Available online at www.sciencedirect.com
ScienceDirect
ICT Express 8 (2022) 588–594
www.elsevier.com/locate/icte

Predicting river water height using deep learning-based features

Punyanuch Borwarnginn[a], Jason H. Haga[b], Worapan Kusakunniran[a,*]

[a] Faculty of Information and Communication Technology, Mahidol University, Nakhon Pathom, Thailand
[b] Digital Architecture Research Center, National Institute of Advanced Industrial Science and Technology, Tsukuba, Ibaraki, Japan

Received 9 August 2021; received in revised form 25 January 2022; accepted 25 March 2022
Available online 1 April 2022

Abstract

The paper presents the river height prediction model using real-world historical sensor data such as rainfall, cumulative rainfall, and river water heights. The study evaluates using a Support Vector Regression, a Long Short-Term Memory, and a combination of a Long Short-Term Memory as the feature extraction and a support vector regression. Through experiments, various future predictions are tested, including a few hours or a day. As expected, RNN achieved the lowest error, but it could not capture rapid changes in river height levels. In comparison, the LSTM-SVR can better represent rapid transient changes in the data by using nonlinear kernels.

Keywords: Water level prediction; Deep learning; Support Vector Regression; LSTM; Feature extraction

1. Introduction

Natural disasters, such as flooding, earthquakes, and landslides, can cause significant losses in terms of economic and structural damage, as well as loss of life [1]. It would require a disaster detection system that possibly provides an early warning to affected areas to reduce these losses. However, monitoring natural disasters is a challenging task. It requires extensive data and reliable methods to analyze and forecast the events. For example, river water heights are intuitively related to the amount of rain if we could associate these data with creating a prediction model.

One specific type of disaster detection that requires a robust and optimal prediction model is for river flooding events. Over the past decades, several works have proposed using probabilistic prediction techniques in hydrology and water resource management. Artificial Intelligence techniques such as Support Vector Machine (SVM), Artificial Neural Networks (ANN), and Multilayer Perceptron (MLP) have been applied to specific tasks such as flood forecasting [2], snow level forecasting [3,4] and water level prediction [5]. Support Vector Machine (SVM) [6] approach has been widely used for classification tasks. Due to its promising results, SVM has the potential to be applied in regression tasks called Support Vector Regression (SVR) [7]. SVM has high potential because of its kernel functions [5]. That can help the SVM find a global optimal solution and allow the data to be mapped in high dimensional space.

Hipni et al. [5] adopted SVM regression to forecast the daily dam water level in the Klang gate, Malaysia. To select the best model, they use four categories including the input scenario, type of SVM, the number of cross-validation and the time lag. The results showed that using both the rainfall $R(t-i)$ and the dam water level $L(t-i)$ was the best input scenario. Type 2 or ν-SVM [8] performed the best results with 5-fold cross-validation. Finally, the best time lag was using the best model with $R(t-2)L(t-2)$ with a 1.64% error rate. The results were compared with the Adaptive Neuro Fuzzy Inference System (ANFIS) [9]. The comparison results reported that their SVM model achieved a higher accuracy. Buyukyildiz et al. [10] investigated five machine learning techniques, Artificial Neural Networks (ANN), Support Vector Regression (SVR), Multilayer Perceptron (MLP), Radial Basis Neural Networks and Adaptive Network Based Fuzzy Inference System (ANFIS). The objective is to estimate monthly water level change in Lake Beysehir, Turkey and they concluded that the SVR model obtained the best estimation. Zhang et al. [11] tried three different techniques, including ANN, SVM and ANFIS for short-term water level prediction

* Correspondence to: Faculty of Information and Communication Technology, Mahidol University, 999 Phuttamonthon 4 Road, Salaya, Nakhon Pathom, 73170, Thailand.
E-mail address: worapan.kun@mahidol.edu (W. Kusakunniran).
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2022.03.012
2405-9595/

in Yaojiang, China. The study showed that ANFIS could avoid information noise regardless of the time lag inputs, however, SVM was stable during rapid changes, such as in the case of typhoon events, leading to peak values of water levels.

In recent years, deep learning techniques such as Convolutional Neural Network (CNN) and Recurrent neural network (RNN) were introduced in water monitoring systems. However, a Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) widely used in prediction tasks, particularly in the time series model. Sahoo et al. [12] investigated the use of LSTM-RNN for low-flow hydrological time series at Basantapur station in the Mahanadi River, India. The study showed that LSTM-RNN outperformed both RNN and naive methods. LSTM was also applied in flood forecasting in [13]. The proposed model forecasted discharge at Hoa Binh Station on the Da River, Vietnam using the flow rate and rainfall from several meteorological and hydrological stations before the Hoa Binh Station was built. The results showed that LSTM achieved over 86% of accuracy for one-day, two-day, and three-day forecasting. Sankaranarayanan et al. [14] proposed Deep Neural Network for flood occurrence using temperature and rainfall intensity. Ren et al. [15] adopted Multilayer Perceptron (MLP) and Recurrent Neural Network (RNN) and built two models for a water level prediction in a target of 2 to 6 h ahead. The experiment results showed that the RNN model was more effective than MLP and achieved 97.05% accuracy.

In this paper, we propose a prediction model using real-world historical data related to river flooding events from several sensors. It is a challenging problem because we do not control the sensor equipment or data acquisition. Therefore, we aim to demonstrate that this imbalanced and noisy data can be used to train prediction models, including traditional techniques and deep learning approaches (RNN, feature extraction, SVR, etc.) and generate a better understanding of how real-world data impacts the performance of these models. As the results demonstrate, RNN achieved the lowest error. However, it cannot capture rapid changes in river height levels. Whereas the combination of RNN and SVR achieved slightly higher error but can better represent and report the rapid transient changes in the data.

The rest of this paper is organized as follows. Section 2 explains the proposed method of predicting river water heights. Section 3 describes the experiments. Then, the results are discussed in Section 4. Finally, the conclusions are drawn in Section 5.

2. Proposed method

The proposed method consists of three main tasks: data acquisition, model training, and prediction. Data acquisition includes the cleansing and preparation of data from real-world public sensors, which will be used for our model's training and test set. The model training process involves feature extraction and classification. We apply the trained model to the test set for the prediction task and evaluate the prediction results. The details of each task are explained in the following sub-sections.

2.1. Data acquisition

In this study, we propose to use real-world data for predicting river water height. The dataset was collected from a Japanese river information website [16] that provides data related to rivers across the country, including such information as river water heights, rainfall levels, cumulative rain levels, snowfall levels, and live video streams from CCTV. We acquired approximately a month of data from 22 November to 20 December 2016. Based on our analysis, it shows that river water height and rainfall are associated, with their corresponding time stamps. An important feature to note is that there are rapid temporal changes in river water height if there is heavy rain. Our sample data shows that river water heights are also affected by a rainfall level and a cumulative rainfall. Therefore, these historical data are used as the inputs to the model with the corresponding historical river water height data at the same point in time. Our goal is to predict a river water height as far into the future as possible for practical use while maintaining accurate predictions. For example, if the model can predict only 10 to 30 min into the future, that may be insufficient for users to understand the information, make decisions, and take actions. These data will be split into different time intervals, such as one hour, six hours, or a day to investigate how effectively the model predicts in different future intervals.

2.2. Prediction model

Because the data contains temporal information and we aim to predict river heights within a specific future time period, this can be viewed as a regression problem. In this study, we investigate three methods, which are conventional techniques for this type of problem: 1) predicting using support vector regression (SVR), 2) deep learning approach using recurrent neural network (RNN), and 3) combination of both techniques by using deep learning for feature extraction and SVR for predicting the result. The first two are conventional techniques for this type of problem, but the third is a modified conventional approach of our own design.

2.2.1. Support Vector Regression (SVR)

In the conventional approach, Support Vector Machine (SVM) [6] is widely used in classification problems and has been successfully applied in many computer vision applications such as face detection, handwriting recognition, and medical image analysis. Therefore, there is an attempt to generalize SVMs for regression problems into Support Vector Regression (SVR). SVR uses kernels and other parameters similar to SVM to control the number of Support Vectors (SV) [17]. SVR has the same characteristics as SVM. SVR can capture nonlinear data using nonlinear kernels. Polynomial and radial basis function kernel (RBF) are defined as in Eq. (1) and (2), respectively.

$$k(x_i, x_j) = (x_i^T x_j + 1)^d \tag{1}$$

$$k(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \tag{2}$$

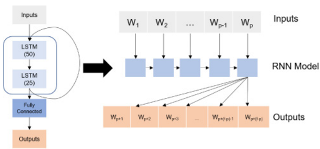


Fig. 1. Our proposed LSTM model and its unfolded model.

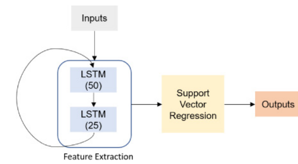


Fig. 2. Our proposed LSTM-SVR model.

where, x_i and x_j are feature vectors in input space, and d is the degree of polynomial. We use SVR for finding a function to estimate the river water height. The input data are nonlinear and have rapid temporal changes. The SVR nonlinear functions such as polynomial and RBF would potentially be able to capture these patterns. SVR is trained using 3 data inputs (river water height, rainfall, cumulative rainfall) that are formed into vectors by time intervals.

2.2.2. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are an extension of RNNs, which are types of neural networks with an internal memory. Therefore, it allows the outputs of the previous time to be used as the input of the current time. RNNs can be viewed as a chain-link network. The structure contains a loop that repeats the same task to all data across the sequence of inputs vector and the output from the previous computation. Therefore, RNNs have been applied in sequential data such as machine translation [18], financial data [19], and regression problems [20,21]. Although RNNs can remember the previous information and have a short memory, there are two typical issues using RNNs. During training, RNNs keep adjusting gradients during backpropagation in many steps. It usually causes exploding problems and gradient vanishing [22]. Long Short-Term Memory was introduced to solve the vanishing gradient problem. LSTM extends the internal memory of RNNs to be able to give longer time lags during training. It extends RNNs with hidden units called the memory cell, allowing RNNs to remember inputs for a long time, similar to computer memory. In LSTM, a memory cell consists of three gates: an input gate, a forget gate, and an output gate [23]. The input gate can determine whether to allow new input, which is similar to read in and write to the memory. The forget gate decides which information is to be deleted from the memory block or uses it to determine outputs at the current timestep in the output gate. The LSTM can be computed as an iteration of all timesteps using the following equations:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (3)$$

$$f_t = \sigma(W_i[h_{t-1}, x_t] + b_f) \quad (4)$$

$$o_t = \sigma(W_i[h_{t-1}, x_t] + b_o) \quad (5)$$

$$\hat{C}_t = \sigma(W_i[h_{t-1}, x_t] + b_c) \quad (6)$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \hat{C}_t \quad (7)$$

$$h_t = o_t \otimes tanh(C_t) \quad (8)$$

where, x_i is the current input at the current time step t, h_{t-1} and C_{t-1} are the hidden state and cell state from the previous timestep, i_t is the input gate, f_t is the forget gate, o_t is the output gate, $\hat{C}_t$ is the internal cell state vector for generating the cell state C_t, h_t is the hidden state and W is weights and b bias.

In this work, we aim to predict the river water height as far as possible into the future while maintaining accurate predictions. It is a time series problem or a regression problem because water heights at future times depend on the historical information of river water heights and rainfall levels. The overview of the proposed model is shown in Fig. 1. The model consists of 2 layers of stacked LSTM networks on 50 hidden nodes and 25 hidden nodes. Given historical data of p hours, we want to predict f hours. All water heights W from the first hour to p hours are used as the inputs to the model, which returns the predicted river water heights W from the next hour $p+1$ to f hours into the future.

2.2.3. Combination of LSTM and SVR

This study proposes the LSTM model as inputs to the support vector regression (SVR). During our proposed LSTM model experiment, we found that due to the rapid temporal changes of water heights over a more extended period, LSTM with the fully connected layer could not capture these changes accurately. While this LSTM model attempts to average between the data points, it still helps identify and extract patterns in other regression techniques. In the conventional techniques, SVR can derive a mapping function of complex nonlinear feature space using RBF and polynomial kernels. Therefore, our proposed LSTM-SVR is shown in Fig. 2. The model uses the same two stacked LSTM layers but without the fully connected layer as the feature extraction. The outputs of LSTM are subsequently input to the SVR to produce the prediction results.

2.3. Evaluation method

Since our model is a regression problem, we use two types of loss functions for evaluating prediction models. These are mean absolute error (MAE) and root mean square error (RMSE). The mean absolute error (MAE) is an average of the absolute error for all data points. The MAE function is defined as below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (9)$$

where y_i is the predicted water level at time i, $\hat{y}_i$ is the actual level at time i, and n is the number of times in 10 min intervals. The root mean square error is the square root of an average of the error difference squared. It is also applied to our evaluation because it is sensitive to outlier values and can be used for selecting the best model. The mathematical function of RMSE is as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{10}$$

3. Experiments

3.1. Dataset

According to the previous section, we acquired data from the Japanese River Website [16], which includes various data from different types of sensors related to river information. We selected historical data of river water heights, instantaneous rainfall levels, and cumulative rain levels. The data also contain river height sensor data, and its corresponding rainfall levels at the closest rain monitoring station for each river. These data are processed and used for training and testing the model.

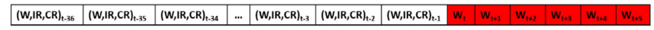
All data are recorded at 10 min intervals. Therefore, there are 6 data points each hour for each sensor. River water height is measured in meters (m). The instantaneous rainfall at a current time is expressed in millimeters (mm), whereas the cumulative rainfall records the accumulated total of rainfall during the start period up to the current period in millimeters (mm). We have approximately a month of these data. After data cleansing and validating, the training set consists of 20 days of data (2,880 data samples), and the remaining data becomes the test set with approximately five days of data (688 data samples). All datasets are normalized and processed into a different vector size in order to train the model. Because of the limited dataset, model overfitting is possible, however, after inspection of the data, we found that all data variables are in similar distributions.[1]

Each model is trained with three features: river water heights, instantaneous rainfall levels, and cumulative rain levels. The feature dimension is based on the period of historical data for each iteration. Each input label is the actual river water heights of the next interval.

3.2. Experiment settings

Our experiments are conducted based on the objective of investigating the furthest predicting period of the model and how much historical information is sufficient in order to give a promising result. Therefore, we set up various scenarios based on different historical data sets and different output periods. The examples of scenario are as follows:

- Given 6 h of historical data and predict 1 h ahead: Since we are using three features, river water height (W), instantaneous rainfall level (IR), cumulative rain level (CR), to predict the next hour ahead. The input feature vector will contain 36 data values from the previous timesteps, and the label will be six water heights for each subsequent timestep. All training samples are processed into these forms, as in Fig. 3(a). Given a current time step (t), historical data of six hours is represented in concatenating all features from t-1 to t-36 and its actual six river water height timesteps (1 h). Therefore, the total training input dimension is [2844, 36, 3] with their size of [2844, 36].
- Given 12 h of historical data and predict 6 h ahead: The input feature vector will contain 72 data values from the previous timesteps, and the label will be 36 river water heights for each subsequent timestep. Therefore, the total training input dimension is [2808, 72, 3] with their label size of [2808, 36], as shown in Fig. 3(b).



(a) Diagram depicting historical data of six hours to predict one hour

(b) Diagram depicting historical data of twelve hours to predict six hours

Fig. 3. Dimensions of a training vector, given current time step (t), historical data is represented by concatenating all features from previous time step (t) to desired timestep (t-d) and actual water height timesteps as its label vector.

Table 1 summarizes the experiment scenarios that we have conducted. For each historical interval, we generated four different periods of prediction outputs: an hour, six hours, 12 h, and 24 h. We individually train three techniques, LSTM, SVR, and the combination of LSTM and SVR, for each time interval. Therefore, 16 models are trained, resulting in a total of 32 models created. LSTM models are implemented using the Tensorflow library with a batch size of 50 and run for 50 epochs. With the combination of LSTM and SVR, the trained LSTM model is used as the feature extraction by dropping the fully connected layer and connecting the output to SVR. The SVR model will take LSTM features as inputs with their corresponding labels. The SVR models are individually trained with three kernels: linear, polynomial, and radial basis function (RBF). The total training features dimension is composed of (1) the total number of input data (n), (2) each of the previous time steps in 10 min intervals (t) and (3) the number of input features (f). In our scenario, there are three features: river water heights, instantaneous rainfall level, and cumulative rain level. The label dimension is the corresponding river water height (w) for each of the previous time steps (t).

4. Results and discussion

As explained in the Experiments section, we trained 32 scenarios for each method and time interval. Our goal was to find a suitable time interval that the prediction model could predict with the optimal error rate. After inspection of the results, we selected the results from six hours and twelve hours to present here for the sake of brevity and clarity. The mean absolute error (MAE) and root mean square error (RMSE) are reported in Table 2. The LSTM model achieves the lowest error rate compared with other techniques for all

[1] Due to space constraints, sample data, exploratory data analysis, and more experimental results are available at https://github.com/punborwarnginn/waterheight_supplements#readme.

Table 1
Different input set up for each experiment.

History data (h)	Input dimension [n,t,f]	Label dimension [w,t]			
		1 h	6 h	12 h	24 h
1	[2874,6,3]	[2874,6]	[2874,36]	[2874,72]	[2874,144]
6	[2844,36,3]	[2844,6]	[2844,36]	[2844,72]	[2844,144]
12	[2808,72,3]	[2808,6]	[2808,36]	[2808,72]	[2808,144]
24	[2736,72,3]	[2736,6]	[2736,36]	[2736,72]	[2736,144]

Table 2
Error results of 6 h and 12 h models.

MAE(m)	Prediction interval							
	6 h				12 h			
Historical data	LSTM	LSTM +SVR			LSTM	LSTM +SVR		
		Linear	RBF	Polynomial		Linear	RBF	Polynomial
6 h	0.012 ± 0.004	0.012 ± 0.007	0.017 ± 0.007	0.019 ± 0.001	0.018 ± 0.005	0.023 ± 0.009	0.024 ± 0.008	0.020 ± 0.004
12 h	0.013 ± 0.003	0.018 ± 0.008	0.016 ± 0.006	0.019 ± 0.002	0.018 ± 0.005	0.022 ± 0.010	0.023 ± 0.008	0.020 ± 0.005

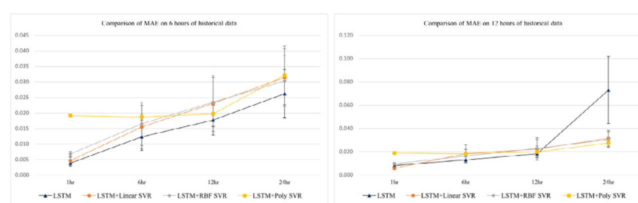
(a) Mean absolute error

RMSE(m)	Prediction interval							
	6 h				12 h			
History data	LSTM	LSTM +SVR			LSTM	LSTM +SVR		
		Linear	RBF	Poly		Linear	RBF	Poly
6 h	0.016 ± 0.004	0.018 ± 0.008	0.019 ± 0.007	0.024 ± 0.002	0.024 ± 0.005	0.029 ± 0.013	0.028 ± 0.010	0.027 ± 0.007
12 h	0.018 ± 0.004	0.025 ± 0.007	0.020 ± 0.010	0.025 ± 0.001	0.024 ± 0.006	0.027 ± 0.011	0.027 ± 0.008	0.025 ± 0.005

(b) Root mean square error

prediction intervals. Although, our proposed method using a trained LSTM as a feature input for an SVR model had a slightly higher average error rate than the LSTM itself, the polynomial SVR model using the previous 6 h of data had the least amount of variation in its results, as demonstrated by having the smallest standard deviation, compared to the other models. Fig. 4 reports the MAE results from using the same 6 h and 12 h of historical data to train four different prediction outputs, one hour, six hours, twelve hours, and 24 h (one day) ahead. We generally found that the future prediction interval should be no greater than the time interval for the historical data for training otherwise the error rate will increase.

Furthermore, the characteristics of our dataset contain some rapid temporal changes in the river water heights. MAE and RMSE cannot accurately capture this information because they evaluate the performance by averaging the difference between the actual and predicted values. Hence, we examine the actual values that returned from the model. Fig. 5 shows example results of predicting 12 h ahead given 12 h of historical data on both the LSTM model and the combination of LSTM with SVR. Although LSTM returns a better error rate, it attempts to average all the points instead of following rapid changes of river water heights. Meanwhile, the combination of LSTM with SVR using the RBF kernel function can capture these temporal patterns. In addition, Table 2 reports the results from training each previous time step of the LSTM model. The results show that selecting the prediction interval should be less than or equal to the previous time interval. For example, if we want to predict 6 h ahead, the training data should be from the preceding 6 h or less. Although the current dataset is limited, the model successfully predicted river water height with minimal error. Further enhancement of the proposed model can be achieved by extending the dataset to cover different time periods and seasons.

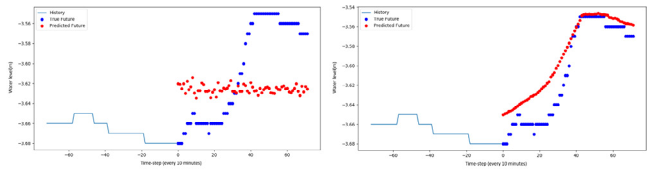


(a) 6 hours (b) 12 hours

Fig. 4. The comparison of MAE for each future prediction interval using the previous 6 h (a) and 12 h (b) of data.

5. Conclusion

This paper proposes a prediction model for river water height using a deep learning feature. Our goal is to use the actual historical data from related sensors such as river water heights, rainfall levels, cumulative rain levels. These data contain some rapid temporal changes, which is a challenging task for the model to predict. Our goal is to predict the river water height as far as possible into the future to simulate the



(a) LSTM (b) LSTM with SVR-RBF

Fig. 5. The 12 h predicted water heights compared with their actual values given 12 h of historical data.

practical use of the system. Hence, we created a total of 32 scenarios for training the model to discover the effect of using different historical data intervals on the prediction accuracy of river water heights in several future time periods. We examined three possible techniques using LSTM, SVR, and a combination of LSTM with SVR. We achieved the following contributions: (1) While the LSTM model achieved a small error rate, it was not able to capture rapid temporal changes in the data but rather acted as a smoothing function. Using an LSTM as the features input for nonlinear SVR achieves better prediction results on these data. (2) As a best practice, the historical data used for training should be from a time interval that is greater than or equal to the prediction time period. (3) In this research work, the actual units of measuring river water height and rainfall are meters and millimeters, respectively. Given the magnitude of the RMSE and MAE is very small, the real-world impact of this could be negligible for rainfall and small for river water height (about +/- 2 cm). As seen in Fig. 5, the river water height changed by more than 10 cm over 2 h, far greater than the intrinsic error rates. Although the proposed method of using LSTM combined with SVR has a slightly larger error rate than using LSTM solely, we believe that capturing the transient events in the data is more critical in real-world scenarios to provide a better assessment of emerging dangers to the public.

CRediT authorship contribution statement

Punyanuch Borwarnginn: Conceptualization, Methodology, Investigation, Validation, Writing – original draft. **Jason H. Haga:** Supervision, Conceptualization, Methodology, Writing – review & editing. **Worapan Kusakunniran:** Supervision, Conceptualization, Methodology, Writing – review & editing.

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.

Acknowledgments

This research project was partially supported by the Faculty of Information and Communication Technology, Mahidol University, Thailand, and the ICT International Collaboration Fund of the National Institute of Advanced Industrial Science and Technology, Thailand.

References

[1] D. Alexander, Natural Disasters, Routledge, 2018, http://dx.doi.org/10.4324/9781315859149.

[2] D.T. Nguyen, S.-T. Chen, Real-time probabilistic flood forecasting using multiple machine learning methods, Water 12 (3) (2020) 787, http://dx.doi.org/10.3390/w12030787.

[3] P. Borwarnginn, W. Kusakunniran, P. Pooyoi, J.H. Haga, Segmenting snow scene from CCTV using deep learning approach, ECTI Trans. Comput. Inform. Technol. 13 (2) (2019) 167–175, http://dx.doi.org/10.37936/ecti-cit.2019132.216323.

[4] Y. Zhan, J. Wang, J. Shi, G. Cheng, L. Yao, W. Sun, Distinguishing cloud and snow in satellite images via deep convolutional network, IEEE Geosci. Remote Sens. Lett. 14 (10) (2017) 1785–1789, http://dx.doi.org/10.1109/LGRS.2017.2735801.

[5] 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) (2013) 3803–3823, http://dx.doi.org/10.1007/s11269-013-0382-4.

[6] R. Fernandes de Mello, M. Antonelli Ponti, R. Fernandes de Mello, M. Antonelli Ponti, Introduction to support vector machines, Mach. Learn. (2018) http://dx.doi.org/10.1007/978-3-319-94989-5_4.

[7] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. (2004) http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88.

[8] P.H. Chen, C.J. Lin, B. Schölkopf, A tutorial on v-support vector machines, Appl. Stoch. Models Bus. Ind. 21 (2) (2005) 111–136, http://dx.doi.org/10.1002/asmb.537.

[9] N. Valizadeh, A. El-Shafie, Forecasting the level of reservoirs using multiple input fuzzification in ANFIS, Water Resour. Manag. 27 (9) (2013) 3319–3331, http://dx.doi.org/10.1007/s11269-013-0349-5.

[10] M. Buyukyildiz, G. Tezel, V. Yilmaz, Estimation of the change in lake water level by artificial intelligence methods, Water Resour. Manag. 28 (13) (2014) 4747–4763, http://dx.doi.org/10.1007/s11269-014-0773-1.

[11] S. Zhang, L. Lu, J. Yu, H. Zhou, Short-term water level prediction using different artificial intelligent models, in: 2016 5th International Conference on Agro-Geoinformatics, Agro-Geoinformatics 2016, Institute of Electrical and Electronics Engineers Inc., 2016, http://dx.doi.org/10.1109/Agro-Geoinformatics.2016.7577678.

[12] B.B. Sahoo, R. Jha, A. Singh, D. Kumar, Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting, Acta Geophys. 67 (5) (2019) 1471–1481, http://dx.doi.org/10.1007/s11600-019-00330-1.

[13] H. Le, J. Lee, Application of long short-term memory (LSTM) neural network for flood forecasting, Water 11 (7) (2019) 1387, http://dx.doi.org/10.3390/w11071387.

[14] S. Sankaranarayanan, M. Prabhakar, S. Satish, P. Jain, A. Ramprasad, A. Krishnan, Flood prediction based on weather parameters using deep learning, J. Water Clim. Change 11 (4) (2019) 1766–1783, http://dx.doi.org/10.2166/wcc.2019.321, arXiv:https://iwaponline.com/jwcc/article-pdf/11/4/1766/831101/jwc0111766.pdf.

[15] T. Ren, X. Liu, J. Niu, X. Lei, Z. Zhang, Real-time water level prediction of cascaded channels based on multilayer perception and recurrent neural network, J. Hydrol. 585 (2020) 124783, http://dx.doi.org/10.1016/j.jhydrol.2020.124783.

[16] Disaster prevention information of river URL http://www.river.go.jp/kawabou/ipTopGaikyo.do.

[17] M. Awad, R. Khanna, Support vector regression, in: Efficient Learning Machines, A Press, Berkeley, CA, 2015, pp. 67–80, http://dx.doi.org/10.1007/978-1-4302-5990-9_4.

[18] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014, http://dx.doi.org/10.3115/v1/d14-1179, arXiv:1406.1078.

[19] S. Selvin, R. Vinayakumar, E.A. Gopalakrishnan, V.K. Menon, K.P. Soman, Stock price prediction using LSTM, RNN and CNN-sliding window model, in: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 1643–1647, http://dx.doi.org/10.1109/ICACCI.2017.8126078.

[20] M. Abdel-Nasser, K. Mahmoud, Accurate photovoltaic power forecasting models using deep LSTM-RNN, Neural Comput. Appl. 31 (7) (2019) 2727–2740, http://dx.doi.org/10.1007/s00521-017-3225-z.

[21] H. Su, E. Zio, J. Zhang, M. Xu, X. Li, Z. Zhang, A hybrid hourly natural gas demand forecasting method based on the integration of wavelet transform and enhanced deep-RNN model, Energy 178 (2019) 585–597, http://dx.doi.org/10.1016/j.energy.2019.04.167.

[22] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[23] A. Graves, Supervised Sequence Labelling with Recurrent Neural Networks, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.