relationships in data for prediction in different areas by modeling (Moosavi et al., 2020). The main purpose of machine learning is to design and develop intelligent applications that can access data and use them in the learning process (Travassos et al., 2020).

Many companies and organizations around the world need new techniques and tools, including machine learning and data mining, to achieve their ideals and fundamentals in dealing with the new challenges of today's ever-evolving world (Dogan & Birant, 2020). In the field of engineering, to identify and solve problems, the testing process is divided into three models: White, Black, and Gray Boxes. The Black Box model does not pay attention to the internal mechanism of a system or a tool and focuses only on the output produced based on the selected input for the operating conditions (Henzinger et al., 2006). In the White Box model, the internal mechanism of a system is also tested (Korel, 1990). Finally, in the Gray Box modeling, the model structure is derived from physical principles by evaluating the parameters through experimental data (Bäumelt

& Dostál, 2020). Currently, there are several hydrological models to simulate this process (C. L. Wu & Chau, 2011), which can be categorized as physical, conceptual, experimental, and artificial intelligence (AI) models (Jothiprakash & Magar, 2009). Streamflow prediction using artificial intelligence techniques can be classified into four classes of data-based regression classification, evolutionary computation, fuzzy sets, and combined (with other models) (Cigizoglu, 2005; Yaseen et al., 2015). From another perspective, these models can be divided into experimental (black box), conceptual (gray box), and physical (white box) models (Worden et al., 2007). The black box (experimental) model is developed without considering the physical processes associated with the watershed. They only operate based on simultaneous input and output time series analysis (Azodi et al., 2020). Conceptual models or gray boxes are based on physical laws and can describe hydrological behavior by empirical expression. Gray box models require less data than physical models and, thanks to the calibration process, demand less computational volume and time (Salcedo-Sanz et al., 2016). Black box does not use physical process information, and its main focus in the modeling process is solely on data. Therefore, it can be said that in the black-box model, the input has a different structure for the forecasting process (Dastorani et al., 2010; Mehr et al., 2015). These models make decisions based on previously stored data and reduce the likelihood of error, hence solving the most complex and difficult problems very quickly without the slightest mistake.

AI models have many advantages in science, especially by reducing repetitive processes and not requiring complex equations (Afan et al., 2015; Deo & Şahin, 2016; Shu & Burn, 2004). ANNs have a special place among AI models. Such models have been increasing due to various merits such as accuracy, convergence speed, and volume of calculations (Pradhan et al., 2020). One of the main advantages of ANNs is that they do not need comprehensive information on the physics of the problem. They also have a very high capability to use incomplete data. Another advantage is that when we have missing data, using these models seems very reasonable (Minns & Hall, 1996). ANNs can be used in different fields of hydrological modeling, e.g., for rainfall-runoff (Adnan et al., 2020; Alizadeh et al., 2020; Kassem et al., 2020; Parisouj et al., 2020; Snieder et al., 2020), sediment (Asheghi & Hosseini, 2020; Banadkooki et al., 2020; Ebtehaj et al., 2020; Ehteram et al., 2020; Meshram et al., 2020; Seo et al., 2020), flood (Cheng et al., 2020; Dtissibe et al., 2020; Kumar & Yadav, n.d.; Kurian et al., 2020; Luu et al., 2020; Obasi et al., 2020), and evapotranspiration (Guan et al., 2020; Malik et al., 2020; Mohamadi et al., 2020; Seifi & Soroush, 2020; L. Wu et al., 2020).

However, there are some drawbacks with the ANNs, and the main issue is finding the optimal values (solution) for weight and bias coefficients (Bashir & El-Hawary, 2009). To overcome this problem, optimization algorithms (metaheuristics) are commonly used to optimize the coefficients. Metaheuristic models use mathematical programming to determine the optimal value of one or more objective functions and include randomly structured search elements that follow empirical instructions; they are often inspired by observations of natural phenomena (McKinney & Lin, 1994; Nicklow et al., 2010). These algorithms examine many factors such as time and speed of convergence and identify how to reach the optimal global solutions. Further research should be conducted to investigate strategies for exiting local optimal solutions and those for increasing the accuracy and efficiency of such models. These algorithms can be applied with small modifications to various optimization problems, hence a significant improvement in finding high-quality solutions to difficult optimization problems. A common feature of such algorithms is local optimization exit mechanisms.

In this research, we use tree-based models to predict streamflow. Decision tree models can be applied to various fields. Random forest is a common decision tree model with applicability to various areas such as streamflow prediction (Abbasi et al., 2020; Araza et al., 2020; Pham et al., 2020; Zeng et al., 2021), flood forecasting (Kim & Kim, 2020; Pahlavan-Rad et al., 2020; Schoppa et al., 2020; Vafakhah et al., 2020), evapotranspiration (H. Chen et al., 2020; Granata et al., 2020; Karimi et al., 2020; Saggi & Jain, 2020; Salam & Islam, 2020), groundwater prediction (Avand et al., 2020; W. Chen, Li, Tsangaratos, et al., 2020; Lahjouj et al., 2020; Norouzi & Moghaddam, 2020; Sachdeva & Kumar, 2020), etc. In the present paper, daily data were utilized to predict streamflow. The variables used included precipitation (R) and discharge (Q). The models were of two categories, standalone and hybrid. The main purpose of this study was to predict the streamflow of the Kurkursar river in Iran via data mining algorithms. Moreover, we intended to evaluate the performance of hybrid models compared to standalone models to determine whether hybrid models would lead to more accurate results. In this study, we seek to compare the performance of single and hybrid models based on decision trees and measure the factors effective on the optimal selection of results to select the best combination for each model and their internal factors based on the physics of the problem is optimized. In this regard, the model composition is determined based on correlation, and the internal factors and parameters of the model will be optimized. We will also try to make a comparison between single and hybrid models and further evaluate their advantages and disadvantages. Finally, the research will be summarized and the success factors in modeling will be addressed and practical suggestions will be provided to increase the research efficiency.

2. Materials and Methods

2.1. Case Study

A catchment is a part of the land where the whole water that has fallen or flowed reaches an endpoint. As we know, Iran is a country located in West Asia. It is the second-largest country in the Middle East. The main catchments of the country include the Caspian Sea, the Persian Gulf and the Sea of Oman, Lake Urmia, the Central Plateau, the Eastern Plateau, and Qaraqom (Sarakhs). The Central Plateau basin has the widest, and the Sarakhs basin has the lowest area. closed or inland basins constitute about 4.73% of the country's area. The Caspian Sea catchment area includes the sub-basins of Aras, Sefidrood, Kurkursar, Lahijan, Haraz, Atrak, and Qarasu. The study area of this research is the Kurkursar basin in Nowshahr city, Mazandaran province, in Northern Iran, with an area of about 75 km^2. In terms of hydrological classification, it is considered as one of the Caspian sub-basins located between the longitudes $51^0 23'28''$ and $51^0 29'33''$ East and latitudes $33^0 36'39''$ and $36^0 29'48''$ North. The average elevation of the Kurkursar watershed is 890 meters, and, according to studies, the slope of the Kurkursar basin is 12.3 degrees in the direction of 111 degrees East. Figure (1) shows the geographical location of the area.

[Fig 1]

2.2. Algorithms and Models

Decision trees are a new and advanced generation of data mining models that have extensively been developed in recent decades. These techniques can discover and extract knowledge from a database and create prediction models (Kazeminezhad et al., 2005). They are now among the most well-known data mining methods and tools for classification and prediction, which, unlike neural networks, generate laws. The decision trees explain their prediction in the form of a set of rules. This research uses eight standalone (RF, RC, RT, REPT, IBK, KStar, M5P, Bagging) and six hybrid models (RS-REPT, RS-RT, RS-M5P, RC-REPT, RC-RT, and RC-RF) based on decision trees.

2.2.1 M5P

As well known, the M5P algorithm (Wang & Witten, 1996) is, in fact, an extended version of M5, which was discovered and developed by Quinlan (1992). Although there are many learning assembly models, there is no doubt that decision tree models have a special place among them. These models have precise performance and are known to be very cheap. Moreover, They show very good performance in terms of regression (Nhu et al., 2020). Another main advantage of decision trees is their quite desirable performance when very large data is in hand with high features and dimensions. Even when there is a great amount of missing data in a project, such models have high technical justification (Behnood et al., 2017). The decision trees create a tree-like structure for prediction by starting with all the instructional examples, selecting the variable that best categorizes them, and forming subcategories. Tree branches result from an experiment performed by the algorithm with intermediate nodes at each stage. Predictions also appear in the tree leaves (Debeljak & Džeroski, 2011). The M5P tree model can numerically predict continuous variables from numerical traits, and the predicted results appear as multivariate linear regression models in the tree leaves (Frank et al., 1998; Wang & Witten, 1996). The division criterion is based on selecting the standard deviation (SDR) of the output values that reach the node as a measure of error. The expected reduction in error is calculated by testing each attribute (parameter) in the node. The SDR is obtained from Equation (1).

$$SDR = \frac{\xi}{|\psi|} \times \beta\ (i) \times \left[sd(\psi) - \sum_{k \in (L,R)} \frac{\psi_K}{|\psi|} \times sd(\psi_K) \right] = sd(\psi) - \sum_k \frac{\psi_K}{|\psi|} \times sd(\psi_K) \quad \textbf{(1)}$$

Where SDR is the standard deviation reduction, ψ represents the series of instances that reach the node, m indicates the number of instances that do not have missing values for this attribute, $\beta\ (i)$ is a corrective factor, and L and R are sets that arise from the division of this attribute.

2.2.2. Random Forest (RF)

Random forest is one of the well-known and widely used algorithms in soft computing and data mining (Breiman, 2001; Chernick, 2002). Using this model is very simple and leads to high accuracy in forecasting with generally desirable results. This algorithm is also applicable to multiclassification and regression (de Santana et al., 2018; Quiroz et al., 2018) since it has a relatively low sensitivity to multicollinearity. This model achieves excellent results with missing and unbalanced data (W. Chen, Li, Xue, et al., 2020; Tsagkrasoulis & Montana, 2018). Each tree branch is identified using a random subset of variables/factors in each node during the RF modeling process. The final result of the modeling process is the average of all the trees (Cutler et al., 2007). To implement the stochastic forest model, it is necessary to define two basic parameters, namely the number of variables (factors) used in each stage of the tree building process (mtree) and the number of trees to be built in the forest (ntree). In order to minimize the generalization error, the mentioned parameters must be optimized (Liaw & Wiener, 2002). Some researchers (e.g., Bryman, 2001; Liaw and Wiener, 2002) have stated in their studies that even one variable (m = 1) can be accurate, while others (e.g., Grömping, 2009) consider at least two variables to be necessary. However, in order to avoid using weaker regressions as a separator, it is better to assume (m = 1,2,3, ...). RF is a set of classification and regression (CART) trees calculated from Equation (2):

$$\{\varphi(\tau, \theta_\xi), \xi = 1,2, \dots, i, \dots\} \quad \textbf{(2)}$$

where φ is random forest classification, τ is an input variable, and $\{\theta_\xi\}$ represents independent and distributed random vector variables used to generate each regression and classification tree. The calculation of important variables is based on the mean Gini coefficient reduction and the mean accuracy reduction. The Gini coefficient is an error that can be deduced from Equation (3):

$$Gini\ \text{coefficient} = 1 - (1 - \sum_C P^2(c|\text{t})) = \sum_{k=1}^{k} \widehat{p_{mk}} \times (1 - \widehat{p_{mk}}) \quad \textbf{(3)}$$

In the above formula, (p_mk)ˆ indicates the probability of correct classification, C is the number of classes, t represents a tree node, and P stands for the relative frequency of c. The Gini coefficient results from multiplying the probability of correct and incorrect classifications (Jiang et al., 2020).

2.2.3. Reduced Error Pruning Tree (REPT)

Decision trees can usually be divided into two types of tree (hierarchical) structure and rules (if-then). If the decision tree is complex, the tree structure and rules may be destroyed (X. Wu & Kumar, 2009). Hence, pruning steps are primarily used for a complex tree to facilitate the interpretation and analysis of results. Furthermore, pruning decision trees is essential in optimization to increase computational efficiency and classification accuracy (Rokach & Maimon, 2008). There are two standard pruning methods: pre-pruning (back pruning) and post-pruning (pruning forward). The pruning method comprises two growth and pruning stages, allowing one to over-fit the data and then prune the grown trees. Post-pruning methods perform better than pre-pruning (Mahmood et al., 2010).

Pruning error reduction (REP) is a post-pruning method for decision trees, and the REPT model, as one of the fastest methods of model training, is a combination of REP and decision tree algorithm. It is developed based on a decision/regression tree to reduce variance (Breslow & Aha, 1997). In the decision tree algorithm, the size of the tree affects the accuracy of data classification. On the other hand, combining two algorithms reduces the synergy in the structure of a decision tree (Sharafati et al., 2019). Therefore, this complexity in the REPT model is reduced by the REP pruning technique, one of the most popular and well-known pruning methods that can target some branches and leaves of trees without affecting the accuracy and precision of the model (Mahmood et al., 2010). The two main advantages of this method include simplifying the tree without reducing accuracy and avoiding the overfitting problem (Khosravi, Mao, et al., 2018; Khosravi, Pham, et al., 2018). The basic REPT relation is given in Equation (4):

$$Gain\ ratio\ (\eta, \xi) = \frac{Entropy\ (\xi) - \sum_{i=1}^{n} \left|\frac{\xi_i}{\xi}\right| Entropy\ (\xi_i)}{-\sum \left|\frac{\xi_i}{\xi}\right| Log2 \left|\frac{\xi_i}{\xi}\right|} \qquad \textbf{(4)}$$

In this regard, the property η belongs to the educational dataset ξ with subsets ξ_i = 1,2,3,…, n.

2.2.4. Bagging (BA)

Bagging is a machine learning method proposed by Breiman (1996). This algorithm increases classification accuracy by combining the classification of randomly generated training sets. It can reduce the variance of the basic algorithms and adjust the estimation to the expected conclusion to improve the accuracy of a model (Dieu Tien Bui et al., 2016; Peters et al., 2002). Also, this method can eliminate the defects of learning components and improve the predictive ability of weak learners (Yin, 2020). Moreover, due to its sensitivity to minor changes in the training data, this technique can increase the accuracy of the prediction results (Shirzadi et al., 2018). Bagging is most useful when regression models with high variance and low bias, such as regression trees, are fully grown (Gweon et al., 2020). It creates multiple instances from the same dataset by modifying the bootstrap technique. Several separate trees are created for the same prediction and used to generate a whole prediction. The final prediction of the process can be obtained by voting or averaging for classification and regression problems (Erdal & Karahanoğlu, 2016; Ribeiro & dos Santos Coelho, 2020).

2.2.5. Random Committee-REPT (RC-REPT)

Random Committee (RC) is a meta-algorithm that possesses classifiers that can be used at the service of the learning power. In the classification process, predictions are made by estimating the average probability, not by voting. This algorithm is used for classification and regression problems depending on the learner base. It can also be combined with other models to form a hybrid classifier model, which in the present study is developed by tree pruning modifiers generating predictions with a direct average probability (Sharafati et al., 2019). They are used to improve the trainability of the model as well as the model that is combined with it (classifier). By this method, a combination of classifier-based methods can be created. For this purpose, WEKA software is used in which, generally, the whole process is divided into two stages of preprocessing and classification. The processing step involves selecting the attribute. In this study, not all attributes of the dataset are necessary for analysis, which leads to a reduction in dimensions and ensures better performance. The second phase involves using machine learning techniques such as random forests, random committees, and random trees to classify samples through voting (Niranjan et al., 2018).

2.2.6. *Random Subspace-REPT (RS-REPT)*

Another method for group learning is Random Subspace (RS). Ho (1995) first proposed the RS model as a comprehensive classical algorithm. This model has many similarities with the bagging model. It seeks to diversify learners in sampling the feature space. All model components are constructed with the same training data, but each feature is selected randomly, leading to the group's diversity. For the most part, the number of attributes in all committee components is at the same level. When it comes to classification, a group decides either by a majority vote or by the weight of votes. Regression is simply done with the average output of the components. This method aims to increase the general accuracy of decision-based classifiers without compromising the accuracy of training data, which is one of the major and most common problems of tree-based classification (Ho, 1998). Many studies perform RS in pairs with different classifiers having the training subsets randomly made from the main training subsets, which is the only difference from the Bagging algorithm. In this method, the complications of each sub-classifier in the final prediction are obtained through the combined voting method (Bertoni et al., 2005).

2.3. *Evaluation criteria*

It is necessary to use evaluation indicators to evaluate a model's prediction performance in any research. Accordingly, in the present study, different statistical criteria are used to assess and compare the performance of the models. These criteria include the coefficient of determination (R^2), root mean square error (RMSE), absolute mean error (MAE), Nash-Sutcliffe efficiency (NSE) coefficient, bias, and the squared ratio of mean squared error to standard deviation (PSR). Table (1) summarizes the results for the above indicators. In addition, the qualitative and quantitative values, as well as their allowable ranges, are specified. “ob” represents the observed values, and “pr” is the calculated or predicted value.

[Table 1]

2.4. Best input combination

Finding the relationship between different variables is a major challenge in this process. It can be simply stated that the preliminary and, at the same time, the most important part in the modeling process is the research to determine the effective factors in the prediction process. Therefore, in the first phase of the research, the effective factors that influence the river discharge are determined, and then, with the help of the Pearson coefficient, the effect of each factor is determined. The most important factors in streamflow prediction include temperature, humidity, precipitation, evapotranspiration, pressure, wind direction, and discharge. However, among these factors, only precipitation and discharge have the greatest impact on the forecasting process, and the effects of other factors can be ignored. The time series in this research are on a daily basis for the data from 1989-2019 (80% for training and 20% for testing). Table (2) shows some statistical parameters for the datasets used in the training and testing phases.

[Table 2]

We determined the correlation between input and output variables with the CC coefficient, shown in Table (3). It should be noted that the input variables include R(t), R(t-1), R(t-2), … , R(t-6), Q(t),Q(t-2),…,Q(t-5), and the output variable is Discharge(Q(t)). Table (4) shows the combinations for the input variables.

[Table 3] & [Table 4]

3. Results

This research aims to predict the streamflow of the Kurkursar river in Iran. First, the required data should be collected and standardized to this aim. The time series of the research data is on a daily basis for precipitation (rainfall) and discharge. After collecting and arranging the data structure, the Weka program was employed to implement the models developed by the University of Waikato. To this end, the data were entered into the program, and the best combination for each model was determined in two phases of training (80%) and testing (20%). The implemented models used in the present study belong to two categories of standalone and hybrid. However, being developed based on the decision tree is their main common trait. Decision trees are helpful when the volume of data is very high. As fully introduced in the previous sections, six standalone and eight hybrid models were used. Table (5) shows the correlation coefficient values in the two phases of training and testing for the standalone models used to choose the best combination.

[Table 5]

All the data were evaluated to determine which combination would be the best solution for each model. Finally, for the M5P model, the combination of model 2 led to the best solution (CC=0.9264) in the test phase. Also, for Random Forest, the combination of model 4 (CC=0.953); Random Tree, model 4 (CC=1.3945); REP Tree, model 4 (CC=1.0334); Bagging, model 4 (CC=1.0314); IBk, model 4 (CC=1.4402); and Kstar, model 4 (CC=1.3323) was identified. Therefore, for all models (except for the M5P model), the combination of model 4 achieved the best results in the test phase.

In the next stage, the value of the coefficient of determination (R^2) was determined, for which the results of the testing phase with the standalone and hybrid models are shown in Table (6). Of note, R^2 is a statistical measure of the data close to the fitted regression line.

[Table 6]

The results in Table (6) show that based on the R^2 coefficient, the M5P model had the best (R^2=0.7839) and the KStar model had the weakest performance (R^2=0.5228) among the standalone and RS-M5P (R^2=0.7613), and RS-RT (R^2=0.6584) had the best and weakest performances, respectively, among the hybrid models.

One of the main goals of modeling is to reduce error. In the present paper, RMSE, MAE, PSR, NSE, and PSR indices were used to evaluate the models, which are shown in Table (7).

[Table 7]

Table 7 shows the BA has the lowest error among standalone models (RMSE=0.5161m^3/sec), and REPT has the highest error (RMSE=1.3664 m^3/sec). Therefore, BA has the best performance, and the REPT model has the weakest performance. Among the hybrid models, RS-M5P has the lowest error and the best performance (RMSE= 0.77176 m^3/sec), and the RS-RT model has the highest error and the weakest performance (RMSE= 0.91089 m^3/sec).

To assess the predictive models used in the current study, several visual comparisons were made through different graphical figures including, time-series plots (Figure 2), scatter plots (Figure 3), and error graphs (Figure 4). From those figures, it can be concluded that among the standalone models, the IBK model is the closest to the observational data in predicting the minimum values. In simpler terms, this model better predicts the minimum values. However, the BA model's weakest performance in predicting the minimum values. In addition, if we look at hybrid models, we will see that the results for predicting minimum values are very close to the observational data. Among these models, the performances of the RS-RT, RS-M5P, and RC-RT models are very close and have almost the same performance. In general, among all the models (both single and hybrid), the M5P model has the best performance, and the K Star model has the weakest performance for the minimum values.

[Figs 2-4]

To assess the probabilistic features of predictive models, their box plot diagrams are shown in Figure (5). From the figure, it is evident that the IBK model is the closest to the observational data, and the RC-REPT hybrid model has the weakest performance in the first quarter (Q_1). However, the performances are slightly different with the median data, with the M5P and RF models having the best performance and being closest to the observational data. Notably, the M5P model is slightly better than the RF model. Furthermore, the RC-REPT model has the poorest performance in predicting midpoints than the observational data. In the third quarter (Q_3), KStar is the best, and RS-RT is the weakest model. Finally, the RT model can predict the maximum data better, and the K Star model is ranked the lowest.

[Fig 5]

The "Bag size percentage" parameter specifies a certain number of samples for each member (classifier) of the group. This parameter is determined by the size of the training set (number of training samples) and by distance, and its value is between 10 and 100. The size of each bag is defined as a percentage of the size of the training set. The number of iterations is a measure to stop the error in Weka. The principle of learning in the network is observed in iterations. The dataset is injected into the algorithm several times, and the algorithm can detect differences in the training data by increasing or decreasing the network parameters. This value is usually assumed to be 10, while some researchers assume it to be 100. The parameter seed is, in fact, a random number. Once its value is fixed, even a random algorithm will behave definitively, and using the same seed will always lead to the same random numbers. There is no definite criterion for determining this parameter, and its optimal value can be calculated by trial and error. Out-of-bag (OOB) error is used to measure the predictive error of random forests, reinforced decision trees, and other machine learning models using Bootstrap aggregation. The bagging model uses sub-sampling with an alternative property to create training examples for model learning. In the present study, optimal coefficients evaluated automatically were measured. The optimized coefficients were tested by trial and error for each model to improve the prediction results. The best-case scenario for the M5P was when the Batchsize value was 100. In this case, the value of RMSE = 1.273 m^3/sec is the best solution. To evaluate the effect of Batchsize with trial and error in the range [5-200], the above parameter is added five units in each step to measure its effects. The study results for the mentioned range show that neither increasing nor decreasing this value reduces the error rate. Therefore, the same initial solution of 100 is chosen as the optimal solution. Assuming that the optimal value of the Batchsize parameter = 100 is constant, we can find the optimal MinnumIstance solution, the value of which is automatically equal to 4. The optimal solution is found by trial and error with [20-1] intervals. The results show that from 1 to 4, it does not have any effect on reducing the error. However, for the value of 5, the error rate increases and reaches 1.466. This error remains constant for the values from 5 to 20. Accordingly, the value of 4 is selected as the optimal solution for the above parameter. If the Build Regression Tree parameter is set to False, the error value will be 1.273, but if it is set to True, the error value will reach 1.7247, which indicates an incremental state. Therefore, the default False state is selected as the optimal model since error reduction is aimed. Moreover, the Debug values and the Do Not Check Capabilities in the false mode lead to the optimal solution. If the value of the Unpruned parameter is set to False, the error value will be 1.273. However, in the True mode, the error reaches 1.2344, which indicates a reduction in the amount of error. Therefore, the optimal solution will be in the True mode. If the value of the above parameter is set to False, the error value will be 1.273 m^3/sec. But if the above parameter is set to True, the error reaches 0.8181 m^3/sec, which indicates a reduction in error. Therefore, the optimal solution is obtained when this parameter is in the True state. Finally, the ultimate solution was the best possible case for the M5P model (Optimized Batchsize=100; MinnumIstance & Num decimal place=4; Build Regression=Tree; Debug=Do Not Check; Capabilities=Save Instances; False, Unpruned & Use Unsmoothed=True). Several trials and errors were carried out to find the optimal coefficients in the RF model. For Bag size percentage, trial and error were performed from 5% to 100%. The results show that by increasing this parameter, the error rate decreases, and the best solution is reached at 100%. Similarly, the trial and error range from 5% to 100% for the Batch size parameter. The results show that increasing or decreasing the value of this parameter does not affect the error rate. Therefore, the same solution of 100% is selected as the optimal one. To calculate the Max Depth parameter, trial and error are performed within the [0-15] interval. The results indicate that the best solution is obtained when its value is equal to zero, and the trend of error from 1 to 15 increases. The best solution for the Num Execution Slots parameter is one, and decreasing or increasing its value does not affect the error rate. In the trial and error, this parameter was maintained within the range of [0-10], and it was observed that the error value remained constant without any significant change. The optimal solution was achieved with 100 repetitions, and the error rate was 0.5466. To find the optimal solution in the [10-200] interval, an error was made after ten repetitions, which was minimized to 40 repetitions. Of note, the error rate up to 60 repetitions had a decreasing trend, but the optimal solution still belonged to repetition 40. For seed parameters, trial and error were performed in two intervals of [0-1] and [0-10]. In the first interval, 0.1 was added in each step to measure the effects, and for the second interval, a unit was added. The optimal state of the model occurred in the first interval, where 0.1 was added. Another result of the trial and error was that increasing the values in the first interval increased the error, but in the second interval, the error remained constant. Increasing this number increased the error rate. Therefore, it can be claimed that increasing the above values causes sensitivity to the error value. There was also a trial and error to optimize the coefficients of the REP Tree model. The best solution for the k value parameter was 1, which reduced the error, while the optimal solution was 0. Increasing this value also increased the error. Increasing the value of the Batchsize parameter did not affect the error. In the trial and error, the range of [5-100] was examined for this parameter. The results showed that the amount of error remained constant within the whole range mentioned; the amount of error did not show sensitivity to Batch size. The best solution for the MaxDepth parameter was zero. The interval [0-100] was considered for trial and error. The amount of error increased at first but decreased significantly over time. The best solution for the MinNum parameter was the default value of zero. In the range of [0-10], trial and error was performed, which showed that the amount of error increased with an increase in the value of this parameter. This increase in error had an upward trend and reached its maximum at the value of 10. Accordingly, the best value for the above parameter was 0.001, with one error occurring within [0-0.01]. In terms of MinVariance Propparameter, error increased with an increase in its value with an upward trend. Accordingly, the error rate was at its lowest level at the zero value for this parameter. For the Num Folds parameter, the value of zero led to the lowest error and reached the optimal solution. Increasing this parameter also increased the error rate. However, this increase did not have an upward trend and was increasing and decreasing intermittently, with the error gradually getting far from the optimal solution. Finally, the results showed that increasing the Seed parameter had no effect on the error rate, and as the value of this parameter increased, the error rate remained constant. Therefore, number one was chosen as the optimal value. To summarize the above discussion, generally, the program had a good performance. However, to improve the model's performance with all parameters, they should be regulated by trial and error to obtain more optimal values. In general, the results of the forecast were desirable. The optimum values of the parameters used in the predictive models are presented in the Table 8.

[Table 8]

4. Discussion

This research aimed to predict streamflow in the Kurkursar catchment in Iran. To this aim, the desired data were collected, which included precipitation and discharge in time series on a daily basis. The correlation coefficient between input and output variables was determined, and different combinations for the model were identified. The models were divided into standalone and hybrid categories. Data were classified into two training and testing classes, and Weka software was used to analyze and evaluate the data.

The presented model had a significant effect on reducing variance. It also largely prevented overfitting. Random forests use the bagging algorithm in the learning process, and to reduce overfitting and variance, we created several trees and some data sets in the first step. In the next step, we combined them with the output of the desired model, resulting in which overfitting was greatly reduced. Generally, the performance meaningfully improves when the missing data are incorporated in the modeling.

Random forests have wide applicability in both classification and regression modes, hence their specific position among engineers. However, these models have weaknesses. First of all, many trees are used in random forests, which leads to increased calculations and reduced speed and accuracy of forecasts. Another common problem with these models is that as the number of trees and the output of the model increase, the training period becomes longer. In this case, the model will try to base its final decision on the most votes, prolonging the training process. As for the M5P model, it is based on creating a tree similar to the traditional decision tree (expressed by CART). The specific difference of this model is in its leaves, which generally follow multiple linear regression. It has a relatively good ability to predict various parameters, but its major disadvantage is that changes in data (even small data) may cause instability in the model structure. With an increase in the number of changes, the response time also increases, adding to the complexity of the problem.

Moreover, it allocates more time to the training process than other random forest models do, and it faces difficulties when predicting continuous values. However, combining this model with bagging in most cases improves the results and increases the accuracy of prediction. Many researchers have used this combination in predicting hydrological processes and have reported favorable results (Duie Tien Bui et al., 2020; Khosravi, Mao, et al., 2018; Melesse et al., 2020).

Another machine learning method used in this article was IBK, which is by nature a lazy learner. It uses linear search algorithms to find the nearest neighbor. The Euclidean distance is also used in this model to evaluate the position of the samples. IBK considers distance from the validation data for weighing estimates of more than one neighbor. Generally, this algorithm has reported good performance in many areas (Angarita-Zapata et al., 2020; Gandhi & Armstrong, 2016; Jabbar & Mohammed, 2020; Khosravi et al., 2019; Pattnaik et al., n.d.; Shabani et al., 2020).

Moreover, for interpreting the performance of this model, we can say that there is no special training course required. More simply, the learning process occurs when we intend to make real predictions and, in this way, the training data is stored and used. This speeds up reaching the solution. This is while other algorithms (such as vector machines, etc.) devote much more time to the training process, hence their prolonged process to reach the solution. To add to the above advantage, because the algorithm does not need training before prediction, new data will be added seamlessly, ensuring that the algorithm's correct operation is not compromised.

Moreover, implementing this algorithm is very easy and fast because it lacks complex and ambiguous parameters. It uses only two parameters, namely the value of K and the distance function. Therefore, it is a very easy algorithm to implement. However, despite all the above, this algorithm also has weaknesses. For example, working with big data in this algorithm is very difficult, and interpretation and analysis in such cases are also unclear. Moreover, with large data, the cost of calculation gets very high. In addition, the distance between the new point and each existing point is very large in this model, which lowers its performance.

5. Conclusion

Four standalone and hybrid models based on the decision tree for rainfall-runoff prediction in the Korkorsar watershed in northern Iran were evaluated in the present study. This study aims to use new decision tree algorithms to predict rainfall-runoff that can be used in other areas of water resources management engineering (such as suspended sediment assessment, flood forecasting, evapotranspiration, etc.). The modeling process indicated that the factor R (t) is the most important determinant of precipitation-runoff. Other cases of importance included Q (t-1), Q (t-2), R (t-1), Q (t-3), etc., respectively. In this study, it can be inferred that the use of different combinations of variables leads to different levels of performance of the models. The findings showed that predictive accuracy reaches its maximum value (maximum predictive power) when utilized variables with the highest CC.

Moreover, the variables with the lowest CC greatly reduced the predictive power of the model. The results of research modeling show that hybrid models performed better than individual models, but it is not possible to comment on their absolute superiority. Therefore, it can be said that they may not be equally successful in all cases.

If these models have good and reliable results for a set of data covering a short period, if the period is longer, modeling accuracy will increase accordingly. Research algorithms (based on the decision tree) can be useful for basins with limited measurement networks and fewer.

Our results demonstrated that the proposed algorithms could be reliable and cost-effective for predicting hydrological processes in water resources management. These models are much more useful and cost-effective for developing countries where the cost of measuring some hydrological parameters is very high. Of course, these results cannot be generalized to all basins and hydrological processes in absolute terms. But without a doubt, it can be said that algorithms have very high power and accuracy in predicting different hydrological processes.

The above models showed acceptable and desirable performance in streamflow prediction. All coefficients were examined by trial and error, and some results were stated in the present article. Standalone models performed well, and while hybrid models were expected to have improved performance, they showed very close results to the former. However, in a qualitative evaluation of the results, all the models were within a good to good range. Therefore, the following suggestions are made for further research:

- Use of other tree models and comparison of the results with the present research;
- Employing the presented models in this study in other hydrological and environmental fields to study their accuracy;
- Comparison between the models here and other models (such as ANN, SVR, ANFIS, gene expression-Bayesian networks, etc.) in terms of accuracy;
- Use of other effective factors in streamflow prediction (temperature, humidity, evaporation, transpiration, etc.) in modeling for determining their effects on the efficiency of models;
- Combining and comparing other different input models;
- Further evaluation of the rate of delay with the parameters; and
- A more comprehensive study of the physics of the problem and the structure of the analyzed models. In this regard, their weaknesses can be identified, and necessary measures are taken to strengthen them.

6- Declarations

Funding : No funding.

Conflict of Interest: The authors declare that they have no conflict of interests.

Availability of data and materials : Please contact the corresponding author for data requests.

Code availability: Please contact the corresponding author for code requests.

Ethics approval: Not applicable.

Consent to participate: Not applicable**.**

Consent for publication: Not applicable**.**

References

Abbasi, M., Farokhnia, A., Bahreinimotlagh, M., & Roozbahani, R. (2020). A hybrid of Random Forest and Deep Auto-Encoder with support vector regression methods for accuracy improvement and uncertainty reduction of long-term streamflow prediction. *Journal of Hydrology*, 125717.

Angarita-Zapata, J. S., Masegosa, A. D., & Triguero, I. (2020). Evaluating automated machine learning on supervised regression traffic forecasting problems. In *Computational Intelligence in Emerging Technologies for Engineering Applications* (pp. 187–204). Springer.

Araza, A., Hein, L., Duku, C., Rawlins, M. A., & Lomboy, R. (2020). Data-driven streamflow modelling in ungauged basins: regionalizing random forest (RF) models. *BioRxiv*.

Avand, M., Janizadeh, S., Tien Bui, D., Pham, V. H., Ngo, P. T. T., & Nhu, V.-H. (2020). A tree-based intelligence ensemble approach for spatial prediction of potential groundwater. *International Journal of Digital Earth*, 1–22.

Bäumelt, T., & Dostál, J. (2020). Distributed agent-based building grey-box model identification. *Control Engineering Practice*, *101*, 104427.

Behnood, A., Behnood, V., Gharehveran, M. M., & Alyamac, K. E. (2017). Prediction of the compressive strength of normal and high-performance concretes using M5P model tree algorithm. *Construction and Building Materials*, *142*, 199–207.

Bertoni, A., Folgieri, R., & Valentini, G. (2005). Bio-molecular cancer prediction with random subspace ensembles of support vector machines. *Neurocomputing*, *63*, 535–539.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breslow, L. A., & Aha, D. W. (1997). Simplifying decision trees: A survey. *Knowledge Engineering Review*, *12*(1), 1–40.

Bui, Dieu Tien, Ho, T.-C., Pradhan, B., Pham, B.-T., Nhu, V.-H., & Revhaug, I. (2016). GIS-based modeling of rainfall-induced landslides using data mining-based functional trees classifier with AdaBoost, Bagging, and MultiBoost ensemble frameworks. *Environmental Earth Sciences*, *75*(14), 1–22.

Bui, Duie Tien, Khosravi, K., Tiefenbacher, J., Nguyen, H., & Kazakis, N. (2020). Improving prediction of water quality indices using novel hybrid machine-learning algorithms. *Science of The Total Environment*, 137612.

Chen, H., Huang, J. J., & McBean, E. (2020). Partitioning of daily evapotranspiration using a modified shuttleworth-wallace model, random Forest and support vector regression, for a cabbage farmland. *Agricultural Water Management*, *228*, 105923.

Chen, W., Li, Y., Tsangaratos, P., Shahabi, H., Ilia, I., Xue, W., & Bian, H. (2020). Groundwater spring potential mapping using artificial intelligence approach based on kernel logistic regression, random forest, and alternating decision tree models. *Applied Sciences*, *10*(2), 425.

Chen, W., Li, Y., Xue, W., Shahabi, H., Li, S., Hong, H., Wang, X., Bian, H., Zhang, S., & Pradhan, B. (2020). Modeling flood susceptibility using data-driven approaches of naïve bayes tree, alternating decision tree, and random forest methods. *Science of The Total Environment*, *701*, 134979.

Chernick, M. R. (2002). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. JSTOR.

Cutler, D. R., Edwards Jr, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology*, *88*(11), 2783–2792.

de Santana, F. B., de Souza, A. M., & Poppi, R. J. (2018). Visible and near infrared spectroscopy coupled to random forest to quantify some soil quality parameters. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, *191*, 454–462.

Debeljak, M., & Džeroski, S. (2011). Decision trees in ecological modelling. In *Modelling complex ecological dynamics* (pp. 197–209). Springer.

Dogan, A., & Birant, D. (2020). Machine learning and data mining in manufacturing. *Expert Systems with Applications*, 114060.

Erdal, H., & Karahanoğlu, İ. (2016). Bagging ensemble models for bank profitability: An emprical research on Turkish development and investment banks. *Applied Soft Computing*, *49*, 861–867.

Faizollahzadeh Ardabili, S., Najafi, B., Shamshirband, S., Minaei Bidgoli, B., Deo, R. C., & Chau, K. (2018). Computational intelligence approach for modeling hydrogen production: A review. *Engineering Applications of Computational Fluid Mechanics*, *12*(1), 438–458.

Frank, E., Wang, Y., Inglis, S., Holmes, G., & Witten, I. H. (1998). Using model trees for classification. *Machine Learning*, *32*(1), 63–76.

Gandhi, N., & Armstrong, L. (2016). Applying data mining techniques to predict yield of rice in Humid Subtropical Climatic Zone of India. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 1901–1906.

Granata, F., Gargano, R., & de Marinis, G. (2020). Artificial intelligence based approaches to evaluate actual evapotranspiration in wetlands. *Science of The Total Environment*, *703*, 135653.

Grömping, U. (2009). Variable importance assessment in regression: linear regression versus random forest. *The American Statistician*, *63*(4), 308–319.

Gupta, P. K., Gupta, P. K., & Gupta, P. K. (1999). *Soil, plant, water and fertilizer analysis*. Agro Botanica.

Gweon, H., Li, S., & Mamon, R. (2020). An Effective Bias-Corrected Bagging Method For The Valuation Of Large Variable Annuity Portfolios. *ASTIN Bulletin: The Journal of the IAA*, *50*(3), 853–871.

Henzinger, B. S. G. T. A., Kannan, Y., Nori, A. V, & Rajamani, S. K. (2006). *SYNERGY: A New Algorithm for Property Checking*.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 832–844.

Ho, T. K. (1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, *1*, 278–282.

Jabbar, A. F., & Mohammed, I. J. (2020). Development of an Optimized Botnet Detection Framework based on Filters of Features and Machine Learning Classifiers using CICIDS2017

Dataset. *IOP Conference Series: Materials Science and Engineering*, *928*(3), 32027.

Jiang, D., Zang, W., Sun, R., Wang, Z., & Liu, X. (2020). Adaptive Density Peaks Clustering Based on K-Nearest Neighbor and Gini Coefficient. *IEEE Access*, *8*, 113900–113917.

Jothiprakash, V., & Magar, R. (2009). Soft computing tools in rainfall-runoff modeling. *ISH Journal of Hydraulic Engineering*, *15*(sup1), 84–96.

Karimi, S., Shiri, J., & Marti, P. (2020). Supplanting missing climatic inputs in classical and random forest models for estimating reference evapotranspiration in humid coastal areas of Iran. *Computers and Electronics in Agriculture*, *176*, 105633.

Kazeminezhad, M. H., Etemad-Shahidi, A., & Mousavi, S. J. (2005). Application of fuzzy inference system in the prediction of wave parameters. *Ocean Engineering*, *32*(14–15), 1709–1725.

Khosravi, K., Barzegar, R., Miraki, S., Adamowski, J., Daggupati, P., Alizadeh, M. R., Pham, B. T., & Alami, M. T. (2019). Stochastic Modeling of Groundwater Fluoride Contamination: Introducing Lazy Learners. *Groundwater*.

Khosravi, K., Mao, L., Kisi, O., Yaseen, Z. M., & Shahid, S. (2018). Quantifying hourly suspended sediment load using data mining models: case study of a glacierized Andean catchment in Chile. *Journal of Hydrology*, *567*, 165–179.

Khosravi, K., Pham, B. T., Chapi, K., Shirzadi, A., Shahabi, H., Revhaug, I., Prakash, I., & Bui, D. T. (2018). A comparative assessment of decision trees algorithms for flash flood susceptibility modeling at Haraz watershed, northern Iran. *Science of the Total Environment*, *627*, 744–755.

Kim, H. Il, & Kim, B. H. (2020). Flood Hazard Rating Prediction for Urban Areas Using Random Forest and LSTM. *KSCE Journal of Civil Engineering*, *24*(12), 3884–3896.

Korel, B. (1990). A dynamic approach of test data generation. *Proceedings. Conference on Software Maintenance 1990*, 311–317.

Lahjouj, A., El Hmaidi, A., Bouhafa, K., & Boufala, M. (2020). Mapping specific groundwater vulnerability to nitrate using random forest: Case of Sais basin, Morocco. *Modeling Earth Systems and Environment*, *6*(3), 1451–1466.

Legates, D. R., & McCabe Jr, G. J. (1999). Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. *Water Resources Research*, *35*(1), 233–241.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22.

Mahmood, A. M., Mrithyumjaya, P. G. V. G. K., & Kuppa, R. (2010). A new pruning approach for better and compact decision trees. *International Journal on Computer Science & Engineering*, *2*(8), 2551–2558.

Melesse, A. M., Khosravi, K., Tiefenbacher, J. P., Heddam, S., Kim, S., Mosavi, A., & Pham, B. T. (2020). River water salinity prediction using hybrid machine learning models. *Water*, *12*(10), 2951.

Moosavi, S. M., Jablonka, K. M., & Smit, B. (2020). The Role of Machine Learning in the Understanding and Design of Materials. *Journal of the American Chemical Society*, *142*(48), 20273–20287.

Moriasi, D. N., Arnold, J. G., Van Liew, M. W., Bingner, R. L., Harmel, R. D., & Veith, T. L.

(2007). Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, *50*(3), 885–900.

Nhu, V.-H., Shahabi, H., Nohani, E., Shirzadi, A., Al-Ansari, N., Bahrami, S., Miraki, S., Geertsema, M., & Nguyen, H. (2020). Daily Water Level Prediction of Zrebar Lake (Iran): A Comparison between M5P, Random Forest, Random Tree and Reduced Error Pruning Trees Algorithms. *ISPRS International Journal of Geo-Information*, *9*(8), 479.

Niranjan, A., Nutan, D. H., Nitish, A., Shenoy, P. D., & Venugopal, K. R. (2018). ERCR TV: Ensemble of random committee and random tree for efficient anomaly classification using voting. *2018 3rd International Conference for Convergence in Technology (I2CT)*, 1–5.

Norouzi, H., & Moghaddam, A. A. (2020). Groundwater quality assessment using random forest method based on groundwater quality indices (case study: Miandoab plain aquifer, NW of Iran). *Arabian Journal of Geosciences*, *13*(18), 1–13.

Pahlavan-Rad, M. R., Dahmardeh, K., Hadizadeh, M., Keykha, G., Mohammadnia, N., Gangali, M., Keikha, M., Davatgar, N., & Brungard, C. (2020). Prediction of soil water infiltration using multiple linear regression and random forest in a dry flood plain, eastern Iran. *CATENA*, *194*, 104715.

Pattnaik, B. S., Pattanayak, A. S., Udgata, S. K., & Panda, A. K. (n.d.). Machine learning based soft sensor model for BOD estimation using intelligence at edge. *Complex & Intelligent Systems*, 1–16.

Peters, A., Hothorn, T., & Lausen, B. (2002). ipred: Improved predictors. *R News*, *2*(2), 33–36.

Pham, L. T., Luo, L., & Finley, A. O. (2020). Evaluation of Random Forest for short-term daily streamflow forecast in rainfall and snowmelt driven watersheds. *Hydrology and Earth System Sciences Discussions*, 1–33.

Quiroz, J. C., Mariun, N., Mehrjou, M. R., Izadi, M., Misron, N., & Radzi, M. A. M. (2018). Fault detection of broken rotor bar in LS-PMSM using random forests. *Measurement*, *116*, 273–280.

Ribeiro, M. H. D. M., & dos Santos Coelho, L. (2020). Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. *Applied Soft Computing*, *86*, 105837.

Rokach, L., & Maimon, O. Z. (2008). *Data mining with decision trees: theory and applications* (Vol. 69). World scientific.

Sachdeva, S., & Kumar, B. (2020). Comparison of gradient boosted decision trees and random forest for groundwater potential mapping in Dholpur (Rajasthan), India. *Stochastic Environmental Research and Risk Assessment*, 1–20.

Saggi, M. K., & Jain, S. (2020). Application of fuzzy-genetic and regularization random forest (FG-RRF): Estimation of crop evapotranspiration (ETc) for maize and wheat crops. *Agricultural Water Management*, *229*, 105907.

Salam, R., & Islam, A. R. M. T. (2020). Potential of RT, Bagging and RS ensemble learning algorithms for reference evapotranspiration prediction using climatic data-limited humid region in Bangladesh. *Journal of Hydrology*, *590*, 125241.

Schoppa, L., Disse, M., & Bachmair, S. (2020). Evaluating the performance of random forest for large-scale flood discharge simulation. *Journal of Hydrology*, *590*, 125531.

Shabani, S., Samadianfard, S., Sattari, M. T., Mosavi, A., Shamshirband, S., Kmet, T., & Várkonyi-Kóczy, A. R. (2020). Modeling pan evaporation using Gaussian process regression K-nearest neighbors random forest and Support Vector machines; comparative analysis. *Atmosphere*, *11*(1), 66.

Sharafati, A., Khosravi, K., Khosravinia, P., Ahmed, K., Salman, S. A., Yaseen, Z. M., & Shahid, S. (2019). The potential of novel data mining models for global solar radiation prediction. *International Journal of Environmental Science and Technology*, *16*(11), 7147–7164.

Shirzadi, A., Soliamani, K., Habibnejhad, M., Kavian, A., Chapi, K., Shahabi, H., Chen, W., Khosravi, K., Thai Pham, B., & Pradhan, B. (2018). Novel GIS based machine learning algorithms for shallow landslide susceptibility mapping. *Sensors*, *18*(11), 3777.

Steinfeld, B., Scott, J., Vilander, G., Marx, L., Quirk, M., Lindberg, J., & Koerner, K. (2015). The role of lean process improvement in implementation of evidence-based practices in behavioral health care. *The Journal of Behavioral Health Services & Research*, *42*(4), 504–518.

Travassos, X. L., Avila, S. L., & Ida, N. (2020). Artificial neural networks and machine learning techniques applied to ground penetrating radar: A review. *Applied Computing and Informatics*.

Tsagkrasoulis, D., & Montana, G. (2018). Random forest regression for manifold-valued responses. *Pattern Recognition Letters*, *101*, 6–13.

Vafakhah, M., Loor, S. M. H., Pourghasemi, H., & Katebikord, A. (2020). Comparing performance of random forest and adaptive neuro-fuzzy inference system data mining models for flood susceptibility mapping. *Arabian Journal of Geosciences*, *13*, 417.

Wang, Y., & Witten, I. H. (1996). *Induction of model trees for predicting continuous classes*.

Wu, C. L., & Chau, K. W. (2011). Rainfall–runoff modeling using artificial neural network coupled with singular spectrum analysis. *Journal of Hydrology*, *399*(3–4), 394–409.

Wu, X., & Kumar, V. (2009). *The top ten algorithms in data mining*. CRC press.

Yin, A. (2020). Equity premium prediction and optimal portfolio decision with Bagging. *The North American Journal of Economics and Finance*, *54*, 101274.

Zeng, X., Schnier, S., & Cai, X. (2021). A data-driven analysis of frequent patterns and variable importance for streamflow trend attribution. *Advances in Water Resources*, *147*, 103799.

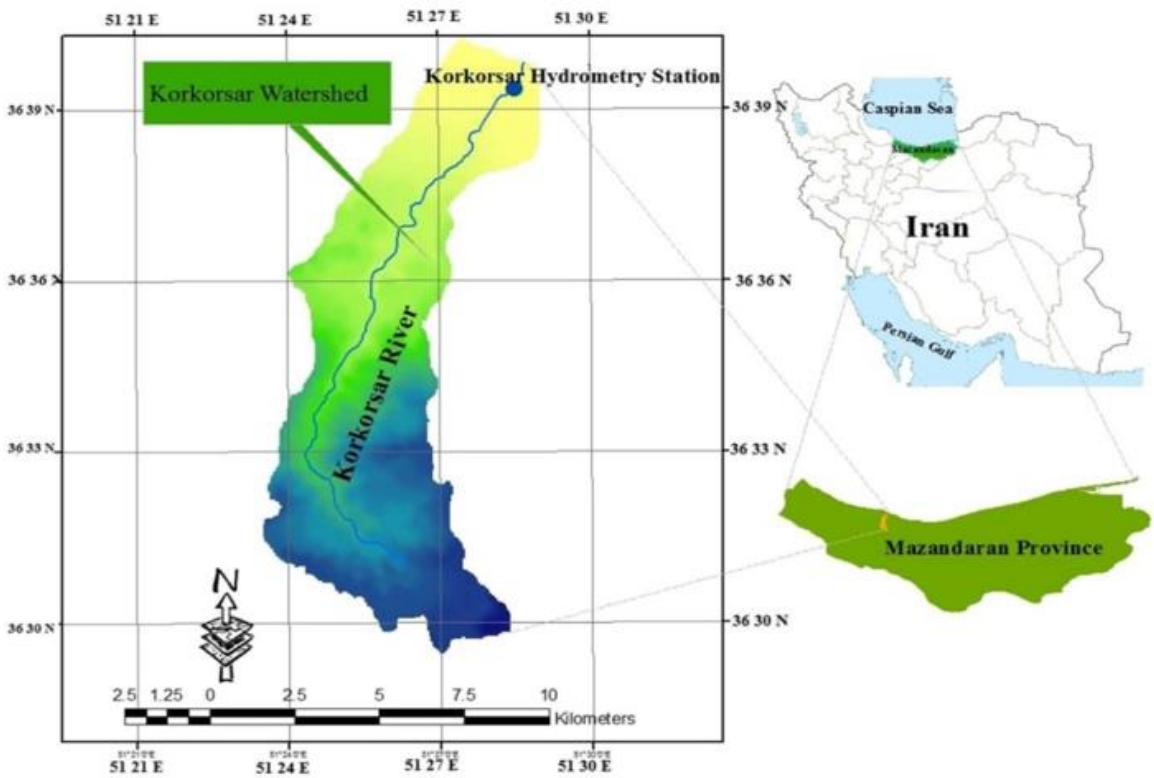
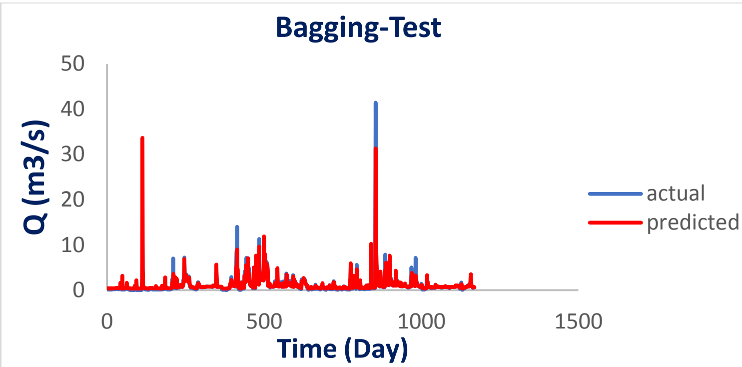
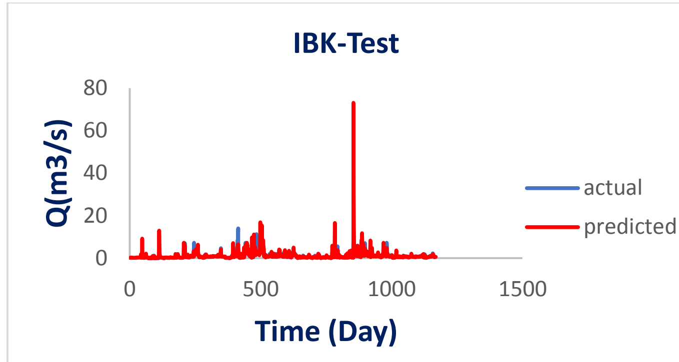
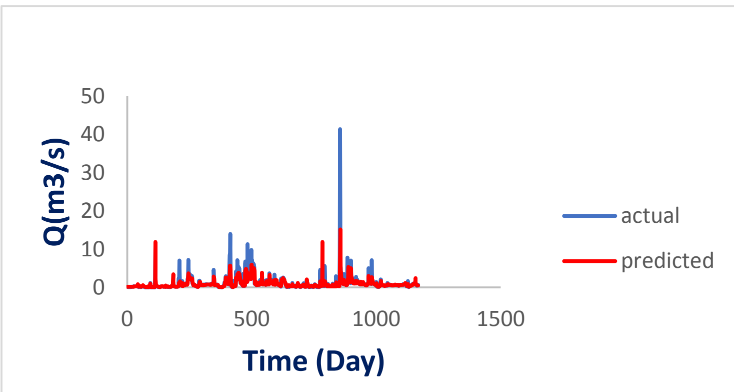
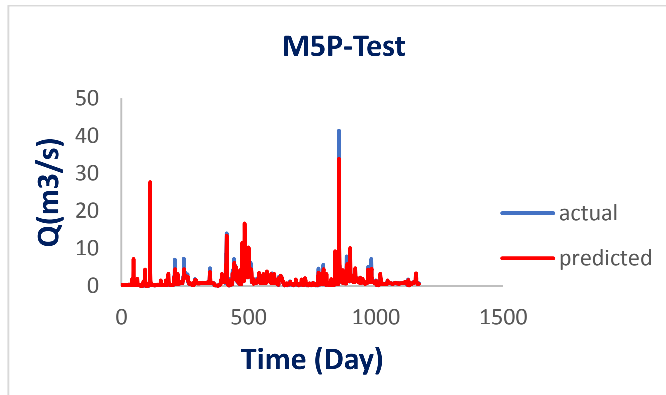


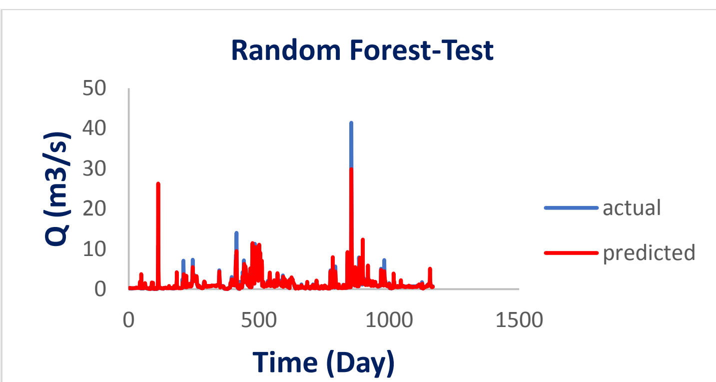
Fig 1. Location of the Kurkursar River

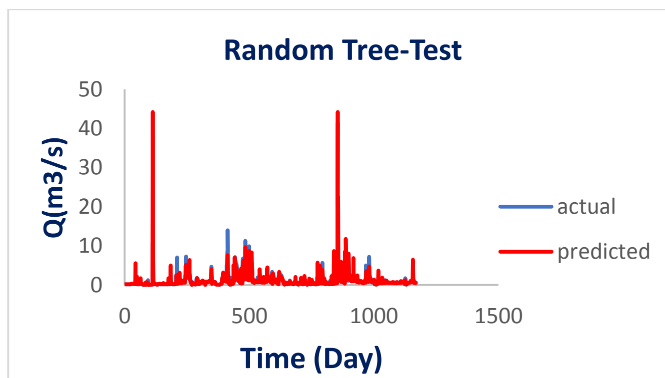

Bagging-Test
Q (m3/s)
Time (Day)
actual
predicted


IBK-Test
Q(m3/s)
Time (Day)
actual
predicted


Q(m3/s)
Time (Day)
actual
predicted


M5P-Test
Q(m3/s)
Time (Day)
actual
predicted


Random Forest-Test
Q (m3/s)
Time (Day)
actual
predicted


Random Tree-Test
Q(m3/s)
Time (Day)
actual
predicted

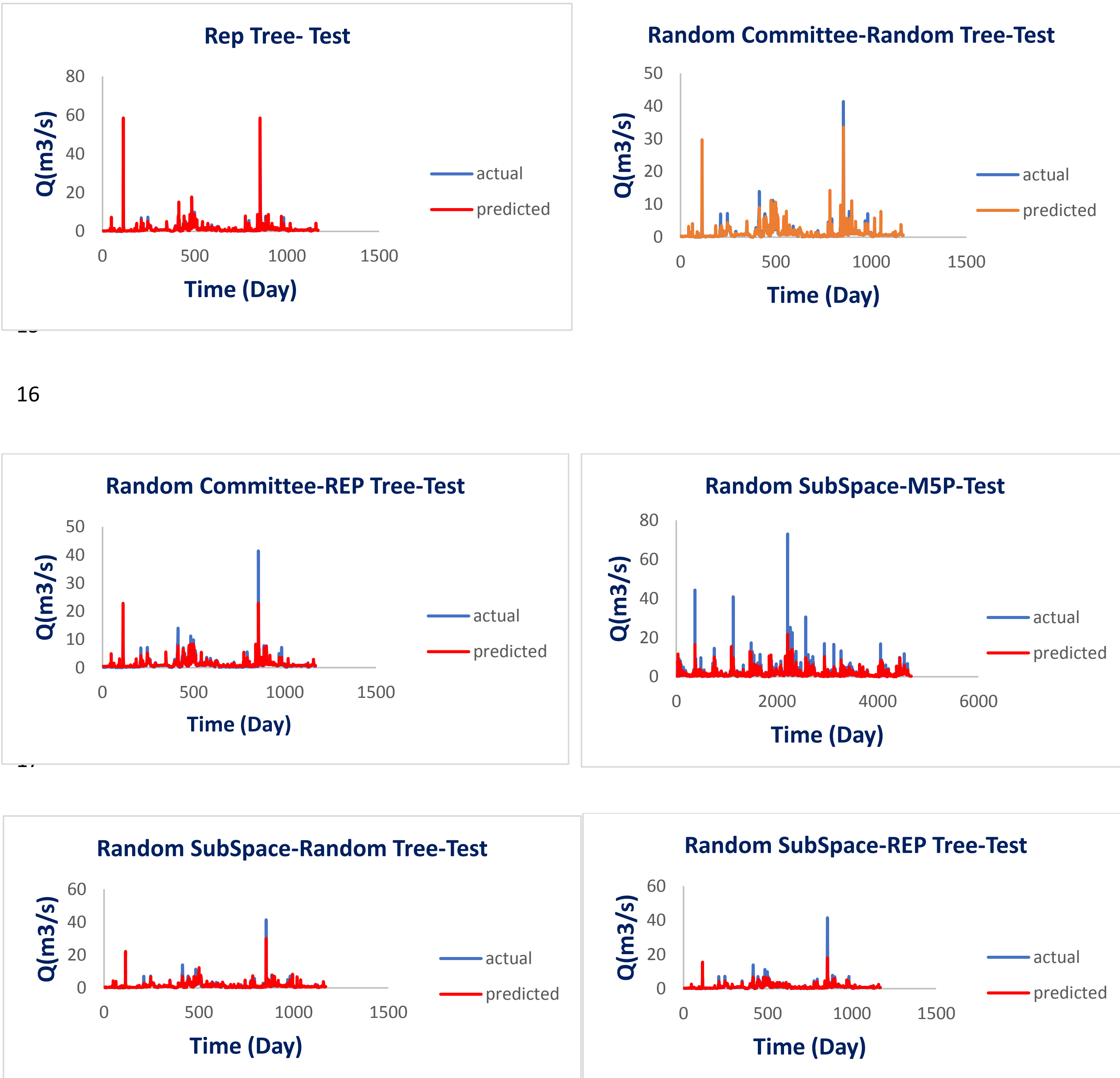
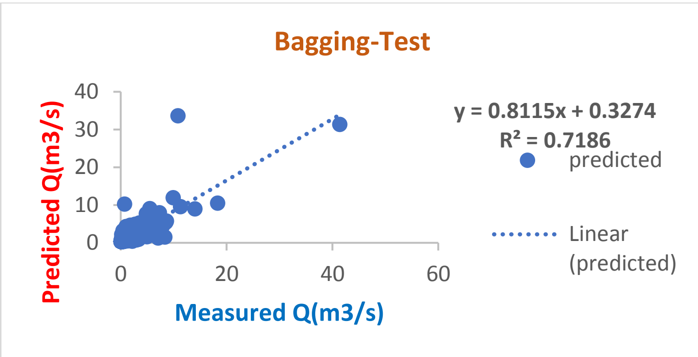
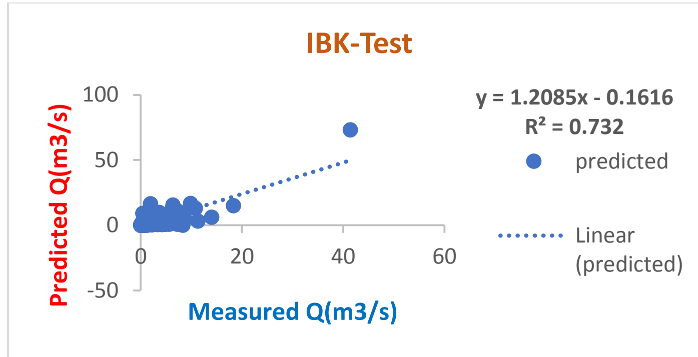
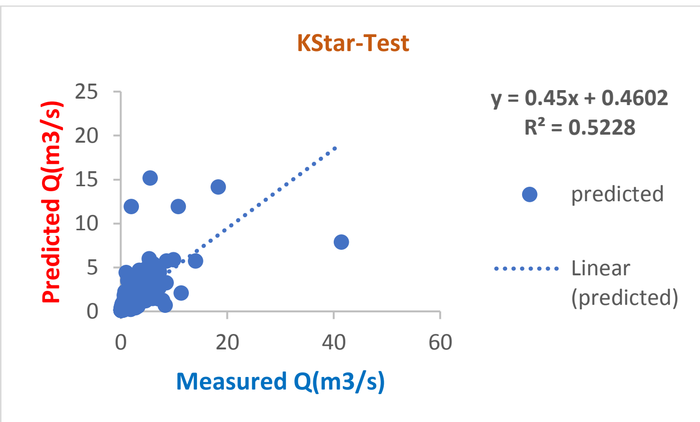
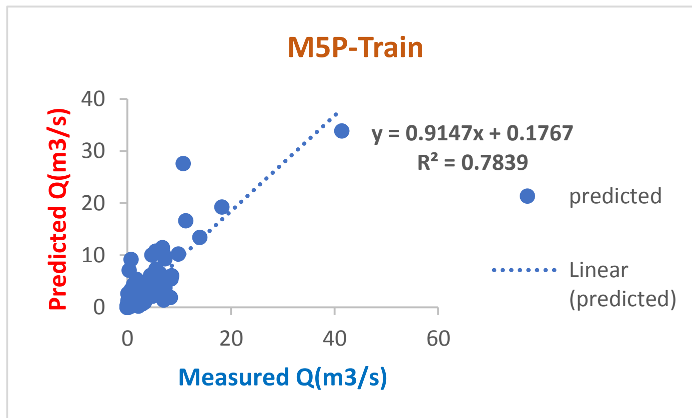


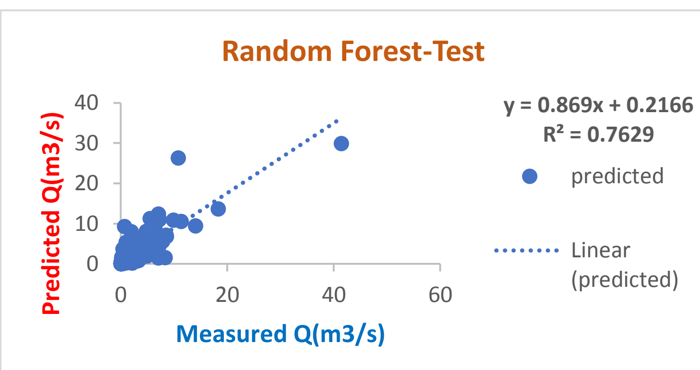
Fig 2. Time variation graphs for the predicted and observed values (testing phase)

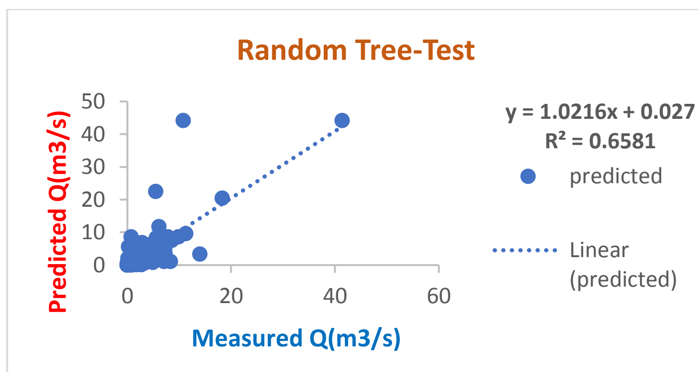

Bagging-Test
y = 0.8115x + 0.3274
R² = 0.7186
predicted
Linear (predicted)
Predicted Q(m3/s)
Measured Q(m3/s)

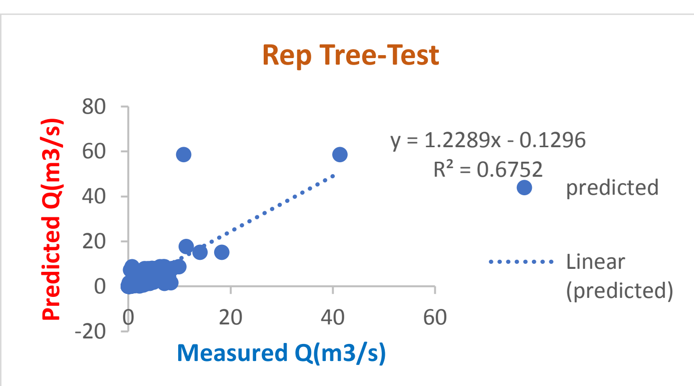

IBK-Test
y = 1.2085x - 0.1616
R² = 0.732
predicted
Linear (predicted)
Predicted Q(m3/s)
Measured Q(m3/s)

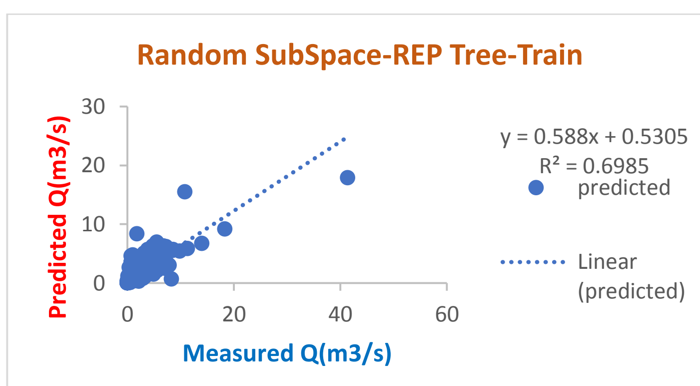

KStar-Test
y = 0.45x + 0.4602
R² = 0.5228
predicted
Linear (predicted)
Predicted Q(m3/s)
Measured Q(m3/s)


M5P-Train
y = 0.9147x + 0.1767
R² = 0.7839
predicted
Linear (predicted)
Predicted Q(m3/s)
Measured Q(m3/s)


Random Forest-Test
y = 0.869x + 0.2166
R² = 0.7629
predicted
Linear (predicted)
Predicted Q(m3/s)
Measured Q(m3/s)


Random Tree-Test
y = 1.0216x + 0.027
R² = 0.6581
predicted
Linear (predicted)
Predicted Q(m3/s)
Measured Q(m3/s)


Rep Tree-Test
y = 1.2289x - 0.1296
R² = 0.6752
predicted
Linear (predicted)
Predicted Q(m3/s)
Measured Q(m3/s)


Random SubSpace-REP Tree-Train
y = 0.588x + 0.5305
R² = 0.6985
predicted
Linear (predicted)
Predicted Q(m3/s)
Measured Q(m3/s)

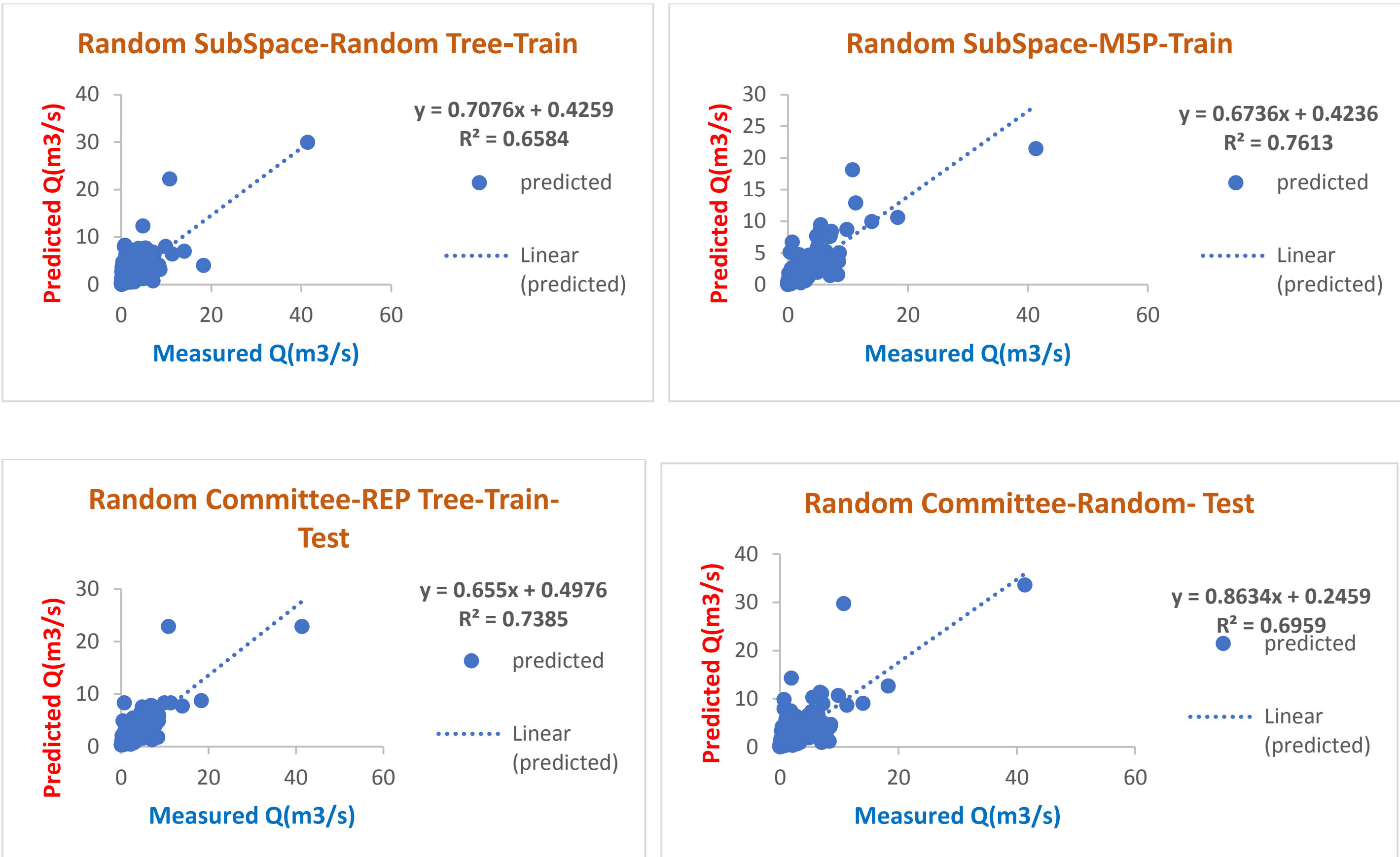
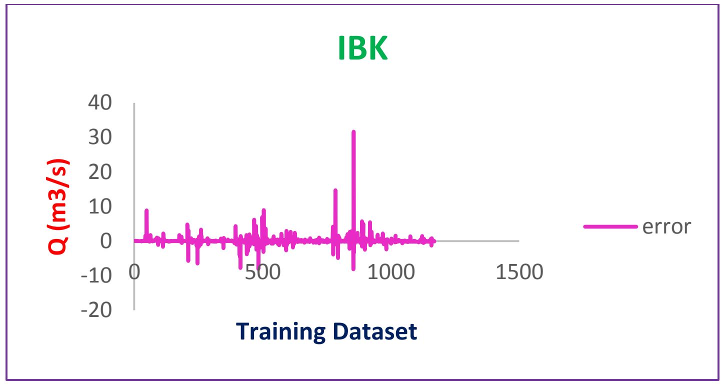
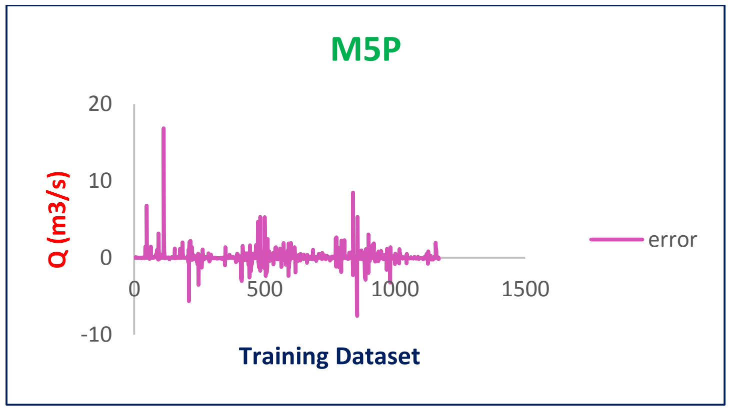


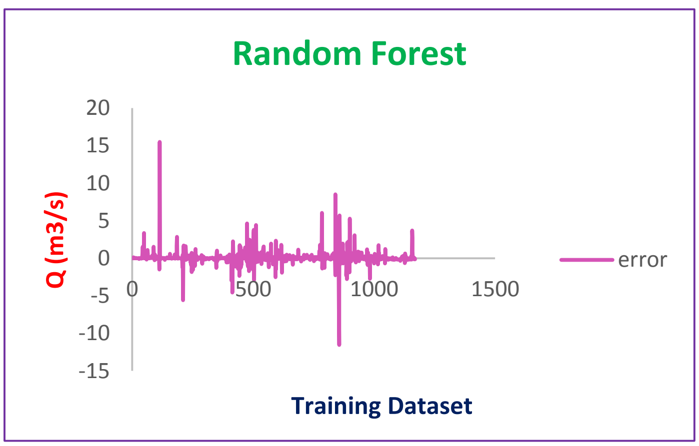
Fig 3. Scatter plots for the predicted and observed values (testing phase)

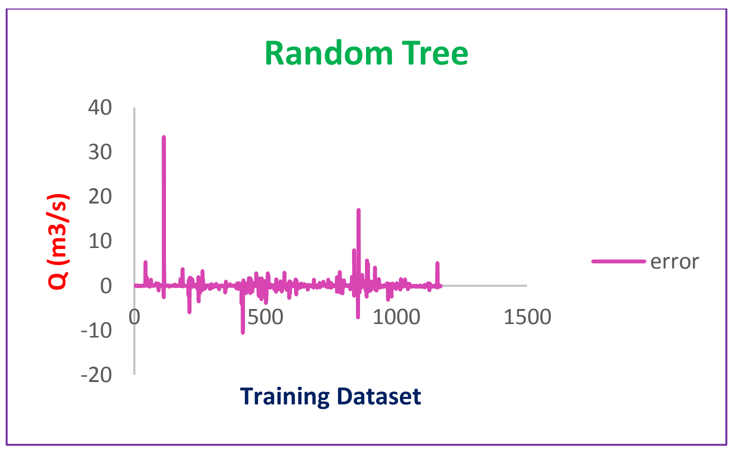

Bagging
30
20
10
0
-10
-20
Q (m3/s)
0
500
1000
1500
error
Training Dataset


IBK
40
30
20
10
0
-10
-20
Q (m3/s)
0
500
1000
1500
error
Training Dataset


K Star
20
10
0
-10
-20
-30
-40
Q (m3/s)
0
500
1000
1500
error
Training Dataset


M5P
20
10
0
-10
Q (m3/s)
0
500
1000
1500
error
Training Dataset


Random Forest
20
15
10
5
0
-5
-10
-15
Q (m3/s)
0
500
1000
1500
error
Training Dataset


Random Tree
40
30
20
10
0
-10
-20
Q (m3/s)
0
500
1000
1500
error
Training Dataset

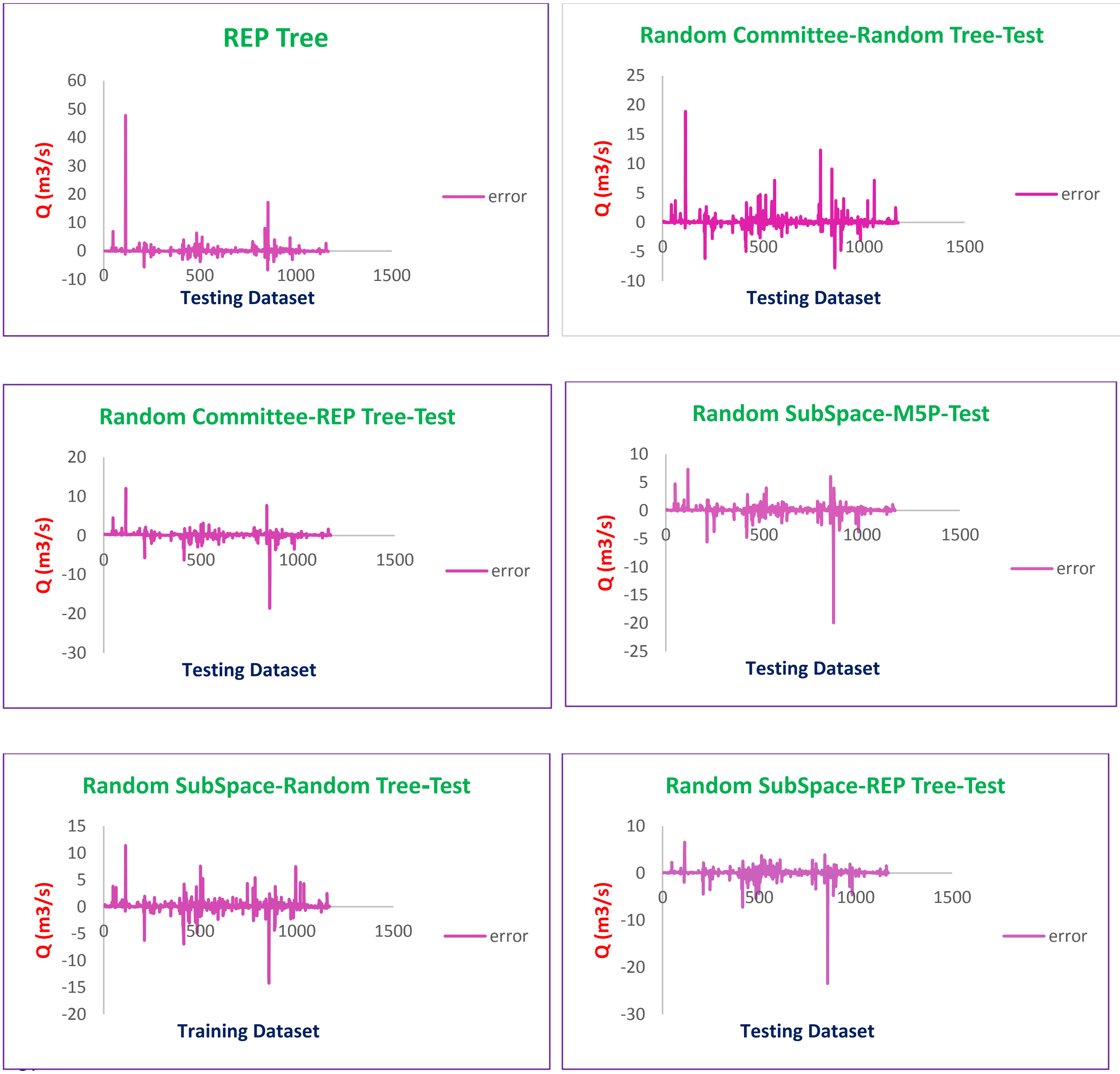
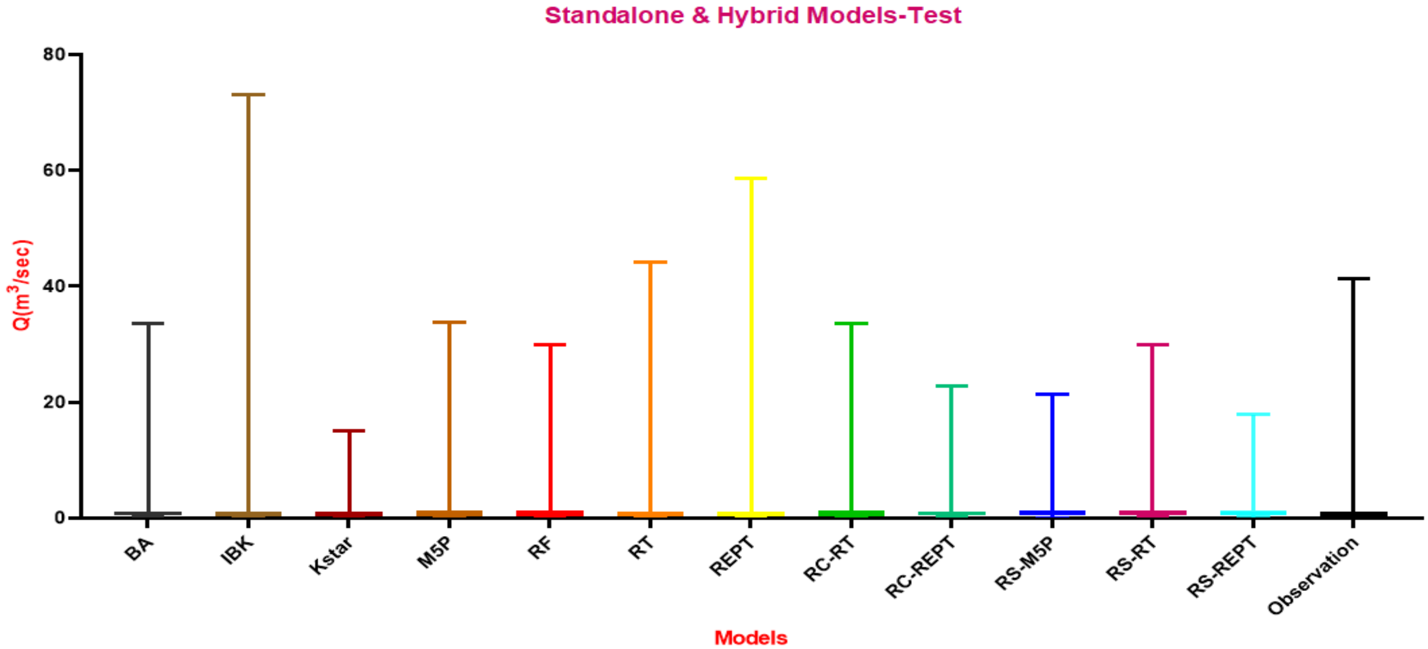


Fig 4. Error graphs for the predicted and observed values (testing phase)



41 **Fig 5.** Box plots for determining the best performance with the applied algorithms

Table

Table 1. Performance indicators for streamflow prediction

Factor	Equation	Factor role	Ref	Range	Performance
R²	$R^2 = 1 - \left(\frac{\sum_{i=1}^{i=N}\left(Q_t^{ob} - Q_t^{pr}\right)^2}{\sum_{i=1}^{i=N}(Q_t^{ob})^2}\right)$	To show the accuracy of prediction	(Faizollahzadeh Ardabili et al., 2018)	$0.7 \leq R^2 \leq 0.1$	Very good
				$0.6 \leq R^2 \leq 0.7$	Good
				$0.5 \leq R^2 \leq 0.6$	Satisfactory
				$0 \leq R^2 \leq 0.5$	Unsatisfactory
RMSE	$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{i=N}\left(Q_t^{pr} - Q_t^{ob}\right)^2}$	To show accuracy	(Faizollahzadeh Ardabili et al., 2018)	-	The lower value is better
MAE	$MAE = \frac{1}{N}\sum_{i=1}^{i=N}\left(Q_t^{pr} - Q_t^{ob}\right)^2$	To show accuracy	(Faizollahzadeh Ardabili et al., 2018)	-	The lower value is better
NSE	$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{i=N}\left(Q_t^{pr} - Q_t^{ob}\right)^2}{\sum_{i=1}^{N}\left(Q_t^{pr} - \overline{Q_t^{pr}}\right)^2}$	Predictive power classification	(Moriasi et al., 2007)	$0.75 < NSE \leq 1.00$	Very good
				$0.65 < NSE \leq 0.75$	Good
				$0.50 < NSE \leq 0.65$	Satisfactory
				$0.4 < NSE \leq 0.50$	Acceptable
				$NSE \leq 0.4$	Unsatisfactory
PBIAS	$\mathrm{PBIAS} = \frac{\sum_{i=1}^{i=N}\left(Q_t^{pr} - Q_t^{ob}\right)}{\sum_{i=1}^{i=N} Q_t^{pr}}$	Predictive power classification	(Legates & McCabe Jr, 1999)	$PBIAS < \pm 10\%$	Very good
				$\pm 10\% \leq PBIAS < \pm 15\%$	Good
				$\pm 15\% \leq PBIAS < \pm 25\%$	Satisfactory
				$PBIAS \geq \pm 25\%$	Unsatisfactory
RSR	$\mathrm{PSR} = \sqrt{\frac{\sum_{i=1}^{i=N}\left(Q_t^{pr} - Q_t^{ob}\right)^2}{\sum_{i=1}^{i=N}\left(Q_t^{pr} - \overline{Q_t^{pr}}\right)^2}}$	Predictive power classification	(Gupta et al., 1999)	$0 \leq RSR \leq 0.50$	Very good
				$0.50 < RSR \leq 0.60$	Good
				$0.60 < RSR \leq 0.70$	Satisfactory
				$RSR > 0.70$	Unsatisfactory

Table 2. Statistical parameters for train and test phases

Dataset		R	Q
Min	Train	0	0.002
	Test	0	0.004
Max	Train	149	73.1
	Test	147	41.4
Mean	Train	3.415	1.298
	Test	3.626	1.115
StdDEV	Train	11.165	2.168
	Test	11.451	1.893

Table 3. Correlation coefficient (CC) between input and output variables

Output variable	Input variable											
	R(t)	R(t-1)	R(t-2)	R(t-3)	R(t-4)	R(t-5)	R(t-6)	Q(t-1)	Q(t-2)	Q(t-3)	Q(t-4)	Q(t-5)
Q(t)	0.563	0.281	0.124	0.0705	0.0681	0.0552	0.0612	0.463	0.297	0.251	0.225	**0.211**

Table 4. Input variable combinations

Model Number	Input variables
1	**R(t)**
2	**R(t), Q(t-1)**
3	**R(t), Q(t-1), Q(t-2)**
4	**R(t), R(t-1), Q(t-1), Q(t-2)**
5	**R(t), R(t-1), Q(t-1), Q(t-2), Q(t-3)**
6	**R(t), R(t-1), Q(t-1), Q(t-2), Q(t-3), Q(t-4)**
7	**R(t), R(t-1), Q(t-1), Q(t-2), Q(t-3), Q(t-4), Q(t-5)**
8	**R(t), R(t-1), Q(t-1), Q(t-2), Q(t-3), Q(t-4), Q(t-5), Q(t-6)**
9	**R(t), R(t-1), R(t-2), Q(t-1), Q(t-2), Q(t-3), Q(t-4), Q(t-5), Q(t-6)**
10	**R(t), R(t-1), R(t-2), R(t-3), Q(t-1), Q(t-2), Q(t-3), Q(t-4), Q(t-5), Q(t-6)**
11	**R(t), R(t-1), R(t-2), R(t-3), R(t-4), Q(t-1), Q(t-2), Q(t-3), Q(t-4), Q(t-5), Q(t-6)**
12	**R(t), R(t-1), R(t-2), R(t-3), R(t-4), R(t-5), Q(t-1), Q(t-2), Q(t-3), Q(t-4), Q(t-5), Q(t-6)**
13	**R(t), R(t-1), R(t-2), R(t-3), R(t-4), R(t-5), R(t-6), Q(t-1), Q(t-2), Q(t-3), Q(t-4), Q(t-5), Q(t-6)**

Table 5. Selection of the best combinations for the standalone models

Model Number	M5P		RF		RT		REPT		BA		IBk		Kstar	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	Test
1	1.7055	1.4074	1.4122	1.7606	1.3493	1.9132	1.5542	1.9707	1.6407	1.4364	1.3516		2.0358	1.7237
2	1.273	0.9264	0.6052	0.0972	0.2677	2.3232	1.0479	1.6731	1.1659	1.1659	0.2771	1.9515	1.6562	1.4353
3	1.254	1.2038	0.5721	0.9549	0.1119	2.4254	1.0201	1.6622	1.1573	1.033	0.1362	1.6317	1.2157	1.3859
4	1.2117	1.2390	0.5603	0.9520	0.0897	1.3945	1.0334	1.0334	1.1451	1.0314	0.2118	1.4402	1.9932	1.3323
5	1.2258	1.2702	0.5551	1.0015	0.0832	1.7810	0.9853	1.7262	1.1259	1.0546	0.3725	1.448	0.6121	1.3295
6	1.2007	1.2855	0.5489	0.9687	0.0727	1.5968	0.985	1.7317	1.1359	1.0782	0.3699	1.4619	0.3553	1.3312
7	1.1911	1.0980	0.5579	0.9695	0.0604	1.8760	0.9813	1.7334	1.140	1.0702	0.3673	1.4438	0.2053	1.3651
8	1.1851	1.0456	0.5510	0.9582	0.0521	1.7433	1.1029	1.7553	1.1368	1.073	0.3671	1.4664	0.1284	1.426
9	1.1612	1.1476	0.5486	0.9591	0.0528	1.1808	1.1027	1.7551	1.1332	1.0754	0.3685	2.1	0.1074	1.5069
10	1.1613	1.1311	0.5778	1.011	0.0586	1.4305	1.0922	1.7686	1.1326	1.0752	0.3686	2.1028	0.0833	1.5061
11	1.1598	1.1333	0.5667	0.9915	0.0475	1.7151	1.0922	1.7686	1.1327	1.0854	0.3668	2.0952	0.0748	1.5251
12	1.1557	1.1375	0.5679	1.0101	0.0459	1.6121	1.0922	1.7686	1.1299	1.0851	0.3719	21039	0.0664	1.5302
13	1.1378	1.1931	0.5650	1.033	0.0458	1.6121	1.0922	1.7686	1.1306	1.0852	0.3653	2.1001	0.0545	1.5379

Table 6. R^2 coefficient for standalone and hybrid models

Model Number	**Models**	**Train**	**Test**
1	BA	0.5485	0.7186
2	M5P	0.6586	0.7839
3	RF	0.9431	0.7629
4	RT	0.9983	0.6581
5	REPT	0.7728	0.6752
6	IBK	0.9905	0.732
7	K Star	0.9398	0.5228
8	RC-RF	0.9497	0.7347
9	RC-RT	0.9997	0.6959
10	RC-REPT	0.7171	0.7385
11	RS-M5P	0.6759	0.7613
12	RS-RT	0.9984	0.6584
13	RS-REPT	0.6027	0.6985

Table 7. Results of the evaluation criteria for standalone and hybrid models (testing phase)

		RMSE	MAE	NSE	PBIAS	PSR
Bagging	Train	1.0685	0.327892	0.701855	-2.968	0.54603
	Test	0.51615	0.092012	0.746796	-10.511	0.50319
IBK	Train	0.21177	0.028575	0.969183	-0.3674	0.17555
	Test	1.17545	0.291303	0.506284	-6.3558	0.70265
K Star	Train	0.61199	0.146363	0.742641	5.27741	0.50731
	Test	1.08512	0.230033	0.579253	13.7364	0.64865
M5P	Train	1.27287	0.35674	-0.11331	0.14091	1.05514
	Test	0.75604	0.215459	0.79575	-7.3175	0.45194
RF	Train	0.56023	0.156762	0.784338	-0.1269	0.4644
	Test	0.77696	0.22041	0.784294	-6.3254	0.46444
RT	Train	0.08982	0.103748	0.994456	-0.0007	0.07446
	Test	1.13815	0.263821	0.53712	-4.5818	0.68035
REPT	Train	1.03332	0.328143	0.266305	0.00898	0.85656
	Test	1.36649	0.245161	0.33276	-11.274	0.81685
RS-REPT	Train	1.44856	0.412903	-0.44184	-0.0008	1.20077
	Test	0.87401	0.264426	0.727041	-6.3691	0.52245
RS-RT	Train	0.0867	0.029795	0.994835	0.00025	0.07187
	Test	0.91089	0.304613	0.703517	-8.9486	0.5445
RS-M5P	Train	1.31697	0.361223	-0.19179	0.24102	1.09169
	Test	0.77176	0.228345	0.787172	-5.3465	0.46133
RC-REPT	Train	1.31823	0.407127	-0.19407	-0.0021	1.09274
	Test	0.80897	0.271162	0.76615	-10.123	0.48358
RC-RT	Train	0.03642	0.021295	0.999088	0.00033	0.03019
	Test	0.90939	0.248427	0.704491	-8.3911	0.54361
RC-RF	Train	0.55407	0.159346	0.789053	0.33768	0.45929
	Test	0.88352	0.324295	0.74151	-3.5204	0.50842

Table 8: Optimum values for models parameters

Parameter	Optimum value										
	BA	IBK	K STAR	M5P	RF	RT	REPT	RC-RT	RS-M5P	RS-REPT	RC-RF
Bag size percentage	100	_	_	_	100	100	_	_	_	_	_
Batch size	100	100	_	_	100	100	100	100	100	100	100
Num decimal places	2	2	2	2	2	2	2	2	2	2	2
Num execution slots	1	_	_	_	1	_	_	1	1	1	1
Num iteration	10	_	_	_	100	_	_	10	10	10	10
Seeds	1	_	_		1	1		1	1	1	
Min num instances	_	_	_	4	_	2	2	_	_	_	_
Build regression tree	_	_	_	False	_	_	_	_	_	_	_
Do not check capabilities	_	False	False	False	_	_	False	False	_	_	False
Debug	_	False	False	False	False	False	False	False	False	False	False
Unpruned	_	_	_	False	_	_	False	_	_	_	_
Use unsmoothed	_	_	_	False	_	_	_	_	_	_	_
Max depth	_	_	_	_	0	0	-1	_	_	_	_
Num feathers	_	_	_	_	_	_	_	_	_	_	_
K value	_	_	0	_		0		_	_	_	_
Min variance Prop	_	_	_	_	_	_	0.001	_	_	_	_
Num folds	_	_	_	_		0	_	_	_	_	_
Global blend	_	_	20	_	_	_	_	_	_	_	_
Entropic auto blend	_	_	False	_	_	_	_	_	_	_	_
Missing mode	_	_	Average column entropy curve	_	_	_	_	_	_	_	_
KNN	_	1	_	_	_	_	_	_	_	_	_
Cross-validation	_	False	_	_	_	_	_	_	_	_	_
Distance weighting	_	No	_	_	_	_	_	_	_	_	_
Subspace size	_	_	_	_	_	_	_	_	0.5	0.5	_
Number of fold	_	_	_	_	_	_	3	_	_	_	_