Heliyon 9 (2023) e17689

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Exploring machine learning algorithms for accurate water level forecasting in Muda river, Malaysia

Muhamad Nur Adli Zakaria [a], Ali Najah Ahmed [a,h,*], Marlinda Abdul Malek [b], Ahmed H. Birima [c], Md Munir Hayet Khan [d], Mohsen Sherif [e,f], Ahmed Elshafie [g]

[a] *Department of Civil Engineering, College of Engineering, Universiti Tenaga Nasional (UNITEN), 43000, Kajang, Selangor, Malaysia*
[b] *Cataclysmic Management and Sustainable Development Research Group (CAMSDE), Department of Civil Engineering, Kulliyyah of Engineering, International Islamic University Malaysia, Malaysia*
[c] *Department of Civil Engineering, College of Engineering, Qassim University, Unaizah, Saudi Arabia*
[d] *Faculty of Engineering & Quantity Surveying, INTI International University (INTI-IU), Persiaran Perdana BBN, Putra Nilai, Nilai, 71800, Negeri Sembilan, Malaysia*
[e] *Civil and Environmental Eng. Dept., College of Engineering, United Arab Emirates University, Al Ain, 15551, United Arab Emirates*
[f] *National Water and Energy Center, United Arab Emirates University, P.O. Box. 15551, Al Ain, United Arab Emirates*
[g] *Department of Civil Engineering, Faculty of Engineering, University of Malaya (UM), 50603, Kuala Lumpur, Malaysia*
[h] *Institute of Energy Infrastructure (IEI) , Universiti Tenaga Nasional (UNITEN), 43000, Selangor, Malaysia*

ARTICLE INFO

Keywords:
Water level
Machine learning
MLP
LSTM
XGBoost
Muda river
Malaysia

ABSTRACT

Accurate water level prediction for both lake and river is essential for flood warning and freshwater resource management. In this study, three machine learning algorithms: multi-layer perceptron neural network (MLP-NN), long short-term memory neural network (LSTM) and extreme gradient boosting XGBoost were applied to develop water level forecasting models in Muda River, Malaysia. The models were developed using limited amount of daily water level and meteorological data from 2016 to 2018. Different input scenarios were tested to investigate the performance of the models. The results of the evaluation showed that the MLP model outperformed both the LSTM and the XGBoost models in predicting water levels, with an overall accuracy score of 0.871 compared to 0.865 for LSTM and 0.831 for XGBoost. No noticeable improvement has been achieved after incorporating meteorological data into the models. Even though the lowest reported performance was reported by the XGBoost, it is the faster of the three algorithms due to its advanced parallel processing capabilities and distributed computing architecture. In terms of different time horizons, the LSTM model was found to be more accurate than the MLP and XGBoost model when predicting 7 days ahead, demonstrating its superiority in capturing long-term dependencies. Therefore, it can be concluded that each ML model has its own merits and weaknesses, and the performance of different ML models differs on each case because these models depends largely on the quantity and quality of data available for the model training.

* Corresponding author. Department of Civil Engineering, College of Engineering, Universiti Tenaga Nasional (UNITEN), 43000, Kajang, Selangor, Malaysia.

E-mail address: mahfoodh@uniten.edu.my (A.N. Ahmed).

https://doi.org/10.1016/j.heliyon.2023.e17689
Received 9 March 2023; Received in revised form 23 June 2023; Accepted 26 June 2023
Available online 29 June 2023
2405-8440/

1. Introduction

Accurate water level prediction for both lake and river is essential for flood warning and freshwater resource management [1]. The change in river level, like other hydrological phenomena, is complex because it involves many factors including rainfall, runoff, watershed and meteorological conditions [2]. In literature, many studies have been conducted utilizing various tools and methods to improve the quality and accuracy of water level prediction [3–5]. These approaches commonly fall into two broad categories – (1) physically based modeling which is process driven and (2) data driven approach using statistical or machine learning techniques. The physical or process-based models provide an in-depth analysis of the behavior of the catchment or simplified simulation of the hydrological process. However to build such models generally requires various input data regarding both hydrological process and the catchment geomorphology. The models also demand an extensive amount of computational efforts, time and sufficient expertise, which make them less efficient in cases where the concern is only to get an accurate prediction and underlying priori knowledge of the hydrological process is not required. Data-driven modeling on the other hand can model the relationship between the input data and the output variables without requiring extensive knowledge about the physical condition of the catchment. These models depend solely on historical observed data and are relatively more efficient in terms of computational cost and time compared to physically based models.

Over the last two decades, machine learning algorithms have been widely used in data driven hydrological modelling in general as an alternative to the conventional statistical modelling technique like the autoregressive integrated moving average (ARIMA) and autoregressive moving average (ARMA) models, which are also known as Box-Jenkins linear stochastic models. Machine learning models are proven to produce better results especially when dealing with high dimensional and non-linear relationships of hydrological variables [6–10]. The algorithms are more flexible, can handle more complex data, and can be used to develop better prediction models. Compared to the Box-Jenkins linear stochastic models, machine learning models such as neural network (NN), random forest (RF) and support vector machine (SVM) have the ability to model non-linear relationship and better at capturing the underlying patterns in the historical data and hence making more accurate predictions.

One of the most popular machine learning algorithms is artificial neural network (ANN) or simply known as neural network (NN). It has been widely applied in many different applications including the prediction of various hydrological variables [11–13]. A deep fully connected NN model with more than one layer is also known as multi-layer perceptron (MLP-NN). Over the years, MLP-NN has been proved to be a successful tool for river water level forecasting application [3,14,15].

Apart from that, there are various other algorithms derived from the basic feedforward neural network model such as the recurrent neural network (RNN) which is a class of NN tailored specifically for temporal data; and convolutional neural network (CNN) which is a regularized MLP-NN used specifically for image recognition applications. One of the latest and best performance NN is the long short-term memory (LSTM). LSTM is a special type of RNN that capable to process the long-term dependencies in data provided, which normal RNN failed to do. LSTM was introduced by Ref. [16] and has proved to be a powerful tool for addressing time-series prediction problems [17]. LSTMs, in comparison to traditional neural networks, can more accurately identify both periodic and chaotic patterns in time series data and learn their long-term dependencies [18]. Many studies have been conducted to apply LSTM to predict and forecast various hydrological variables [19–23]. For water level prediction application, LSTM has been applied to predict river level, groundwater level, lake level and reservoir water level [24]. applied LSTM to develop water level prediction models using of the upstream and downstream of Yeojubo in Gyeonggi-do, Republic of South Korea. They compared the LSTM prediction model with the one developed using gated recurrent unit (GRU) method. The results shows that the both LSTM model and GRU produces good prediction and can be used for flood management [25]. compared LSTM, deep neural network (DNN) and complex network models to predict water level in the Phan Rang River Basin of Nihn Thuan, Vietnam. They showed that all the models used including LSTM-based models provide good performance in predicting water level at the study location.

Another type of machine learning is tree-based models. Tree-based models are computationally cheap compared to other machine learning models. The most widely used tree-based models are classification and regression trees (CART) [26] and random forest (RF) [27–29] which is an ensemble tree-based algorithm. One of the latest and the best tree-based models is the extreme gradient boosting (XGBoost). It was first introduced by Ref. [30]. In general, XGBoost is a decision tree-based ensemble algorithm that utilizes gradient boosting framework. It leverages strengths from both decision tree and gradient boosting algorithms and uses additive training strategies to consider all the outputs of weak learners to create a strong learner. XGBoost offers a wide variety of tuning parameters for cross validation, regularization, and missing values; and almost consistently outperforms other algorithms in various applications. Some of the successful implementations of XGBoost modeling in hydrological applications include in water level prediction [31], river flow [32], wave height prediction [33] groundwater levels [34]. For water level prediction, the application of this algorithm is considered relatively new compared to NN and LSTM [34]. used XGBoost model to predict groundwater levels in Selangor, Malaysia and comparing it with ANN and Support Vector Regression (SVR) models. They concluded that XGBoost model provides more accurate prediction result and gives consistent performance and smaller error during training and testing of the model [31]. used XGBoost-based hybrid models to predict water level in the Jungrang urban basin in South Korea. The result shows that the XGBoost-based models outperformed other tree-based models in multistep-ahead water level prediction.

The use of machine learning has been proved to produce a simpler model with better prediction to serve as the alternative to the physically based and statistical based data driven models in water level prediction. Note that these black box models, as sophisticated as they can be, cannot fully replace the physically based model which are more robust and can provide more insight to the prediction. However, in some applications that only concern about getting accurate predictions in short time without the need to understand the underlaying hydrological process, machine learning models are the best option. Therefore, in the current study, three different ML algorithms – multilayer perceptron neural network (MLP-NN), the long short-term memory neural network (LSTM) and extreme gradient boosting (XGBoost) were utilized to develop daily water level prediction models in Muda River, Malaysia. The river is one of the main freshwater resources to Kedah and Penang, and crucial for the agriculture, industrial and domestic use for both states. An improved water level prediction could provide a better freshwater management plan for the local authorities and agencies, apart from flood mitigation plan since some parts of the river basin are prone to yearly flood.

As a case study, this paper intends to test the performance of the selected stand-alone ML models at the study location by using limited amount of data. These three models were selected as a benchmark for ML algorithms as these models were considered amongst the most popular ML algorithms. The current study also intends to explore the impact of adding meteorological data as an input to river level prediction models. Hence, different input combinations were used utilizing historical river water level and meteorological data at the study area. The performance of these models was evaluated using different set of performance metrics. Finally, the models' reliability was further evaluated in predicting the changes in the water level for different time horizons up to 7 days ahead. The contribution of the study includes the development of three widely used ML algorithms to predict river water level at the chosen study location including XGBoost algorithm which is relatively new and has only been applied in a few studies related to river level prediction. In addition, this study is conducted using limited amount of data. In data-driven models, extensive amount of data are proven to produce more accurate results. However, in some area with data scarcity and missing data, this will a limitation and hence this study intends to explore how good the selected models are in predicting the river water level with the mentioned limitation. In additional to that, the primary aim of this study is to explore the potential for predicting water levels even in the absence of precipitation and river flow discharge data.

This paper is structured as follows: Section 2 is dedicated for data and methodology. It describes the study area, dataset, brief introduction of the machine learning algorithms used and the model development strategy. The results are presented in Section 3, along with analysis and discussion. Section 4 is for conclusion of the study.

2. Data and methodology

2.1. Study area and data description

The Muda River is an important water source for the Kedah state in Malaysia, stretching 178 km and draining a total area of 4219 km^2. It originates from the Ulu Muda forest, a mountainous area near the border with Thailand, and flows westward, passing through the cities of Baling, Sik, Kulim and Kuala Muda, where the second largest city in the state, Sungai Petani, is located. The Muda River is a key water source for agricultural activities and water supply in Kedah, as well as the Penang state, which shares a border with Kedah. According to Ghani et al. (2010) [35], floods in the Muda River basin occur almost every year during the wet season, leading to

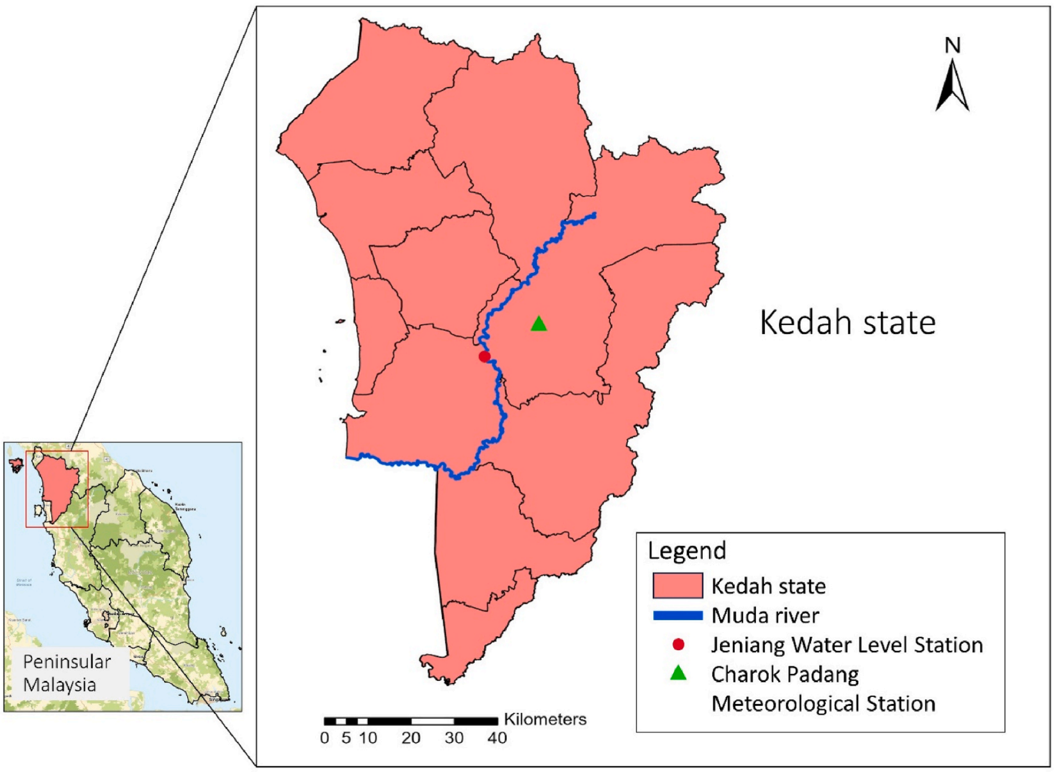

Fig. 1. Map showing the location of Jeniang Water Level Station at Sungai Muda and Charok Padang Meteorological Station, Kedah, Malaysia.

millions of ringgit in losses of paddy, the main agricultural activity in the basin area. Fig. 1 shows the location of the study area along with the water level and meteorological stations involves in the study.

In this study, daily water level and meteorological data from 2016 to 2018 was obtained from the Department of Irrigation and Drainage Malaysia (JPS) and Malaysia Meteorological Department (MetMalaysia) respectively. Table 1 presents a statistical summary of the dataset used in the present study. After preprocessing and imputing missing data using regression imputation method, the total amount of data available for modelling process is 1097 from 1 January 2016 to 31 December 2018. The whole data series was further split into training and testing dataset using 80/20 ratio. Further details on the data preprocessing and data split is discussed in **Section 2.5**.

2.2. Multi-layer perceptron neural network (MLP-NN)

The MLP-NN is a fundamental type of artificial neural network, made up of several layers (including an input layer, an output layer, and one or more hidden layers) of interconnected processing units, called neurons. This model is fully connected, meaning each neuron is linked to every neuron in the adjacent layers. It is a feed-forward network, meaning information flows from input to output in one direction only. The three primary factors that define the MLP-NN are its architecture (including the number of layers, neurons in each layer, and connections between them), the method used to determine connection weights, and the activation functions employed. Fig. 2 displays the MLP-NN's architecture. This model can be viewed as a function that maps an input vector x to an output y ,Eq. (1).

$$y = f_{MLP}(x) \tag{1}$$

Each neuron's input-output relationship in the hidden layer can be represented in Eq. (2) as:

$$y = g\left(\sum_j w_j x_j + b\right) \tag{2}$$

The j neuron's output from the preceding layer is denoted as x_j, while the link's weight between the present and j nodes is denoted as w_j. The bias of the current node is represented as b, and f denotes its activation function. Each hidden layer neuron calculates a weighted sum of its previous layer's inputs, along with a bias b. This value is then passed through an activation function g, resulting in an output for that particular neuron. The performance of the MLP-NN model is mostly dependent on its architecture, including the number of hidden layers and their neurons, and the activation functions used.

2.3. Long short-term memory (LSTM)

Long short-term memory (LSTM) [16] is a form of recurrent neural network (RNN), which, unlike the standard feed-forward neural network, contains inter-neuron loops. RNN includes a recurrent hidden unit that processes sequential data, with the output of each time step being used as the input for the subsequent time step. LSTM was designed specifically to address the vanishing gradient problem encountered in RNNs.

Fig. 3. (a) shows the schematic representation of LSTM algorithm structure and layers in addition to the input layer, fully connected layers and the output layer, LSTM network consists of one or more LSTM layers comprise memory cells that enable the network to determine when to forget previous hidden states and when to update hidden states with new data. The structure of an LSTM unit or cell is depicted in Fig. 3. (b). The LSTM unit is equipped with three primary gates: the input gate i_t, which regulates the flow of incoming information; the forget gate f_t, which controls the quantity of information retained from the previous memory state; and the output gate o_t, which governs the flow of outgoing data.

2.4. Extreme gradient boosting (XGBoost)

XGBoost was first proposed by Ref. [30]. It utilizes classification and regression tree (CART) to fit samples of training data. Each CART is associated with an autonomous decision rule for a binary tree, and each leaf node yields a predictive value. The output of the algorithm is the sum of the corresponding node values for a given input. The tree ensemble model used in XGBoost is trained in additive manner until stopping criteria are satisfied. Assuming that the model is composed of K CARTs, the model can be expressed in Eq. (3) as

Table 1
Statistical summary of the overall daily water level (Jeniang station) and meteorological data (Charok Padang station) used in this study.

Statistic	Jeniang Station	Charok Padang Station			
	Water level (m)	MSL Pressure (hPa)	Dry bulb temperature (°C)	Relative humidity (%)	Mean wind speed (m/s)
Minimum	20.06	1003.8	23.0	64	3.40
Maximum	25.20	1014.2	33.1	100	5.60
Mean	20.97	1009.0	27.50	94.95	4.47
Std. Deviation	0.67	1.57	1.51	5.17	0.31
Variance	0.44	2.46	2.27	26.72	0.09
Skewness	2.11	−0.36	0.69	−3.32	0.43

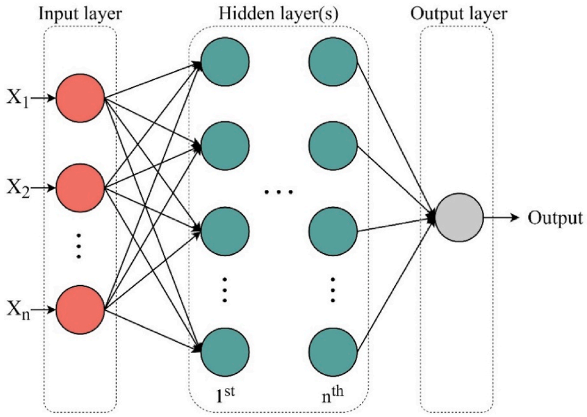

Fig. 2. MLP-NN architecture.

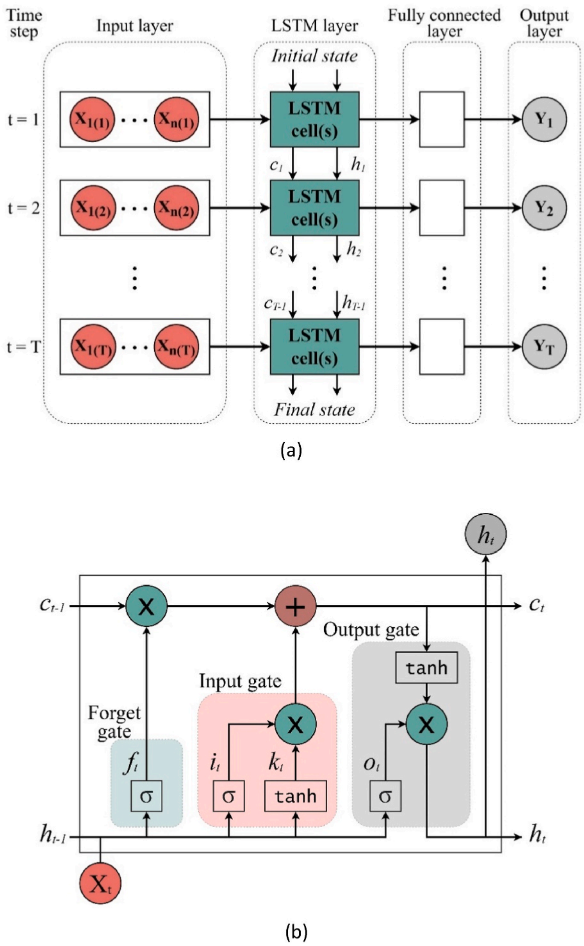

Fig. 3. (a) Schematic representation of LSTM algorithm structure and layers; **(b)** the structure of the LSTM cell: i_t, f_t, o_t are the input gate, forget gate and output gate, respectively; h_t is the cell output, C_t is the cell state.

follow:

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{3}$$

where $\widehat{y_i}$ represents the predicted value, f_k denotes an individual regression tree (specifically a Classification and Regression Tree, or CART), x_i represents the input vector, K represents the number of regression functions (i.e., the number of CARTs), and F represents the entire range of possible f_k s (i.e., the CART space). Like other machine learning techniques, the objective function involves the sum of the loss function and the regularization term. In XGBoost training, the regularized objective function presented below (Eqs. (4) and (5)) is minimized:

$$L(\varphi) = \sum_{i=1}^{n} l(y_i, \widehat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{4}$$

where

$$\Omega(\mathrm{f}) = \gamma\mathrm{T} + \frac{1}{2}\lambda||\omega||^2 \tag{5}$$

Here, l denotes the loss function for computing the difference between the actual value y_i and the predicted value $(\widehat{y_i})$, Ω serves as the regularization term for penalizing model complexity and mitigating overfitting, γ reflects the complexity of each leaf, T represents the number of leaves in a decision tree (or CART), λ is the parameter for balancing the penalty, and ω denotes the vector of scores for leaves. The model is trained in an additive manner. Let $\widehat{y_i}^{(t-1)}$ be the prediction of the i-th instance (i-th CART) at the t-th iteration, a new function f_t is added to minimize the following objective, :

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \widehat{y_i}^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \tag{6}$$

in order to streamline the optimization of Eq. (6), the loss function is expressed using the second-order Taylor series, as shown in Eq. (7):

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(c)}{2!}(x-a)^2 \tag{7}$$

By letting x be $L^{(t)} = \sum_{i=1}^{n} l(y_i, \widehat{y_i}^{(t-1)} + f_t(x_i)) + \Omega(f_t)$ and a be $\widehat{y_i}^{(t-1)}$, the objective function can be optimized as:

$$L^{(t)} \simeq \sum_{i=1}^{n}\left[l\left(y_i, \widehat{y_i}^{(t-1)} + g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right]\right. + \Omega(f_t) \tag{8}$$

where $g_i = \delta_{\widehat{y}^{(t-1)}} l(y_i, \widehat{y_i}^{(t-1)})$, and $h_i = \delta^2_{\widehat{y}^{(t-1)}} l(y_i, \widehat{y_i}^{(t-1)})$ are first and second order gradient statistics of the loss function, respectively. Eliminating the constant components in Eq. (8) produces the following simplified objective for step t:

$$\underline{L}^{(t)} = \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_t) \tag{9}$$

Let $I_j = \{i|q(x_i) = j\}$ as the instance set of leaf j, expanding Ω in **Eq. (9)** results in the followings:

$$\underline{L}^{(t)} = \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T}\omega_j^2 = \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)\omega_j + \frac{1}{2}\left(\sum_{i\in I_j} g_i + \lambda\right)\omega_j^2\right] + \gamma T \tag{10}$$

By computing the derivatives of Eq. (10) with respect to ω_t and equating them to zero gives the optimal weight $\omega*_t$ of leaf j as the following (Eq. (11)):

$$\omega*_t = \frac{\Sigma_{i\in I_j} g_i}{\Sigma_{i\in I_j} h_i + \lambda} \tag{11}$$

The ideal value can be computed as follows in Eq. (12):

$$\underline{L}^{(t)} = -\frac{1}{2}\sum_{j=1}^{T}\frac{\left(\Sigma_{i\in I_j} g_i\right)^2}{\Sigma_{i\in I_j} h_i + \lambda} + \gamma T \tag{12}$$

Given that I_L and I_R represent the sets of instances for the left and right nodes, respectively, following the split, and that $\mathrm{I} = I_L \cup I_R$, the loss reduction resulting from the split is determined by:

$$L_{split} = \frac{1}{2}\left[\frac{(\Sigma_{i\in I_L} g_i)^2}{\Sigma_{i\in I_L} h_i + \lambda} + \frac{(\Sigma_{i\in I_R} g_i)^2}{\Sigma_{i\in I_R} h_i + \lambda} - \frac{(\Sigma_{i\in I} g_i)^2}{\Sigma_{i\in I} h_i + \lambda}\right] - \gamma \tag{13}$$

in practice, Eq. (13) serves as the equation for calculating the loss reduction gain and is employed to evaluate potential split candidates.

2.5. Model development and hyperparameters tuning

In the current study, the development of water level prediction model is divided into three different stages as depicted in Fig. 5: (1) data preprocessing and preparation, (2) model training and testing and (3) model comparison. The aim of each model is to predict water level at time *t*, WL(*t*) using various input combinations that include the observations of different selected variables at previous time lags.

The first stage includes the data preprocessing, input selection, data normalization, data reshape and data partition as the part of the data preparation for the model training process. Input selections and combinations are important in machine learning models because they directly influence the model structure, thus helping the model in the training process to better understand the data and make more accurate predictions. By combining different variables, the model can gain insight into complex relationships between the variables. This can help the model to better identify patterns and hence make better predictions. Combining input variables can also help reduce the number of features the model has to consider, which can help improve its performance. In this study, based on the input variables, the models can be divided into two subcategories which is univariate and multivariate models. For univariate models, only water level is considered as the input along with its lag values while for multivariate models, different meteorological variables are considered in addition to water level, along with their selected lag values. To identify the number of lags to consider for each variable, correlation analysis was performed between their respective time series and water level time series. Fig. 4 shows partial autocorrelation function (PACF) correlogram of the water level time series in which the lag times have been chosen in this study. Based on the PACF correlogram, a considerable correlation can be seen up to lag 4. Accordingly, for univariate models, four combinations of input were considered. For multivariate models, cross correlation between each input variable and water level were considered. Table 2 shows the input combination used in this study.

Before applying the data to the model, the dataset is normalized in the range of 0 and 1 because the multivariate data comprises of different unit and ranges of values. This is also to ensure an effective network training process. Furthermore, in the current study, different ML model is expecting different format of input data – MLP-NN in 2D matrix, LSTM is expecting data in 3D matrix, and XGBoost in Dmatrix format. The whole dataset was further divided into training and testing dataset. Data split is important in machine learning model building to avoid the problem of overfitting. In the literature, there is no fix guideline regarding the data split. In the present study, 80/20 ratio is chosen following the common practice in literature [6,8], and it is also proven to be the one of the most optimum data split ratios.

The second stage in the model development process is the model training and testing. With regards to MLP-NN, different hidden layers were used in each of the proposed model as can be seen in Table 3. In this study Adam Optimizer is used because it is an adaptive learning rate method, which means that it uses the gradients of the current mini-batch to adjust the learning rate. It is computationally efficient and has been shown to work well in a wide range of deep learning applications. Adam Optimizer also has the benefit of being less sensitive to hyperparameter settings than other optimizers, making it a more reliable choice when training deep learning models.

The hyper-parameters of LSTM include the number of layers, the number of neurons in each layer, the type of activation functions used, the type of recurrent connections, the type of input/output connections, the learning rate, the dropout rate, and the regularization rate. These hyper-parameters can be tuned by using techniques such as grid search, random search, or evolutionary algorithms. Additionally, the hyper-parameters can also be tuned by manually adjusting them based on the results of model evaluation. Table 4 shows the optimum hyperparameters of the LSTM model obtained after the hyperparameter tuning for each input combination.

XGBoost algorithms provides large range of hyperparameters that can be tuned to improve the model performance as can be seen in

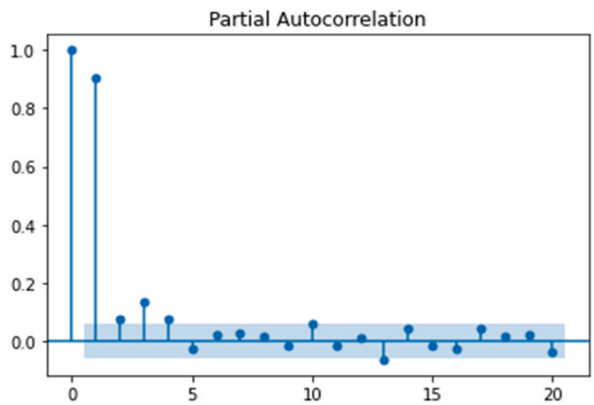

Fig. 4. Partial autocorrelation function (PACF) of the water level time series. Number of lag is in days.

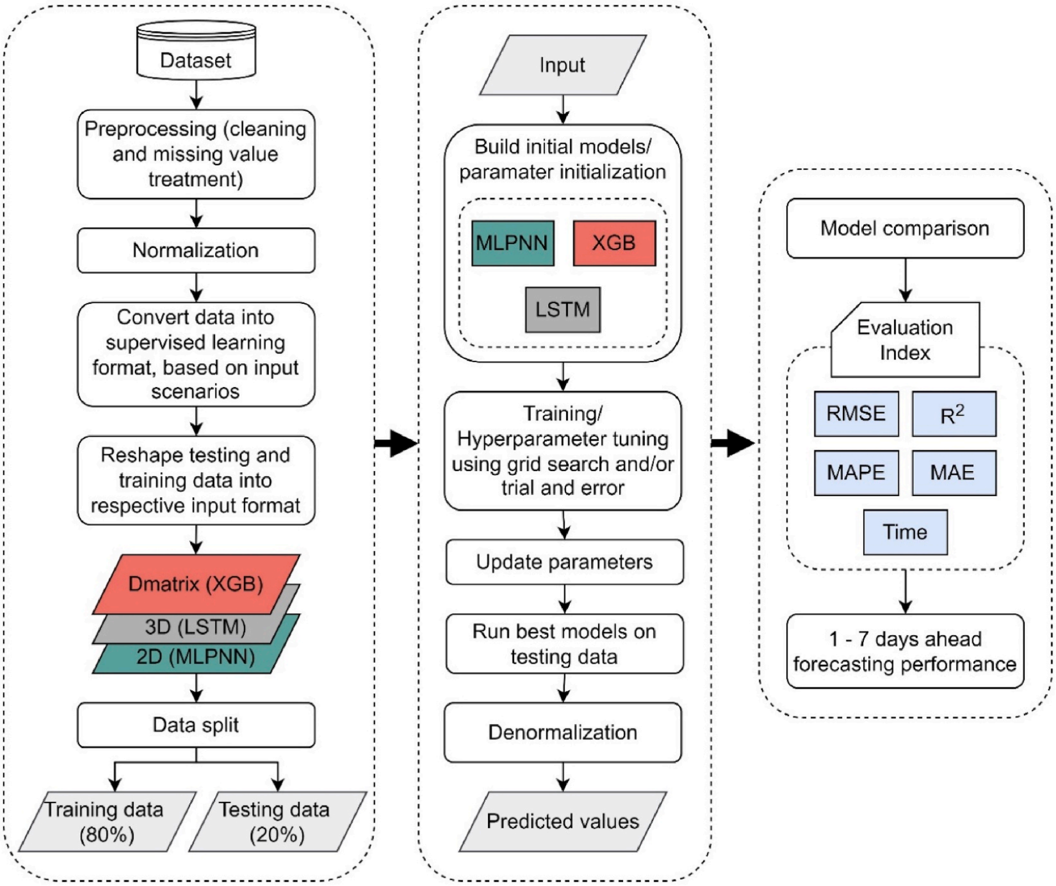

Fig. 5. Flow chart of the study.

Table 2
Input combination (WL: Water level, P: Pressure, T: Temperature, H: Relative humidity and WS: Mean wind speed).

Input combination		Lags	Input	Output
Univariate (water level only)	U1	1 lags	WL(t-1)	WL(t)
	U2	2 lags	WL(t-1), WL(t-2)	WL(t)
	U3	3 lags	WL(t-1), WL(t-2), WL(t-3)	WL(t)
	U4	4 lags	WL(t-1), WL(t-2), WL(t-3), WL(t-4)	WL(t)
Multivariate (water level & meteorological data)	M1	1 lags	WL(t-1), P(t-1), T(t-1), H(t-1), WS(t-1)	WL(t)
	M2	2 lags	WL(t-1), P(t-1), T(t-1), H(t-1), WS(t-1), WL(t-2), P(t-2), T(t-2), H(t-2), WS(t-2)	WL(t)
	M3	3 lags	WL(t-1), P(t-1), T(t-1), H(t-1), WS(t-1), WL(t-2), P(t-2), T(t-2), H(t-2), WS(t-2), WL(t-3), P(t-3), T(t-3), H(t-3), WS(t-3)	WL(t)
	M4	4 lags	WL(t-1), P(t-1), T(t-1), H(t-1), WS(t-1), WL(t-2), P(t-2), T(t-2), H(t-2), WS(t-2), WL(t-3), P(t-3), T(t-3), H(t-3), WS(t-3), WL(t-4), P(t-4), T(t-4), H(t-4), WS(t-4)	WL(t)
	M5	(selected lags based on PACF and CCF)	WL(t-1), WL(t-3), WL(t-2), P(t-11), T(t-11), H(t-7), WS(t-4)	WL(t)

Table 5. In this study, parameters tuning is conducted by using grid search in Python to test a range of values for each parameter, one at a time. For instance, at the beginning of the tuning process, to find the best value of Parameter A, several models were built using a range of values of Parameters A, while keeping other parameters constant. To tune the first parameter, the rest of the parameters were kept at their default values. For the next grid search, the parameters were updated based on the previous search results (i.e using the best value of the particular parameter).

2.6. Model performance indicator

To evaluate the performances of each model, the following metrics were used (Eq. 14–17:

Table 3
MLP-NN hyperparameters tuning results for different input combinations.

Input combination	Batch size	Epoch	Optimizer	Activation (hidden)	Activation (output)	Hidden neuron	Learning rate	Test RMSE
U1	32	250	adam	relu	Linear	15	0.001	0.327
U2	32	250	adam	relu	Linear	30	0.001	0.354
U3	32	250	adam	relu	Linear	30	0.001	0.334
U4	32	250	adam	relu	Linear	5	0.001	0.312
M1	32	250	adam	relu	Linear	4	0.001	0.321
M2	32	250	adam	relu	Linear	25	0.001	0.360
M3	32	250	adam	relu	Linear	5	0.001	0.346
M4	32	250	adam	relu	Linear	5	0.001	0.326
M5	32	250	adam	relu	Linear	11	0.001	0.350

Table 4
LSTM hyperparameters tuning results for different input combinations.

Input combination	LSTM layer	Unit	Batch	Optimizer	Learning rate	Test RMSE
U1	1	10	64	adam	0.001	0.319
U2	1	5	64	adam	0.001	0.361
U3	1	35	64	adam	0.001	0.320
U4	1	25	64	adam	0.001	0.321
M1	1	15	64	adam	0.001	0.340
M2	1	20	64	adam	0.001	0.362
M3	1	40	64	adam	0.001	0.353
M4	1	30	64	adam	0.001	0.352
M5	1	25	64	adam	0.001	0.360

Table 5
XGBoost hyperparameters tuning result for different input combinations.

Input combination	Max_depth	Min_child_weight	eta	subsample	colsample	Number of boost round	Test RMSE
U1	5	2	0.3	0.9	1.0	14	0.357
U2	5	6	0.1	1.0	1.0	112	0.383
U3	7	4	0.3	0.8	1.0	11	0.381
U4	9	4	0.3	0.8	1.0	8	0.378
M1	11	4	0.3	1	1	13	0.414
M2	5	4	0.2	1	1	19	0.411
M3	7	4	0.3	1	0.9	9	0.397
M4	9	1	0.2	0.8	0.9	18	0.354
M5	6	4	0.05	1	0.9	109	0.367

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(W_{o,i} - W_{f,i}\right)^2} \tag{14}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|W_{o,i} - W_{f,i}\right| \tag{15}$$

$$R^2 = \frac{\sum_{i=0}^{N}\left(W_{f,i} - W_{o,i}\right)^2}{\sum_{i=0}^{N}\left(W_{f,i} - \widehat{W}_{o,i}\right)^2} \tag{16}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{W_{f,i} - W_{o,i}}{W_{o,i}}\right| \tag{17}$$

where N is the total number of observations, $W_{o,i}$ and $W_{f,i}$ is the i th value of the observed and forecasted water level respectively. Root mean square error (RMSE) is a measure of how well a model predicts a given set of values. It measures the average difference between predicted values and the actual values. The lower the RMSE, the better the model is at predicting the data. Mean absolute error (MAE) is another measure of how well a model predicts a given set of values. It measures the average difference between predicted values and the actual values. The lower the MAE, the better the model is at predicting the data. Coefficient of determination (R^2) is a measure of how well a model explains the variance in a dataset. It measures how close the predicted values are to the actual values. The higher the

R^2, the better the model is at explaining the variance in the data. Mean absolute percentage error (MAPE) is a measure of how well a model predicts a given set of values. It measures the average difference between predicted values and the actual values as a percentage of the actual values. The lower the MAPE, the better the model is at predicting the data. The Friedman test is a non-parametric statistical test used to analyze differences between groups when the dependent variable is measured [36–38]. While the Friedman test has its advantages, it also has some limitations, one of which is that it requires a relatively larger sample size. Another constraint in the current study is the limited availability of data. Therefore, only the three mentioned indices were used in this study.

3. Results and discussion

3.1. Water level prediction model performance for 1-day ahead prediction

Table 6 shows the training and testing performances for MLP-NN models. Based on Table 6, for univariate models, adding number of lags as input decrease the performance and does not significantly improve the result except at U4 (input combination U4) where the best performance is obtained when using four lags as input with testing and training RMSE of 0.312 and 0.229 respectively. Similarly, for multivariate, adding number of lags as input decreases the result and the best performance is obtained when using one lag as input (input combination M1) with testing RMSE of 0.321 and training RMSE 0.226 for M4 respectively. Fig. 6 shows the forecasted water level of the MLP-NN models compare to the actual water level. From the figure, MLP-NN can predict the data patterns very well especially at the base water level. However, the models slightly underpredict the peak water level.

Table 7 shows the final training and testing performances for LSTM models. Based on Table 7, for univariate, adding number of lags as input does not significantly improve the result and the best performance is obtained when using only one lag as input (input combination U1) with testing and training RMSE of 0.319 and 0.233 respectively. Although the model performance has dropped at U2, it is noticeable that increase in performance for next lags at U3 & U4 are observed. For multivariate as well, adding number of lags as input does not significantly improve the result and the best performance is obtained when using one lags of variables as input (input combination M4) with testing and training RMSE of 0.340 and 0.229 respectively. Model M5 which is based on selected lags of various parameters which were selected based on partial autocorrelation and cross correlation analysis, did not performed poorer. Fig. 7 depicts the performance of LSTM models versus the actual water level.

Table 8 shows the results for XGBoost models. Based on Table 8, for univariate, adding number of lags as input does not significantly improve the result and the best performance is obtained when using only one lag as input (input combination U1) with testing and training RMSE of 0.357 and 0.203 respectively. Although the model performance has dropped at U2, it is noticeable that increase in performance for next lags at U3 & U4 are observed. On the contrary, for multivariate, adding number of lags as input does significantly improve the result and the best performance is obtained when using four lags as input (input combination M4) with testing and training RMSE of 0.354 and 0.088 respectively. Model M5 which is based on selected lags of various parameters based on PACF and CCF values, did not perform better than M4. In addition, it can be seen from Fig. 8 that all XGBoost models uncapable of capturing the extreme event.

Fig. 9(a–d) shows the comparison of forecasting performances of the three ML algorithms with different input combinations in term of RMSE, R^2, MAE and MAPE. From the figures, it can be seen that for univariate, the best performances found among MLP-NN, LSTM and XGBoost are respectively U4, U1 & U1 with RMSE values of 0.312, 0.319 and 0.357. Hence MLP-NN has the best performance of all. On the contrary, for multivariate input combinations best performances found for M1, M1 and M4 across MLP-NN, LSTM and XGBoost with RMSE values of 0.321, 0.340 and 0.354. Again MLP-NN has the best of all input combinations of multivariate models and univariate, U4 (using four lags as input) performed better than any other models developed. Hence, such technique will be vital for water level prediction of river which provides valuable information for multiple sectors, including disaster management, water resource planning, environmental conservation, and infrastructure development.

Time is an important factor when running machine learning models because it affects the accuracy and cost of the model. A longer running time can lead to a more accurate model, but it also increases the cost associated with the model. Additionally, longer running times can lead to slower response times for users interacting with the model. Finally, faster running times can allow for the model to be more frequently updated, which can lead to better performance. It can be seen from Table 9 that even though the lowest reported performance was reported by the XGBoost, it is the faster of the three algorithms followed by LSTM then MLP-NN.

Table 6
MLP-NN final model performance.

Input combination	Training				Testing				Time (s)
	RMSE	MAE	R^2	MAPE	RMSE	MAE	R^2	MAPE	
U1	0.231	0.144	0.798	0.007	0.327	0.191	0.858	0.009	17.657
U2	0.233	0.150	0.795	0.007	0.354	0.220	0.834	0.010	16.867
U3	0.230	0.155	0.800	0.007	0.334	0.211	0.852	0.010	17.475
U4	0.229	0.143	0.802	0.007	0.312	0.189	0.871	0.009	16.859
M1	0.232	0.141	0.797	0.007	0.321	0.182	0.863	0.008	16.961
M2	0.225	0.135	0.809	0.006	0.360	0.204	0.828	0.009	17.741
M3	0.231	0.141	0.798	0.007	0.346	0.193	0.841	0.009	17.023
M4	0.226	0.137	0.808	0.006	0.326	0.182	0.859	0.008	17.053
M5	0.230	0.135	0.801	0.006	0.350	0.198	0.839	0.009	17.133

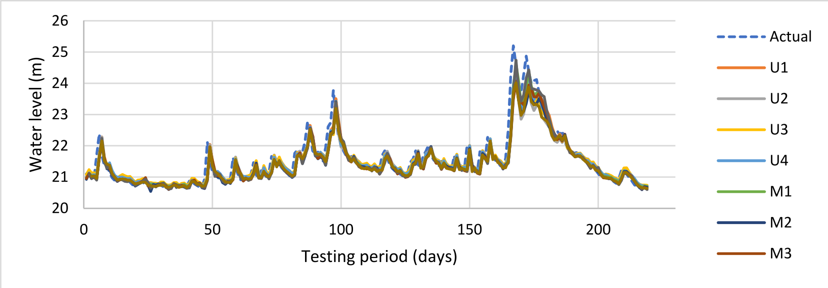

Fig. 6. Performance of MLP-NN models for 1-day ahead river level prediction.

Table 7
LSTM final model performance.

Input combination	Training				Testing				Time (s)
	RMSE	MAE	R^2	MAPE	RMSE	MAE	R^2	MAPE	
U1	0.233	0.141	0.794	0.007	0.319	0.178	0.865	0.008	16.255
U2	0.244	0.151	0.776	0.007	0.361	0.201	0.827	0.009	17.952
U3	0.231	0.140	0.799	0.007	0.320	0.185	0.864	0.008	21.957
U4	0.229	0.141	0.802	0.007	0.321	0.191	0.863	0.009	22.175
M1	0.229	0.139	0.802	0.007	0.340	0.195	0.847	0.009	16.802
M2	0.227	0.142	0.806	0.007	0.362	0.215	0.826	0.010	19.050
M3	0.225	0.142	0.809	0.007	0.353	0.212	0.834	0.010	21.054
M4	0.225	0.144	0.809	0.007	0.352	0.217	0.836	0.010	22.009
M5	0.230	0.140	0.801	0.007	0.360	0.214	0.829	0.010	17.693

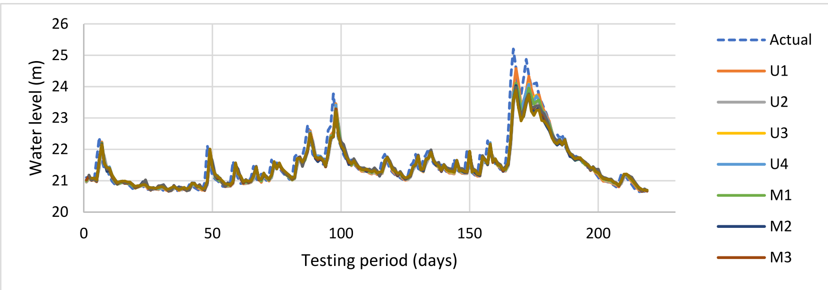

Fig. 7. Performance of LSTM models for 1-day ahead prediction.

Table 8
XGBoost final models performance.

Input combination	Training				Testing				Time (s)
	RMSE	MAE	R^2	MAPE	RMSE	MAE	R^2	MAPE	
U1	0.203	0.127	0.845	0.006	**0.357**	**0.210**	**0.831**	**0.009**	0.232
U2	0.184	0.110	0.872	0.005	0.383	0.226	0.805	0.010	0.863
U3	0.180	0.121	0.878	0.006	0.381	0.221	0.808	0.010	0.264
U4	0.209	0.166	0.835	0.008	0.378	0.244	0.811	0.011	0.448
M1	0.124	0.081	0.942	0.004	0.414	0.253	0.772	0.011	0.399
M2	0.161	0.109	0.902	0.005	0411	0.262	0.776	0.012	0.320
M3	0.168	0.129	0.893	0.006	0.397	0.263	0.791	0.012	0.319
M4	0.088	0.066	0.971	0.003	**0.354**	**0.220**	**0.834**	**0.010**	0.672
M5	0.136	0.087	0.931	0.004	0.367	0.225	0.823	0.010	1.119

3.2. Water level prediction up to 7-days-ahead

Finally, in order to evaluate the performance of LSTM and MLP-NN models, different time horizons up to seven days were utilized. The results indicated that the LSTM model performed better for up to seven-days ahead forecasting compared to the MLP-NN model as

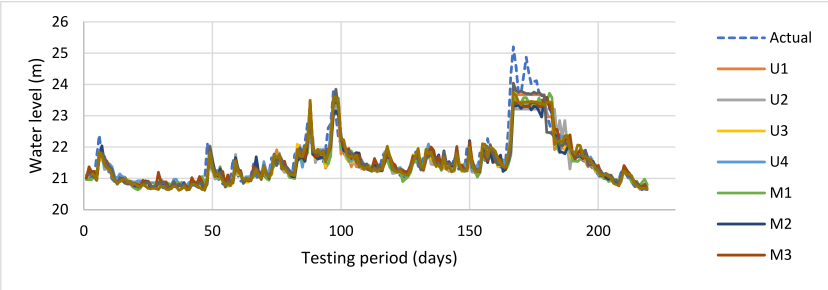

Fig. 8. Performance of XGBoost models for 1-day ahead prediction.

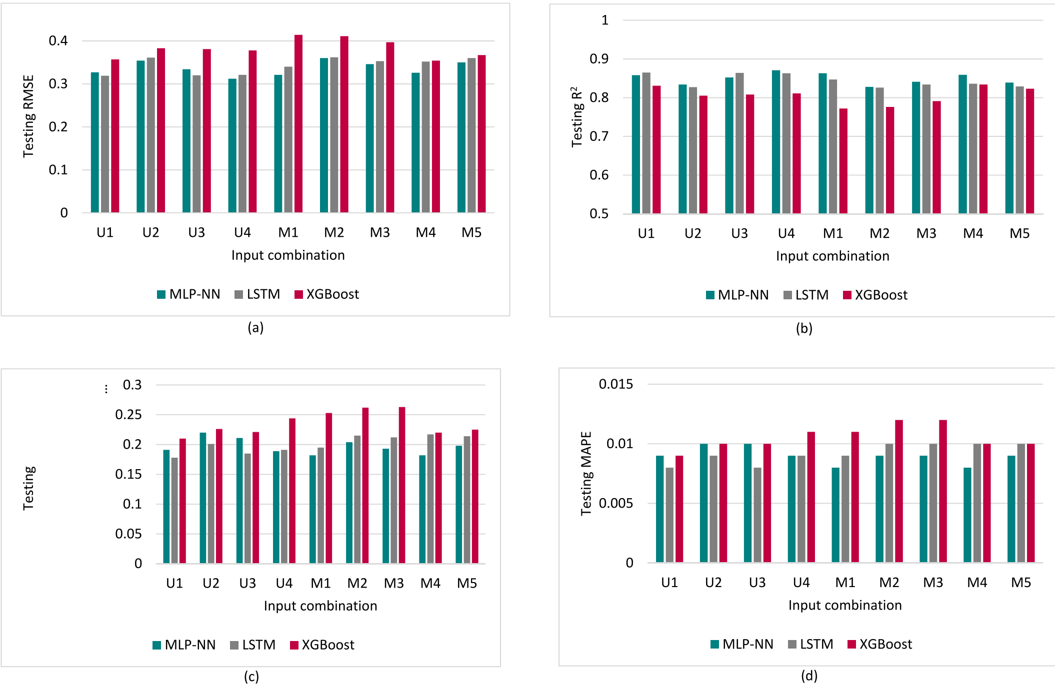

Fig. 9. MLP-NN, LSTM and XGBoost model comparison for 1-day ahead forecasting performance using different performance metrics: (a) testing RMSE; (b) testing R^2; (c) testing MAE; (d) testing MAPE.

Table 9
Time running for each of the proposed model.

	Time (s)								
Model	**U1**	**U2**	**U3**	**U4**	**M1**	**M2**	**M3**	**M4**	**M5**
MLP-NN	17.657	16.867	17.475	**16.859**	16.961	17.741	17.023	17.053	17.133
LSTM	**16.255**	17.952	21.957	22.175	16.802	19.05	21.054	22.009	17.693
XGBoost	0.232	0.863	0.264	0.448	0.399	0.32	0.319	0.672	**1.119**

can be seen in Fig. 10(a–d)].

The findings in this proved that LSTM can be reliable model in predicting water level with different time horizon and might robust as other proposed model in literature such as Radial Basis Function RBF [4], Boosted Decision Tree Regression (BDTR) [39]and Gaussian Process Regression GPR [40]. With such accuracy and reliability, LSTM can be used for real time water level forecasting. Such too will enable better decision-making, enhances safety, and contributes to the sustainable management of water resources and river ecosystems.

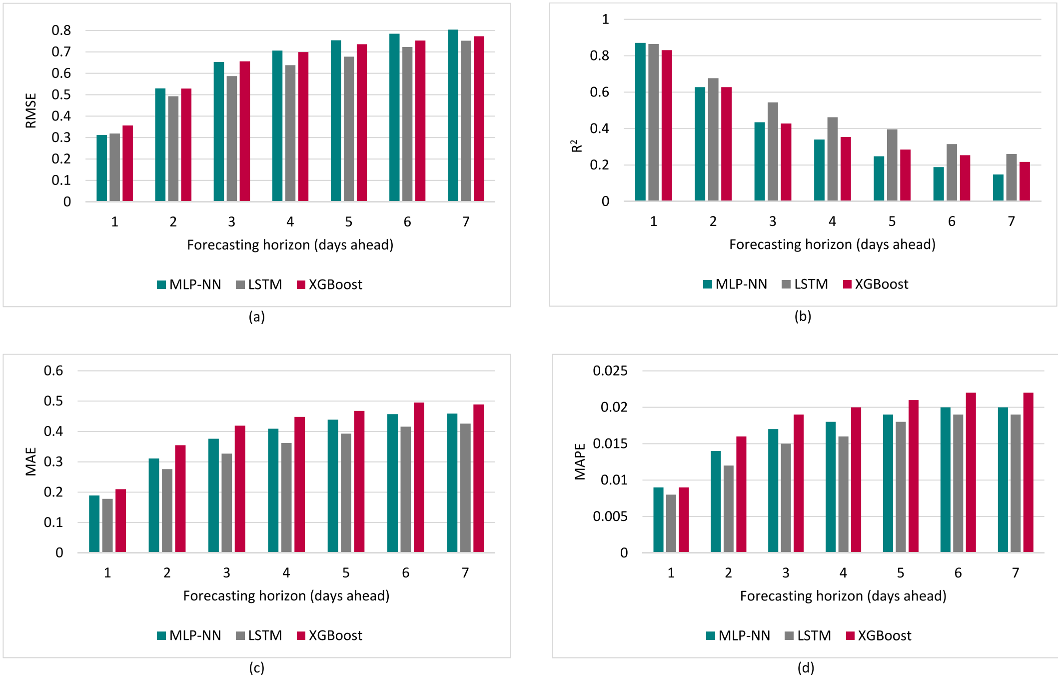

Fig. 10. Forecasting performance for MLP-NN, LSTM and XGBoost models across different forecasting horizon using different metrics: (a) RMSE; (b) R^2; (c) MAE; (d) MAPE.

4. Conclusion

Accurate water level forecasting is essential for flood warning and freshwater resource management, and in the current study, three machine learning algorithms (XGBoost, MLP and LSTM) were applied to develop water level forecasting models for Muda River, Malaysia. The models were tested using a limited amount of daily water level and meteorological data from 2016 to 2018. The results showed that the MLP model was the most accurate, with an overall accuracy score of 0.871, followed by the LSTM model (0.865) and XGBoost (0.831). No significant improvement was seen after incorporating meteorological data. Although XGBoost had the lowest reported performance, it was the fastest algorithm due to its advanced parallel processing capabilities and distributed computing architecture. The LSTM model was found to be more accurate in predicting 7 days ahead, indicating its superiority in capturing long-term dependencies. Thus, machine learning algorithms can be used to develop models for predicting water levels in rivers and streams. These models can be used to better understand the dynamics of water flow, how it will respond to changes in precipitation, and how it will affect water availability, water quality, and water-related infrastructure. In additional to that, it can provide insights into the most appropriate methods for predicting water levels, as well as inform decision-making by providing timely and accurate estimates of water levels. In the future, machine learning could be used to better understand the impact of climate change on water levels, as well as be used to predict extreme events and conditions, such as floods or droughts.

Funding

Not applicable.

Author contribution statement

Muhamad Nur Adli Zakaria: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Ali Najah Ahmed: Performed the experiments; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Marlinda Abdul Malek: Contributed reagents, materials, analysis tools or data; Conceived and designed the experiments; Wrote the paper.

Ahmed Hussein Birima: Analyzed and interpreted the data; Wrote the paper.

Md. Munir Hayet Khan: Performed the experiments; Analyzed and interpreted the data; Wrote the paper.

Mohsen Sherif: Contributed reagents, materials, analysis tools or data; Analyzed and interpreted the data; Wrote the paper.

Ahmed El-Shafie: Conceived and designed the experiments; Wrote the paper.

Data availability statement

Data will be made available on request.

Additional information

No additional information is available for this paper.

Declaration of competing 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.

Acknowledgements

This research was supported by the Ministry of Education (MOE) through Fundamental Research Grant Scheme (FRGS/1/2020/TK0/UNITEN/02/16). The authors would like to acknowledge their access to data from the Department of Irrigation and Drainage Malaysia (JPS) and the Malaysia Meteorological Department (MetMalaysia).

References

[1] A. Hipni, A. El-shafie, A. Najah, O.A. Karim, A. Hussain, M. Mukhlisin, Daily forecasting of dam water levels: comparing a support vector machine (SVM) model with adaptive neuro fuzzy inference system (ANFIS), Water Resour. Manag. 27 (10) (Aug. 2013) 3803–3823, https://doi.org/10.1007/s11269-013-0382-4.
[2] M. Sapitang, W.M. Ridwan, K. Faizal Kushiar, A. Najah Ahmed, A. El-Shafie, Machine learning application in reservoir water level forecasting for sustainable hydropower generation strategy, Sustainability 12 (15) (2020), https://doi.org/10.3390/su12156121, 6121, Jul.
[3] A.N. Ahmed, et al., Water level prediction using various machine learning algorithms: a case study of Durian Tunggal river, Malaysia, Engineering Applications of Computational Fluid Mechanics 16 (1) (Dec. 2022) 422–440, https://doi.org/10.1080/19942060.2021.2019128.
[4] M.N.A. Zakaria, M.A. Malek, M. Zolkepli, A.N. Ahmed, Application of artificial intelligence algorithms for hourly river level forecast: a case study of Muda River, Malaysia, Alex. Eng. J. 60 (4) (Aug. 2021) 4015–4028, https://doi.org/10.1016/j.aej.2021.02.046.
[5] Q.B. Pham, B. Mohammadi, R. Moazenzadeh, S. Heddam, R.P. Zolá, A. Sankaran, V. Gupta, I. Elkhrachy, K.M. Khedher, D.T. Anh, Prediction of lake water-level fluctuations using adaptive neuro-fuzzy inference system hybridized with metaheuristic optimization algorithms, Appl. Water Sci. 13 (1) (2023) 13.
[6] M. Zounemat-Kermani, et al., Neurocomputing in surface water hydrology and hydraulics: a review of two decades retrospective, current status and future prospects, J. Hydrol. (Amst.) 588 (Sep. 2020) 125085, https://doi.org/10.1016/j.jhydrol.2020.125085. January.
[7] S. Zhu, H. Lu, M. Ptak, J. Dai, Q. Ji, Lake water-level fluctuation forecasting using machine learning models: a systematic review, Environ. Sci. Pollut. Control Ser. 27 (36) (Dec. 2020) 44807–44819, https://doi.org/10.1007/s11356-020-10917-7.
[8] Sarmad Dashti Latif, Ali Najah Ahmed, Streamflow prediction utilizing deep learning and machine learning algorithms for sustainable water supply management, Water Resour. Manage. (2023) 1–15.
[9] S. Karimi, J. Shiri, Ö. Kisi, T. Xu, Forecasting daily streamflow values: assessing heuristic models, Nord. Hydrol 49 (3) (2018) 658–669, https://doi.org/10.2166/nh.2017.111.
[10] S. Dazzi, R. Vacondio, P. Mignosa, Flood stage forecasting using machine-learning methods: a case study on the parma river (Italy), Water (Basel) 13 (12) (2021), https://doi.org/10.3390/w13121612, 1612, Jun.
[11] Z. He, X. Wen, H. Liu, J. Du, A comparative study of artificial neural network , adaptive neuro fuzzy inference system and support vector machine for forecasting river flow in the semiarid mountain region, J. Hydrol. (Amst.) 509 (2014) 379–386, https://doi.org/10.1016/j.jhydrol.2013.11.054.
[12] Ö. Kisi, A.M. Nia, M.G. Gosheh, M.R.J. Tajabadi, A. Ahmadi, Intermittent streamflow forecasting by using several data driven techniques, Water Resour. Manag. 26 (2012) 457–474, https://doi.org/10.1007/s11269-011-9926-7.
[13] H. Badrzadeh, R. Sarukkalige, A.W. Jayawardena, Impact of multi-resolution analysis of artificial intelligence models inputs on multi-step ahead river flow forecasting, J. Hydrol. (Amst.) 507 (Dec. 2013) 75–85, https://doi.org/10.1016/j.jhydrol.2013.10.017.
[14] S. Alvisi, G. Mascellani, M. Franchini, A. Bardossy, Water level forecasting through fuzzy logic and artificial neural network approaches, Hydrol. Earth Syst. Sci. 10 (2006) 1–17.
[15] T. Zhou, Z. Jiang, X. Liu, K. Tan, Research on the long-term and short-term forecasts of navigable river's water-level fluctuation based on the adaptive multilayer perceptron, J. Hydrol. (Amst.) 591 (May) (2020), https://doi.org/10.1016/j.jhydrol.2020.125285, 125285, Dec.
[16] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.
[17] Y. Hu, L. Yan, T. Hang, J. Feng, Stream-Flow Forecasting of Small Rivers Based on LSTM, 2020.
[18] S. Mouatadid, J.F. Adamowski, M.K. Tiwari, J.M. Quilty, Coupling the maximum overlap discrete wavelet transform and long short-term memory networks for irrigation flow forecasting, Agric. Water Manag. 219 (Jun. 2019) 72–85, https://doi.org/10.1016/j.agwat.2019.03.045. August 2018.
[19] R. Qiu, et al., River water temperature forecasting using a deep learning method, J. Hydrol. (Amst.) 595 (Apr. 2021) 126016, https://doi.org/10.1016/j.jhydrol.2021.126016. December 2020.
[20] C. Hu, Q. Wu, H. Li, S. Jian, N. Li, Z. Lou, Deep learning with a long short-term memory networks approach for rainfall-runoff simulation, Water (Basel) 10 (11) (2018), https://doi.org/10.3390/w10111543, 1543, Oct.
[21] F. Kratzert, D. Klotz, C. Brenner, K. Schulz, M. Herrnegger, Rainfall-runoff modelling using long short-term memory (LSTM) networks, Hydrol. Earth Syst. Sci. 22 (11) (2018) 6005–6022, https://doi.org/10.5194/hess-22-6005-2018.
[22] X.-H. Le, H.V. Ho, G. Lee, S. Jung, Application of long short-term memory (LSTM) neural network for flood forecasting, Water (Basel) 11 (7) (2019), https://doi.org/10.3390/w11071387, 1387, Jul.
[23] I.-F. Kao, Y. Zhou, L.-C. Chang, F.-J. Chang, Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting, J. Hydrol. (Amst.) 583 (Apr. 2020) 124631, https://doi.org/10.1016/j.jhydrol.2020.124631. January.
[24] M. Cho, C. Kim, K. Jung, H. Jung, Water level prediction model applying a long short-term memory (LSTM)–Gated recurrent unit (GRU) method for flood prediction, Water (Basel) 14 (14) (2022), https://doi.org/10.3390/w14142221, 2221, Jul.
[25] D. Kim, H. Han, W. Wang, H.S. Kim, Improvement of deep learning models for river water level prediction using complex network method, Water (Basel) 14 (3) (2022) 466, https://doi.org/10.3390/w14030466. Feb.
[26] H.I. Erdal, O. Karakurt, Advancing monthly streamflow prediction accuracy of CART models using ensemble learning paradigms, J. Hydrol. (Amst.) 477 (Jan. 2013) 119–128, https://doi.org/10.1016/j.jhydrol.2012.11.015.
[27] L. Schoppa, M. Disse, S. Bachmair, Evaluating the performance of random forest for large-scale flood discharge simulation, J. Hydrol. (Amst.) 590 (September) (2020), https://doi.org/10.1016/j.jhydrol.2020.125531, 125531, Nov.

[28] M. Abbasi, A. Farokhnia, M. Bahreinimotlagh, R. Roozbahani, A hybrid of Random Forest and Deep Auto-Encoder with support vector regression methods for accuracy improvement and uncertainty reduction of long-term streamflow prediction, J. Hydrol. (Amst.) 597 (2021), https://doi.org/10.1016/j.jhydrol.2020.125717, 125717, Jun.

[29] Q.B. Pham, D.A. Tran, N.T. Ha, A.R. Islam, R. Salam, Random forest and nature-inspired algorithms for mapping groundwater nitrate concentration in a coastal multi-layer aquifer system, J. Clean. Prod. 343 (2022) 130900.

[30] T. Chen, C. Guestrin, XGBoost, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, Aug. 2016, pp. 785–794, https://doi.org/10.1145/2939672.2939785.

[31] D.H. Nguyen, X. Hien Le, J.Y. Heo, D.H. Bae, Development of an extreme gradient boosting model integrated with evolutionary algorithms for hourly water level prediction, IEEE Access 9 (2021) 125853–125867, https://doi.org/10.1109/ACCESS.2021.3111287.

[32] L. Ni, et al., Streamflow forecasting using extreme gradient boosting model coupled with Gaussian mixture model, J. Hydrol. (Amst.) 586 (2020), 124901, https://doi.org/10.1016/j.jhydrol.2020.124901. March.

[33] H. Hu, A.J. van der Westhuysen, P. Chu, A. Fujisaki-Manome, Predicting Lake Erie wave heights and periods using XGBoost and LSTM, Ocean Model. 164 (2021), https://doi.org/10.1016/j.ocemod.2021.101832. May.

[34] A. Ibrahem Ahmed Osman, A. Najah Ahmed, M.F. Chow, Y. Feng Huang, A. El-Shafie, Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia, Ain Shams Eng. J. 12 (2) (Jun. 2021) 1545–1556, https://doi.org/10.1016/j.asej.2020.11.011.

[35] A.A. Ghani, R. Ali, N.A. Zakaria, Z. Abu Hasan, C.K. Chang, M.S.S. Ahamad, A temporal change study of the Muda River system over 22 years, Int. J. River Basin Manag. 8 (1) (2010), https://doi.org/10.1080/15715121003715040.

[36] M.S.M. Kasihmuddin, S.Z.M. Jamaludin, M.A. Mansor, H.A. Wahab, S.M.S. Ghadzi, Supervised learning perspective in logic mining, Mathematics 2022 10 (6) (Mar. 2022) 915, https://doi.org/10.3390/MATH10060915. Page 915, vol. 10.

[37] N.E. Zamri, S.A. Azhar, M.A. Mansor, A. Alway, M.S.M. Kasihmuddin, Weighted random k satisfiability for k=1,2 (r2SAT) in discrete hopfield neural network, Appl. Soft Comput. 126 (Sep. 2022) 109312, https://doi.org/10.1016/J.ASOC.2022.109312.

[38] S.S.M. Sidik, N.E. Zamri, M.S.M. Kasihmuddin, H.A. Wahab, Y. Guo, M. Asyraf Mansor, Non-Systematic weighted satisfiability in discrete hopfield neural network using binary artificial bee colony optimization, Mathematics 2022 10 (7) (Apr. 2022) 1129, https://doi.org/10.3390/MATH10071129. Page 1129, vol. 10.

[39] M. Sapitang, W.M. Ridwan, K. Faizal Kushiar, A. Najah Ahmed, A. El-Shafie, Machine learning application in reservoir water level forecasting for sustainable hydropower generation strategy, Sustainability 12 (15) (2020), https://doi.org/10.3390/su12156121, 6121, Jul.

[40] A.N. Ahmed, et al., Water level prediction using various machine learning algorithms: a case study of Durian Tunggal river, Malaysia, Engineering Applications of Computational Fluid Mechanics 16 (1) (2022) 422–440, https://doi.org/10.1080/19942060.2021.2019128.