water

MDPI

Article

Supervised Machine Learning for Estimation of Total Suspended Solids in Urban Watersheds

Mohammadreza Moeini, Ali Shojaeizadeh and Mengistu Geza *

The Department of Civil and Environmental Engineering, South Dakota School of Mines and Technology, Rapid City, SD 57701, USA; mohammadreza.moeini@mines.sdsmt.edu (M.M.); ali.shojaeizadeh@mines.sdsmt.edu (A.S.)
* Correspondence: Stu.Geza@sdsmt.edu

Abstract: Machine Learning (ML) algorithms provide an alternative for the prediction of pollutant concentration. We compared eight ML algorithms (Linear Regression (LR), uniform weighting k-Nearest Neighbor (UW-kNN), variable weighting k-Nearest Neighbor (VW-kNN), Support Vector Regression (SVR), Artificial Neural Network (ANN), Regression Tree (RT), Random Forest (RF), and Adaptive Boosting (AdB)) to evaluate the feasibility of ML approaches for estimation of Total Suspended Solids (TSS) using the national stormwater quality database. Six factors were used as features to train the algorithms with TSS concentration as the target parameter: Drainage area, land use, percent of imperviousness, rainfall depth, runoff volume, and antecedent dry days. Comparisons among the ML methods demonstrated a higher degree of variability in model performance, with the coefficient of determination (R^2) and Nash–Sutcliffe (NSE) values ranging from 0.15 to 0.77. The Root Mean Square (RMSE) values ranged from 110 mg/L to 220 mg/L. The best fit was obtained using the AdB and RF models, with R^2 values of 0.77 and 0.74 in the training step and 0.67 and 0.64 in the prediction step. The NSE values were 0.76 and 0.72 in the training step and 0.67 and 0.62 in the prediction step. The predictions from AdB were sensitive to all six factors. However, the sensitivity level was variable.

Keywords: stormwater quality; urban watersheds; machine learning algorithms; total suspended solids

check for updates

Citation: Moeini, M.; Shojaeizadeh, A.; Geza, M. Supervised Machine Learning for Estimation of Total Suspended Solids in Urban Watersheds. *Water* **2021**, *13*, 147. https://doi.org/10.3390/w13020147

Received: 11 November 2020
Accepted: 8 January 2021
Published: 10 January 2021

1. Introduction

Urbanization causes the degradation of stormwater quality and limits the industrial, agricultural, hydropower, and recreational use of water bodies [1]. Stormwater runoff contains suspended solids, nutrients, trace organic compounds, heavy metals, and pathogens discharged into natural water bodies, impairing ecosystems, and human health [2,3]. Total Suspended Solids (TSS) have an effect on water temperature, dissolved oxygen levels, and clarity. The stormwater discharges that wash the surface of the construction site can produce a massive volume of suspended solids [4]. Additionally, other pollutants such as heavy metals and phosphorus can be attached to TSS and be carried through stormwater runoff [5]. Thus, the significance of TSS concentration has led researchers toward the development of modeling approaches, such as empirical models, Physically Based Models (PBMs), and data-driven models. PBMs involve complex numerical techniques and require significant computational time. Whereas empirical models and PBMs may continue to be used in the near future, data-driven approaches will begin to play a significant role in the hydrologic and water quality analysis as big data become available and as computing power improves.

Some of the models use an empirical approach to predict pollutant load. Source Loading and Management Model for Windows (WinSLAMM) is an example of an empirical model for stormwater quantity and quality [6,7]. WinSLAMM requires a significant amount of information about the site's geographic location, site development characteristics (e.g., impervious or pervious area), soil type, land use, and rainfall amount. It also requires specifics about the drainage system (e.g., grass swales, infiltration trench) and the fraction of the area covered by the drainage system. Runoff should be estimated before the evaluation of stormwater quality. WinSLAMM uses estimates of particulate solids concentration with respect to the source area (e.g., paved parking) and land use type (e.g., residential and commercial). Pollutant loads are estimated as a product of runoff volume and source area concentration estimates [8,9].

Some of the PBMs that use a physically based approach for runoff estimation apply empirical approaches to estimate pollutant concentration. An example of such a model is the commonly applied stormwater management model (SWMM). SWMM can be applied for both event-based and continuous simulation in urban watersheds [10,11]. Although SWMM modeling is mostly applied for the prediction of stormwater runoff, it has also been used to estimate pollutant concentration [12]. TSS concentration in SWMM is simulated based on an empirical buildup and wash-off model or the Event Mean Concentration (EMC). The buildup wash-off model involves the accumulation of TSS load during dry periods and wash-off by the first storm [13]. The buildup process is described using exponential, power, or saturation functions [10]. The lack of generally accepted parameters for the buildup and wash-off functions leads to higher uncertainty in water quality prediction [14].

The investigations demonstrate that PBMs can provide guidance on the design and management of water resources. However, these models have been criticized as overparameterized, overly complex, data-intensive, and difficult to use, limiting their broader use. Also, PBMs require the prediction of runoff with reasonable accuracy before simulating pollutant concentrations [5,15,16]. Pertaining to the novelty of Artificial Intelligence (AI), researchers have attempted to take advantage of AI methods. The Machine Learning (ML) technique, one area for the application of AI, is also increasingly implemented in water resources and environmental engineering [17–20] and in many other areas such as the medical and economic fields [21,22]. Given the limitations of empirical models and PBMs, ML techniques should be considered as alternative approaches for the management of urban stormwater and estimation of runoff and water quality. The use of empirical approaches and constant concentration (EMC) for water quality prediction include assumptions that could potentially limit the accuracy of such models. ML approaches are relatively computationally efficient and cost-effective compared to empirical models and PBMs [23]. ML algorithms work based on numerical or categorical relationships between features and target values rather than physical relations between inputs and outputs. ML approaches allow for the potential use of historical data on input features and target values that have been collected over several years and move toward the data-driven prediction of stormwater pollutants.

ML techniques have been frequently applied for the estimation of runoff based on rainfall data. Their use for the prediction of water quality parameters such as TSS or nutrient concentration has been very limited. The first step in applying ML methods is to identify relevant variables, called features, and how these features affect the target value (e.g., water quality). The authors of Refs. [24,25] investigated the possibility of using an Artificial Neural Network (ANN) and Linear Regression (LR) to estimate TSS concentration. They used Unmanned Aerial Vehicle Images (UAV) for extracting TSS data from water bodies. The results demonstrated that the ANN estimated TSS with an R^2 value of 0.84 during the training step and 0.57 during the prediction step. The authors of Ref. [26] used multiple ML algorithms, such as Multiple Linear Regression, Polynomial Regression, Random Forest, Gradient Boosting Algorithm, and Support Vector Machines, to predict a Water Quality Index (WQI) for various lakes. The best results were obtained using the Gradient Boosting Algorithm, with an R^2 value of 0.74 during the training step. The authors of Ref. [27] used Regression Tree (RT) and Support Vector Regression (SVR) for the estimation of specific indicators of stormwater quality. They estimated target parameters, such as biochemical oxygen demand (BOD5), chemical oxygen demand (COD), total suspended solids (TSS), and total dissolved solids (TDS), in the stormwater treatment network based on features such as land use and volume of runoff. The SVR method predicted BOD5, COD, TSS, and TDS with better accuracy ($R^2 > 0.8$) than the RT algorithm ($R^2 > 0.7$). These studies demonstrate the potential for the application of ML techniques for the management of urban water resources. The methods were used to estimate water quality targets (e.g., TSS concentration) based on quantitative relationships between features and targets. ML methods could provide important insights about the quantitative relations between environmental factors or features (e.g., land use and antecedent dry days) and target values (e.g., TSS or nutrient concentration), especially when there is an extensive database [28]. Several of the previous studies have focused on the application of ML algorithms to assess streamflow and water quality in the river network. ML methods have rarely focused on pollutant concentration in urban stormwater runoff. In this study, we attempted comprehensively investigate the development and application of data-driven ML algorithms as an alternative to the physically based modeling approach to estimate TSS concentration from urban watersheds. We implemented a new approach where we used comprehensive supervised ML techniques with six single and two ensemble models using six influential factors (drainage area, land use, imperviousness area, rainfall depth, runoff volume, and antecedent days) and one target (TSS concentration). We introduced a sensitivity analysis for evaluating the relevance of the six factors and their relative contribution towards improving the prediction accuracy of the ML algorithms. We used indices such as R^2, NSE, and RMSE to compare the performance of the ML algorithms and identify the best fitting algorithm.

2. Materials and Methods

2.1. Data Source

The input datasets were acquired from the National Stormwater Quality Database (NSQD) (Version 4.02, 2015). This database is the result of an effort by the Environmental Protection Agency (EPA) to compile stormwater quality data in the United States. The Center for Watershed Protection at the University of Alabama was responsible for managing and evaluating the data from the National Pollutant Discharge Elimination System (NPDES) and the Municipal Separate Storm Sewer (MS4) System. The first version of this database was released in 2003. The database included data from more than 3700 storm events collected by 66 agencies from 17 different states in the US [29,30]. The latest version of this database (4.02-2015), used in this study, includes the concentration of different pollutant types, such as TSS, phosphorous, and other pollutants, collected from different EPA rainfall zones. Other data types in the database used as input/features in our ML analysis include drainage area, land use categories, percent imperviousness, antecedent dry days, rainfall depth, and runoff volume [30,31]. We identified input features that may affect TSS concentration. However, the major limitation was finding sufficient data on all the features that could be used to develop an ML model for estimation of the target (TSS concentration). Most of the feature dataset had missing data. Thus, the database was sorted with respect to the features (e.g., rainfall depth, land use, and antecedent dry days) and the target value (TSS concentration) to select datapoints with a complete dataset that included all the features with no missing values.

2.2. Data Preprocessing

Identification and selection of the most important environmental factors or features (e.g., land use and imperviousness area) for the estimation of TSS concentration is an important step in the application of ML techniques. Different approaches can be used to carry out feature identification and selection, such as a t-test, p-value test, and a review of the literature [32]. This study evaluated the critical factors described in earlier studies to identify and select relevant features and extended the list to include new features. One of the features included in this study is the drainage area. The volume of stormwater runoff [33,34] and percent imperviousness [35–37] were also cited as relevant factors [33,35–37]. Imperviousness influences stormwater runoff, temperature, and the rate of pollutant degradation [35]. Rainfall patterns, intensity, depth [38], and antecedent dry days [39,40] were also stated as predictors of TSS concentration. We used rainfall depth as an input feature in this study since the duration and intensity of the rainfall event were not available. Previous studies have demonstrated that antecedent dry days have a crucial effect on the buildup and pollutant concentrations in urban stormwater [35,41]. Thus, this parameter was also included as one of the features in this study. Land use is another parameter that affects water quality. TSS concentration was observed to change significantly with changes in land use [38,40,42]. Land use was categorized into residential, institutional, commercial, industrial, and freeway. A summary of available data considered in this study is provided in Table 1.

Table 1. Summary of features and target values.

Features and Target Value		Description	Unit
Drainage area		Area contributing to discharge and pollutant loading	km^2
Land use	Residential	Percent of the area under each land use category	%
	Institutional		
	Commercial		
	Industrial		
	Open Space		
	Freeway		
Imperviousness		Percent impervious	%
Antecedent dry days		The number of dry days before the storm event	days
Rainfall depth		Rainfall depth	mm
Runoff volume		The volume of runoff during a specific event	m^3
TSS		Measured TSS concentration used as a target value	mg/L

The collinearity among the features was evaluated. As shown in Table 2, the collinearity among the features was generally low except for the rainfall depth and the runoff volume. The runoff volume was directly related to rainfall depth.

Table 2. Correlation matrix.

Features	Drainage Area	Land Use	Imperviousness	Antecedent Dry Days	Rainfall Depth	Runoff Volume
Drainage Area	1	−0.14	−0.04	−0.12	−0.07	0.31
Land use	−0.14	1	0.09	0.017	0	−0.08
Imperviousness	−0.04	0.09	1	0.22	0.04	−0.28
Antecedent dry days	−0.12	0.017	0.22	1	−0.03	−0.03
Rainfall depth	−0.07	0	0.04	−0.03	1	0.46
Runoff volume	0.31	−0.08	−0.28	−0.03	0.46	1

The NSQD database has datapoints with respect to the six mentioned features. We selected 530 datapoints with consistent features and the corresponding target value (TSS concentrations) from 7 different states. The selection was based on whether that dataset contained all the six features and the target value (TSS concentration).

2.3. ML Model Structure

The ML algorithm within an open-source python-based software, named Orange (Version 3.26), was implemented for the analysis after compiling the input dataset. The workflow in Figure 1 describes the three major stages involved in running ML algorithms. Stage 1 relates to the input data, in which the dataset consisting of the features (e.g., drainage area and land use) and the target value (TSS concentration) was compiled as an input dataset. The next step in stage 1 involved dividing the data into the training and prediction datasets. We used 66% of the dataset for the training step and 34% for the prediction step. Stage 2 represents the core of the ML analysis, where ML algorithms were used to analyze the quantitative relationships between the features and the target value. At this stage, eight different ML approaches were applied to the input data. Details about each ML method are presented in subsequent sections. Stage 3 is the output step, where the training results were tested using a 10-fold cross-validation approach. If reasonable estimates of the target value were obtained in the training step based on the R^2, NSE, and RMSE values, the trained ML models were further tested using a separate dataset in the prediction step to determine the accuracy and ability of each ML method to make inferences or generalizations. If the training results did not demonstrate good performance, the internal specific parameters, such as weights, number of nodes, number of neighbors, and depth of the tree, for each ML model were readjusted to obtain better results. Additional information about the evaluation process is included in the results and discussion section. Figure 1 shows the workflow of the model.

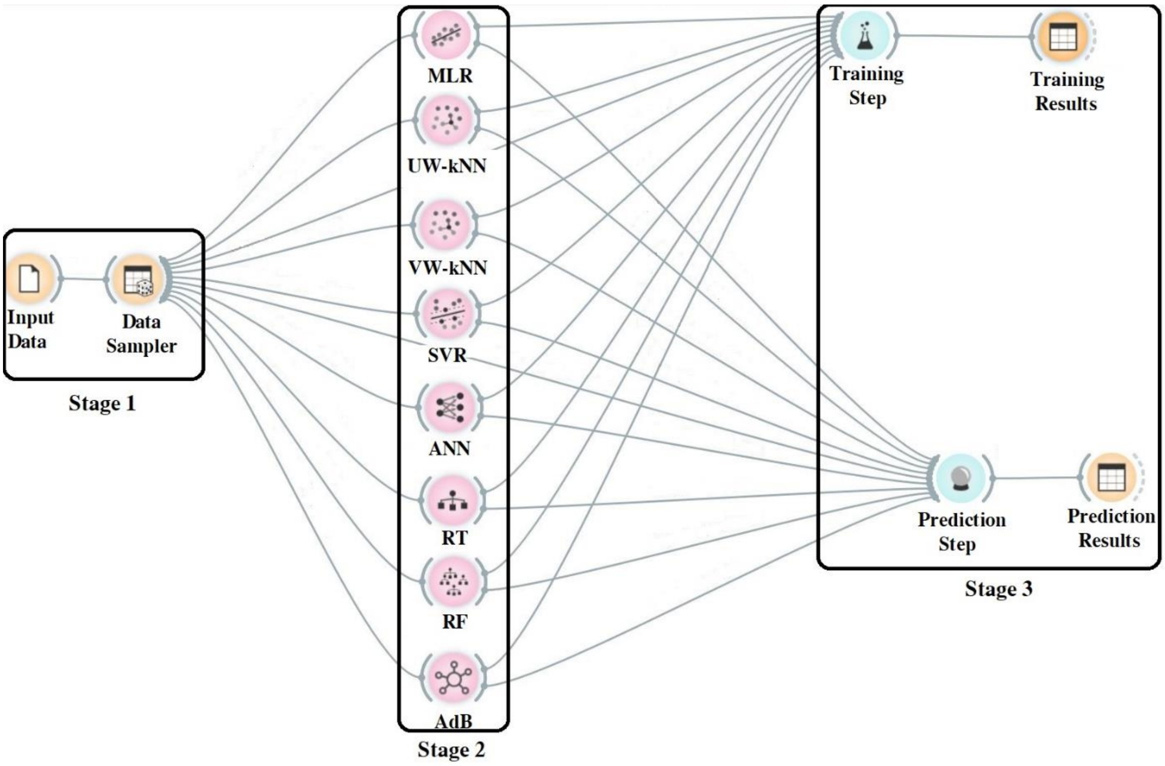

Figure 1. Schematics of the modeling steps in this study.

2.4. Machine Learning (ML) Methods

Machine learning techniques are broadly classified into three distinct groups: Supervised, semi-supervised, and unsupervised algorithms. The supervised learning algorithm includes two steps: Training and prediction. The dataset is divided into two subsets, where the first subset is used for training, and the second subset is used for the prediction step. The training or labeling step helps the model identify the pattern and relationship between features (e.g., land use and imperviousness) and the target value (TSS concentration). The trained model and the relationships developed between input features and target values in the training step are further assessed in the prediction step using a prediction dataset to check the accuracy and generalization ability of the model [43,44]. In contrast, unsupervised learning does not involve a training step. Semi-supervised learning includes both supervised and unsupervised learning processes. Semi-supervised learning techniques use part of the training data for supervised learning and part of the dataset for unsupervised learning. Supervised machine learning algorithms were applied in this study [45,46].

Under supervised learning, there are two types of learning algorithms, described as eager and lazy learner algorithms. Eager learning algorithms learn the pattern from the training dataset and apply it to the prediction dataset, while lazy learning algorithms can perform the training and prediction steps simultaneously. These types of algorithms construct a general form of a trained model, using the training dataset that can be applied to a prediction dataset [47,48]. Supervised machine learning algorithms can also be categorized into linear and nonlinear learners [22]. LR methods are well-known linear methods. The kNN, DT, RF, AdB, and ANN algorithms are regarded as nonlinear algorithms, while the SVR method can be considered as both linear and nonlinear. The complexity of an ML model is related to the model structure and model parameters (e.g., hidden layers and weighting factors in ANN algorithm, or considering leaves and nodes in RT) used to capture the patterns in target values. Increasing the number of model parameters increases the complexity and improves the ability of the model to find the relation between features and target values [49–51].

Complexity improves accuracy in both the training and prediction steps, although in some cases, complex nonlinear models have been observed to decrease the accuracy in the prediction step [51]. In the prediction step, the relation between features and the target values is evaluated by applying the trained model (using a training dataset) to a prediction dataset. The prediction dataset is anticipated to have similar properties, features, and target data structure as the training dataset. Therefore, ML models are expected to detect the pattern and relation between the features and target values, learned in the training step, in the prediction dataset. However, this might not happen for some models, because the algorithm may stick to the details in the training dataset that are absent in the prediction dataset, leading to an overfitting problem. On the contrary, the simpler model (e.g., linear models) might not be able to capture the details in the patterns of the training dataset as accurately as complex models, leading to an underfitting problem and poor performance during the prediction step [45,46]. Thus, it can be challenging to find the optimum level of complexity to prevent underfitting and overfitting problems [49]. ML algorithms can also be applied as a single model or as a combination of ML techniques, called ensemble modeling, presented in Section 2.4.1.

2.4.1. Single-Model ML Methods

Single-learner ML models have one ML method in their structure. In this study, we applied six different single ML algorithms: LR, uniform weighting kNN, variable weighting kNN, RT, SVR, and ANN. The Single-Model ML methods are presented here. The Ensemble-Model methods are presented in the subsequent section.

1. Linear Regression (LR)

Linear Regression (LR) is a well-known approach used in many different fields [52]. The LR model can be applied for single- or multiple-variable problems. Most of the phenomena in nature may be better explained by multiple variables than a single variable [53].

Thus, this approach is traditionally divided into two different techniques: Simple and multiple regression models [54,55]. LR methods describe the relationship between the dependent and independent variables, referred to as a target value and features, respectively, in the ML literature.

2. k-Nearest Neighbor (kNN)

The k-Nearest Neighbor (kNN) is a nonparametric ML method for regression problems [56]. The kNN model stores feature values from all the datapoints (target values) in the input dataset and uses those values to find the similarities between the training dataset and prediction dataset. It uses feature similarity between training and prediction datasets to find similar datapoints (target values) and predict the target values for the prediction dataset. The estimated target value is assigned based on how closely it resembles the values of the training dataset. Two approaches are available for estimating the target value in the kNN algorithm: A uniform weighting averaging approach (UW-kNN) and a variable weighting approach (VW-kNN). In the UW-kNN method, the average of the target of the k-Nearest Neighbors is calculated with the same weighting values. In the VW-kNN algorithm, the inverse distance value is assigned as a weight for each of the k-Nearest Neighbors, and the target value is obtained using a weighted average procedure [57,58]. The kNN regression algorithm uses distance functions, such as Euclidean, Manhattan, and Minkowski, to calculate the distance between datapoints [3,59,60].

3. Support Vector Regression (SVR)

The Support Vector Regression (SVR) was developed based on the support vector machine algorithm using the ε-intensive loss function. The ε-intensive loss function within the SVR process ignores errors that are less than ε, which creates two margins for the fitted function. The error refers to the difference between the calculated and measured values [22,61]. SVR is able to account for any types of nonlinearity mapping using a limited dataset and is able to make a good generalization based on the statistical theories. The method has been frequently applied because of its performance and robust theoretical principles [60]. The SVR method attempts to find the best linear regression for a nonlinear mapping function with multiple features. This algorithm uses a kernel function to map all input data to a high dimensional plane to find the hyperplane with minimum distances to all available datapoints [62].

4. Artificial Neural Network (ANN)

The Artificial Neural Network (ANN) is a powerful computational method capable of capturing nonlinear patterns in functional relationships between features and targets [15,63,64]. The ANN algorithms consist of three basic layers: Input, hidden, and output. Each layer consists of a different number of neurons. The number of neurons in the input layer depends on the number of features, while the number of neurons in the output layer depends on the number of target values [65,66]. Therefore, changing the number of features and target values in the dataset will affect the number of neurons in the input and output layers. The number of hidden layers and neurons needs to be assigned. However, there is no specific method for finding an optimum number of hidden layers and neurons in the hidden layers [67]. Each layer in ANN consists of different numbers of neurons. The neurons in different layers are connected to each other by weighted paths between layers. The multiple layer perceptron (MLP) algorithm, one of the ANN procedures, was used to perform a one-layer efficient neural network in this study [68,69].

5. Regression Tree (RT)

The Regression Tree (RT) is one of the well-known ML algorithms for solving regression problems [70]. The RT algorithm follows two main steps. The first step is a splitting process to subdivide the dataset using a set of decision rules, and the second step is the pruning process [3,71]. The start node in the splitting process, called the root node, consists of all datapoints. Each node in the splitting process selects one of the input features. The splitting process starts using one of the features for the root node and splits the dataset to the left and right branches. Each of the branches indicates a range of values for the selected feature. Thus, the left and right node (operational node) consist of datapoints fitted to the range of each branch. This process continues by considering the least Sum of Squared Residuals (SSR; between the measured and SSR, in which the minimum SSR refers to the most efficient tree) [70,72]. The pruning process is the second method besides SSR, which controls the growing process in the RT algorithm. Two of the well-known criteria for pruning is assigning a maximum depth for the tree and considering a minimum number of datapoints remaining in each node. The final estimation for the target value is the average of datapoints remaining in the last nodes (prediction nodes) [72,73].

2.4.2. Ensemble-Model ML Methods

Ensemble models use multiple single ML methods to perform the training and prediction steps. Ensemble modeling is further divided into three categories: Bagging, boosting, and stacking or blending. Bagging and boosting are two ensemble methods that use multiple similar single algorithms in their structure, while a stacking model can include both single and ensemble methods inside its structure [74]. This study demonstrates the application of bagging and boosting ensemble models for the estimation of TSS concentration.

1. Bagging Ensemble Models

A bagging ensemble model allows the use of an unlimited number of similar single learners inside its structure. This algorithm uses a bootstrap sampling procedure to create multiple random subsamples from the training dataset with replacement [75,76]. Random Forest (RF) is a nonparametric bagging ML method for performing regression analysis [24,77]. The RF method, proposed by the authors of [76], combines several individual RTs (n "tree") to produce more accurate results compared to a single RT algorithm [78,79]. Because of its ensemble structure, it prevents instabilities in the final results that may arise from overfitting, usually observed in a single RT algorithm [26,76,80]. In the RF method, individual trees randomly choose a subset of features as their roots and nodes through the splitting process described in the RT method. Each learner uses a specific bootstrapped dataset obtained using the bootstrap sampling process to develop an RT structure [76]. Therefore, each one of the single learners produces an estimate of target values. In the RF method, a uniform weighting averaging approach is applied to target values obtained from the single learners and reported as the final estimated target value for each datapoint in the training dataset [75,76,81].

2. Boosting Ensemble Models

A boosting ensemble model can use multiple similar single learners inside its structure. The boosting algorithm follows the same process of bootstrap sampling as the bagging algorithm [75,82,83]. Adaptive Boosting (AdB) is a boosting algorithm that is able to use multiple RT learners as its basic learners. This algorithm requires assigning the number of basic learners (n). The basic learners within the AdB algorithm are evaluated iteratively based on the outcomes. The AdB starts the process by creating basic learners (RT) with lower accuracy and improves the subsequent learners based on the results obtained by previous learners. During this process, the algorithm defines weights for the datapoints. AdB starts with a uniform distribution of weights for each datapoint in the subsample. Then, as the process continues, different weights are assigned to each of the datapoints based on their importance to the accuracy of the model. The RT learners trained based on bootstrapped datasets are tested using the original training dataset to identify datapoints that were not predicted accurately. Thus, the AdB places preference on those datapoints predicted inaccurately to improve the accuracy of the next learner [84,85]. This systematic process helps to enhance the accuracy of the AdB method. The final estimate of the target value is a weighted average of the values obtained using individual learners [82,83,86].

3. Results

3.1. Training and Prediction Steps

Supervised ML techniques were used for the prediction of TSS concentration in urban watersheds. Supervised ML algorithms can be trained for specific datasets and used for the prediction step. The study utilized a dataset obtained from NSQD. The dataset from NSQD was divided into two subsets. We used 66% of the dataset randomly selected by the Orange software (version 3.26) for the training step and 34% for the prediction step. All eight algorithms were trained using 350 rows. Ten-fold cross-validation was used for testing the results of the training step. The accuracy of the algorithms in the training step was tested based on indices such as the Coefficient of Determination (R^2) [45], Nash–Sutcliffe (NSE) [14,87], and Root Mean Square (RMSE) [27,88]. Inference about the ability of the algorithms for generalization and accuracy was tested further by applying the trained model to the prediction dataset with 180 datapoints. If satisfactory results were not obtained during the prediction step, the training step was rerun iteratively by changing specific input parameters in each ML method. Indices, such as R^2, NSE, and RMSE, were used to assess model performance in both the training and prediction steps.

The R^2 is the square of the correlation coefficient (r) given by

$$r = \frac{k\left(\sum_{i=1}^{k} P_i O_i\right) - \left(\sum_{i=1}^{k} P_i\right)\left(\sum_{i=1}^{k} O_i\right)}{\sqrt{\left(k\sum_{i=1}^{k} P_i^2 - \left(\sum_{i=1}^{k} P_i\right)^2\right)\left(k\sum_{i=1}^{k} O_i^2 - \left(\sum_{i=1}^{k} O_i\right)^2\right)}}, \quad (1)$$

The second index, the Nash–Sutcliffe equation, is given by

$$NSE = \left(1 - \frac{\sum_{i=1}^{k} (P_i - O_i)^2}{\sum_{i=1}^{k} (\overline{O} - O_i)^2}\right) \quad (2)$$

The RMSE, which was used to calculate the difference between the measured values and calculated values, is given by

$$RMSE = \left(\sqrt{\frac{\sum_{i=1}^{k} (P_i - O_i)^2}{k}}\right), \quad (3)$$

where P_i is the predicted value for the i^{th} row, O_i is the measured data, $\overline{O}$ is the average of the measured values, and k is the total number of the datapoints in a dataset.

For each ML method, we generated four different plots where the first and the second plot displayed the pattern between model predicted target values and measured data obtained from NSQD for the training and prediction datasets, respectively. The third and the fourth plots displayed the R^2, NSE, and RMSE values that demonstrated the performance of each ML algorithm in predicting the target value (TSS concentration) for the training and prediction steps.

3.2. Single ML Algorithm

3.2.1. Multiple Linear Regression (MLR)

The scatterplots in Figure 2a,b show the results obtained using MLR. The R^2 values were 0.21 and 0.29 for the training and prediction steps, respectively. The scatterplots depicted poor fit, particularly for TSS values less than 200 mg/L. The NSE values were 0.15 and 0.27 for the training and prediction steps, respectively. The RMSE values obtained were 220 mg/L and 210 mg/L.

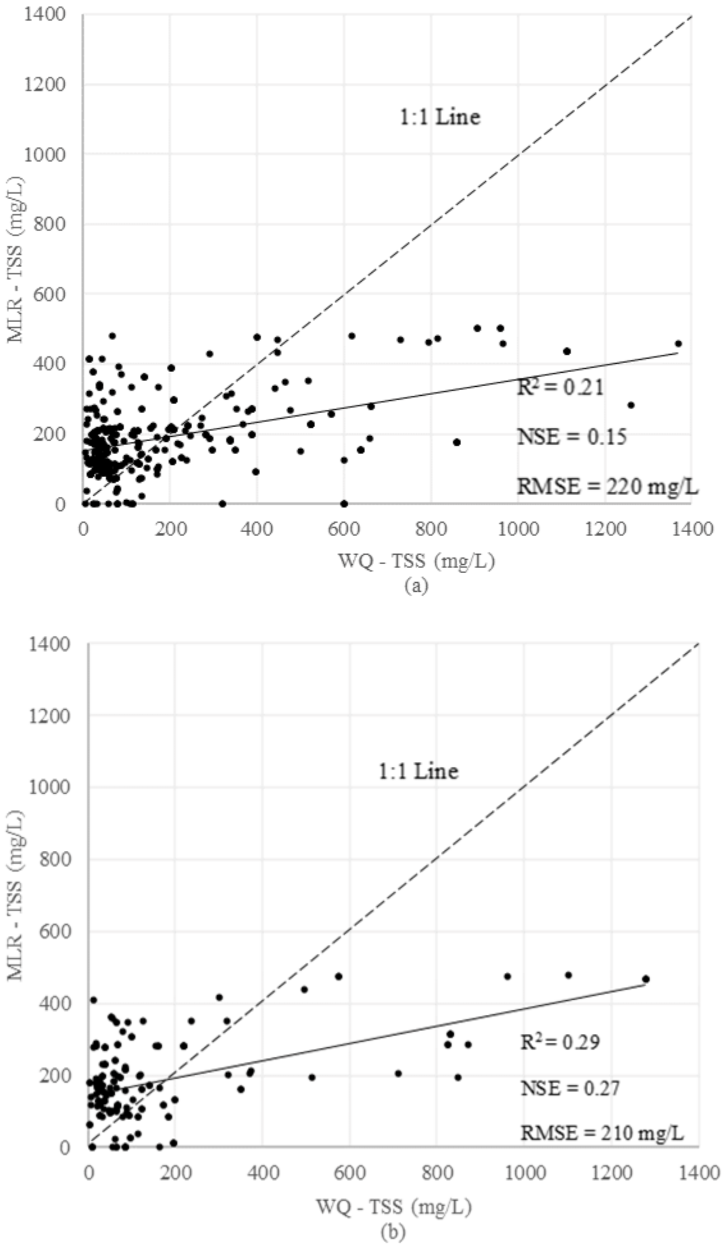

Figure 2. Correlation between the measured values and Multiple Linear Regression (MLR)-predicted Total Suspended Solids (TSS) concentration in (**a**) the training step and (**b**) the prediction step.

3.2.2. k-Nearest Neighbors (kNN)

Three neighbors were used for the uniform weighting method, while 45 neighbors were used for the VW-kNN. Increasing the number of neighbors in a uniform weighting method could improve performance. However, the weighting approach also has a significant effect on the accuracy of the outcome. Figure 3a,b show the R^2, NSE, and RMSE values obtained using the two algorithms. For the training step, the uniform weighting method predicted the target value with R^2, NSE, and RMSE values of 0.71, 0.70, and 160 mg/L, respectively. The R^2, NSE, and RMSE values for the variable weighting method were 0.77, 0.75, and 110 mg/L, respectively.

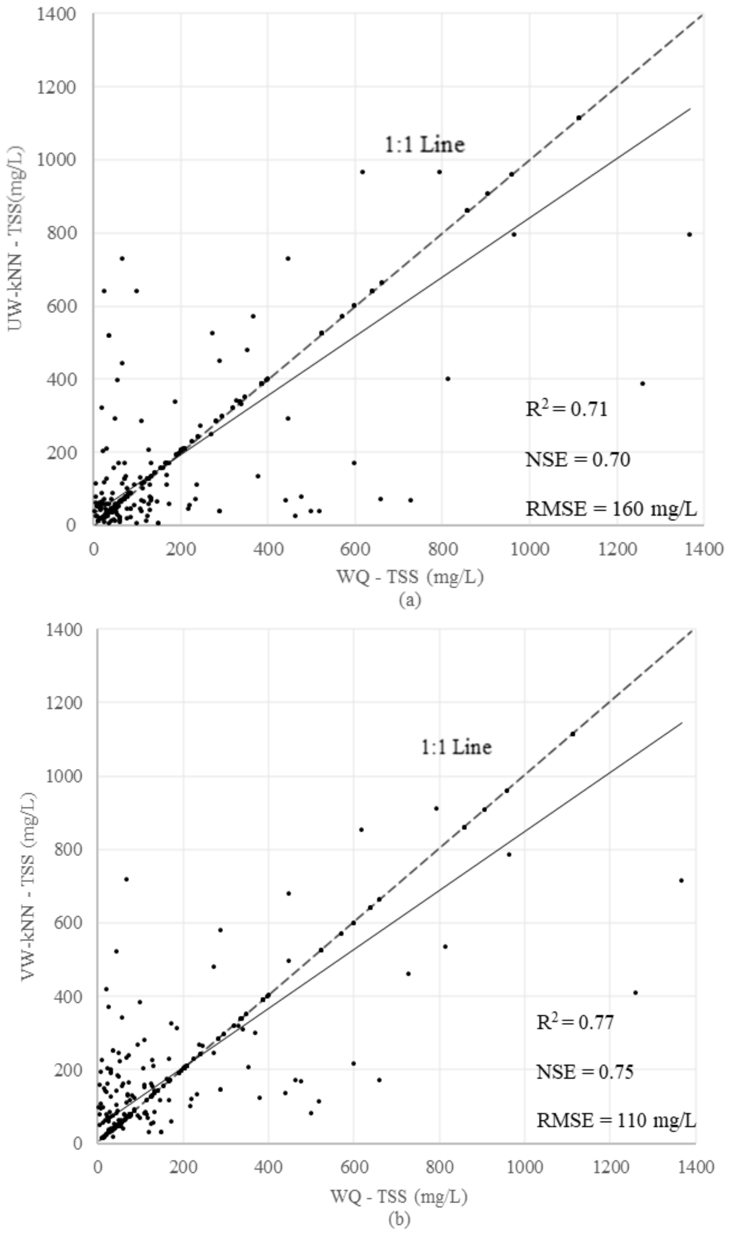

Figure 3. Correlation between the measured values and predicted TSS in the training step for (**a**) uniform weighting k-Nearest Neighbors (UW-kNN) and (**b**) variable weighting (VW)-kNN.

Both models were able to capture the patterns in the prediction step. However, the variable weighting method demonstrated better performance with an R^2 value of 0.56 compared to the uniform weighting method with an R^2 value of 0.36. Thus, the VW-kNN method could enhance the results by reducing the effect of overfitting in the uniform weighting method. The UW-kNN method, with an NSE value of 0.34 and RMSE value of 200 mg/L (Figure 4a), had lower accuracy compared to VW-kNN, with a higher value of NSE (0.55) and a lower RMSE (170 mg/L) value (Figure 4b).

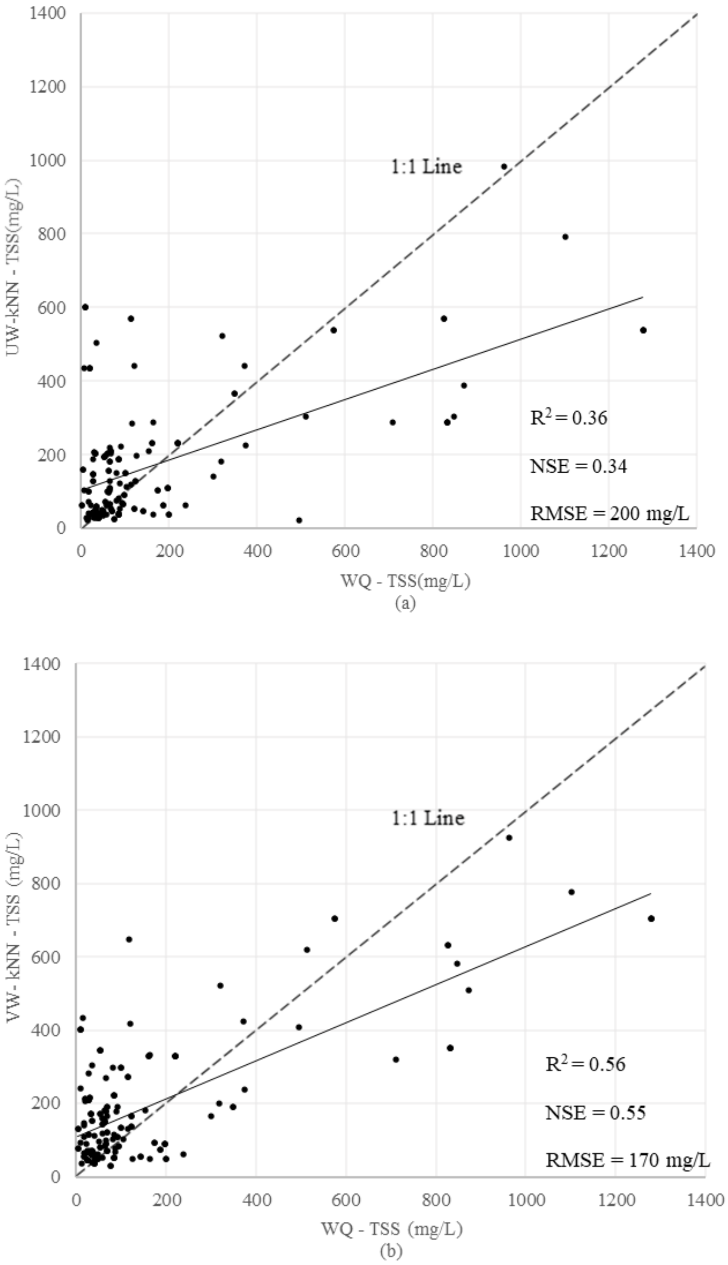

Figure 4. Correlation between the measured values and predicted TSS in the prediction step for (**a**) UW-kNN and (**b**) VW-kNN.

3.2.3. Support Vector Regression (SVR)

The SVR method was one of the methods used in this study. The regression cost value was assigned as 1.4, and the ε value was 0.01. A nonlinear kernel function with a coefficient of 0.63 was used for the mapping process. The scatterplots for training and prediction steps are presented in Figure 5a,b. The model had good performance across all indices in both the training and prediction steps, with R^2 and NSE values of 0.65 and 0.58, respectively. The RMSE values were 140 mg/L and 160 mg/L, respectively, for the training and the prediction steps.

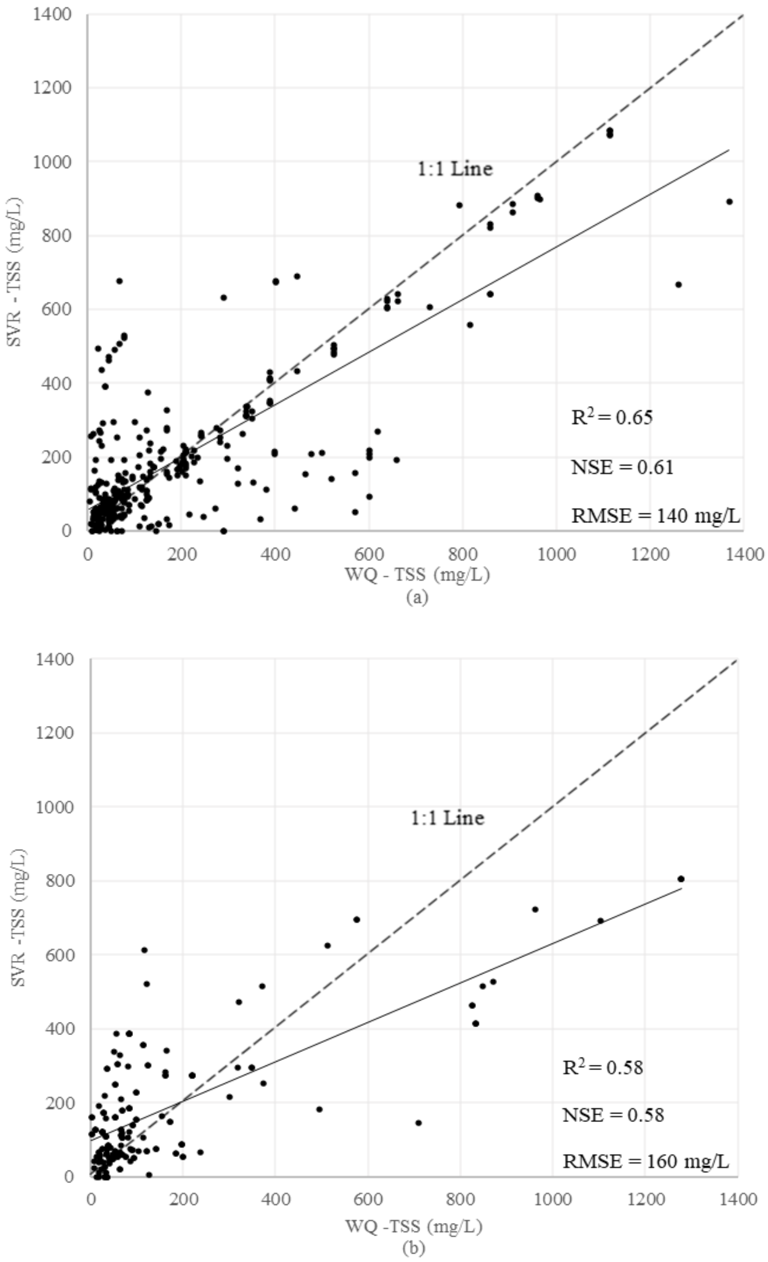

Figure 5. Correlation between the measured values and Support Vector Regression (SVR)-predicted TSS concentration in (**a**) the training step and (**b**) the prediction step.

3.2.4. Artificial Neural Network (ANN)

MLP backpropagation was applied in the ANN, which means that the previous iteration results were used to improve the results of the next iteration. There is no specific method for finding an optimum number of hidden layers and neurons in the hidden layers [67]. Therefore, a trial-and-error method was used to find the optimum number of hidden layers and neurons. We did preliminary testing to determine the number of hidden layers and neurons by varying the number of hidden layers and neurons within a range of 1 to 100 and 1 to 500, respectively. The best result was obtained using 1 hidden layer and 230 neurons. The ReLu function, used as an activation function, produced good results. A quasi-Newton method-based solver with a value of 0.06 for regulation, which penalizes the worst weights in the process to prevent overfitting and underfitting problem, and a maximum number of iterations of 100,000 was used to reach the optimum weighting values. The R^2 value for the training step was considerably better than the value in the prediction step. The scatterplots in Figure 6 show good performance in the training step and poor performance in the prediction step. The performance indices were R^2 values of 0.75 and 0.49, NSE values of 0.73 and 0.49, and RMSE values of 120 mg/L and 180 mg/L, respectively, for the training and the prediction steps.

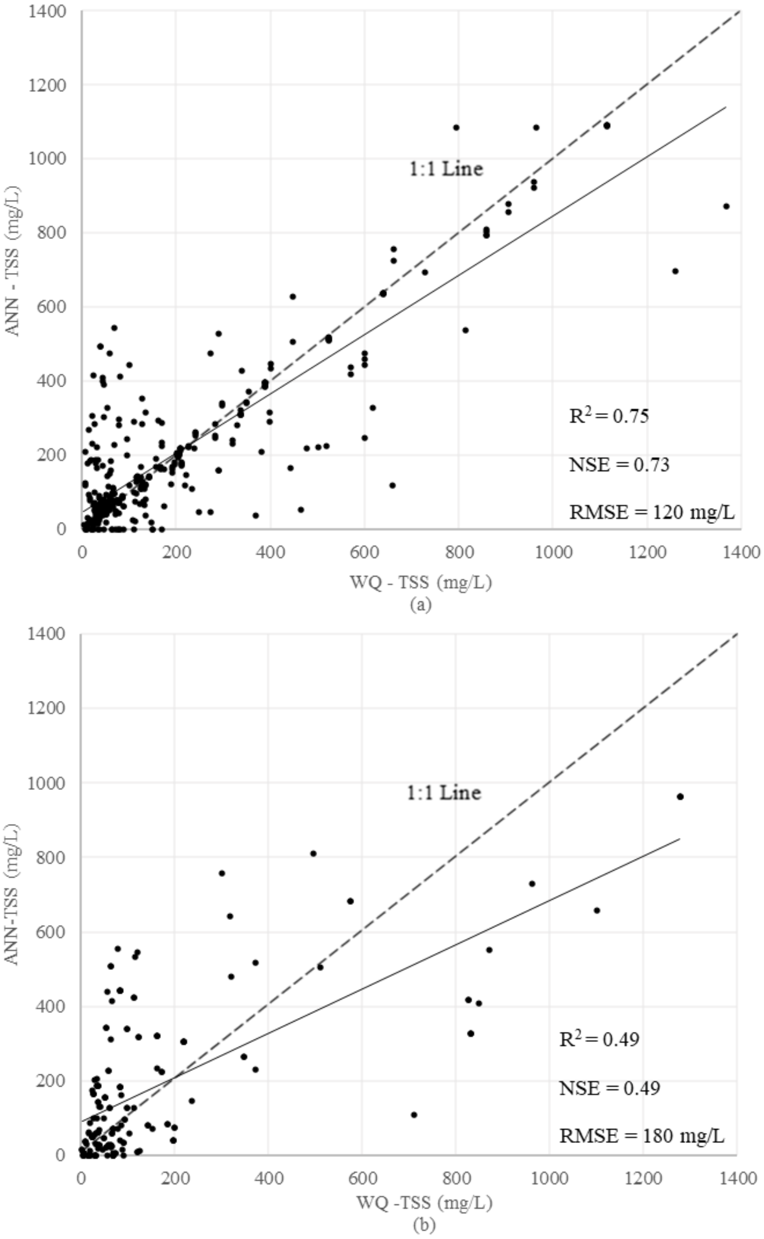

Figure 6. Correlation between the measured values and Artificial Neuron Network (ANN)-predicted TSS concentration in (**a**) the training step and (**b**) the prediction step.

3.2.5. Regression Tree (RT)

Regression Tree (RT) is a widely used ML method due to its simplicity. This study implemented a pruning process to help the RT algorithm avoid overfitting problems. Therefore, a maximum depth value of 10 was assigned for the trees. The RT method predicted TSS concentrations with R^2 values of 0.61 and 0.34, NSE values of 0.53 and 0.30, and RMSE values of 160 mg/L and 210 mg/L for the training and the prediction steps, respectively, as shown in Figure 7a,b. The RT method was unable to make a good generalization. Thus, the method could not be regarded as a good algorithm for the estimation of TSS concentration.

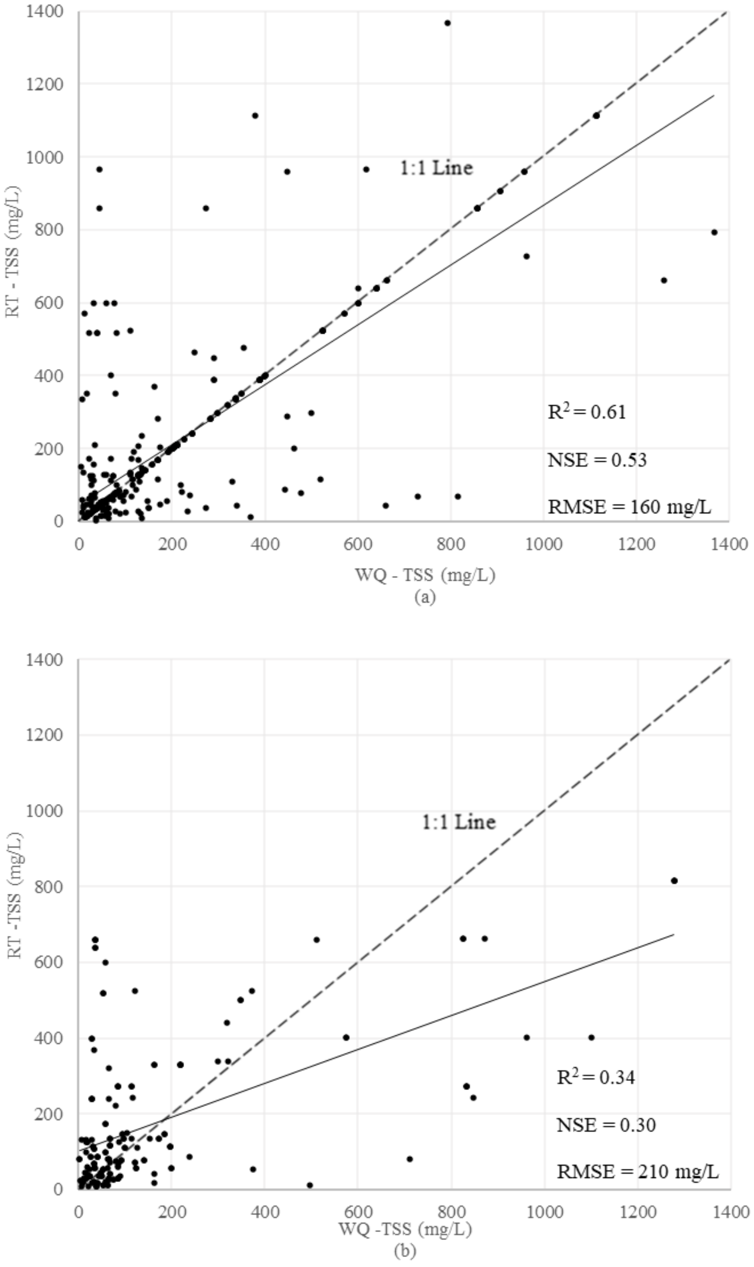

Figure 7. Correlation between the measured values and Regression Tree (RT)-predicted TSS concentration in (**a**) training step and (**b**) prediction step.

3.3. *Ensemble Modeling*

3.3.1. Bagging Model—Random Forest (RF)

An RF model with three features ("m" try) in each iteration and 500 ("n" tree) weak learner RTs was used. Generally, increasing the number of basic learners after 100 did not change the result significantly. However, based on the literature, 500 basic learners would be a reasonable number of single learners within the RF structure [76,81]. We did preliminary testing to determine the number of basic learners using 100, 200, 400, 500, and 1000 basic learners. The best results were obtained for 500 RTs.

The R^2 values indicate that the algorithm can reasonably predict the target value at both higher and lower TSS concentrations, as shown in Figure 8. The R^2, NSE, and RMSE values were 0.77, 0.76, and 110 mg/L, respectively, for the training step (Figure 8a) and 0.67, 0.62, and 150 mg/L, respectively, for the prediction step (Figure 8b).

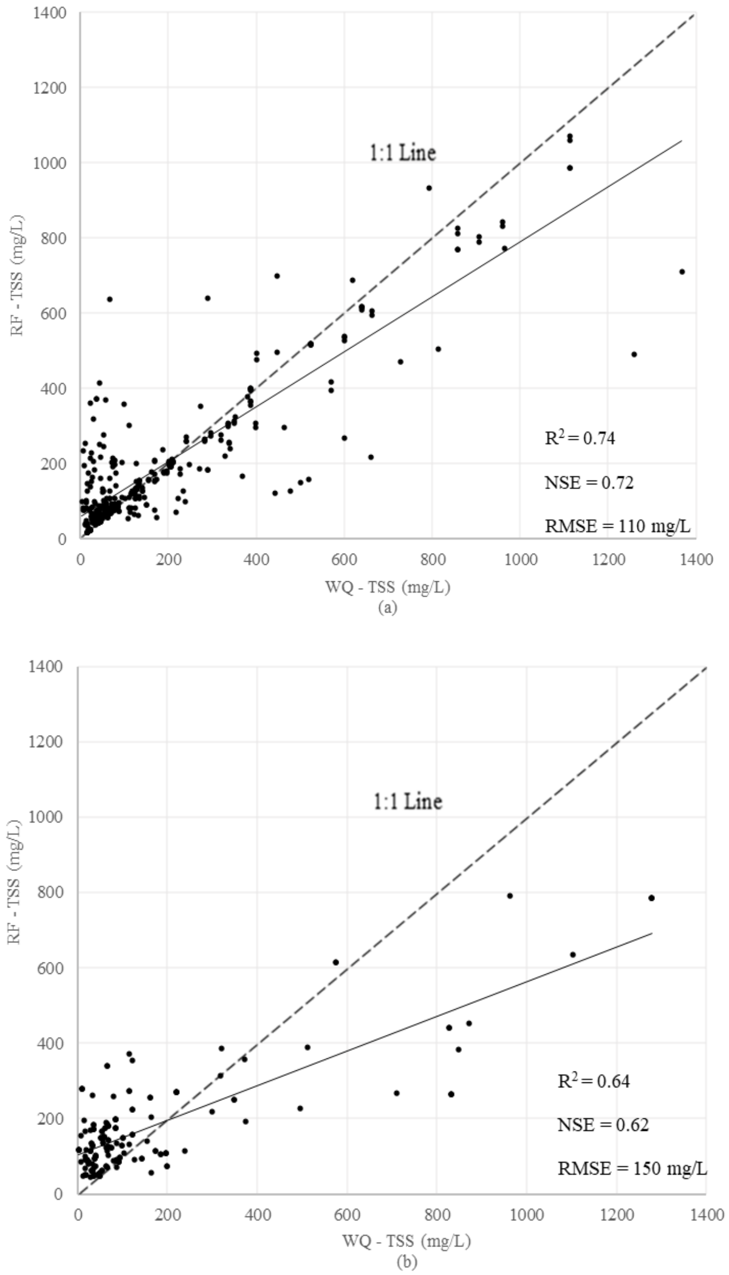

Figure 8. Correlation between the measured values and Random Forest (RF)-predicted TSS concentration in (**a**) training step and (**b**) prediction step.

3.3.2. Boosting Model—Adaptive Boosting (AdB)

This study implemented an AdB algorithm that used 30 RTs as basic learners with a learning rate of 1, which used the results obtained from the last single learner. Additionally, it used an exponential regression loss function to fit the dataset. The model performed well across all indices, with R^2 values of 0.74 and 0.64, NSE values of 0.72 and 0.62, and RMSE values of 110 mg/L and 150 mg/L, respectively, for the training and prediction steps. However, TSS concentration estimates were relatively more accurate in the training step, as shown in Figure 9a, compared to the prediction step, as shown in Figure 9b.

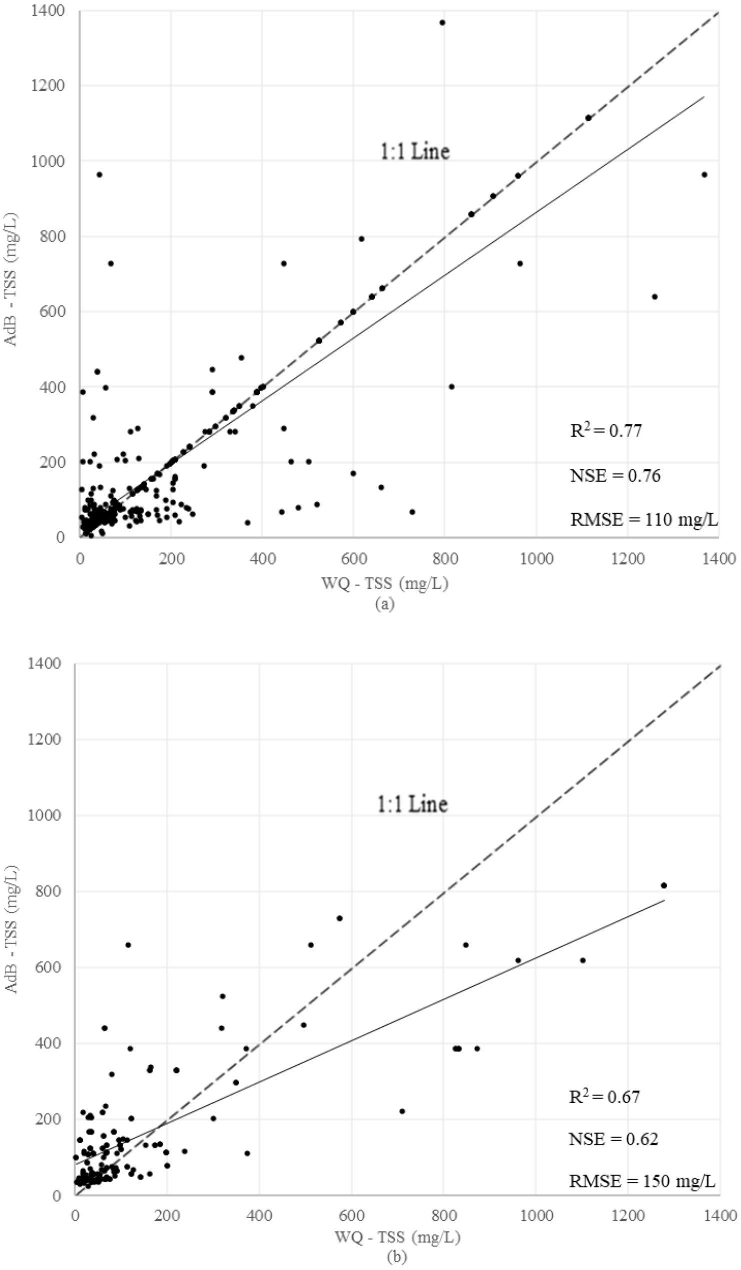

Figure 9. Correlation between the measured values and Adaptive Boosting (AdB)-predicted TSS concentration in (**a**) training step and (**b**) prediction step.

4. Discussion

4.1. Single, Bagging, and Boosting Models

The Multiple Linear Regression model was used for the training and prediction steps to estimate TSS concentration using six different features. Figure 2a shows the efficiency of linear regression in capturing patterns in the measured values data in the training step. The results demonstrate that the Multiple Linear Regression model failed to mimic the patterns in the measured values of TSS concentration. This could be attributed to the underfitting problem that arose from the simplicity and linearity of the algorithm. The MLR model was not able to find the relationship between the features and the target during the training step. Thus, the model could not predict TSS concentration in the prediction step, as shown in Figure 2b. These results indicate that MLR is not a robust model for the estimation of TSS concentration.

Two different kNN methods were used: One with a fewer number of neighbors with uniform weighting and one with a larger number of neighbors with variable weighting based on the distance of each datapoint to its neighbors. Both approaches demonstrated good performance in the training step, as shown in Figure 3a, mimicking the overall patterns, particularly for high TSS concentrations. However, the performance of the VW-kNN method in the lower concentration ranges was slightly better than the UW-kNN. A fewer number of neighbors forced the algorithm to capture the details in the training dataset. Therefore, this process was more likely to fall into an overfitting problem. Moreover, in UW-kNN, the same weight was assigned to the farthest and the nearest neighbor. Nevertheless, there might have been differences between the farthest and nearest neighbor with respect to all the features. As a result, the algorithm could not mimic the general pattern observed in the training dataset. This inability led to a poor fit in the prediction step.

The VW-kNN process resulted in a relatively better fit than the UW-kNN approach in both the training and prediction steps. Thus, using a larger number of neighbors and a variable weighting approach helped the algorithm capture the general pattern observed in the dataset in the training step. The scatterplots in Figure 4 show that the variable weighting method performed better, not only in predicting low concentrations, but also in high TSS concentrations when compared to the uniform weighting method in the prediction step. The uniform and variable weighting kNN methods were compared using other statistical measures (NSE and RMSE).

The SVR algorithm captured the general pattern in both the training and prediction steps. The scatterplots for the training and prediction steps are presented in Figure 5a,b). The model had good performance across all indices in both the training and prediction steps in terms of R^2, NSE, and RMSE values. The SVR algorithm could have better performance in high and mid-range TSS concentration compared to the lower value. Although the SVR had higher R^2 and NSE values in the prediction step compared to the VW-kNN, the VW-kNN had better performance in the training step compared to the SVR. Despite the relatively better performance in prediction steps compared to other models, such as MLR and UW-kNN, the performance in the prediction step was not as good as the training step.

The ANN performed well on the training dataset. However, it could not accurately mimic the pattern in the prediction step, which could have been due to an overfitting problem in the training step resulting from the complexity of the ANN model structure. Overfitting suggests that the ANN model learned and applied details in the training dataset that might have been absent in the prediction dataset. Another factor contributing to the overfitting problem could have been the limited number of datapoints. Having a larger number of datapoints may have reduced the overfitting problem and may have led to better performance in both the training and prediction steps [51].

The next algorithm that was utilized in this study was the RT method. The algorithm captured the general pattern in both the training and prediction steps. However, RT could not capture the pattern for several of the datapoints, especially in the prediction step, which is attributable to an overfitting problem. The RT method can be used inside the structure of ensemble bagging and ensemble boosting algorithm to improve the results obtained in the training and prediction step.

The RF method, which was developed based on the RT, was one of the best-performing algorithms implemented in this study in terms of estimating the TSS concentration. The model was efficient and could capture the patterns in the measured values data for both high and low TSS concentration ranges in the training dataset. In the prediction step, the algorithm captured the pattern but not as accurate as it did in the training step. The algorithm performed slightly better in lower and mid-range TSS concentration compared to higher concentration values in both the training and prediction steps. The reason could have been the relatively larger number of events in the lower and mid-range concentrations. Furthermore, the results show that RF was more efficient than RT, which was used as a basic learner in RF. These results demonstrate that ensemble modeling improved the performance of the RT algorithm compared to the application of the RT algorithm as a single learner.

As stated earlier, AdB can improve the results of single learners that failed to capture patterns in the data due to an overfitting problem. The model was able to recognize the patterns in both the training and prediction datasets. The scatterplots for the training and prediction steps are presented in Figure 9a,b. The scatterplots demonstrate that the AdB method predicted TSS concentrations, with relatively higher R^2 and NSE values compared to the single learners presented earlier. The algorithm did not perform with the same level of accuracy across all the ranges of TSS concentration. The algorithm was able to capture patterns in the TSS concentration for the prediction step for lower, mid-level, and high concentrations of TSS.

All the applied methods in this study were compared by considering the relative R^2 and relative RMSE values in the prediction step, which demonstrated the benefits of using ensemble modeling. The R^2 values were normalized by dividing all the R^2 values by the maximum R^2 value to obtain values between "0" and "1." A value of "1" for a model refers to the highest level of accuracy. A value near "0" represents the lower level of accuracy of (Figure 10a). The RMSE values were normalized by dividing the minimum RMSE value by all the RMSE values to obtain values between "0" and "1." A value of "1" for a model refers to the highest level of accuracy. A value near "0" represents the lower level of accuracy (Figure 10b).

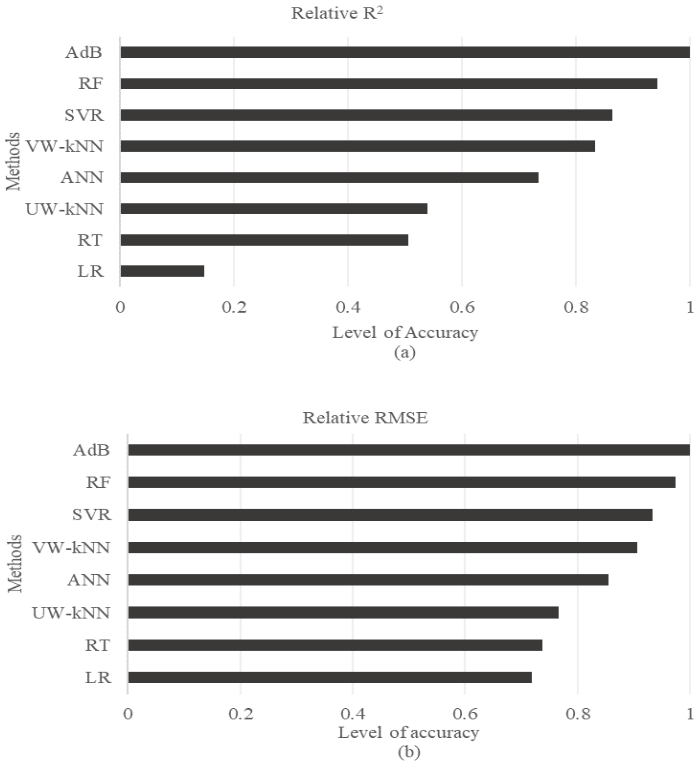

Figure 10. Relative comparison between (**a**) the coefficient of determination (R^2) values (**b**) the Root Mean Square Error (RMSE) values in the prediction step.

The AdB and RF method used multiple RT algorithm as its basic learner. The RT method was unable to perform well when applied individually. However, the boosting and bagging process considerably improved the performance compared to the performance under the individual application. The ensemble structure in AdB and RF allowed the establishment of a robust correlation between a feature and target values in the training step, which led to limiting overfitting and underfitting problems and better results in the prediction step. However, the AdB method performed better than RF, attributable to the boosting process that helped to improve the limitation of each basic learner and subsequent basic learners.

4.2. Sensitivity Analysis

Six different features were considered for the application of supervised ML algorithms to reproduce the pattern in TSS concentration observations retrieved from the NSQD. Further investigation was conducted to assess the relative importance of each feature on the performance of the ML model. Thus, a sensitivity analysis was implemented using one of the best-performing methods, the AdB method. We developed a methodology where we excluded one feature at a time during the training step and assessed how the feature affected the performance of the model. Figure 11 shows the sensitivity results for the six features used in the AdB algorithm. The significance of each feature was assessed by comparing the difference in R^2 values with and without the feature. Then, the data were normalized by dividing the differences by the maximum difference in R^2 values to obtain values between "0" and "1," with a value of "1" for a feature indicating the highest level of sensitivity. A value near "0" value represents the lower level of sensitivity of the output to a feature.

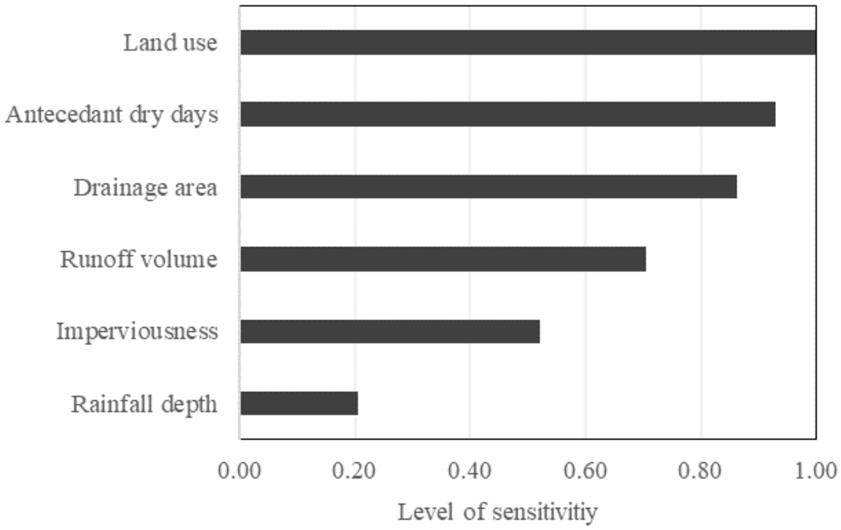

Figure 11. Sensitivity analysis results of AdB.

The AdB outcomes were sensitive to several of the features. However, there were differences in the level of sensitivity. The models showed higher sensitivity to land use data, followed by an antecedent number of dry days and drainage areas. The AdB outcomes were sensitive to both rainfall depth and runoff volume, with a sensitivity higher to runoff volume than rainfall depth. Runoff volume was more relevant to TSS concentration than rainfall depth, likely because runoff is directly linked to detachment and transport of sediment [89]. Thus, when available, runoff volume data could be a better predictor of TSS concentration that rainfall data. The results showed that AdB outcomes were sensitive to both land use and imperviousness, with a higher sensitivity to land use. Land use is relatively more sensitive because of the greater potential for the generation of relatively more sediment load from pervious areas than for wash-off from the impervious areas [89]. Percent imperviousness increases runoff volume, which is responsible for detachment, wash-off, and transport. The AdB also showed that the results obtained by this model were sensitive to antecedent dry days. The amount of solids deposited over the impervious area increases with an increasing number of dry days [89]. The drainage area is also an important factor because it is directly related to the runoff volume and peak. The sensitivity analysis demonstrated all of the factors used in our analysis were important to the prediction of TSS.

5. Conclusions

This study investigated the application of supervised machine learning algorithms for urban stormwater quality management. Version 4.02 of the National Stormwater Quality Database (NSQD) was used to extract event-based data on TSS concentration, and associated site features such as drainage area, land use, percent of impervious, antecedent dry days, rainfall depth, and runoff volume. We compiled 530 datapoints from NSQD containing all the features and target values with no missing data. We used 66% of the dataset as input for the training step and 34% for the prediction step.

Eight different ML algorithms were compared: Linear Regression, Regression Tree, Support Vector Regression, Random Forest, Adaptive Boosting, variable weighting k-Nearest Neighbor, uniform weighting k-Nearest Neighbor, and Artificial Neural Network. Linear Regression failed in both the training and prediction steps. However, all the other methods showed a good performance in both the training and prediction steps except the uniform weighting k-Nearest Neighbor and Artificial Neural Network algorithms. These two methods (UW-kNN and ANN) had a good performance in the training step but failed in the prediction step. The highest R^2 and NSE and the lowest RMSE in both the training and prediction steps, indicators of good performance, were obtained by the RF and AdB algorithms. Moreover, a sensitivity analysis demonstrated that the prediction accuracy of AdB was sensitive to all input features.

It was demonstrated that machine learning methods are plausible approaches to the prediction of TSS concentration. A limitation identified in some of the models, poor performance in the prediction step, is attributed to overfitting and underfitting problems. Thus, these limitations can be addressed with the choice of appropriate models and the use of sufficient data points. The approach could benefit from the expansion of the NSQD dataset. Thus, with further enhancement of the ML methods and data sources, ML methods have the potential for application across regions. Future efforts to enhance model accuracy should consider the use of hybrid methods or ensemble models.

Author Contributions: Conceptualization, M.M., M.G., A.S., methodology, M.M., A.S., M.G.; investigation, M.M., Writing—original draft preparation, M.M.; writing—review and editing, A.S., M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the South Dakota Board of Regents Competitive Research Grant (CRG) and Civil and Environmental Engineering, South Dakota School of Mines and Technology.

Institutional Review Board Statement: This study does not involve humans, Ethical review not applicable for this study.

Informed Consent Statement: There is no consent required for this study.

Data Availability Statement: The National Stormwater Quality Database used in this study is freely available data source. The data is available at https://www.bmpdatabase.org/nsqd.html.

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

References

1. Danades, A.; Pratama, D.; Anggraini, D.; Anggriani, D. Comparison of accuracy level K-nearest neighbor algorithm and support vector machine algorithm in classification water quality status. In Proceedings of the 2016 6th International Conference on System Engineering and Technology (ICSET), Bandung, Indonesia, 3–4 October 2016; pp. 137–141.
2. Bedient, P.B.; Lambert, J.L.; Springer, N.K. Stormwater pollutant load-runoff relationships. *J. Water Pollut. Control Fed.* **1980**, *52*, 2396–2404.
3. Jeung, M.; Baek, S.; Beom, J.; Cho, K.H.; Her, Y.; Yoon, K. Evaluation of random forest and regression tree methods for estimation of mass first flush ratio in urban catchments. *J. Hydrol.* **2019**, *575*, 1099–1110. [CrossRef]
4. Patil, S.S.; Barfield, B.J.; Wilber, G.G. Turbidity modeling based on the concentration of total suspended solids for stormwater runoff from construction and development sites. In Proceedings of the World Environmental and Water Resources Congress 2011, Palm Springs, CA, USA, 22–26 May 2011; pp. 477–486.
5. Peters, J.; De Baets, B.; Verhoest, N.E.C.; Samson, R.; Degroeve, S.; De Becker, P.; Huybrechts, W. Random forests as a tool for ecohydrological distribution modelling. *Ecol. Model.* **2007**, *207*, 304–318. [CrossRef]

6. Young, B.N.; Hathaway, J.M.; Lisenbee, W.A.; He, Q. Assessing the runoff reduction potential of highway swales and WinSLAMM as a predictive tool. *Sustainability* **2018**, *10*, 2871. [CrossRef]
7. Pitt, R. WinSLAMM Instruction. 2013. Available online: http://www.winslamm.com/docs/WinSLAMM%20Model%20Algorithms%20v7.pdf (accessed on 10 December 2020).
8. Bachhuber, J.A.; Mattfield, K. Quantifying Urban Stormwater Pollutant Loads and Management Costs Within the Lower Fox River Basin. *Proc. Water Environ. Fed.* **2009**, *2009*, 600–605. [CrossRef]
9. Pitt, R. Calibration of WinSLAMM. Available online: http://winslamm.com/docs/WinSLAMM%20calibration%20Sept (accessed on 9 February 2020).
10. Rossman, L.A.; Dickinson, R.E.; Schade, T.; Chan, C.C.; Burgess, E.; Sullivan, D.; Lai, F.-H. SWMM 5-the Next Generation of EPA's Storm Water Management Model. *J. Water Manag. Model.* **2004**, *16*, 339–358. [CrossRef]
11. Zoppou, C. Review of urban storm water models. *Environ. Model. Softw.* **2001**, *16*, 195–231. [CrossRef]
12. Niazi, M.; Nietch, C.; Maghrebi, M.; Jackson, N.; Bennett, B.R.; Tryby, M.; Massoudieh, A. Storm water management model: Performance review and gap analysis. *J. Sustain. Water Built Environ.* **2017**, *3*, 04017002. [CrossRef]
13. Charbeneau, R.J.; Barrett, M.E. Evaluation of methods for estimating stormwater pollutant loads. *Water Environ. Res.* **1998**, *70*, 1295–1302. [CrossRef]
14. Tu, M.-C.; Smith, P. Modeling pollutant buildup and washoff parameters for SWMM based on land use in a semiarid urban watershed. *Water Air Soil Pollut.* **2018**, *229*, 121. [CrossRef]
15. Azari, B.; Tabesh, M. Optimal design of stormwater collection networks considering hydraulic performance and BMPs. *Int. J. Environ. Res.* **2018**, *12*, 585–596. [CrossRef]
16. Moeini, M.; Zahraie, B. Monthly Water Balance Modeling By Linking Hydro-Climatologic And Tank Groundwater Balance Models. *Iran Water Resour. Res.* **2018**, *14*, 59–70.
17. Chang, F.-J.; Guo, S. Advances in hydrologic forecasts and water resources management. *Water* **2020**, *12*, 1819. [CrossRef]
18. Chang, F.-J.; Hsu, K.; Chang, L.-C. *Flood Forecasting Using Machine Learning Methods*; MDPI: Basel, Switzerland, 2019.
19. Hu, J.-H.; Tsai, W.-P.; Cheng, S.-T.; Chang, F.-J. Explore the relationship between fish community and environmental factors by machine learning techniques. *Environ. Res.* **2020**, *184*, 109262. [CrossRef]
20. Kao, I.-F.; Zhou, Y.; Chang, L.-C.; Chang, F.-J. Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting. *J. Hydrol.* **2020**, *583*, 124631. [CrossRef]
21. Kashani, A.R.; Gandomi, M.; Camp, C.V.; Gandomi, A.H. Optimum design of shallow foundation using evolutionary algorithms. *Soft Comput.* **2020**, *24*, 6809–6833. [CrossRef]
22. Liang, J.; Li, W.; Bradford, S.A.; Šimůnek, J. Physics-Informed Data-Driven Models to Predict Surface Runoff Water Quantity and Quality in Agricultural Fields. *Water* **2019**, *11*, 200. [CrossRef]
23. May, D.B.; Sivakumar, M. Prediction of urban stormwater quality using artificial neural networks. *Environ. Model. Softw.* **2009**, *24*, 296–302. [CrossRef]
24. Álvarez-Cabria, M.; Barquín, J.; Peñas, F.J. Modelling the spatial and seasonal variability of water quality for entire river networks: Relationships with natural and anthropogenic factors. *Sci. Total Environ.* **2016**, *545*, 152–162. [CrossRef]
25. Guimarães, T.T.; Veronez, M.R.; Koste, E.C.; Souza, E.M.; Brum, D.; Gonzaga, L.; Mauad, F.F. Evaluation of regression analysis and neural networks to predict total suspended solids in water bodies from unmanned aerial vehicle images. *Sustainability* **2019**, *11*, 2580. [CrossRef]
26. Ahmed, U.; Mumtaz, R.; Anwar, H.; Shah, A.A.; Irfan, R.; García-Nieto, J. Efficient water quality prediction using supervised Machine Learning. *Water* **2019**, *11*, 2210. [CrossRef]
27. Granata, F.; Papirio, S.; Esposito, G.; Gargano, R.; De Marinis, G. Machine learning algorithms for the forecasting of wastewater quality indicators. *Water* **2017**, *9*, 105. [CrossRef]
28. Karamouz, M.; Mojahedi, S.A.; Ahmadi, A. Interbasin water transfer: Economic water quality-based model. *J. Irrig. Drain. Eng.* **2010**, *136*, 90–98. [CrossRef]
29. Maestre, A.; Pitt, R.E.; Williamson, D. Nonparametric statistical tests comparing first flush and composite samples from the national stormwater quality database. *J. Water Manag. Model.* **2004**. [CrossRef]
30. Pitt, R.; Maestre, A.; Morquecho, R. The National Stormwater Quality Database (NSQD, Version 1.1). In Proceedings of the 1st Annual Stormwater Management Research Symposium Proceedings, Orlando, FL, USA, 16 February 2004; pp. 13–51.
31. Maestre, A.; Pitt, R.E. Identification of significant factors affecting stormwater quality using the national stormwater quality database. *J. Water Manag. Model.* **2006**, *13*, 287–326. [CrossRef]
32. Aiken, L.S.; West, S.G.; Pitts, S.C.; Baraldi, A.N.; Wurpts, I.C. Multiple linear regression. In *Handbook of Psychology*, 2nd ed.; Wiley Online Library: Hoboken, NJ, USA, 2012; Volume 2.
33. McCarthy, D.T.; Hathaway, J.M.; Hunt, W.F.; Deletic, A. Intra-event variability of Escherichia coli and total suspended solids in urban stormwater runoff. *Water Res.* **2012**, *46*, 6661–6670. [CrossRef] [PubMed]
34. Azizi, K.; Attari, J.; Moridi, A. Estimation of discharge coefficient and optimization of Piano Key Weirs. In *Labyrinth and Piano Key Weirs III–PKW 2017*; CRC Press: Boca Raton, FL, USA, 2017; p. 213.
35. Barnes, K.B.; Morgan, J.; Roberge, M. *Impervious Surfaces and the Quality of Natural and Built Environments*; Department of Geography and Environmental Planning, Towson University: Towson, MD, USA, 2001.

36. Brodie, I.M.; Dunn, P.K. Suspended particle characteristics in storm runoff from urban impervious surfaces in Toowoomba, Australia. *Urban Water J.* **2009**, *6*, 137–146. [CrossRef]
37. Uygun, B.Ş.; Albek, M. Determination effects of impervious areas on urban watershed. *Environ. Sci. Pollut. Res.* **2015**, *22*, 2272–2286. [CrossRef]
38. Pizarro, J.; Vergara, P.M.; Morales, J.L.; Rodríguez, J.A.; Vila, I. Influence of land use and climate on the load of suspended solids in catchments of Andean rivers. *Environ. Monit. Assess.* **2014**, *186*, 835–843. [CrossRef]
39. Gong, Y.; Liang, X.; Li, X.; Li, J.; Fang, X.; Song, R. Influence of rainfall characteristics on total suspended solids in urban runoff: A case study in Beijing, China. *Water* **2016**, *8*, 278. [CrossRef]
40. King, J.K.; Blanton, J.O. Model for predicting effects of land-use changes on the canal-mediated discharge of total suspended solids into tidal creeks and estuaries. *J. Environ. Eng.* **2011**, *137*, 920–927. [CrossRef]
41. Shakya, S.; Tamaddun, K.A.; Stephen, H.; Ahmad, S. Urban Runoff and Pollutant Reduction by Retrofitting Green Infrastructure in Storm Water Management System. In Proceedings of the World Environmental and Water Resources Congress 2011, Palm Springs, CA, USA, 22–26 May 2019; pp. 93–104.
42. Karathanasis, A.D.; Potter, C.L.; Coyne, M.S. Vegetation effects on fecal bacteria, BOD, and suspended solid removal in constructed wetlands treating domestic wastewater. *Ecol. Eng.* **2003**, *20*, 157–169. [CrossRef]
43. Al Hasan, M.; Chaoji, V.; Salem, S.; Zaki, M. Link prediction using supervised learning. In Proceedings of the SDM06: Workshop on Link Analysis, Counter-Terrorism and Security, Bethesda, MD, USA, 19 April 2006; pp. 798–805.
44. Hardt, M.; Price, E.; Srebro, N. Equality of opportunity in supervised learning. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 3315–3323.
45. Springenberg, J.T. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv* **2015**, arXiv:1511.06390.
46. Tan, M.; Le, Q.V. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv* **2019**, arXiv:1905.11946.
47. Solomatine, D.P.; Maskey, M.; Shrestha, D.L. Eager and lazy learning methods in the context of hydrologic forecasting. In Proceedings of the 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada, 16–21 July 2006; pp. 4847–4853.
48. Wei, C.-C. Comparing lazy and eager learning models for water level forecasting in river-reservoir basins of inundation regions. *Environ. Model. Softw.* **2015**, *63*, 137–155. [CrossRef]
49. García-Callejas, D.; Araújo, M.B. The effects of model and data complexity on predictions from species distributions models. *Ecol. Model.* **2016**, *326*, 4–12. [CrossRef]
50. Yao, Y.; Xiao, Z.; Wang, B.; Viswanath, B.; Zheng, H.; Zhao, B.Y. Complexity vs. performance: Empirical analysis of machine learning as a service. In Proceedings of the 17th ACM SIGCOMM Internet Measurement Conference (IMC 2017), London, UK, 1–3 November 2017; pp. 384–397.
51. Li, P.; Zha, Y.; Shi, L.; Tso, C.-H.M.; Zhang, Y.; Zeng, W. Comparison of the use of a physical-based model with data assimilation and machine learning methods for simulating soil water dynamics. *J. Hydrol.* **2020**, *584*, 124692. [CrossRef]
52. Montgomery, D.C.; Peck, E.A.; Vining, G.G. *Introduction to Linear Regression Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2012; Volume 821.
53. Marill, K.A. Advanced statistics: Linear regression, part II: Multiple linear regression. *Acad. Emerg. Med.* **2004**, *11*, 94–102. [CrossRef]
54. Almeida, A.M.d.; Castel-Branco, M.M.; Falcao, A.C. Linear regression for calibration lines revisited: Weighting schemes for bioanalytical methods. *J. Chromatogr. B* **2002**, *774*, 215–222. [CrossRef]
55. Shojaeizadeh, A.; Geza, M.; McCray, J.; Hogue, T.S. Site-scale integrated decision support tool (i-DSTss) for stormwater management. *Water* **2019**, *11*, 2022. [CrossRef]
56. Liang, J.; Yang, Q.; Sun, T.; Martin, J.D.; Sun, H.; Li, L. MIKE 11 model-based water quality model as a tool for the evaluation of water quality management plans. *J. Water Supply Res. Technol. AQUA* **2015**, *64*, 708–718. [CrossRef]
57. Kohli, S.; Godwin, G.T.; Urolagin, S. Sales Prediction Using Linear and KNN Regression. In *Advances in Machine Learning and Computational Intelligence*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 321–329.
58. Saini, I.; Singh, D.; Khosla, A. QRS detection using K-Nearest Neighbor algorithm (KNN) and evaluation on standard ECG databases. *J. Adv. Res.* **2013**, *4*, 331–344. [CrossRef] [PubMed]
59. Khan, M.M.R.; Arif, R.B.; Siddique, M.A.B.; Oishe, M.R. Study and observation of the variation of accuracies of KNN, SVM, LMNN, ENN algorithms on eleven different datasets from UCI machine learning repository. In Proceedings of the 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), Dhaka, Bangladesh, 13–15 September 2018; pp. 124–129.
60. Wang, X.; Ma, L.; Wang, X. Apply semi-supervised support vector regression for remote sensing water quality retrieving. In Proceedings of the 2010 IEEE International Geoscience and Remote Sensing Symposium, Honolulu, HI, USA, 25–30 July 2010; pp. 2757–2760.
61. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
62. Liu, S.; Tai, H.; Ding, Q.; Li, D.; Xu, L.; Wei, Y. A hybrid approach of support vector regression with genetic algorithm optimization for aquaculture water quality prediction. *Math. Comput. Model.* **2013**, *58*, 458–465. [CrossRef]

63. Najah, A.; El-Shafie, A.; Karim, O.A.; El-Shafie, A.H. Application of artificial neural networks for water quality prediction. *Neural Comput. Appl.* **2013**, *22*, 187–201. [CrossRef]
64. Singh, K.P.; Basant, A.; Malik, A.; Jain, G. Artificial neural network modeling of the river water quality—A case study. *Ecol. Model.* **2009**, *220*, 888–895. [CrossRef]
65. Fotovvati, B.; Balasubramanian, M.; Asadi, E. Modeling and Optimization Approaches of Laser-Based Powder-Bed Fusion Process for Ti-6Al-4V Alloy. *Coatings* **2020**, *10*, 1104. [CrossRef]
66. Graupe, D. *Principles of Artificial Neural Networks*; World Scientific: Singapore, 2013; Volume 7.
67. Yegnanarayana, B. *Artificial Neural Networks*; PHI Learning Pvt. Ltd.: New Delhi, India, 2009.
68. Rizwan, J.M.; Krishnan, P.N.; Karthikeyan, R.; Kumar, S.R. Multi layer perception type artificial neural network based traffic control. *Indian J. Sci. Technol.* **2016**, *9*, 1–6. [CrossRef]
69. Boughrara, H.; Chtourou, M.; Amar, C.B.; Chen, L. Facial expression recognition based on a mlp neural network using constructive training algorithm. *Multimed. Tools Appl.* **2016**, *75*, 709–731. [CrossRef]
70. Sutton, C.D. Classification and regression trees, bagging, and boosting. *Handb. Stat.* **2005**, *24*, 303–329.
71. Ließ, M.; Glaser, B.; Huwe, B. Uncertainty in the spatial prediction of soil texture: Comparison of regression tree and Random Forest models. *Geoderma* **2012**, *170*, 70–79. [CrossRef]
72. Hasanipanah, M.; Faradonbeh, R.S.; Amnieh, H.B.; Armaghani, D.J.; Monjezi, M. Forecasting blast-induced ground vibration developing a CART model. *Eng. Comput.* **2017**, *33*, 307–316. [CrossRef]
73. Golecha, Y.S. Analyzing Term Deposits in Banking Sector by Performing Predictive Analysis Using Multiple Machine Learning Techniques. Masters Thesis, National College of Ireland, Dublin, Ireland, 2017.
74. Rajadurai, H.; Gandhi, U.D. A stacked ensemble learning model for intrusion detection in wireless network. *Neural Comput. Appl.* **2020**. [CrossRef]
75. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
76. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
77. Wu, D.; Wang, H.; Seidu, R. Smart data driven quality prediction for urban water source management. *Future Gener. Comput. Syst.* **2020**, *107*, 418–432. [CrossRef]
78. Liaw, A.; Wiener, M. Classification and regression by randomForest. *R news* **2002**, *2*, 18–22.
79. Ok, A.O.; Akar, O.; Gungor, O. Evaluation of random forest method for agricultural crop classification. *Eur. J. Remote Sens.* **2012**, *45*, 421–432. [CrossRef]
80. Adam, E.M.; Mutanga, O.; Rugege, D.; Ismail, R. Discriminating the papyrus vegetation (Cyperus papyrus L.) and its co-existent species using random forest and hyperspectral data resampled to HYMAP. *Int. J. Remote Sens.* **2012**, *33*, 552–569. [CrossRef]
81. Smith, R.G.; Majumdar, S. Groundwater Storage Loss Associated With Land Subsidence in Western United States Mapped Using Machine Learning. *Water Resour. Res.* **2020**, *56*, e2019WR026621. [CrossRef]
82. Al-Stouhi, S.; Reddy, C.K. *Adaptive Boosting for Transfer Learning using Dynamic Updates*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 60–75.
83. Freund, Y.; Schapire, R.; Abe, N. A short introduction to boosting. *J. Jpn. Soc. Artif. Intell.* **1999**, *14*, 1612.
84. Duan, S.; Yang, W.; Wang, X.; Mao, S.; Zhang, Y. Forecasting of grain pile temperature from meteorological factors using machine learning. *IEEE Access* **2019**, *7*, 130721–130733. [CrossRef]
85. Mousavi, S.S.; Firouzmand, M.; Charmi, M.; Hemmati, M.; Moghadam, M.; Ghorbani, Y. Blood pressure estimation from appropriate and inappropriate PPG signals using A whole-based method. *Biomed. Signal Process. Control* **2019**, *47*, 196–206. [CrossRef]
86. Rojas, R. *AdaBoost and the Super Bowl of Classifiers a Tutorial Introduction to Adaptive Boosting*; Freie University: Berlin, Germany, 2009.
87. Nash, J.E.; Sutcliffe, J.V. River flow forecasting through conceptual models part I—A discussion of principles. *J. Hydrol.* **1970**, *10*, 282–290. [CrossRef]
88. Willmott, C.J. Some comments on the evaluation of model performance. *Bull. Am. Meteorol. Soc.* **1982**, *63*, 1309–1313. [CrossRef]
89. Yuan, Q.; Guerra, H.B.; Kim, Y. An investigation of the relationships between rainfall conditions and pollutant wash-off from the paved road. *Water* **2017**, *9*, 232. [CrossRef]