

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/351295005>

Application of the Machine Learning LightGBM Model to the Prediction of the Water Levels of the Lower Columbia River

Article in Journal of Marine Science and Engineering · May 2021

DOI: 10.3390/jmse9050496

CITATION

1

READS

218

6 authors, including:



Min Gan

Hohai University

11 PUBLICATIONS 19 CITATIONS

[SEE PROFILE](#)



Shunqi Pan

Cardiff University

134 PUBLICATIONS 1,519 CITATIONS

[SEE PROFILE](#)



Yong ping Chen

Hohai University

95 PUBLICATIONS 420 CITATIONS

[SEE PROFILE](#)



Chen Cheng

Hohai University

9 PUBLICATIONS 14 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Project

A non-stationary tidal analysis toolbox S_TIDE [View project](#)



Project

PhD thesis: Process modelling of the impact of beach nourishment on coastal morphology [View project](#)

Article

Application of the Machine Learning LightGBM Model to the Prediction of the Water Levels of the Lower Columbia River

Min Gan ^{1,2,3}, Shunqi Pan ³ , Yongping Chen ^{1,2,*}, Chen Cheng ², Haidong Pan ^{4,5}  and Xian Zhu ^{1,2}

¹ State Key Laboratory of Hydrology-Water Resources & Hydraulic Engineering, Nanjing 210098, China; GanM@cardiff.ac.uk (M.G.); zhuxian@hhu.edu.cn (X.Z.)

² College of Harbor, Coastal, and Offshore Engineering, Hohai University, Nanjing 210098, China; gordanchan@hhu.edu.cn

³ Hydro-Environmental Research Centre, School of Engineering, Cardiff University, Cardiff CF243AA, UK; PanS2@cardiff.ac.uk

⁴ Key Laboratory of Physical Oceanography, Qingdao Collaborative Innovation Center of Marine Science and Technology, Ocean University of China, Qingdao 266003, China; panhaidong@stu.ouc.edu.cn

⁵ Qingdao National Laboratory for Marine Science and Technology, Qingdao 266100, China

* Correspondence: ypchen@hhu.edu.cn



Citation: Gan, M.; Pan, S.; Chen, Y.; Cheng, C.; Pan, H.; Zhu, X. Application of the Machine Learning LightGBM Model to the Prediction of the Water Levels of the Lower Columbia River. *J. Mar. Sci. Eng.* **2021**, *9*, 496. <https://doi.org/10.3390/jmse9050496>

Academic Editor: Georgios Sylaios

Received: 22 March 2021

Accepted: 26 April 2021

Published: 3 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: water level prediction; estuarine tides; lower Columbia River; machine learning; LightGBM; NS_TIDE

1. Introduction

In recent decades, estuaries have been intensively developed and used for a wide range of economic activities [1]. In estuaries, water temperature, salinity, turbidity and sediment transport change daily in response to the tides [2]. Accurately predicting the water levels in estuaries is important, in particular under extreme conditions, to allow sufficient time for the authorities to issue a forewarning of an impending flood and to implement early evacuation measures [3].

Ocean and coastal tides are usually predictable as they have periodic signals. Their tide properties (amplitudes and phase) can be easily obtained from tide levels by using classical harmonic analysis methods such as the T_TIDE model [4], or extracted from tide prediction models such as the Oregon State University Tidal Inversion Software [5]. However, when tides propagate toward the nearshore zone, shallow water effects can significantly affect tide properties, as shallow water tide constituents can be generated during the on-shore propagation of ocean tides [6]. Moreover, tide properties can be further altered when tides enter into an estuary due to the effects caused by the river discharge from upstream. In addition, changes of estuarine width can cause energy variation of tides,

while the estuarine water depth can change the propagation speed of tide waves. Due to the complex estuarine environment, estuarine water level variations contain strong nontide components, making the analysis of estuarine tides more challenging [7].

In the past decades, substantial research has been focused on the nonlinear interaction between the river discharge and tides, such as the works of Jay [8] and Godin [9]. Especially in recent years, advanced tools aimed at improving the predictions of estuarine tides have been developed. Matte et al. [7] developed the nonstationary tidal harmonic analysis model (NS_TIDE) to consider the time-dependent tide properties (amplitudes and phases) caused by the nonlinear interaction between tides and river discharge. Similarly, Pan et al. [10] proposed the enhanced harmonic analysis model (S_TIDE), which calculates time-dependent tide properties by using the interpolation method without the input of river discharge. These tools achieved a significant higher analytic accuracy than the classical harmonic analysis model. Therefore, they have been widely used in the analysis of estuarine tides, or even modified to make them more suitable for local estuary environments [11–16]. These studies provided a good generalization about the nonstationary nature of estuarine tides from the view of physical processes.

At present, mathematical descriptions of the nonlinear interaction between tides and river discharge in estuaries commonly adopt a linear simplification in predicting the tides, or predict them numerically using hydrodynamic models. The linear simplification or numerical solutions, which rely on accurate topography data and boundary conditions, can limit the prediction accuracy. As an alternative, data-driven models provide a good supplement to physical process-based models. Data-driven models require no specific mathematical assumptions about tide levels and are more suitable to simulate the nonlinear or unknown relationships between the inputs and outputs.

A number of previous studies have achieved the success of accurately predicting tide levels both in coastal and estuary zones through data-driven methods [17–20]. Some studies also showed that data-driven models, such as the artificial neural network (ANN) method, could achieve better performance than classical harmonic analysis when water level variation contains nontide components [21–23]. However, to the best of our knowledge, previous application of data-driven models to tide level predictions mainly focused on coastal tides, whilst their applications to estuarine tides were mainly used for short-term prediction.

Supharatid [20] constructed a prediction model at the River Chao Phraya in the Gulf of Thailand by using the ANN method to predict the mean monthly estuarine tide level. Chang and Chen [24] used the radial basis function neural network to predict an-hour-ahead water levels in the Tanshui River during the typhoon periods. Similar works were done by Tsai et al. [25] in predicting the water levels in 1–3 hourly intervals in the Tanshui River. Chen et al. [26] compared the model performance of the ANN model with 2-D and 3-D hydrodynamic models in predicting the water levels of the Danshui River estuary in northern Taiwan. Their results showed that the ANN model achieved similar accuracy as 2-D and 3-D hydrodynamic models, or even better performance during the extremely high flow periods. More recently, Yoo et al. [27] applied the Long Short-Term Memory method to predict the water level of Hangang River, South Korea. Their results showed that the model prediction accuracy becomes unacceptable when the prediction length is longer than 3 h. Chen et al. [28] integrated the NS_TIDE model [7] with an autoregressive model to improve the short-term (within 48 h) prediction of the water levels in the Yangtze estuary.

With the growing research interests in artificial intelligence and the concept of big data, more data-driven methods, especially machine learning and deep learning methods, have been developed [29]. These methods have been widely used in a large number of areas such as economics [30], geoscience [23,31,32], and medicine [33]. Riazi [23] also successfully applied the deep learning approach to accurately predict the coastal tide level, which clearly indicated the applicability of machine learning methods in analyzing the estuarine tides.

In 2017, Microsoft® proposed a fast, distributed, high performance gradient boosting framework: LightGBM [34]. Consequently, the LightGBM framework, which belongs to decision tree algorithms, has been frequently used in solving the problems of classification [35] and regression [33,36]. Considering the effective performance of the LightGBM framework in regression, this study aimed to develop the LightGBM model for predicting estuarine water levels. Appropriate selection of input parameters and the optimization of the hyperparameters are fully described and carried out. The lower reach of the Columbia River was selected as the research area because the long-term hydrographic field data are publicly available. The results from the LightGBM model were further compared with those from the commonly-used NS_TIDE model [7]. The spatially varied contribution of the parameters in the input layer of the LightGBM model in the lower Columbia River was also investigated in this study.

The remaining part of this paper is organized as follows. The algorithm and the modelling process of the LightGBM model are described in Section 2, while the information of the research area and the measurements are given in Section 3. Model training and testing results are presented in Sections 4 and 5, respectively. A detailed discussion is given in Section 6, while the main conclusions are presented in Section 7.

2. Model Description

2.1. The Algorithm of the LightGBM Model

The LightGBM model is an open-source framework originally proposed by Microsoft® [34]. It is a decision-tree-based algorithm, which divides the parameters in the input layer into different parts and thereby constructs the mapping relationship between the inputs and outputs. Figure 1A shows the main feature of the LightGBM model, which uses leaf-wise-tree growth instead of the widely used level-wise-tree growth to speed up the training. In terms of the level-wise-tree growth strategy, the tree structure grows level by level. As shown in the figure, assuming that the tree depth is D , the number of the nodes at the final level is 2^D . Compared with the level-wise-tree growth strategy, the leaf-wise-tree growth strategy grows tree based on the node where the largest error reduction can be obtained. In other words, the leaf-wise-tree growth strategy is greedier than the level-wise-tree growth strategy. The tree nodes of the leaf-wise-tree algorithm are usually less than the tree nodes of the level-wise-tree algorithm with the same tree depth. A typical example is shown in Figure 1B, illustrating the leaf-wise-tree growth for breast cancer patients through the input parameters of gender and age groups. In comparison with the level-wise-tree growth algorithm, the leaf-wise tree growth algorithm can considerably reduce the number of tree nodes, so that the training process can be significantly speeded up when the dataset is large. However, when the dataset is small, the leaf-wise tree growth algorithm tends to over-fit because of its greedier algorithm, while the level-wise tree growth algorithm is relatively more stable.

More specifically, the LightGBM model uses the Gradient Boosting Decision Tree (GBDT) algorithm to reach a more accurate prediction. As conceptualized in Figure 1C, assuming X and Y are input and output (targets) for regression, the target of the 1st tree (Tree 1 in Figure 1C) is Y . If the estimation of the 1st tree is T_1 , the input and output (targets) of the 2nd tree (Tree 2 in Figure 1C) become X and $Y - T_1$. More specifically, the target (output) of each tree comes from the tree model's errors of the previous tree, while the input is kept the same. The GBDT algorithm generates the final predictions by using the ensemble predictions of a series of trees. With the construction of more and more trees, the ensemble estimation can be gradually close to the output Y . During the construction of each decision tree, the parameters that determine the tree structure and influence the iteration process are determined through cross-validation.

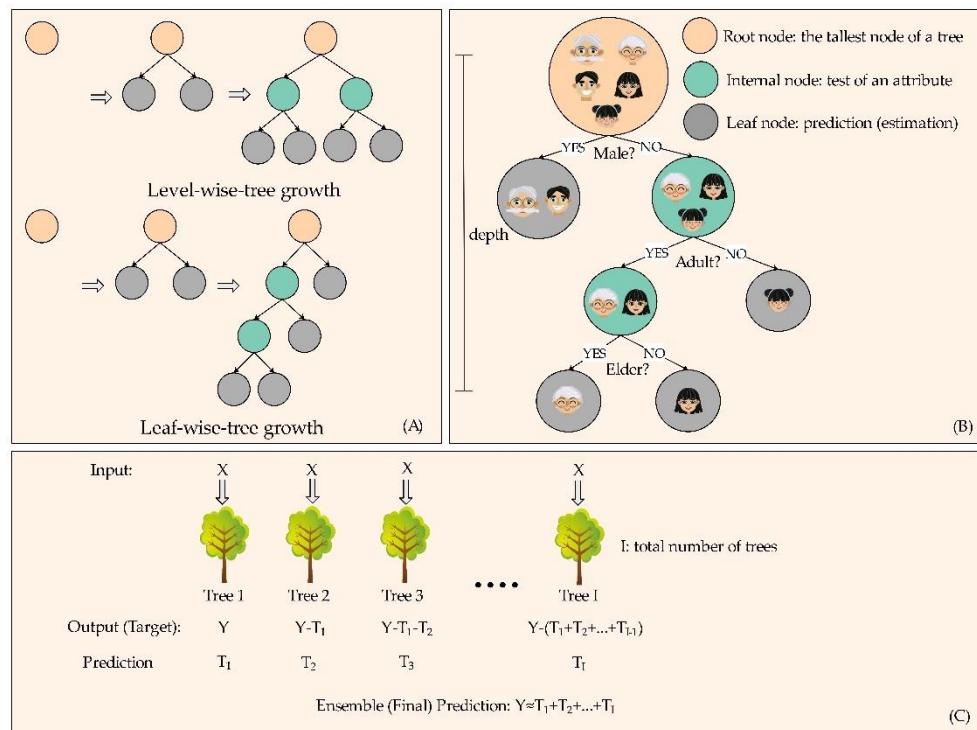


Figure 1. Schematic diagram of the LightGBM model: (A) growth tree structures; (B) an example of leaf-wise-tree growth conceptual algorithm and (C) Gradient Boosting Decision Tree algorithm.

Another commonly-used GBDT-based model is the XGBoost model [37]. It has gained much popularity and attention in recent years, such as the studies of Shi et al. [38] and Dong et al. [39]. The XGBoost model uses the level-wise-tree growth algorithm to generate the tree nodes. Compared with other GBDT-based models such as the XGBoost model, the LightGBM model also provides another three algorithms to accelerate its training, namely histogram-based, gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) algorithms. The histogram-based algorithms [34] convert the sorted dataset of the parameters in the input layer into a histogram with a specified number of data intervals or bins [36] (Fan et al., 2019), as shown in Figure 2A. After this transformation, each data interval (bin) shares the same index. This algorithm allows the LightGBM model to require a much lower memory consumption while considerably accelerating the training speed. However, compared with the XGBoost model, the split point of the dataset found by the LightGBM model may be less accurate than that of the XGBoost model because the LightGBM traverses the data bins (Figure 2A) while the XGBoost model directly traverses the dataset. As shown in Figure 2B, the GOSS algorithm [34] approximates the sum of sample data by using the top $a\%$ descending-sorted data points and part randomly selected data ($b\%$) from the remaining data ($100\% - a\%$). The GOSS algorithm reduces the number of data instances, while the accuracy of the learned decision trees can be still kept [34]. The EFB algorithm reduces the parameter size by merging some parameters in the input layer, which also improves the training speed. Taking Figure 2C as an example, the repeated zero-value points are in the same bin (Figure 2A). However, the location of the nonzero values of Parameter 1 and Parameter 2 in Figure 2C are mutually exclusive. Merging them does not change the distribution of the output (target) dataset but can reduce the parameters' size in the input layer. Through these algorithms, the LightGBM model has the advantage of a faster training speed and is suitable for handling large-scale datasets.

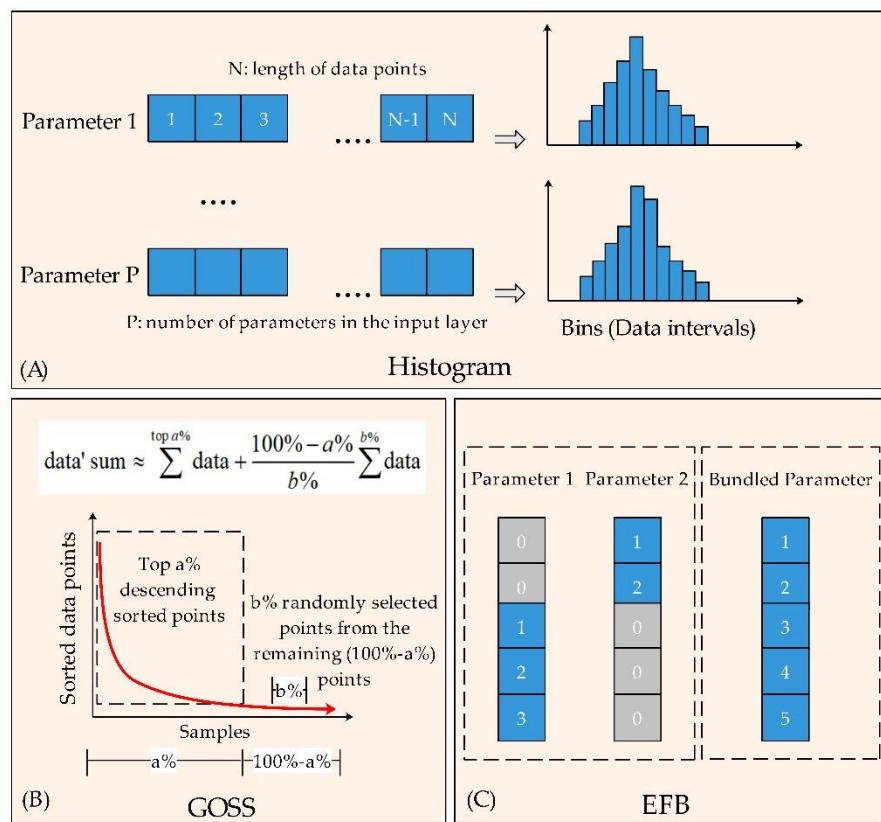


Figure 2. (A) Histogram, (B) gradient-based one-side sampling, and (C) exclusive feature bundling algorithms provided in the LightGBM model to accelerate the training.

Taking the estuarine water levels as an example in this study, they can be estimated by the LightGBM model as:

$$\eta_I = \sum_{i=1}^I T_i(X, \theta_i) \quad (1)$$

where η_I (unit: m) is the estimated estuarine water level by the LightGBM model; T_i (unit: m) is the estimation of the water level from the i th tree, in which X is the input parameter dataset and θ_i is the learned parameter set of the i th tree, and I is the total number of decision trees.

During the learning stage, T_i can be determined by minimizing a loss (error) function, L , against the observation data, η_{obs} . For the 1st tree, T_1 can be estimated with the following equation:

$$T_1 = \operatorname{argmin}\{L(\eta_{obs}, \eta_1)\} \quad (2)$$

where argmin is the condition of the function achieving the minimum value between the observed and an initial estimation, η_1 . For the i th tree ($i > 1$) tree, the square error (SE) is adopted as the loss function, $L(\eta_i) = (\eta_{obs} - \eta_i)^2$, to evaluate the model errors, with the introduction of a target function, Obj_i , as shown below:

$$Obj_i = L(\eta_i) + \Omega(T_i) \quad (3)$$

where the target function of the i th tree, Obj_i (unit: m^2), contains two parts, one represents the loss function for evaluating model accuracy, $L(\eta_i)$ and the other, $\Omega(T_i)$ represents model robustness, which is a regular function expressed as:

$$\Omega(T_i) = \alpha \sum_{j=1}^J |\omega_j| + \frac{1}{2} \beta \sum_{j=1}^J \omega_j^2 \quad (4)$$

where α and β are hyperparameters, which are used to control the learning process, to be tuned at the hyperparameters optimization stage, j is the index of the leaf node, while J is the total number of leaf nodes of the i th decision tree and ω_j (unit: m) is the model estimation at the j th leaf node. The first and second terms on the right-hand side of Equation (4) are respectively the Lasso Regression (L1 regularization) and Ridge Regression (L2 regularization) [40]. Both are used to improve the model stability (avoid over-fitting) by adding different penalty terms. If $\alpha = 0$ and $\beta = 0$, Equation (3) becomes the ordinary square error function. The i th tree is determined by finding the smallest Obj_i following the least square error approach. In other words, the determination of the decision tree structure not only considers the model accuracy but also takes into account the model's stability.

The LightGBM model pertains to the concept of GBDT, which uses the second-order Taylor expansion to estimate the target function. More specifically, the i th target function is estimated by using the learning results from the $(i - 1)$ th process:

$$Obj_i = L(\eta_{i-1}) + gT_i(X, \theta_i) + \frac{1}{2}hT_i^2(X, \theta_i) + \Omega(T_i) \quad (5)$$

$$g = \frac{\partial L(\eta_{i-1})}{\partial \eta_{i-1}} \quad (6)$$

$$h = \frac{\partial^2 L(\eta_{i-1})}{\partial \eta_{i-1}^2} \quad (7)$$

where g and h are the first and second derivations of $L(\eta_{i-1})$ with respect to η_{i-1} . Since $L(\eta_{i-1})$ is known from the $(i - 1)$ th learning process, the target function can be written as:

$$Obj_i = \sum_{j=1}^J \left[(G_j \omega_j + \alpha | \omega_j |) + \frac{1}{2} (H_j + \beta) \omega_j^2 \right] \quad (8)$$

$$G_j = \sum_{t=1}^N g^t \quad (t \in \text{leaf node } j) \quad (9)$$

$$H_j = \sum_{t=1}^N h^t \quad (t \in \text{leaf node } j) \quad (10)$$

where ω_j at each leaf node on the i th tree is the unknown variable to be solved, the superscript t is the sample index and N is the total number of sample points. In Equation (8), except for ω_j , all other parameters are specified or can be obtained from the $(i - 1)$ th learning process. In order to obtain the smallest Obj_i , we have:

$$\frac{\partial Obj_i}{\partial \omega_j} = 0 \quad (11)$$

which yields the following:

$$\omega_j = -sgn(G_j) \frac{G_{\alpha,j}}{H_j + \beta} \quad (12)$$

$$G_{\alpha,j} = \max\{0, (|G_j| - \alpha)\} \quad (13)$$

where sgn is the sign function. When ω_j in Equation (12) is determined, the objective function in Equation (8) can be re-written as:

$$Obj_i = -\frac{1}{2} \sum_{j=1}^J \left(\frac{G_{\alpha,j}^2}{H_j + \beta} \right) \quad (14)$$

To statistically evaluate the tree performance at each leaf node, a score function for the j th leaf node is introduced as:

$$V_j = \frac{G_{\alpha,j}^2}{H_j + \beta} \quad (15)$$

It is clear that Equation (15) is a subterm of Equation (14), indicating the model's error reduction contributed from the j th leaf node. Equation (15) can assist the LightGBM model in selecting the most suitable node to split further. If two new leaf nodes are generated from the j th leaf node, the difference of V_j after the split, V_d , can be calculated:

$$V_d = \left(\frac{G_{L,\alpha,j}^2}{H_{L,j} + \beta} + \frac{G_{R,\alpha,j}^2}{H_{R,j} + \beta} \right) - \frac{G_{\alpha,j}^2}{H_j + \beta} \quad (16)$$

where $G_{L,\alpha,j}$ ($H_{L,j}$) and $G_{R,\alpha,j}$ ($H_{R,j}$) are the values of G (H) on the left and right nodes newly-generated from the j th node. Equation (16) statistically compares the model's error reduction when new leaf nodes are generated from different nodes. After examining all the parameters in the input layer and all the leaf nodes, the LightGBM model chooses the most suitable parameter to split from the leaf node where the largest V_d can be obtained.

2.2. Construction of the LightGBM Model

2.2.1. Input and Output Setting

The input dataset of the LightGBM model should contain the factors that influence the variation of estuarine water levels. These input factors should also be predicted when they are used to forecast estuarine water levels. The effective input parameters include the variables related to the tide potential or the time series of tides constituents from ocean tides or coastal tides. However, due to the strong nonlinear interaction between tides and river discharge, these factors may not be enough. As shown in the works of Kukulka and Jay [41,42], the upstream river discharge and tide range near the estuary mouth are important factors in analyzing estuarine tides. Supharatid [20] also included them as variables of the input layer of the ANN model. Therefore, the LightGBM model also includes the river discharges and tide range as the parameters in the input layer.

Figure 3 shows the schematic map of the input and output layers of the LightGBM model in this study. In order to both consider the riverine force and marine force factors, similar to the works of Jay et al. [43] and Matte et al. [7], upstream river discharge (Q) and tide range (R) near the estuary mouth were selected in the input layer of the LightGBM model. Moreover, the upstream river discharge was low-passed to smooth the data. In the semi-diurnal tide regime, tide range R is the greater tide range within one lunar day. Referred from the works of Lee [18] and Lee et al. [44], the time series of tide constituents were also included. The tide constituents whose signal-noise-ratio values are larger than 2 in the results of the T_TIDE model [4] were selected as the input of the LightGBM model. Harmonic analysis was used to preliminarily filter the insignificant tide constituents and can reduce the computation time of the LightGBM model. In the LightGBM model, it selects the important parameters from the input layer based on finding the parameter that contributes the largest error reduction (Equation (16)).

2.2.2. Hyperparameters Setting

In the LightGBM model, a number of hyperparameters were used, but many of them can be adjusted to improve the model performance for different applications as the hyperparameters may significantly influence the model performance. Therefore, hyperparameters tuning is an essential process in the construction of machine learning models. The hyperparameters tuned in this study and their related meanings are listed in Table 1. Only the hyperparameters listed in Table 1 were tuned. The other hyperparameters that can further slightly influence the training process, such as setting the minimal H_j on the leaf node and the minimal number of data inside one bin, were kept as default. Generally, the core

hyperparameters for regression are listed in Table 1. More details of the hyperparameters of the LightGBM model can be found in the LightGBM documentation [45].

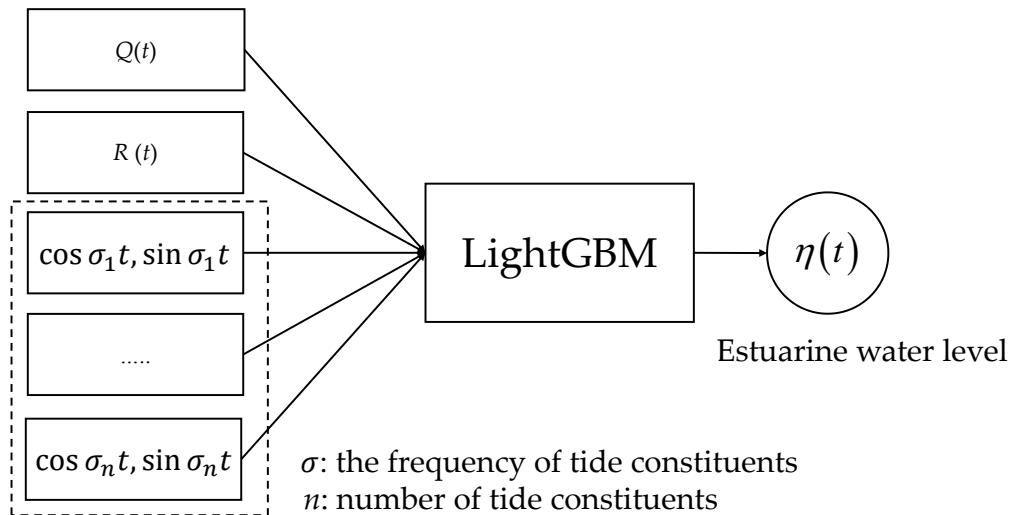


Figure 3. Schematic map of the input and output layers of the LightGBM model used in this study.

Table 1. The main hyperparameters of the LightGBM model tuned in this study.

Hyperparameters	Description and Usage	Type	Range	Default
learning_rate	Control the shrinkage rate; smaller value indicates a smaller iteration step.	double	>0.0	0.1
num_iterations	Number of iterations (trees).	int	≥ 0	100
max_depth	Limit the max depth for a tree model.	int	case-based	
num_leaves	Control the maximum number of leaves of a decision tree.	int	1–131,072	31
max_bin	Control the max number of bins (data intervals) when the dataset of a parameter in the input layer is transformed to a histogram (Figure 2A).	int	1–255	255
min_data_in_leaf	Minimal number of data in one leaf.	int	≥ 0	20
feature_fraction	The proportion of the selected parameters to the total number of the parameters in the input layer.	double	0.0–1.0	1.0
bagging_fraction	The proportion of the selected data to the total data size.	double	0.0–1.0	1.0
bagging_freq	Frequency of re-sampling the data when bagging_fraction is smaller than 1.0.	int	≥ 0	0
lambda_l1	The value of α in Equation (4).	double	≥ 0.0	0
lambda_l2	The value of β in Equation (4).	double	≥ 0.0	0
min_gain_to_split	Indicating the minimal error reduction to conduct the further split; corresponding to the minimal value of Equation (16)	double	≥ 0.0	0

2.2.3. Optimal Hyperparameters Searching

Figure 4 shows a diagram of constructing the optimal LightGBM model in this study. The main process was to optimize the training process of the LightGBM model by adjusting the hyperparameter values as listed in Table 1. The numerical range of the values of the hyperparameters in Figure 4 is specified according to preliminary tests, and they can be varied when this model is applied to other research areas. The hyperparameters in Table 1 are divided into four steps to be sequentially optimized (Figure 4) when suitable learning_rate and num_iterations are specified. The optimization sequence of these hyperparameters was determined according to their mutual relationship and the importance of their influence on the LightGBM model.

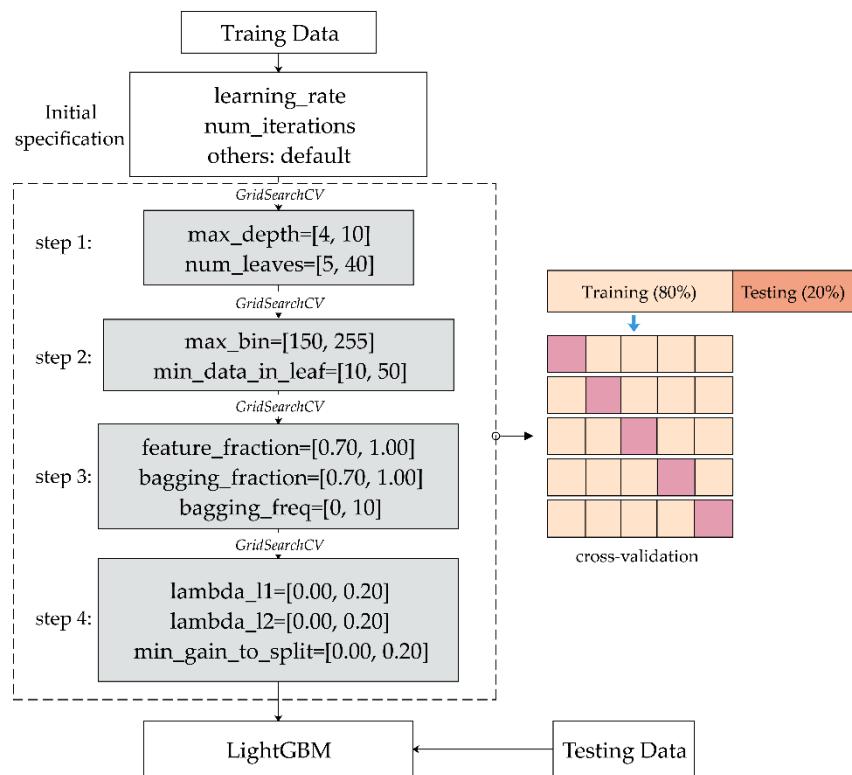


Figure 4. Flow chart of conducting the optimization of hyperparameters (ranges are indicated in the square brackets) of the LightGBM model.

In determining the optimal combination of hyperparameters, the GridSearchCV function in the Scikit-learn Python module [46] was used. The GridSearchCV function uses the concept of the grid search method [47] to determine the optimal hyperparameters. On the basis of the specified range of hyperparameters, the GridSearchCV function forms all possible combinations and then performs a cross-validation process (Figure 4) to search the optimal hyperparameter combinations.

In the specification of the GridSearchCV function, a negative mean squared error (NMSE) was selected as the scoring criterion function to evaluate the model performance. In addition, five-fold cross-validation was specified to optimize the hyperparameters. More specifically, the initial training data in Figure 4 were further averagely divided to five groups, four of which were sequentially selected as the training data in the hyperparameters optimization period and the remaining group was used to test the model. Therefore, five cross-validation tests were conducted in each combination of hyperparameters. The highest NMSE score in the cross-validation tests represents the smallest model error, indicating the best combination of the hyperparameters in the given range.

The LightGBM model was constructed from the opensource framework of Microsoft® [34] and the related codes were run on the Python 3.7 environment.

3. Study Area & Data

3.1. The Lower Columbia River

The Columbia River is located in a region of North America (Figure 5) and flows into the North Pacific Ocean. The Columbia River is the third largest river in the U.S. state [14,15] and the largest river in the Pacific northwest region of North America. Its annual average flow is about $7500 \text{ m}^3/\text{s}$ [48]. Three variation patterns can be found in the river discharge records of the Columbia River [48]: First, it has a significant seasonal variation pattern; second, some high-flow events can happen in the winter and early spring seasons; third, due to the actions of hydraulic electrogenerating, it has a periodical variation pattern during low-flow periods. The research area of this study was the lower Columbia

River whose area ranges from the estuary mouth to 235 km landward along the river course [7]. It can be seen from Figure 5 that there is a tributary river (Willamette River) entering into the lower Columbia River. It is the largest tributary into the lower Columbia River, and its average flow is about $950 \text{ m}^3/\text{s}$ [7]. The tides in the Columbia River are classified as mixed tides. The amplitude ratio of semidiurnal tides to diurnal tides at the estuary mouth is around 1.5 [48]. Both the river discharge from the Columbia River and the Willamette River can affect the tides in the lower Columbia River. Under the influence of river discharge, the tides in the lower Columbia River present significant nonstationary characteristics. Former studies [10,14,41–43] clearly showed that the lower Columbia River is a suitable natural laboratory for the analysis of estuarine tides.

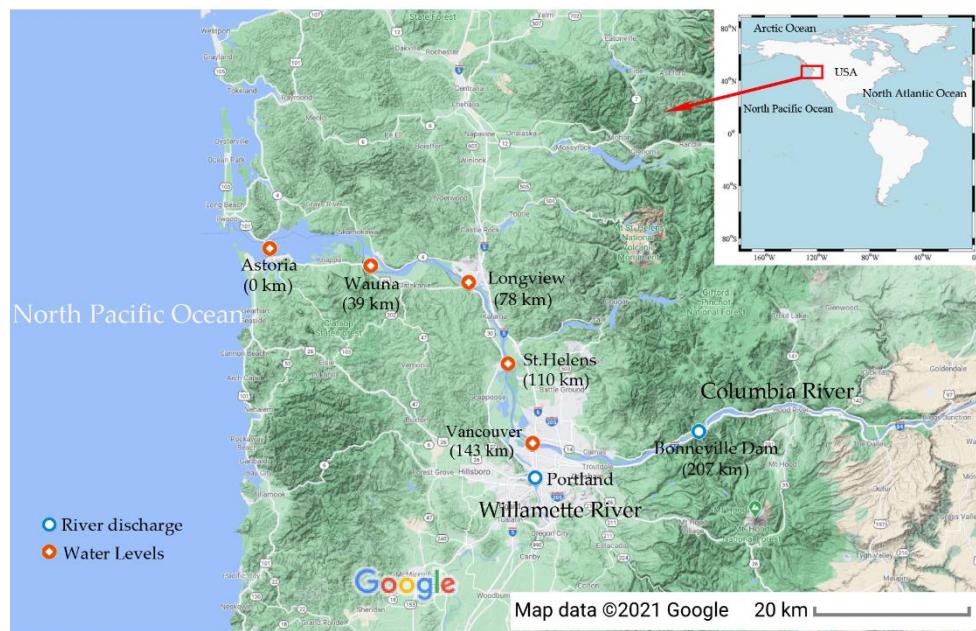


Figure 5. Map of the lower Columbia River and the locations of the hydrometric stations (modified from Google Map by M_Map software package [49]).

3.2. Data

The hydrometric stations used in this study are also shown in Figure 5. Hourly water level measurements at Astoria, Wauna, Longview, St.Helens, and Vancouver stations were collected from the National Oceanic and Atmospheric Administration (NOAA) website [50]. The duration of the water level measurements at all stations was from 2003 to 2020, except for Wauna station, where data between 2003 and 2006 was unavailable. In addition, the synchronized hourly river discharge data of the Columbia River at the Bonneville Dam (denoted as Q_c hereafter) and the Willamette River at Portland (denoted as Q_w hereafter) over the same period (2003–2020) were downloaded from the website of the U.S. Geological Survey (USGS) [51].

The date and time of all the measurements were adjusted to the Greenwich Mean Time (GMT), while the mean sea level datum was specified as the uniform datum when downloading the water levels data from the NOAA. The collected measurements of the water levels are shown in Figure 6A, while the measured river discharges are shown in Figure 6B. For the discharges of the Willamette River at Portland, it can be seen that the river discharge is under the influence of tides from the downstream, as evidenced by the negative values. From the collected data of this study, in about 70% of the whole period, the values of the Q_w were no more than 20% of the values of the Q_c . However, there were also a few periods that the values of Q_w exceeded Q_c , as shown in Figure 6B.

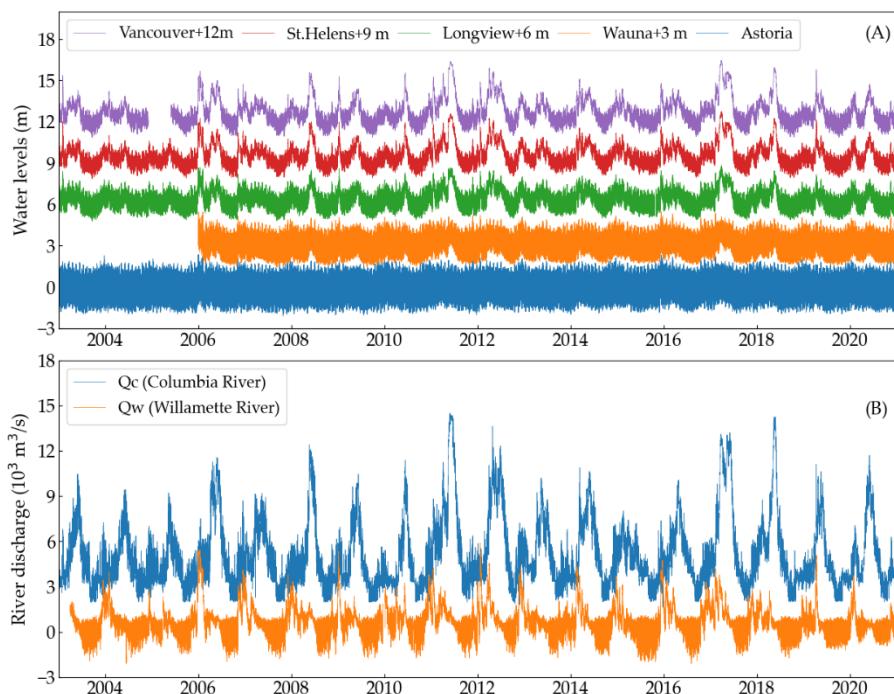


Figure 6. (A) Measured water levels at five stations along the lower Columbia River, and (B) measured river discharge of the lower Columbia River at the Bonneville Dam and the Willamette River at the Portland.

The Astoria station is the most seaward station in the study site and is assumed under little or no influence from the upstream river discharge. Therefore, it was selected as the reference station to provide the tide range information, while Bonneville Dam and Portland stations were used as the reference stations to provide the upstream river discharge information. The measured water levels at Wauna, Longview, St.Helens and Vancouver stations were used for training and testing the LightGBM model.

3.3. Data Preparation

The measured water levels at all stations were divided into two parts: 80% of all the data from January 2003 to June 2017, together with the tide range at Astoria and the river discharges at Bonneville Dam and Portland in the same period, was used for training, whilst the remaining data measured during June 2017–December 2020 (20% of the data) was used for model predictions. As shown in Figure 6A, the measured water level at Wauna station for training and testing was slightly less due to the unavailability. Wauna station used the same data division ratio for training (80%) and testing (20%). The short data length of Wauna station may have had only a minor impact on the results because the tides at Wauna station were less affected by upstream river discharge than Vancouver, St.Helens, and Longview stations (Figure 6A). The tides at Wauna station mainly present the characteristics of astronomic tides, while the river discharge primarily influences the tides in flood season.

4. Model Training

The training process of the LightGBM model was conducted in two stages. First, to specify the appropriate values of learning_rate and num_iterations, their influence on the model performance and the computational time of the LightGBM model was tested. With the other hyperparameters being defaulted, the RMSE values of the LightGBM model in the training period with learning_rate varying from its default value (0.1) to 0.05 and 0.01 were tested. The value of num_iterations of the LightGBM model is the iteration value when the model errors stop to decrease with the specified learning_rate.

The comparison of the RMSE values between the water levels predicted by the LightGBM model and measurements during the training period at each station is shown in Figure 7. It can be seen from Figure 7 that a smaller learning_rate led to better model performance (as indicated by the smaller RMSE values) at St.Helens, Longview and Wauna stations. However, at Vancouver station, the RMSE values when learning_rate = 0.05 were slightly smaller than that when learning_rate=0.01. Overall, the influence of learning_rate did not significantly influence the model performance. The difference between the RMSE values of each station with different learning_rate was within 0.01 m.

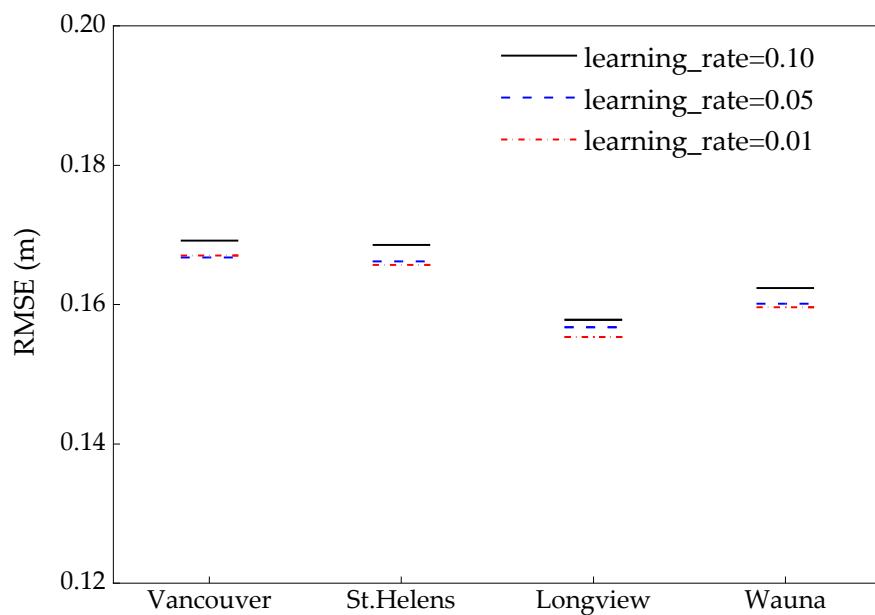


Figure 7. Comparison of the RMSE values of the predicted water levels from the LightGBM model during the training period when hyperparameters were not optimized, but with different learning_rate.

Taking the results at Vancouver station as an example, Figure 8 plots the variation of the RMSE values of the LightGBM model during each iteration with different learning_rate values. When learning_rate = 0.10, the LightGBM model used 184 iterations to reach the convergence state of the RMSE values. When learning_rate was reduced to 0.05 and 0.01, the iterations, respectively, increased to 380 and 1829. The results in Figures 7 and 8 show that a smaller learning_rate did not significantly improve the model accuracy but considerably increased the model iterations (computation time). According to the preliminary test results in Figures 7 and 8, learning_rate at each station was specified as 0.05 during the subsequent training and testing. Meanwhile, the corresponding num_iterations values at each station were, respectively, 380 (Vancouver), 291 (St.Helens), 339 (Longview) and 374 (Wauna).

The optimization of max_depth and num_leaves at Vancouver station was used as an instance to show the optimization process of the remaining hyperparameters. Parameters max_depth and number_leaves controlled the complexity of a decision tree. In the methods using the level-wise-tree growth strategy as shown in Figure 1A, such as the XGBoost model [37], num_leaves was 2^D (assuming D = max_depth). However, as the LightGBM model uses the leaf-wise-tree growth strategy [34], the depth of the growth tree generated could be deeper than the tree generated by the level-wise-tree growth strategy with the same value of num_leaves. In this study, max_depth was set in a range of from 4 to 10, while num_leaves was specified in a range of from 5 to 40 and also smaller than 2^D to avoid over-fitting.

All combinations of max_depth and number_leaves are shown as a grid graph in Figure 9 with their related NMSE values at the cross-validation period at Vancouver station. Indicated by the size (larger size represents better performance) and the color of the data points, it can be seen that the grid point, where max_depth = 5 and num_leaves = 17,

had the highest score (NMSE), indicating the best model performance during the hyperparameters optimization period. Compared with the default value of num_leaves (31), the results shown in Figure 9 suggest that a simpler (fewer leaf nodes) tree structure could achieve better model performance. However, it should be noted that the mutually restrictive relationship of max_depth or num_leaves prevented the NMSE values from presenting apparent monotonous relationships with max_depth or num_leaves. max_depth restricted the up limit number of tree leaf nodes, while num_leaves restricted max_depth to no more than num_leaves-1. Figure 9 mainly shows the process of searching the optimal hyperparameters based on all the hyperparameters' combinations in their specified range. During the subsequent optimization of the other hyperparameters (Step 2–Step 4 in Figure 4), the values of max_depth and num_leaves at Vancouver station were set as 5 and 17, respectively.

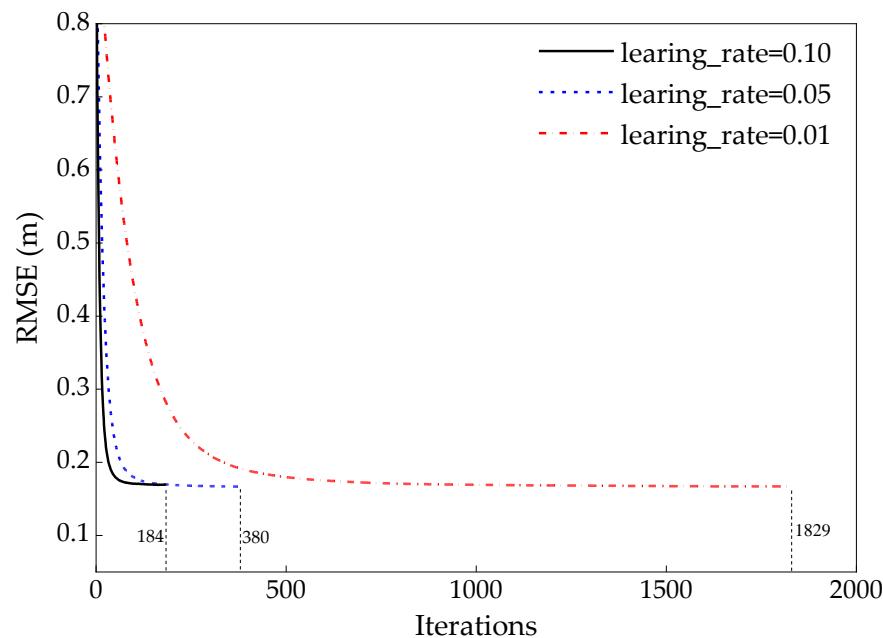


Figure 8. The variation of the RMSE values of the predicted water levels from the LightGBM model during the training period at Vancouver station with the increase of iterations.

The remaining hyperparameters were essential in improving model performance and accelerating the model iterations. Judged by the NMSE values, all the optimized hyperparameters are listed in Table 2. With the hyperparameters in Table 2, the final LightGBM model could be trained.

Table 2. The optimal hyperparameters at each station.

Hyperparameters	Stations			
	Wauna	Longview	St.Helens	Vancouver
num_iterations	374	339	291	380
max_depth	5	5	8	5
num_leaves	21	13	18	17
max_bin	165	190	195	194
min_data_in_leaf	17	16	23	19
feature_fraction	0.72	0.79	1.00	0.97
bagging_fraction	0.70	0.67	0.88	0.72
bagging_freq	1	6	8	4
lambda_l1	0.016	0.008	0.000	0.000
lambda_l2	0.060	0.100	0.052	0.138
min_gain_to_split	0.104	0.136	0.068	0.008

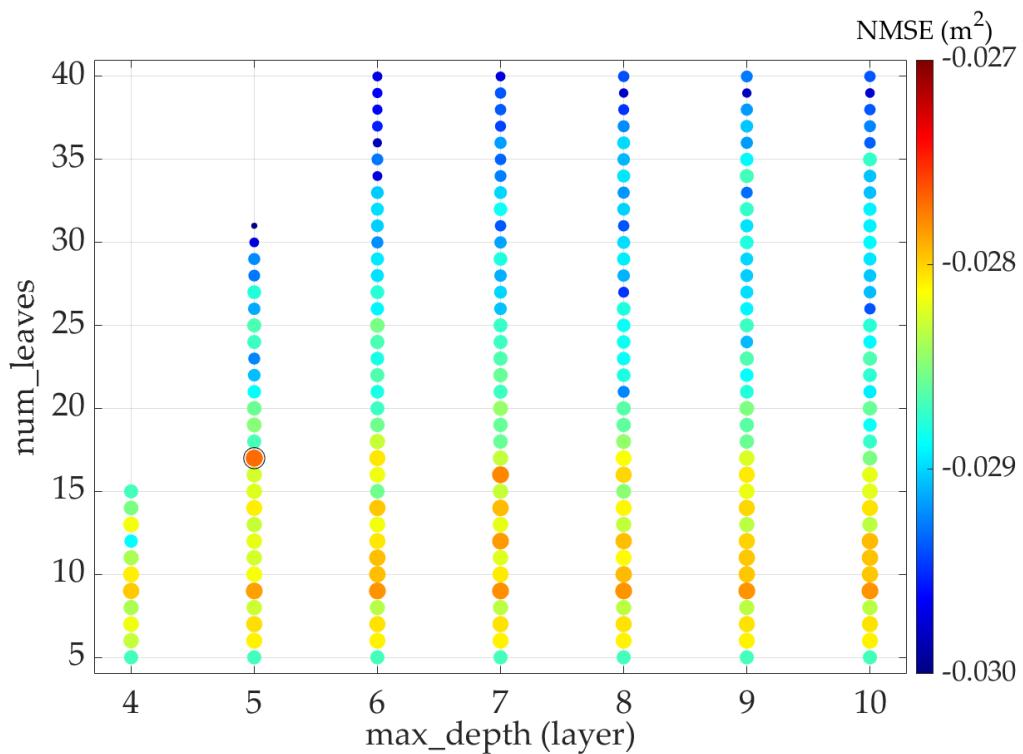


Figure 9. The variation of the NMSE values with different combinations of `max_depth` and `num_leaves` values at Vancouver station.

5. Model Testing

The testing data (20% of measurements) at each station were used to examine the model accuracy. Figure 10 shows the comparison of the water levels predicted by the LightGBM model with measurements at four stations. In general, the model results of the LightGBM model show a reasonable accuracy relative to the measurements. However, it is clear that the discrepancy seems to become larger when the water levels were high at Vancouver and St.Helens stations. This illustrates that the errors of the LightGBM model were larger in flood seasons at the upstream tide reach stations (Vancouver and St.Helens). However, the differences between the results of the LightGBM model and the measurements at Longview and Wauna stations were more randomly (or evenly) distributed.

In the time domain, Figure 11A shows the time series of the predicted water levels against the measurements at the Vancouver station during 2019 and 2020, while Figure 11B plots the model errors and the synchronous river discharge of the Qc (Columbia River). Again, the predicted water level from the LightGBM model generally agrees well with the measurements, as shown in Figure 11A. It is clear from Figure 11B that the average error limits between the predicted and measured water levels were mostly in the range of ± 0.5 m. However, larger errors up to approximately ± 1.0 m appeared when river discharge was strong in flood season, such as the periods between April to May of 2019 and May to June of 2020. The larger errors of the LightGBM model in these two periods both appeared during the rising flood period. In the rising flood period of 2019, the river discharge met a neap tide (Figure 11B), and the model errors mainly came from the phase difference between the model results and the measurements. However, in the rising flood period of 2020, the river discharge met a spring tide (Figure 11B). The model errors mainly sourced from the over-estimation of model results. This means the model errors of the LightGBM in the rising flood period tended to present a different pattern when the tidal condition was different.

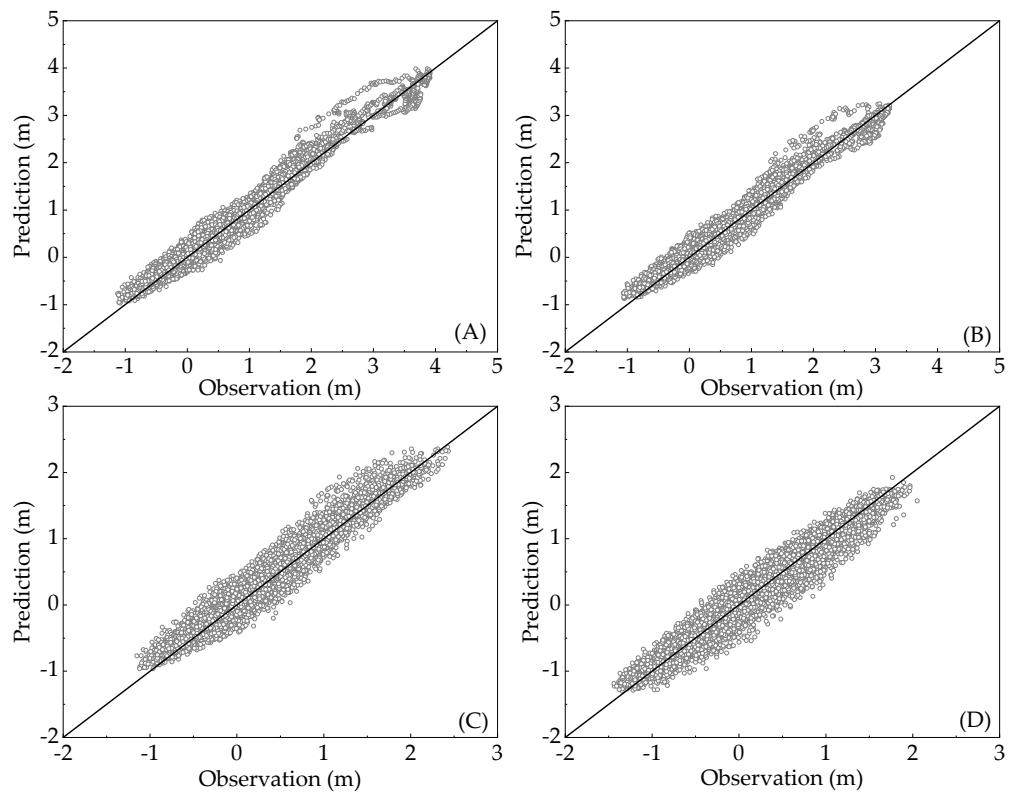


Figure 10. Comparisons of the predicted water levels from the LightGBM model with the measurements at (A) Vancouver, (B) St.Helens, (C) Longview and (D) Wauna stations.

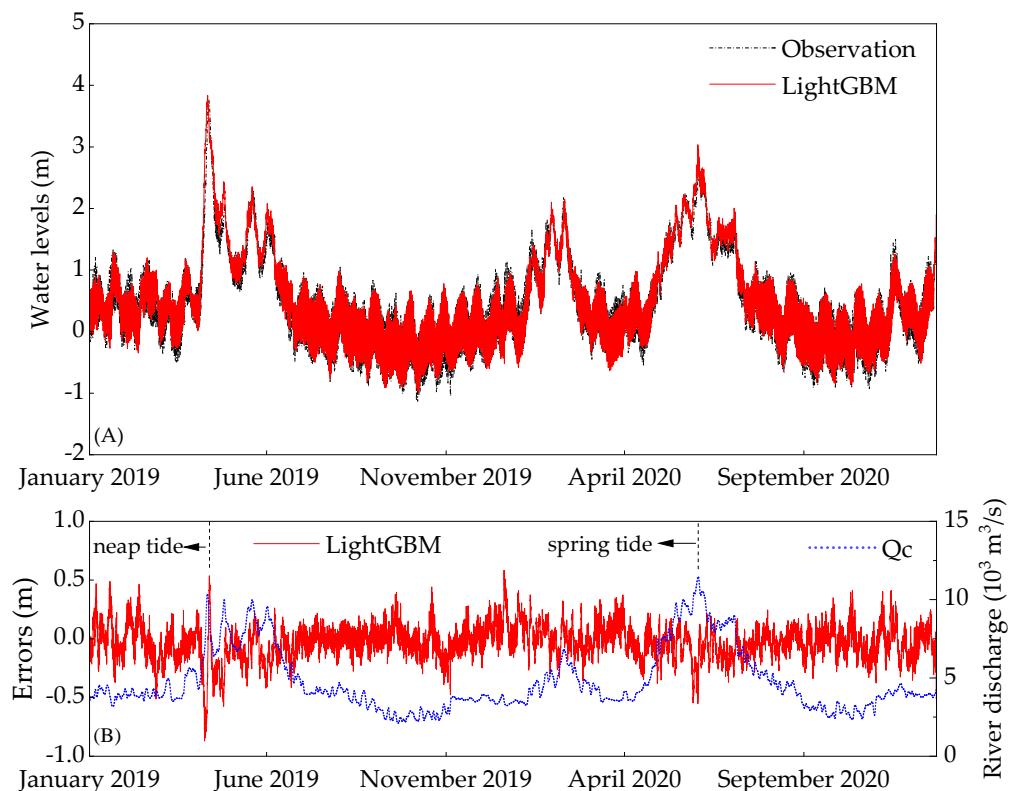


Figure 11. (A) Comparison of the LightGBM model results and the measurements at Vancouver station, and (B) the corresponding errors of the LightGBM model, the measured river discharge of the Columbia River.

To compare the performance of the LightGBM model at different tide reaches when river discharge was significant in 2019, Figure 12 compares the predicted and measured water levels particularly for the period between 1 April and 22 April in 2019, together with the river discharges from the Columbia River (Q_c) at all stations. Generally, the errors mainly arose when the phase of the water levels predicted by the LightGBM model was earlier than the measurements. Meanwhile, the phase difference was more significant at Vancouver and St.Helens stations than at Longview and Wauna stations. In the LightGBM model, the synchronized upstream river discharge and the water levels at each station were used, respectively, as the input and output of the LightGBM model. There was actually a propagation time for the upstream river discharge flowing to each station. In other words, there was a time lag in terms of the effect of the upstream river discharge on tides. Without considering the time lag, the signals of the water levels predicted by the LightGBM models could be earlier than the measurements. The time lag between the model results and the measurements was up to about one day at Vancouver and St.Helens stations and became smaller at Longview station (Figure 12C), but insignificant at Wauna station (Figure 12D) because the influence of river discharge on tides gradually became weaker in the landward direction.

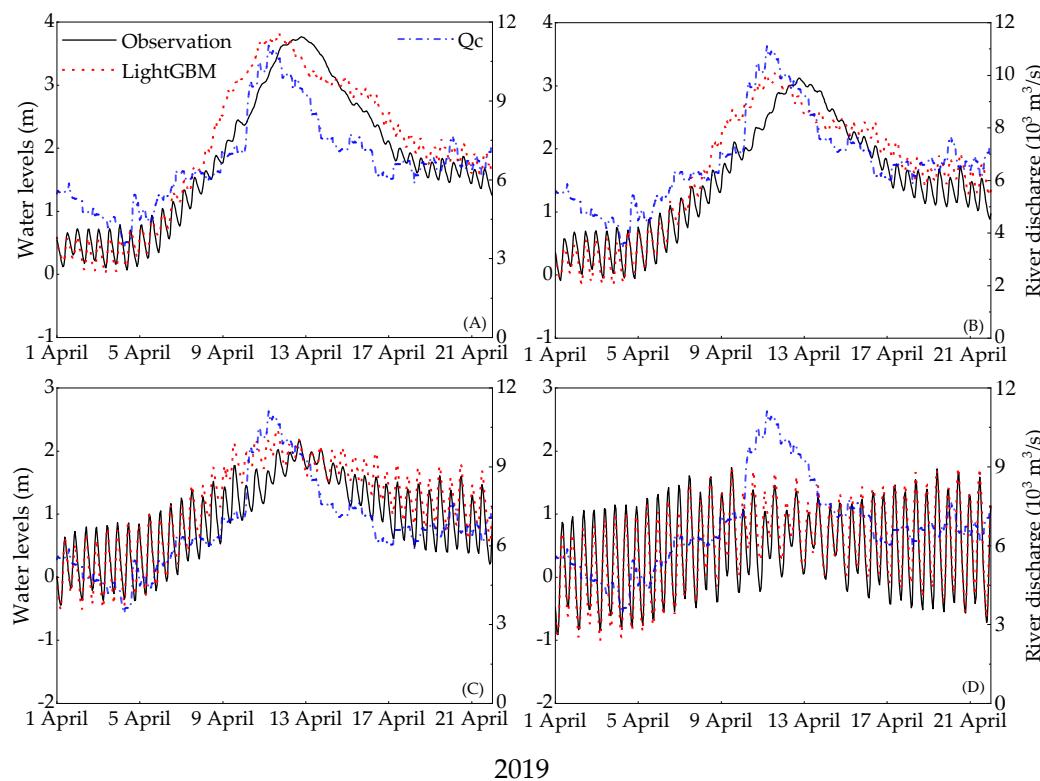


Figure 12. Comparison of the LightGBM model results and the measurements in April of 2019 at: (A) Vancouver, (B) St.Helens, (C) Longview and (D) Wauna stations.

For further comparison of the LightGBM model results and the measurements in other periods, Figure 13 compares the monthly maximum water levels from the LightGBM models and the measurements at each station. It is clear from Figure 13 that the error limits of the LightGBM model results relative to the measurements were all in the range of about ± 0.5 m. This clearly shows that the LightGBM model can give an accurate prediction of the monthly maximum water levels. However, due to the inability to consider the propagation time of the upstream river discharge or even tides, the errors of the LightGBM can be larger during the periods of flood season, such as the model errors at Vancouver station (Figure 11B).

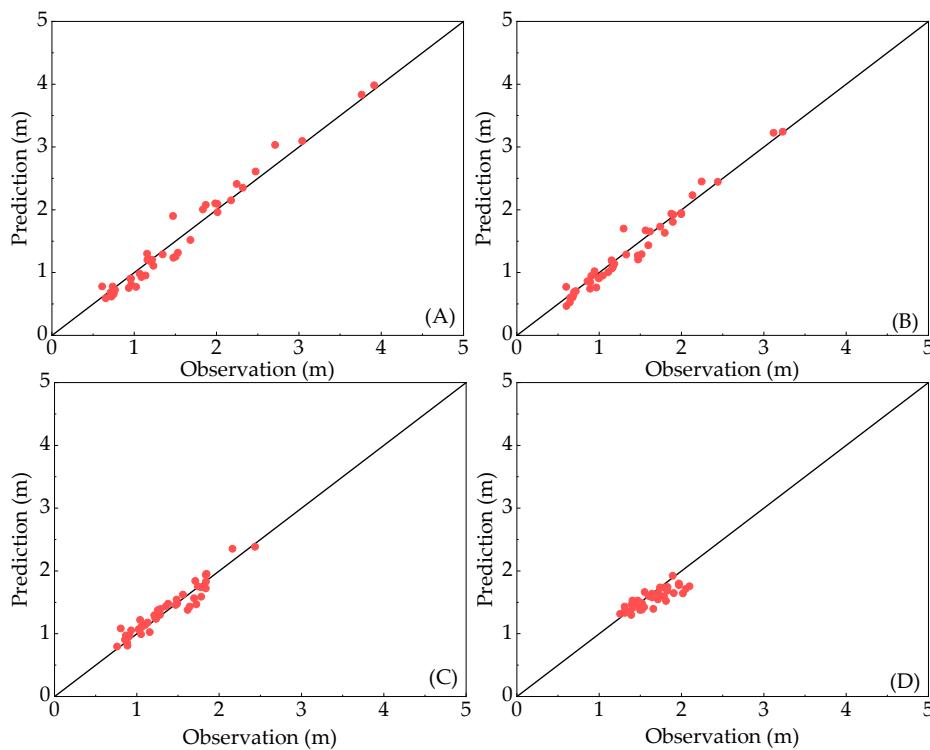


Figure 13. Comparison of the monthly maximum water levels of the LightGBM model results and the measurements at: (A) Vancouver, (B) St.Helens, (C) Longview and (D) Wauna stations.

Although the LightGBM model had relatively larger errors in the flood season, its accuracy was still comparable with the physics-based models. For statistical comparison, the performance of the LightGBM model was further compared with the NS_TIDE model [7,13] because of its typical application in the analysis of estuarine tides. The details of the NS_TIDE can be found in Matte et al. [7]. The specification of the NS_TIDE model at the lower Columbia River is referred to the works of Matte et al. [7] and Pan et al. [14]. The same data division as the LightGBM model was used to examine the model performance of the NS_TIDE model. Table 3 compares their models' performance by using the performance indicators of the maximum absolute error (MAE), RMSE, correlation coefficient (CC), and the nondimensional skill score (SS) based on the method of Murphy [52]. An SS value close to 1 represents a better model performance. It can be seen that the RMSE values of the LightGBM model were all smaller than the NS_TIDE model, which indicates that the LightGBM model statistically performed better than the NS_TIDE model. The MAE and the values of CC and SS of the LightGBM model were, respectively, smaller and larger than the NS_TIDE model. The results in Table 3 illustrate that the LightGBM model could achieve better performance than the NS_TIDE model.

Table 3. Comparison of the performance indicators of the LightGBM and the NS_TIDE models.

Stations	LightGBM/NS_TIDE			
	MAE (m)	RMSE (m)	CC	SS
Vancouver	0.87/1.09	0.14/0.16	0.987/0.983	0.972/0.965
St.Helens	0.82/0.97	0.14/0.15	0.982/0.978	0.963/0.956
Longview	0.75/0.90	0.14/0.15	0.975/0.972	0.941/0.938
Wauna	0.72/0.82	0.14/0.16	0.977/0.975	0.955/0.947

6. Discussion

6.1. Parameter Contribution to the LightGBM Model

During the propagation of tide waves toward the upstream direction in an estuary, nonlinear interaction between tides and river discharge intensifies, as well as the tide constituents themselves, so that the overall influence of the river discharge on estuarine tides can vary spatially. These characteristics make the parameters in the input layer (Figure 3) contribute differently in constructing the tree nodes of the LightGBM model at different tide reaches. Figure 14 shows the contribution of the top 15 parameters in the input layer at each station, including two river discharges (Q_c and Q_w), tide range at Astoria (R) and 12 tide constituents from statistical analysis. The parameter contribution represents the ratio of the times of the parameter used in tree splitting to the total times of all the parameters used in tree splitting. A larger value of the parameter contribution means that it has been used more times to split the nodes, indicating more significant parameter importance. For each tide constituent, the cosine and sine parts are used as the input parameters, but the sum of the parameter contribution of the cosine and sine parts of a tide constituent is shown in Figure 14. It is clear from Figure 14 that the river discharges Q_c and Q_w consistently rank the top three at each station. The tide range at Astoria (R) ranks between 7th–13th at those stations for its contribution. In respect of tide constituents, their ranks are basically related to their tide amplitudes. For example, the M2 tide constituent, which is the most significant astronomic tide constituent, ranks 3rd–5th. Although the LightGBM model plays the role of a black box, its selections of the parameters to the construction of the decision trees is still related to the physic background of the lower Columbia River.

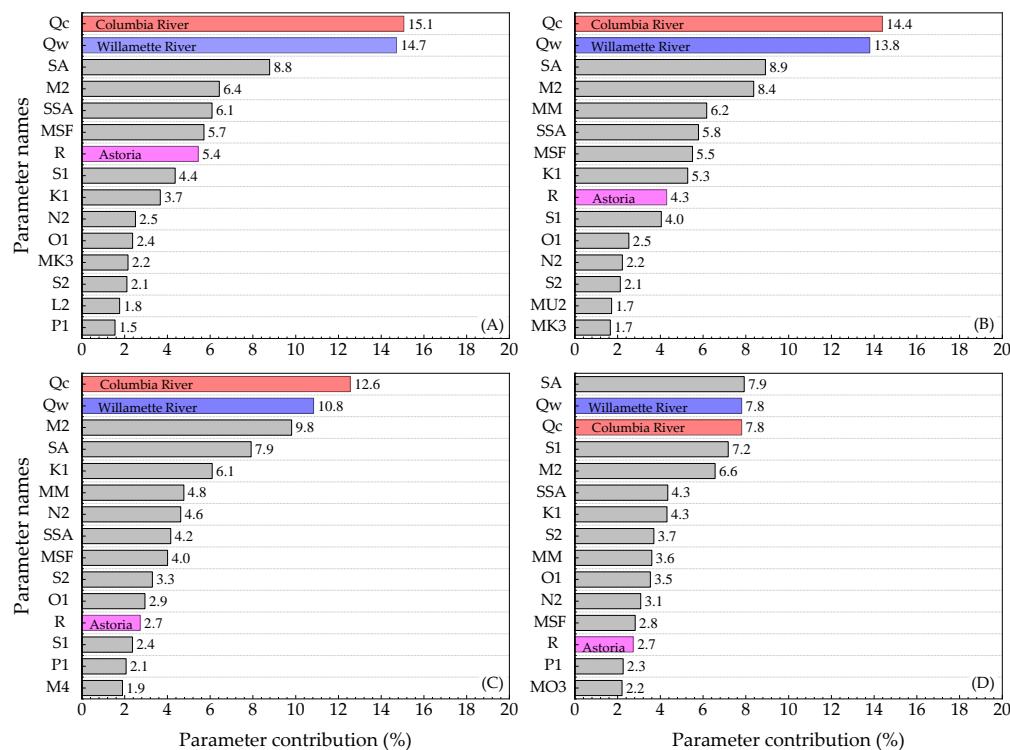


Figure 14. The top 15 parameters of the input layer of the LightGBM model at: (A) Vancouver, (B) St.Helens, (C) Longview, and (D) Wauna stations.

6.2. The Subtide Constituents

It can be seen from Figure 14 that the SA, SSA, MSF, and MM tide constituents are in the top 15 of the most important parameters. To further analyze these subtide constituents, their amplitudes obtained from the measurements are plotted in Figure 15 and compared with the M2 tide constituent. These subtide constituents generally become more significant

in the landward direction. Near the estuary mouth where the reference station (Astoria) is located, the amplitudes of these sub-tide constituents are insignificant except the SA tide constituent (0.10 m). The strength of sub-tides increasing in the landward direction was also reported in previous studies [12,28,53,54]. For example, MSF (MM) can become more energetic from the nonlinear interaction of the M2 and S2 (M2 and N2) tide constituents during the landward propagation of tide waves. Moreover, the lower frequency of sub-tides relative to the tides such as diurnal and semidiurnal tides favors their propagation because sub-tides are affected by smaller friction [8].

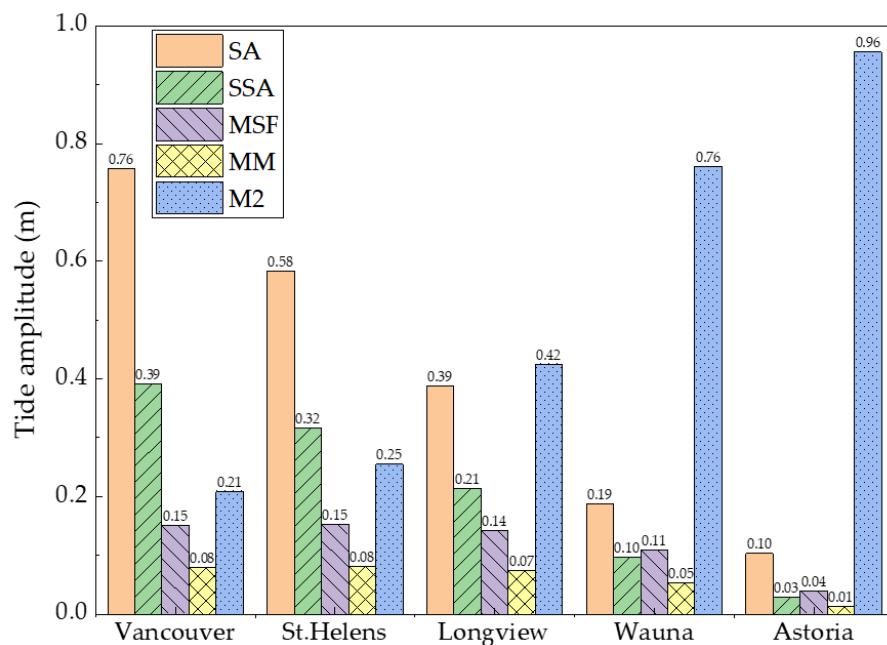


Figure 15. Amplitudes of the highest important sub-tide constituents and the M2 tide constituent at each station.

The upstream river discharge can also directly affect the water level variation. The upstream river discharge is annually and seasonally varied, making the water level fluctuation also annual and seasonal. The stronger effect of upstream river discharge on water levels along the upstream direction can be reflected in the increase of the SA and SSA tide amplitudes as their periods are annual and semiannual, respectively. The amplitudes of the SA and SSA tide constituents increase by about 7 and 13 times from Astoria station to Vancouver station, which is comparable with the variation ratio of the M2 tide constituent. It should be noted that the amplification of the SA and SSA tide constituents mainly comes from the effect of the variation pattern of the influence of river discharge on water levels. This makes their physical meaning different from the SA and SSA tide constituents in coastal or ocean zone. Figure 15 illustrates that the strong inland propagation of sub-tide waves is also exited in the Lower Columbia River and responds to the high ranks of the sub-tide constituents in Figure 14.

7. Conclusions

In this study, a novel prediction model was built based on the machine learning LightGBM framework to accurately predict the water levels in a tide-affected estuary and gain a better understanding of the nonlinear interaction between tides and river discharge in estuarine waters. The model was fully trained with the data obtained from the lower Columbia River to establish the optimal parameters required. The model results were also compared with those from the NS_TIDE model.

In the input layer of the LightGBM model, the river discharge from the lower Columbia River at the Bonneville Dam and the Willamette River at Portland, the tide ranges near

the estuary mouth (Astoria), and the tide constituents are used. 80% of the field data were used for training and optimizing the model, and the remaining 20% of measurements were used for testing the model performance.

The accuracy of the LightGBM model was statistically evaluated by RMSE, MAE, CC, and SS. The RMSE value of the LightGBM model was 0.14 m, which shows higher accuracy than those from the NS_TIDE model (0.15–0.16 m). The MAE values (0.72–0.87 m), the CC values (0.975–0.987), and the SS values (0.941–0.972) of the LightGBM model also indicated a comparable prediction accuracy relative to the MAE values (0.82–1.09 m), the CC values (0.972–0.983), and the SS values (0.938–0.965) of the NS_TIDE model. It was also revealed that the predicted water levels from the LightGBM model in flood season exhibited a phase lag affected by the upstream river discharge.

The results clearly demonstrate that the LightGBM model is a powerful machine learning tool for predicting the water levels in the fluvial flow affected estuaries, such as the Lower Columbia River and maybe any other mega-estuaries worldwide, with satisfactory accuracy. The LightGBM model can reveal a new perspective to the prediction of estuarine water levels.

Author Contributions: Conceptualization, M.G., S.P. and Y.C.; Data curation, M.G. and H.P.; Formal analysis, M.G., S.P., Y.C., C.C., H.P. and X.Z.; Funding acquisition, M.G. and Y.C.; Investigation, M.G., S.P., Y.C. and H.P.; Methodology, M.G., S.P., Y.C. and C.C.; Software, M.G., S.P. and C.C.; Supervision, S.P. and Y.C.; Validation, M.G., S.P. and C.C.; Writing—original draft, M.G., S.P., Y.C., C.C., H.P. and X.Z.; Writing—review & editing, M.G., S.P., Y.C., C.C., H.P. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the National Natural Science Foundation of China [Grant No: 51620105005], the Fundamental Research Funds for the Central Universities of China [Grant No: 2018B635X14, B200204017], and the Postgraduate Research & Practice Innovation Program of Jiangsu Province [Grant No: KYCX18_0602].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The water level and river discharge data used in this study can be obtained from the National Oceanic and Atmospheric Administration (NOAA) website (<https://www.noaa.gov/>, accessed on 27 April 2021) and the U.S. Geological Survey (USGS) website (<https://www.usgs.gov/>, accessed on 27 April 2021), respectively. The LightGBM model used in this study is based on the open-source framework of Microsoft® available at <https://github.com/microsoft/LightGBM/tree/master/python-package> (accessed on 27 April 2021).

Acknowledgments: The first author would like to acknowledge the financial support from the China Scholarship Council (CSC) under PhD exchange program with Cardiff University [201906710022]. The LightGBM model was run on the Supercomputing Wales Hawk clusters, funded by the European Regional Development Fund (ERDF). The authors would also like to thank Pascal Matte for providing codes of the NS_TIDE model.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Savenije, H.H.G. Prediction in ungauged estuaries: An integrated theory. *Water Resour. Res.* **2015**, *51*, 2464–2476. [[CrossRef](#)]
2. Garvine, R.W. The distribution of salinity and temperature in the connecticut river estuary. *J. Geophys. Res.* **1975**, *80*, 1176–1183. [[CrossRef](#)]
3. Chau, K.W. A split-step particle swarm optimization algorithm in river stage forecasting. *J. Hydrol.* **2007**, *346*, 131–135. [[CrossRef](#)]
4. Pawlowicz, R.; Beardsley, B.; Lentz, S. Classical tidal harmonic analysis including error estimates in MATLAB using T_TIDE. *Comput. Geosci.* **2002**, *28*, 929–937. [[CrossRef](#)]
5. Egbert, G.D.; Erofeeva, S.Y. Efficient inverse modeling of barotropic ocean tides. *J. Atmos. Ocean. Technol.* **2002**, *19*, 183–204. [[CrossRef](#)]
6. Gallo, M.N.; Vinzon, S.B. Generation of overtides and compound tides in Amazon estuary. *Ocean Dyn.* **2005**, *55*, 441–448. [[CrossRef](#)]

7. Matte, P.; Jay, D.A.; Zaron, E.D. Adaptation of classical tidal harmonic analysis to nonstationary tides, with application to river tides. *J. Atmos. Ocean. Technol.* **2013**, *30*, 569–589. [[CrossRef](#)]
8. Jay, D.A. Green's law revisited: Tidal long-wave propagation in channels with strong topography. *J. Geophys. Res.* **1991**, *96*, 20585. [[CrossRef](#)]
9. Godin, G. The propagation of tides up rivers with special considerations on the upper Saint Lawrence River. *Estuar. Coast. Shelf Sci.* **1999**, *48*, 307–324. [[CrossRef](#)]
10. Pan, H.; Lv, X.; Wang, Y.; Matte, P.; Chen, H.; Jin, G. Exploration of Tidal-Fluvial interaction in the Columbia River Estuary Using S_TIDE. *J. Geophys. Res. Oceans* **2018**, *123*, 6598–6619. [[CrossRef](#)]
11. Cai, H.; Yang, Q.; Zhang, Z.; Guo, X.; Liu, F.; Ou, S. Impact of river-tide dynamics on the temporal-spatial distribution of residual water level in the Pearl River channel Networks. *Estuaries Coasts* **2018**, *41*, 1885–1903. [[CrossRef](#)]
12. Gan, M.; Chen, Y.; Pan, S.; Li, J.; Zhou, Z. A modified nonstationary tidal harmonic analysis model for the Yangtze estuarine tides. *J. Atmos. Ocean. Technol.* **2019**, *36*, 513–525. [[CrossRef](#)]
13. Matte, P.; Secretan, Y.; Morin, J. Temporal and spatial variability of tidal-fluvial dynamics in the St. Lawrence fluvial estuary: An application of nonstationary tidal harmonic analysis. *J. Geophys. Res. Oceans* **2014**, *119*, 5724–5744. [[CrossRef](#)]
14. Pan, H.D.; Guo, Z.; Wang, Y.Y.; Lv, X.Q. Application of the EMD method to river tides. *J. Atmos. Ocean. Technol.* **2018**, *35*, 809–819. [[CrossRef](#)]
15. Pan, H.; Lv, X. Reconstruction of spatially continuous water levels in the Columbia River Estuary: The method of Empirical Orthogonal Function revisited. *Estuar. Coast. Shelf Sci.* **2019**, *222*, 81–90. [[CrossRef](#)]
16. Zhang, W.; Cao, Y.; Zhu, Y.; Zheng, J.; Ji, X.; Xu, Y.; Wu, Y.; Hoitink, A.J.F. Unravelling the causes of tidal asymmetry in deltas. *J. Hydrol.* **2018**, *564*, 588–604. [[CrossRef](#)]
17. Chang, H.; Lin, L. Multi-point tidal prediction using artificial neural network with tide-generating forces. *Coast. Eng.* **2006**, *53*, 857–864. [[CrossRef](#)]
18. Lee, T.L. Back-propagation neural network for long-term tidal predictions. *Ocean Eng.* **2004**, *31*, 225–238. [[CrossRef](#)]
19. Lee, T.L.; Jeng, D.S. Application of artificial neural networks in tide-forecasting. *Ocean Eng.* **2002**, *29*, 1003–1022. [[CrossRef](#)]
20. Supharatid, S. Application of a neural network model in establishing a stage–discharge relationship for a tidal river. *Hydrol. Process.* **2003**, *17*, 3085–3099. [[CrossRef](#)]
21. Cox, D.T.; Tissot, P.; Michaud, P. Water level observations and short-term predictions including meteorological events for entrance of galveston bay, Texas. *J. Waterw. Port Coast. Ocean Eng.* **2002**, *128*, 21–29. [[CrossRef](#)]
22. Liang, S.X.; Li, M.C.; Sun, Z.C. Prediction models for tidal level including strong meteorologic effects using a neural network. *Ocean Eng.* **2008**, *35*, 666–675. [[CrossRef](#)]
23. Riazi, A. Accurate tide level estimation: A deep learning approach. *Ocean Eng.* **2020**, *198*, 107013. [[CrossRef](#)]
24. Chang, F.; Chen, Y. Estuary water-stage forecasting by using radial basis function neural network. *J. Hydrol.* **2003**, *270*, 158–166. [[CrossRef](#)]
25. Tsai, C.C.; Lu, M.C.; Wei, C.C. Decision tree-based classifier combined with neural-based predictor for water-stage forecasts in a river basin during typhoons: A case study in Taiwan. *Environ. Eng. Sci.* **2012**, *29*, 108–116. [[CrossRef](#)]
26. Chen, W.B.; Liu, W.C.; Hsu, M.H. Comparison of ANN approach with 2D and 3D hydrodynamic models for simulating estuary water stage. *Adv. Eng. Softw.* **2012**, *45*, 69–79. [[CrossRef](#)]
27. Yoo, H.J.; Kim, D.H.; Kwon, H.; Lee, S.O. Data driven water surface elevation forecasting model with hybrid activation function—A case study for Hangang River, South Korea. *Appl. Sci.* **2020**, *10*, 1424. [[CrossRef](#)]
28. Chen, Y.P.; Gan, M.; Pan, S.Q.; Pan, H.D.; Zhu, X.; Tao, Z.J. Application of Auto-Regressive (AR) analysis to improve short-term prediction of water levels in the yangtze estuary. *J. Hydrol.* **2020**, *590*, 125386. [[CrossRef](#)]
29. Zhang, Q.C.; Yang, L.T.; Chen, Z.K.; Li, P. A survey on deep learning for big data. *Inf. Fusion* **2018**, *42*, 146–157. [[CrossRef](#)]
30. Sun, X.L.; Liu, M.X.; Sima, Z.Q. A novel cryptocurrency price trend forecasting model based on LightGBM. *Financ. Res. Lett.* **2020**, *32*, 101084. [[CrossRef](#)]
31. Zhu, S.L.; Hrnjica, B.; Ptak, M.; Choiński, A.; Sivakumar, B. Forecasting of water level in multiple temperate lakes using machine learning models. *J. Hydrol.* **2020**, *585*, 124819. [[CrossRef](#)]
32. Zhu, S.L.; Ptak, M.; Yaseen, Z.M.; Dai, J.Y.; Sivakumar, B. Forecasting surface water temperature in lakes: A comparison of approaches. *J. Hydrol.* **2020**, *585*, 124809. [[CrossRef](#)]
33. Chen, C.; Zhang, Q.M.; Ma, Q.; Yu, B. LightGBM-PPI: Predicting protein-protein interactions through LightGBM with multi-information fusion. *Chemom. Intell. Lab. Syst.* **2019**, *191*, 54–64. [[CrossRef](#)]
34. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*; Morgan Kaufmann Publishers: San Mateo, CA, USA, 2017; pp. 3146–3154.
35. Dev, V.A.; Eden, M.R. Formation lithology classification using scalable gradient boosted decision trees. *Comput. Chem. Eng.* **2019**, *128*, 392–404. [[CrossRef](#)]
36. Fan, J.L.; Ma, X.; Wu, L.F.; Zhang, F.C.; Yu, X.; Zeng, W.Z. Light gradient boosting machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data. *Agric. Water Manag.* **2019**, *225*. [[CrossRef](#)]
37. Chen, T.Q.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [[CrossRef](#)]

38. Shi, X.; Wong, Y.D.; Li, M.Z.; Palanisamy, C.; Chai, C. A feature learning approach based On XGBoost for driving assessment and risk prediction. *Accid. Anal. Prev.* **2019**, *129*, 170–179. [[CrossRef](#)]
39. Dong, W.; Huang, Y.; Lehane, B.; Ma, G. XGBoost algorithm-based prediction of concrete electrical resistivity for structural health monitoring. *Autom. Constr.* **2020**, *114*, 103155. [[CrossRef](#)]
40. Demir-Kavuk, O.; Kamada, M.; Akutsu, T.; Knapp, E.W. Prediction Using Step-Wise L1, L2 Regularization and Feature Selection for Small Data Sets with Large Number of Features. *BMC Bioinform.* **2011**, *12*, 412. Available online: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-12-412> (accessed on 27 April 2021). [[CrossRef](#)]
41. Kukulka, T.; Jay, D.A. Impacts of Columbia River discharge on salmonid habitat: 1. A nonstationary fluvial tide model. *J. Geophys. Res. Ocean.* **2003**, *108*, 3293. [[CrossRef](#)]
42. Kukulka, T.; Jay, D.A. Impacts of Columbia River discharge on salmonid habitat: 2. Changes in shallow-water habitat. *J. Geophys. Res. Ocean.* **2003**, *108*, 3294. [[CrossRef](#)]
43. Jay, D.A.; Leffler, K.; Degens, S. Long-term evolution of Columbia River tides. *J. Waterw. Port Coast. Ocean Eng.* **2011**, *137*, 182–191. [[CrossRef](#)]
44. Lee, T.; Makarynskyy, O.; Shao, C. A combined harmonic analysis–artificial neural network methodology for tidal predictions. *J. Coast. Res.* **2007**, *23*, 764–770. [[CrossRef](#)]
45. LightGBM’s Documentation. Available online: <https://lightgbm.readthedocs.io/en/latest/> (accessed on 27 April 2021).
46. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. [[CrossRef](#)]
47. Bergstra, J.; Bengio, Y. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305. Available online: <https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a> (accessed on 27 April 2021).
48. Jay, D.A.; Flinchem, E.P. Interaction of fluctuating river flow with a barotropic tide: A demonstration of wavelet tidal analysis methods. *J. Geophys. Res.* **1997**, *102*, 5705–5720. [[CrossRef](#)]
49. Pawlowicz, R. “M_Map: A Mapping Package for MATLAB”, Version 1.4m, [Computer Software]. 2020. Available online: www.eoas.ubc.ca/~{}rich/map.html (accessed on 27 April 2021).
50. National Oceanic and Atmospheric Administration. Available online: <https://www.noaa.gov/> (accessed on 27 April 2021).
51. U.S. Geological Survey (USGS). Available online: <https://www.usgs.gov/> (accessed on 27 April 2021).
52. Murphy, A.H. Skill scores based on the mean square error and their relationships to the correlation coefficient. *Mon. Weather Rev.* **1988**, *116*, 2417–2424. [[CrossRef](#)]
53. Guo, L.C.; Wegen, M.V.D.; Jay, D.A.; Matte, P.; Wang, Z.B.; Roelvink, D.; He, Q. River-tide dynamics: Exploration of nonstationary and nonlinear tidal behavior in the Yangtze River estuary. *J. Geophys. Res. Oceans* **2015**, *120*, 3499–3521. [[CrossRef](#)]
54. Guo, L.; Zhu, C.; Wu, X.; Wan, Y.; Jay, D.A.; Townend, I.; Wang, Z.B.; He, Q. Strong inland propagation of low-frequency long waves in river estuaries. *Geophys. Res. Lett.* **2020**, *47*, e2020GL089112. [[CrossRef](#)]