Advances in Water Resources 142 (2020) 103656

Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres

Combining statistical machine learning models with ARIMA for water level forecasting: The case of the Red river

Thi-Thu-Hong Phan[a,*], Xuan Hoai Nguyen[b,c,*]

[a] Department of Computer Science, Faculty of Information Technology, Vietnam National University of Agriculture, Hanoi, Vietnam
[b] Faculty of Information Technology, Ho Chi Minh University of Technology (HUTECH), VietNam
[c] AI Academy, Hanoi, Vietnam

ARTICLE INFO

Keywords:
Water level forecasting
Hybrid model
Statistical machine learning
ARIMA
Long-term univariate time series

ABSTRACT

Forecasting water level is an extremely important task as it allows to mitigate the effects of floods, reduce and prevent disasters. Physically based models often give good results but they require expensive computational time and various types of hydro-geomorphological data to develop the forecasting system. Alternatively, data driven forecasting models are usually faster and easier to build.

During the past decades, statistical machine learning (ML) methods have greatly contributed to the advancement of data driven forecasting systems that provide cost-effective solutions and better performance. Meanwhile, Autoregressive integrated moving average (ARIMA) is one of the famous linear statistical models for time series forecasting. In this paper, we propose a hybrid approach that takes advantages of linear and nonlinear models. The proposed method combines statistical machine learning algorithms and ARIMA for forecasting water level.

Observed water level of the Red river at the Vu Quang, Hanoi (3 hourly sampled from 2008 to 2017) and Hung Yen hydrological stations (hourly collected data from 2008 to 4/2015) are used to evaluate the performance of different methods. Experimental results on these 3 real big datasets show the effectiveness of our proposed hybrid models.

1. Introduction

Accurate forecasting of water level in rivers and lakes is essential for warning floods and water resource management. Water level data obtained from hydrological stations usually have a time series structure, hence researchers often use time series hydrological prediction models to forecast future data. Using past data to predict future water level (future behavior), hidden information can be disclosed which is of great significance for mitigating the effects of floods, reducing or preventing disasters and managing water resource.

In general, there are two lines of approaches for building the water level forecasting models that are process based numerical prediction and data-driven. The process based models (HEC-RAS (Dang and De Smedt, 2017), MIKE-11 (Patro et al., 2009; Kumar et al., 2019), MIKE-21 (Zhu et al., 2013; Tran Anh et al., 2018, etc.) usually provide precise results and fully describe the nature of physical phenomena. However, they are computationally expensive requiring various types of hydro-geomorphological data such as topography, geology, conduction roughness, cross-section (Fatichi et al., 2016), to name but a few to develop forecasting systems. They also demand sufficient expertise to interpret the results and require huge volumes of data which might be hard to satisfy in the case of water level and flow (Di et al., 2014; Fatichi et al., 2016; Zhong et al., 2017a). An alternative way for building forecasting models is to use data-driven approaches (Riad et al., 2004; Wu et al., 2009; Ghumman et al., 2011; Peng et al., 2017; Gjika et al., 2019). Data driven forecasting models aim to reveal relationship between features or hidden information in the data using (mostly) only information from available data. They are often faster and easier to build and therefore useful for real-time rivers/lakes flow forecasting with accurate water level prediction. Data driven models are mainly built from data using statistical and machine learning techniques.

Hydrological data are usually affected by numerous external factors, which make their time series contain both linear and nonlinear behaviors (components). Autoregressive integrated moving average (ARIMA) is one of the most popular and successful linear statistical models for time series forecasting (Ghimire, 2017; Birylo et al., 2018). Meanwhile, statistical machine learning (ML) methods have been widely applied in building forecasting models for nonlinear time series (Wu et al., 2009; Peng et al., 2017; Yaseen et al., 2018; Gjika et al., 2019). Combining these two techniques for the sake of better time series forecasting is sensible and, in fact, there have been numerous studies in

* Corresponding authors.
E-mail addresses: ptthong@vnua.edu.vn (T.-T.-H. Phan), nx.hoai@hutech.edu.vn (X.H. Nguyen).

https://doi.org/10.1016/j.advwatres.2020.103656
Received 4 December 2019; Received in revised form 3 June 2020; Accepted 5 June 2020
Available online 9 June 2020

the literature on such hybrid approach giving an affirmative answer (Chen and Wang, 2007; Khashei and Bijari, 2010; Yan and Zou, 2013; Wongsathan and Jaroenwiriyapap, 2016; Pannakkong et al., 2017; Zhong et al., 2017a; Peng et al., 2017; Mohan and Reddy, 2018; Yaseen et al., 2018; Mousavi-Mirkalaei and Banihabib, 2019; Temr et al., 2019; Xie and Lou, 2019). The success of these hybrid models in different fields of application has inspired us to utilize this approach for building water level forecasting models and test it on the case of the Red river. In particular, we propose a data driven hybrid model building process with two components for modeling linear and non-linear parts of water level time series using ARIMA and non-parametric statistical learning. We hypothesize that by using two types of models for capturing linear and non-linear parts in water level time series, hidden patterns in the time series data could be better revealed compared to the single type of model approach. We shall test and empirically prove our hypothesis by solving the water level forecasting problem for the Red river with three real big datasets obtained from three hydrological stations.

The rest of this paper is organized as follows: The next section overviews the related works in the literature. Sections 3 describes in details our proposed method. Data representation, pre-processing, evaluation metrics, and experiment results are presented and discussed in Section 4. The last section (section 5) concludes the paper and highlights some possible future works.

2. Related works

ARIMA is one of the most well-known and effective linear statistical models for time series forecasting. In (Birylo et al., 2018), the authors utilized ARIMA model to predict ground water level requiring three parameters: precipitation, surface runoff and evapotranspiration. Prediction results for twelve months showed that ARIMA models obtained a good performance. Similarly, Ghimire (2017) developed ARIMA model to forecast the river discharges in the US with significant success. Work in Mirzavand and Ghazavi (2015) showed the successful application of five time series models: moving-average (MA), auto regressive moving average (ARMA), ARIMA and seasonal ARIMA (SARIMA) and combination of several time series models to forecast groundwater level. Notably, the experiment results indicated that combining time series models really improved the accuracy of ground water level forecasts. Valipour et al. (2013) made a comparison of the predictive performance of ARMA, ARIMA and the autoregressive artificial neural networks (ANN) in forecasting the monthly inflow of Dez dam reservoir. The results clearly showed that ARIMA model is better than ARMA in the 12 past months but is not as good as autogressive ANN for the past 60 months. Yu et al. (2017) investigated ARIMA model to predict daily water level of three stations in the middle reaches of the Yangtze river. It is found that the accuracy of the ARIMA model decreases as the forecasting horizon increases, ie. it gives good results for the short-term predictions but not for long-term water level forecasts. The authors also pointed out that the nonlinear and non-stationary nature of the time series may lead to the uncertainty of using directly and only ARIMA model. Therefore, it is necessary to combine different types of models to improve the predictive performance.

In the recent years, (non-parametric) statistical machine learning (ML) methods have greatly contributed to the advancement of forecasting systems that provide cost-effective solutions with satisfying performance using historical time series of water level data. Nguyen et al. (2015) compared the performance of three statistical machine learning models LASSO, Random Forest (RF) and Support Vector Regression (SVR) for forecasting water level of the Mekong river. SVR produced good results with mean absolute error as 0.486(m) for 5-lead-day (acceptable error for a flood forecasting model is in $[0.5m, 0.75m]$). Garcia et al. (2016) investigated RF algorithm to predict the water level on two different stations of Cagayan River basin in Philippines. The correlations between predicted water level and ground truth data indicated the favorable predictive performance of the approach and suggested that it can be implemented for other stations on the major river basins across the Philippines. Pasupa and Jungjareantrat (2016) made prediction of water level on Chao Phyra river in Thailand using a number of different statistical machine learning methods, namely, linear regression (LR), Kernel regression (KL), SVR, K-nearest neighbor (KNN), and Random Forest (RF). The SVR model with radial basis function kernel and 72-hour past time series data generated the best prediction results with the least error and better than the previous approach used by the Royal Thai Navy. In other works, SVR also demonstrated its ability to predict river flows (Garsole and Rajurkar, 2015; Adnan et al., 2018; Bafitlhile and Li, 2019). Yang et al. (2017) conducted forecasts for water level time series on Taiwans Shimen reservoir using five statistical machine learning methods (RBF network, Kstar, KNN, RF, and Random Tree). The experimental results showed that RF improved forecasting performance over the other methods. RF also achieved better results than other statistical machine learning methods such as SVM, Artificial Neural Networks (ANNs), decision tree (DT) when predicting daily water levels (Wang et al., 2018; Choi et al., 2019). Hipni et al. (2013) applied SVM to forecast the daily dam water level of the Klang gate in Malaysia in comparison with Adaptive Neuro Fuzzy Inference System (ANFIS). The results demonstrated clearly that SVM is better than ANFIS. In the other study, (Khan and Coulibaly, 2006) predicted long-term lake water level using SVM, Multilayer Perceptron (MLP), and seasonal autoregressive model (SAR)). The best of these three approaches is SVM.

ANNs have been widely applied as a statistical machine learning method in the field of hydrology such as in modeling water flow, assessing water quality, and forecasting water level (Toro et al., 2013; Kim and Seo, 2015; Kasiviswanathan et al., 2016; Hamid et al., 2019). Recently, deep and recurrent neural networks (deep learning) have made great success in numerous fields of applications catching attention from both academia and industry (LeCun et al., 2015). One of such networks, called Long short-term memory (LSTM), has been successfully applied in solving problems in hydrology such as flood forecasting (Le et al., 2019) and lake water level prediction (Hrnjica and Bonacci, 2019; Xu et al., 2019).

Hybrid models are capable of taking advantages of each component model in order to achieve improved modeling accuracy and flexibility (Zhang, 2003). Recently, numerous hybrid models have been proposed to enhance the forecast performance of hydrological time series

(Di et al., 2014; Seo et al., 2015; Zhong et al., 2017a; Yaseen et al., 2018; Chen et al., 2018; Yaseen et al., 2018; Rezaie-Balf et al., 2019; Mousavi-Mirkalaei and Banihabib, 2019; Nazir et al., 2019). Seo et al. (2015) investigated two hybrid models, namely, wavelet-based artificial neural network (WANN) and wavelet-based adaptive neuro-fuzzy inference system (WANFIS). WANN was proven to be an effective tool for predicting water levels and achieving better results than other conventional forecasting models. In (Zhong et al., 2017b), a hybrid ANN-Kalman filtering was proposed to predict daily water level for MaAnshan station. The forecasting results of the hybrid model were overall favorable comparing with ANN. In (Mousavi-Mirkalaei and Banihabib, 2019), the authors coupled ARIMA and Nonlinear auto-regressive exogenous to forecast daily Urban Water Consumption (UWC) for Tehran Metropolis. The experiments clearly showed that the hybrid model, which combined both linear and nonlinear models, had higher accuracy in forecasting UWC. Xu et al. (2019) developed an hybrid model combining ARIMA and RNN (Recurrent Neural Network) for water level prediction. The experimental results indicated that the combined model can better capture the overall trend and amplitude fluctuation. An other conjunction of ARIMA and SVR was proposed to anticipate daily average water level data of Liuhe station. The proposed algorithm showed a good anti-jamming performance to data and good accuracy for water level forecasting.

Motivated by the success of hybrid forecasting models, we propose a hybrid approach that combines ARIMA[1] with different statistical machine learning methods to capture the linear and non-linear components of the time series separately. The novel method is then tested on the real datasets of the Red river for hourly water level forecasting.

3. Methods

3.1. Autoregressive integrated moving average model, ARIMA

ARIMA, introduced in (Box and Jenkins, 1976), is one of the most popular statistical linear models for forecasting univariate time series. The main idea is that time series can be decomposed into present, past values and random errors. Hence, ARIMA is a combination of auto-regression AR(p) (an additive linear function of p past observations), moving average MA(q) (q random errors), and d which is an integer making a series to be stationary. The ARIMA(p, q, d) can be represented as:

$$\Delta^d y(t) = c + \sum_{j=1}^{p} \alpha_j \times y(t-j) + \epsilon(t) + \sum_{j=1}^{q} \beta_j \times \epsilon(t-j) \quad (1)$$

Where $\Delta = (1 - B)$, B is the 'Backward' operator and $By(t) = y(t-1)$, $y(t)$ is the observation data at time t, c is the constant, $\alpha_1, \ldots, \alpha_p$ are the auto-regressive parameters, $\epsilon(t)$ is the white noise at time t, and $\beta_1, \ldots, \beta_q$ are the moving average coefficients.

Fitting an ARIMA model involves 4 steps as follows:

- Identification of the ARIMA(p, d, q) structure;
- Estimation of parameters;
- Diagnostic checking on the estimated residuals;
- Forecasting future values based on the known data.

The autocorrelation function (ACF) and the partial autocorrelation function (PACF) of the data are used to find out the order q and p of the ARIMA model (Box and Jenkins, 1976). Besides, a number of other methods for selecting ARIMA order have been proposed based on AIC (Akaikes information criterion) (Shibata, 1976), MDL (minimum description length) (Ljung, 1986; Hurvich and Tsai, 1989), AIC and BIC (Aho et al., 2014) or using fuzzy systems (Haseyama and Kitajima, 2001).

The strength of ARIMA is its ability in making a non-stationary time series to stationary time series by differencing it d times. Stationary is essential as it makes the prediction practical and useful. Therefore, before fitting an ARIMA model, we often need data transformation if the data contains trend component hetero scedasticity. A time series is stationary when its mean, variance, and covariance (at different lags) are constant over time. The α and β coefficients are estimated so that the overall measure of errors is minimal. In the diagnostic checking step, statistical assumptions about the errors of the model are examined by some diagnostic statistics and plots of the residuals.

3.2. Random forest, RF

Since the 1990s, ensemble learning models have been studied to improve the performance of classification and regression algorithms. The purpose of the aggregation models is to reduce the variance and/or bias error components of the learning algorithms. Bias is the conceptual error of learning model and variance is an error due to the variability of the model compared to the randomness of the data samples. Random forests (RF), proposed in (Breiman, 2001), is one of the most successful ensemble learning methods. It is an extension of bagging (bootstrap aggregation) to build dissimilar trees that develops the idea of random subspace sampling. Random forest includes multiple single trees, each of which is built on a random sample of the training data. The algorithm creates fully grown trees without pruning to keep the bias low. Each random forest tree is learned on a bootstrap sample set, and at each node, a random subset of attributes are considered for splitting. The randomness makes diversity among the trees and allows to control the low correlation between trees in the forest. So, naturally, they are more accurate and stable as more trees are added. The Random forrest algorithm (Fig. 1) can be briefly described as follows:

- From a training dataset (learning set, LS), with m samples and n variables (features), construct independently T decision trees.
- The t^{th}-decision tree model is built on the t^{th}- bootstrap sample set from LS.
- At each inner node, randomly choose n' variables ($n' < < n$) and calculate the best partition based on these n' variables.
- Un-pruned trees are built with the maximum depth.

Then, each tree provides a prediction value y and the final prediction value is obtained by aggregating the results given by T individual trees from the forest. The RF prediction is

$$\hat{y} = \frac{1}{T} \sum_{i=1}^{T} \hat{f}_i(x) \quad (2)$$

where, $x \in R^n$ is a new input, T is the number of trees in the forest, $\hat{f}_i(x)$ is the prediction of unknown value y of input x generated from the i tree ($i = 1..T$).

When building a tree, an additional tuning parameter is the number of candidate variables selected for node splitting at each iteration, called *mtry*, and $1 < mtry < n$. An objective of selecting $mtry < n$ is to reduce the computational time. Normally, *mtry* is chosen as $mtry = \sqrt{n}$ for classification problems and $mtry = n/3$ for regression ones. Another parameter to consider is *ntree*. Trees usually tend to be unstable with high variance. In the random forest algorithm, number of trees (*ntree*) is used to reduce the variance.

3.3. Support vector regression, SVR

Support Vector Regression (SVR) is the regression version of Support Vector Machines (SVM), a statistical machine learning algorithm based on statistical learning theory developed in (Vapnik, 1995). The basic idea of SVM is to find an optimal hyper-plane for linearly separable patterns in a high dimensional space where features are mapped onto. There are more than one hyper-plane satisfying this criterion. The task is to discover the one that maximizes the margin around the separating hyper-plane (Fig. 2). This is done with the helps of the support vectors which are the data points that lie closest to the decision surface and have direct bearing on the optimum location of the decision surface.

The linear regression estimating function can be illustrated as follows:

$$f(x) = w^T x + b \quad (3)$$

where w is the weight vector, b is the bias and x is the input vector. SVMs can be extended for classification or regression problems that are not linearly separable by transforming original data into a new space using Kernel functions. The new space, called feature space, is usually high dimensional so that the classes become linearly separable.

3.4. K-nearest neighbors, KNN

K-nearest neighbors (Altman, 1992), a (non-parametric) statistical learning method, has been widely used for both classification and regression problems. This algorithm predicts values of any new data points using a similarity measure (i.e distance function). This means that a new data sample is assigned a value based on how close it is to the points in the training set. In detail, for each sample of a test set, we compute the distance (e.g Euclidean, Manhattan, or Minkowski) between that sample and all samples in the training set and then we choose the K closest

[1] In the Section 3.6.1, we shall explain why ARIMA is chosen ahead of other statistical models to capture the linear component of the time series in this paper.

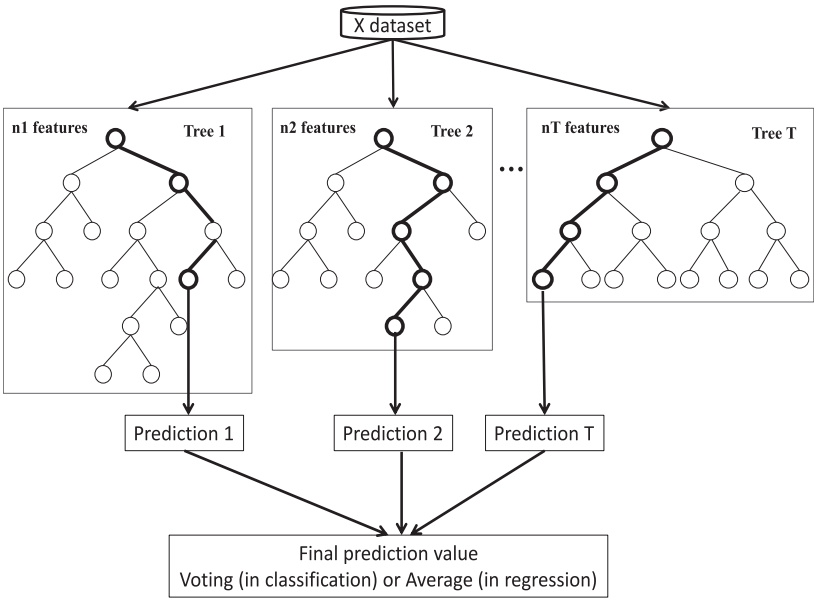

Fig. 1. Diagram of the Random forest building process.

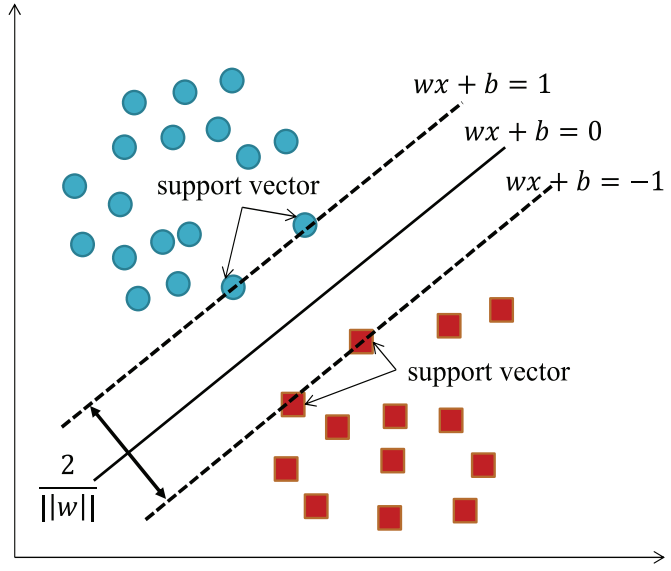

Fig. 2. Linear classification with SVM.

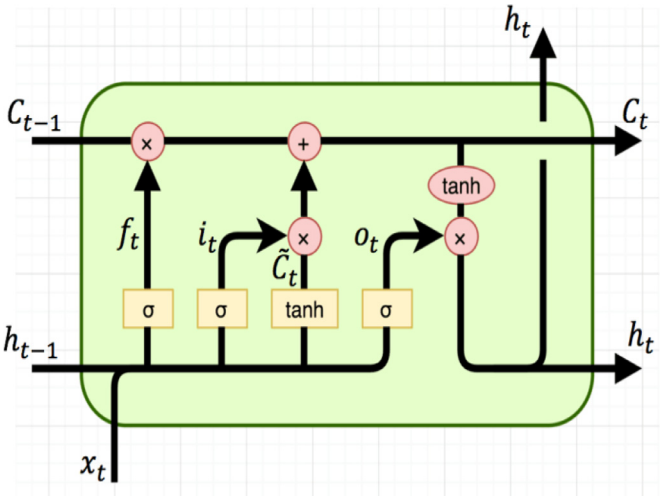

Fig. 3. Internal architecture of a LSTM cell (Christopher, 2015).

training data samples. The prediction value is the average of the target output values of these K nearest neighbors.

3.5. Long short term memory, LSTM

Long Short Term Memory (LSTM) networks, introduced in (Hochreiter and Schmidhuber, 1997), are a special type of recurrent neural networks (RNN) that are capable of learning long-distance dependencies. They have been successfully applied to numerous sequence learning problem and have become the popular building block of numerous well-known deep learning systems. LSTM can remember information from a long previous time and transmit it to the next cell. They can determine which information is important to learn. That is, their intrinsic ability can be memorized without any explicit intervention.

The LSTM network consists of many linked LSTM memory cells and the architecture of each cell is shown in Fig 3.

The idea of LSTM is that, in addition to the hidden state h, the cell internal state c and the three gates including forget gate (f_t), input gate (i_t) and output gate (o_t) are taken into account. At each time step t, a gate always gets the same input values x_t and h_{t-1} obtained from the output of the previous memory cell (at time $t-1$). Each gate filters information for different purposes:

- Forget gate: Removing unnecessary information from the internal state cell.
- Input gate: Selecting what necessary information is added to the internal cell state.
- Output gate: Determining what information from the internal state cell is used as the output.

3.6. The proposed method

Time series data usually consists of linear part as well as nonlinear part (Zhang, 2003). In fact, there is no universal model which can successfully achieve both linear and nonlinear relationships. Linear statistical models, such as ARIMA, might not be good at modeling the nonlinear relationships in the time series but it is sufficient for modeling linear component (Ömer Faruk, 2010; Valenzuela et al., 2008). Meanwhile, (non-parametric) statistical machine learning models such as SVM, RF,

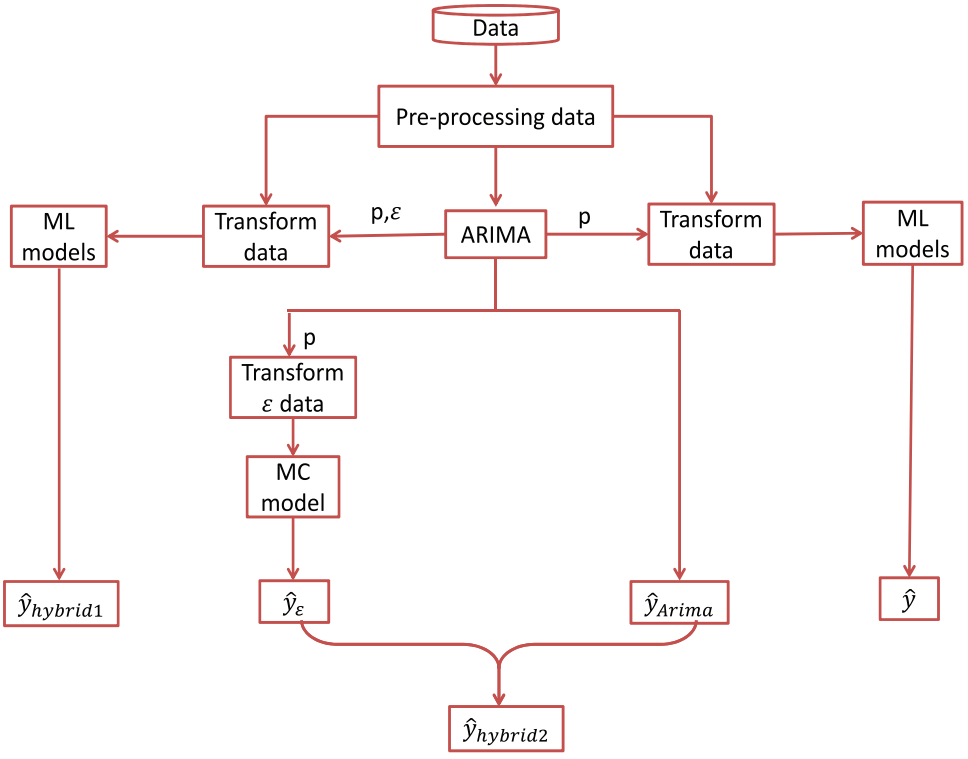

Fig. 4. Flowchart of the proposed method for univariate time series forecasting.

KNN or LSTM can model any nonlinear components (universal approximators). Therefore, in the hope of obtaining better forecasting results, we propose the hybrid models based on the idea of separate modeling of linear and nonlinear components of time series. Before presenting our approach in detail, we give our arguments and justifications for the individual algorithms chosen to develop the novel hybrid approach.

3.6.1. Selection of component models

In the literature, there are a number of linear statistical models for time series modelling, notably, ARIMA family (ARIMA, ARMA, Seasonal ARMA (SARMA), AR, etc.), Gegenbauer ARMA (GARMA, (Woodward et al., 2017)). ARIMA family and GARMA models work under the assumption of stationary processes (Valipour et al., 2013; Zhang, 2003; Woodward et al., 2017), which means that the mean of the series and the covariance among its observations do not change over time. However, ARIMA model can generally fit the non-stationary time series based on the ARMA model, with a differencing process, which effectively transforms the non-stationary data into stationary one. Furthermore, Valipour et al. (2013) pointed out that ARIMA has a better performance than ARMA because of making time series stationary, in both training and forecasting phases. In addition, in the training step of ARIMA, we always take into account the seasonal factor as the property of our data. PARMA (periodic ARMA) models are devoted to model the time series whose mean, variance, and covariance function vary with seasons (Anderson et al., 2013). They are usually used for data having cyclostationary property (Chaari et al., 2015). Thus, PARMA could handle to some extent the non-stationary of time series (if and only if it coincides with seasons). All of the datasets used in this paper have a seasonality component (See Table 2 and Fig. 5) that makes data are non-stationary. Therefore, ARIMA, PARMA, and GAMRMA are the most relevant candidates for modeling the linear component of our time series data, in which, ARIMA has been credited, in the literature, to be the most suitable of the three for handling non-stationary time series. To confirm the suitability of ARIMA, we have conducted the experiments to compare PARMA, GARMA and ARIMA on our time series of water level described in Section 4. The results (not shown in full here to save the space) consistently confirmed the superiority of ARIMA based on the performance indicators described in Section 4.2. For instance, Table 1 depicts the average forecast errors of ARIMA, PARMA, GAMRMA for 12h-ahead on Hanoi time series. It is clear that, ARIMA was significantly better than the others on all performance indicators. Therefore, in this paper, ARIMA is chosen to develop our proposed hybrid approach.

Table 1
Average 12h-ahead forecasting results generating by ARIMA, GARMA and PARMA.

Size	Method	Sim	MAE	RMSE	FSD	R	NSE
12h	GARMA	0.46	38.03	39.99	0.95	0.76	-35.4
	PARMA	0.61	17.79	19.73	1.6	0.78	-4.36
	ARIMA	**0.65**	**15.07**	**17.3**	**1.01**	**0.61**	**-3.08**

To capture the non-linear component of time series, RF, SVM, KNN, and LSTM methods are utilized. We have selected these methods for a number of reasons. First, they are all (non-parametric) statistical learning methods that have relatively firm theoretical foundations (eg. with theoretical guarantee on learnability, learning consistency, and universality). Second, they are representatives of large and popular classes of statistical machine learning techniques - lazy learning (KNN), recurrent neural networks (LSTM), kernel based methods (SVM), and ensemble learning methods (RF). Third, they have been successfully applied in modeling/ learning hydrological time series (Nguyen et al., 2015; Yang et al., 2017; Wang et al., 2018; Choi et al., 2019; Hipni et al., 2013; Khan and Coulibaly, 2006; Fernandez-Delgado et al., 2014).

3.6.2. Details of the proposed method

Fig. 4 describes all steps of our proposed method for forecasting univariate time series. The objective of the proposed method is to take into account the advantages of the different forecasting models in terms of type (i.e. linear and non-linear) and complexity (i.e. single model and hybrid model).

Table 2
Characteristics of the water level time series.

No	Dataset name	Period	# Samples	Trend (Y/N)	Seasonal (Y/N)	Frequency
1	Vu Quang	2008–2017	29,224	N	Y	3 hours
2	Hanoi	2008–2017	29,224	N	Y	3 hours
3	Hung Yen	2008–2015	64,061	N	Y	hourly

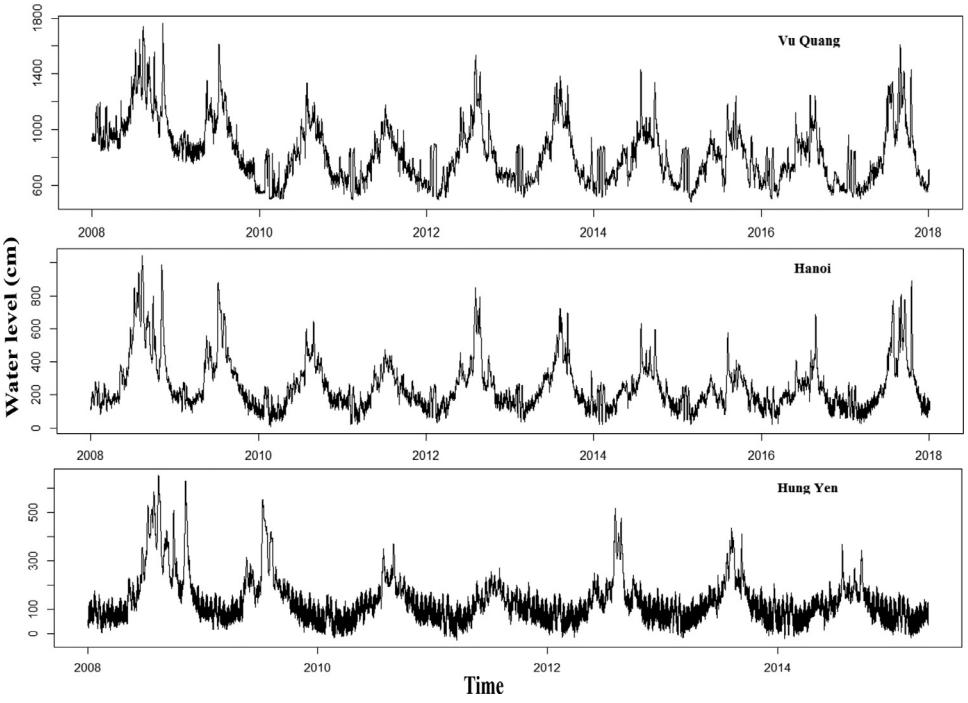

Fig. 5. Water level at three hydrology stations of the Red river.

A time series can be considered as a combination of two components: linear and nonlinear ones (Zhang, 2003). That is,

$$y(t) = L(t) + N(t) \tag{4}$$

where $L(t)$ and $N(t)$ are the linear and nonlinear components of the time series. Both components (i.e. $L(t)$, $N(t)$) must be estimated from time series data.

The proposed method involves three stages: i) linear modeling, (ii) linear-nonlinear modeling, and (iii) forecasting future values. In the first stage, we apply the ARIMA model to extract the linear part of time series. Then, in the second stage, the residuals obtained from the ARIMA and the lags values of the time series are used as the input for training statistical machine learning models. Finally, in the third stage, future values are estimated from different hybrid models. The details of these steps are described in the following:

- Linear modelling
  ARIMA is conducted on the whole time series to get the predicted values $\hat{L}(t)$ and their residuals. The residuals of this step are used as a part of the input in the second stage (hybrid models). Specifically, in this study, we utilize the order p obtained from the ARIMA model to identify the optimal lag time for the inputs of the hybrid models. This allows us to transform one-dimensional time series into p-dimension data, so that statistical machine learning methods could be applied for forecasting univariate time series.[2]
  It is noted that, rather than heuristically chosen, the order p is obtained in a data-driven manner from the process of building ARIMA model (ACF (auto-correlation function) and PACF (partial auto-correlation function) are performed to determine different time lags. The calculation of p with ARIMA is fast. Therefore, we do not need extra work, which could be unnecessarily expensive, to tune p for each statistical machine learning model.
- Linear-Nonlinear modelling
  The residuals are important in selecting appropriate linear models because a model is not completely linear if it still contains nonlinear component in its residuals. Normally, we cannot detect any nonlinear component when analyzing the residuals and in fact there is no statistical method that allows to determine nonlinear autogressive relationships (Mousavi-Mirkalaei and Banihabib, 2019). Therefore, we take into account the residuals by using statistical machine learning models in order to discover the nonlinear relations in the residuals.
  The residual from ARIMA model is calculated by:

$$\epsilon(t) = y(t) - \hat{L}(t) \tag{5}$$

  where $\epsilon(t)$ is the residual and $\hat{L}(t)$ is the forecast value of ARIMA model at time t. In order to discover the nonlinear relationships of the time series, $\epsilon(t)$ is could be fitted by statistical machine learning models such as KNN, SVM, RF and LSTM. In this study, we build and test two types of hybrid models as follows:

  - Model 1

$$\hat{y}_{hybrid1}(t) = f(y(t-1), \ldots, y(t-p), \epsilon(t-1), \epsilon(t-2)) \tag{6}$$

  where f a nonlinear function learnt from data by the machine learning models. Here, we perform some pre-experiments on our data to find an appropriate stop of ϵ with different values $(1, 2, 3, \ldots, p)$. The results showed that when the past window is greater than 2 the predicted values are not better and it requires

[2] These p-dimensional data are also used for training single statistical machine learning methods (to compute $\hat{y}$ in Fig. 4) for comparison with the hybrid approaches in Section 4.

longer computational time. Therefore, the past window of ϵ is empirically chosen as 2 in this paper.

- Model 2

At this stage, we forecast the nonlinear data as:

$$\hat{y}_{\epsilon}(t) = f(\epsilon(t-1), \ldots, \epsilon(t-p)) \tag{7}$$

where f a nonlinear function learnt from data by the statistical machine learning methods and ϵ is the residuals obtained from stage 1. Therefore, the final combined forecast will be:

$$\hat{y}_{hybrid2}(t) = \hat{y}_{\epsilon}(t) + \hat{y}_{arima}(t) \tag{8}$$

- Forecasting: The developed models in the previous step are used to forecast water level (in our study is for the Red river).

4. Experimental results and discussions

In this section, we first describe the datasets and their pre-processing stage in our experiments, then, define the performance evaluation metrics, and finally discuss the experimental results with key findings and observations.

4.1. Data representation and pre-processing

In this study, water level data from three hydrology stations of the Red river (the main river in the Northern Delta of Vietnam), namely Vu Quang, Hanoi, and Hung Yen are used to test the proposed hybrid approach and compare with well-known machine learning methods. Fig. 5 presents the data of the three selected stations.

- Vu Quang and Hanoi datasets:
The water level of these two datasets were collected at the Vu Quang and Hanoi hydrology stations in Vietnam. The samples were taken from 01 January 2008 to 31 December 2017 with different frequencies. The reason is that on normal days without rainfall or flood, the frequency of sampling is low, for example, with 4, 6, 7 or 8 times per day or no data available. However, on rainy days, the frequency of sampling is much more such as with 10, 11, 12, 18, 19, 20, 21, 22, or even 23 times a day. Therefore, before applying time series forecasting algorithms to predict the water level, it is necessary to re-sample data at equal time intervals and pre-process them. For these time series, we normalized the sampling frequency to 8 times per day at 1h, 4h, 10h, 13h, 16h, 19h, 22h. Consequently, there are two cases:

  - *The data sampling a day is less than 8 times or no data.*
  In this case, we consider the periods of non-sampling or days without data as missing data. Then, we use imputation methods such as linear interpolation and/or moving average to fill in the missing data. For the large consecutive missing data (i.e. a full day missing or more), the result will be a straight line when applying linear interpolation. So in order to capture the dynamism of the data, we utilize DTWBI method (Phan et al., 2017) that allows to complete consecutive missing data while taking into account the dynamism of data.
  - *The data sampling a day is more than 8 times.*
  We base on the sampling times to recalculate the water level with the time aforementioned. This means we perform re-sampling 8 times a day at 1h, 4h, 7h, 10h, 13h, 16h, 19h, and 22h.

- Hung Yen dataset:
This dataset was collected at the Hung Yen hydrology station every hour from 01/01/2008 to 23/04/2015 (Fig. 5). It also contains some missing data points and we use the interpolation method to complete those missing.

The characteristics of the three datasets are summarized in Table 2.

4.2. Performance evaluation indicators

After the prediction phase, six evaluation metrics were used to assess different models, they are Sim, MAE, RMSE, R score, FSD and NSE. These metrics were selected as they possess different properties that are important to efficiently understand the performance of forecasting models from different angles. They are defined as follows:

1. Similarity: defines the percentage of similar values between the predicted values y and the actual values x. It is calculated as:

$$Sim(y,x) = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{1+\frac{|y_i - x_i|}{\max(x)-\min(x)}} \tag{9}$$

Where T is the number of forecasting values. A higher similarity (Sim value $\in [0, 1]$) highlights a better ability of the method for the forecasting task.

2. MAE: The Mean Absolute Error between the predicted values y and the actual ones, x, is computed as:

$$MAE(y,x) = \frac{1}{T}\sum_{i=1}^{T}|y_i - x_i| \tag{10}$$

A lower MAE value means better performance method for the prediction task.

3. RMSE: The Root Mean Square Error is defined as the average squared difference between the forecast values y and the respective true values x. This indicator is very useful for measuring overall precision or accuracy. In general, the most effective method would have the lowest RMSE.

$$RMSE(y,x) = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(y_i - x_i)^2} \tag{11}$$

4. R score: is determined as the correlation coefficient between two variables y and x. This indicator makes it possible to assess the quality of a forecasting model. A method presents better performance when its R score is higher ($R \in [0, 1]$).
5. FSD: The Fraction of Standard Deviation is defined as:

$$FSD(y,x) = 2 * \frac{|SD(y) - SD(x)|}{SD(y) + SD(x)} \tag{12}$$

This fraction points out whether a method is acceptable or not. For a forecasting method, when FSD value approaches 0, it is impeccable.

6. NSE: The Nash Sutcliffe efficiency is used to evaluate the predictive ability of hydrological models. The NSE values range from -∞ to 1, with higher values mean better fit between observed and forecast water level (Nash and Sutcliffe, 1970).

$$NSE = 1 - \frac{\sum_{i=1}^{T}(x_i - y_i)^2}{\sum_{i=1}^{T}(x_i - \overline{x_i})^2} \tag{13}$$

4.3. Results and discussions

To evaluate and compare all tested methods, the whole collected data were divided into two parts for training and testing. With Vu Quang and Hanoi time series, the training datasets were those from 2008 to 2015 accounted for 80% of the data, and the remaining observed samples (20%) from 2016 to 2017 were used to assess the forecasting models. With Hung Yen dataset, the data from 2008 to 2013 were used for training models and the remaining data (2014–2015) were utilized to test the learnt models.

The problem of water level forecasting for the Red river was tested with ARIMA, aforementioned statistical machine learning methods, namely, KNN, SVR, RF, LSTM, and the proposed hybrid models. In this paper, we develop multi-step ahead prediction models based on one-step ahead prediction. This means that to forecast a value $\hat{y}(t)$ at time t, p previous values in the past, $y(t-1), y(t-2), \ldots, y(t-p)$, are used. In

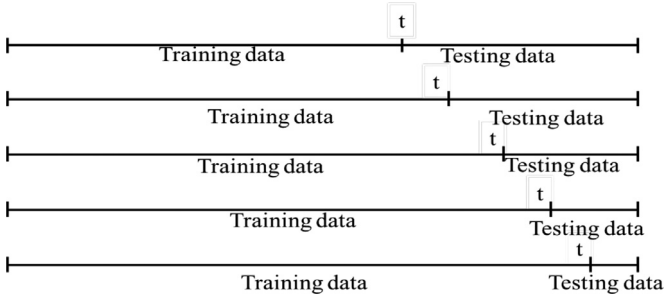

Fig. 6. Training ARIMA model at different times.

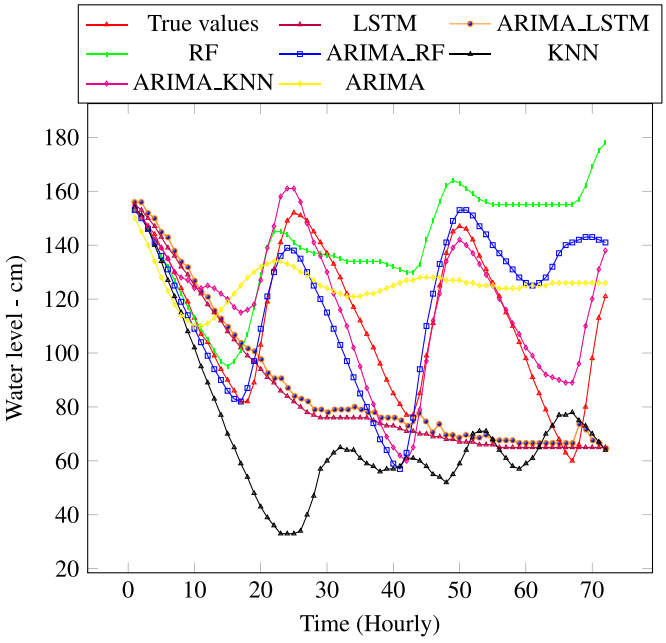

Fig. 7. Visual comparison of 72h-ahead predicted values (with hourly frequency) using different forecasting methods with true values on Hung Yen series on February 13, 2014.

the next prediction step $\hat{y}(t)$ is used as one of the input values to predict the next value $\hat{y}(t+1)$, and so on.[3]

All single ML and hybrid models were trained one time on the training datasets and then the learnt models were applied to forecast at different time t on the testing sets. However, for ARIMA, in order to get better results, all the whole past history values until time $t-1$ were used to build the model, then this model was used to forecast T values from time t. So, for each forecast at a different time, the retrain of ARIMA model was needed. The process for ARIMA is shown in Fig. 6, it depicts how the training samples are utilized to make the model.

To implement the hybrid methods, the first step is to capture the linear, nonlinear components of the time series data and the order p. In the next phase, for the Model 1, the original data and nonlinear parts are combined as in Eq. 6 to build new data used for developing hybrid forecasting methods. For the Model 2, nonlinear data are used directly (Eq. 7) by the ML algorithms. With this model, the final predictive result is a combination of linear prediction generated by ARIMA and nonlinear prediction generated by ML methods.

Determining the parameters of these models plays an important role. These parameters influence the forecasting accuracy significantly. Here, when building a machine learning method, we used 5-fold

[3] It is noted that by doing that way the prediction error could get accumulated overtime.

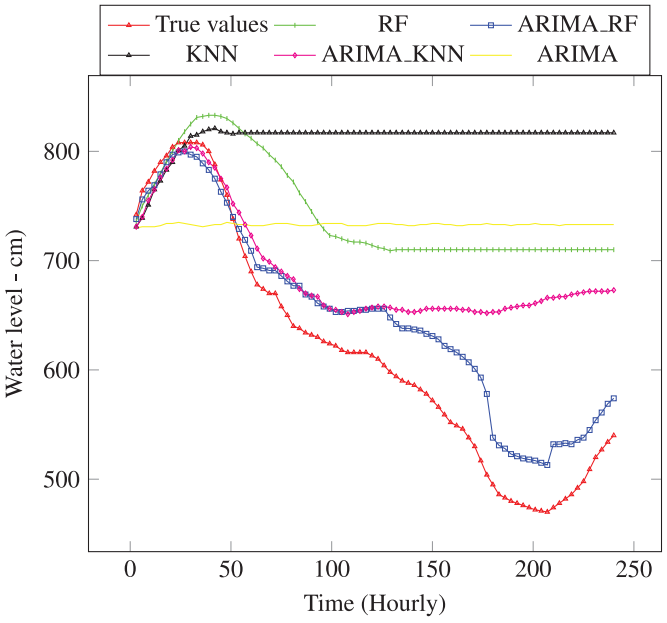

Fig. 8. Visual comparison of 10 days-ahead predicted values (with 3 hourly frequency) using different forecasting methods with true values on Hanoi series.

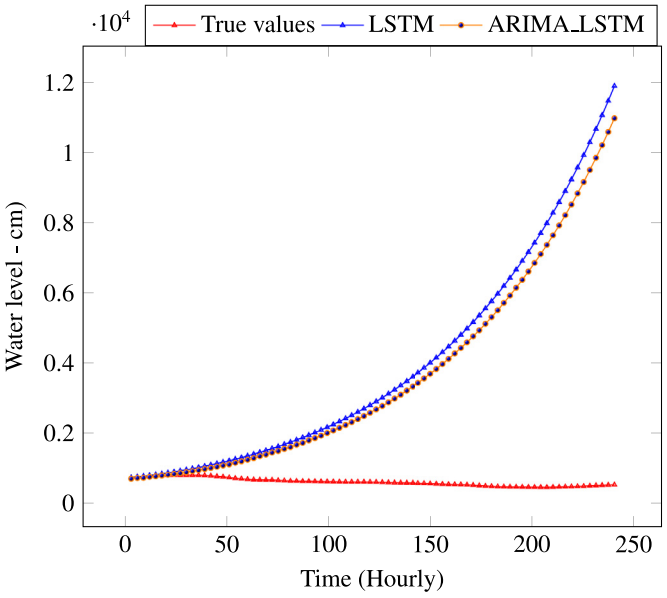

Fig. 9. Visual comparison of 10 days-ahead predicted values (with 3 hourly frequency) using different forecasting methods with true values on Hanoi series.

cross-validation to find the optimal parameters of each model. For ARIMA model, we apply auto.arima() from the forecast R package (Hyndman and Khandakar, 2008) to find the parameters p, d, and q. We also take into account the seasonality component for all datasets when training ARIMA models. Table 3 shows our parameter settings.

Tables 4, 5 and 6 give the details on the average forecast results of all tested methods at 20 different random times for the Vu Quang, Hanoi and HungYen datasets. Different forecast horizons were used to assess the forecasting performance of the proposed hybrid approaches and other methods. For Vu Quang and Hanoi time series, the forecasting models were established for 12h, 24h, 48h, 72h and 5-days ahead time intervals. For Hung Yen data, these models were applied to predict 6h, 12h, 24h, 48h, 72h and 5-days ahead. The best results for each forecasting horizon are highlighted in bold. It can be seen that the hybrid

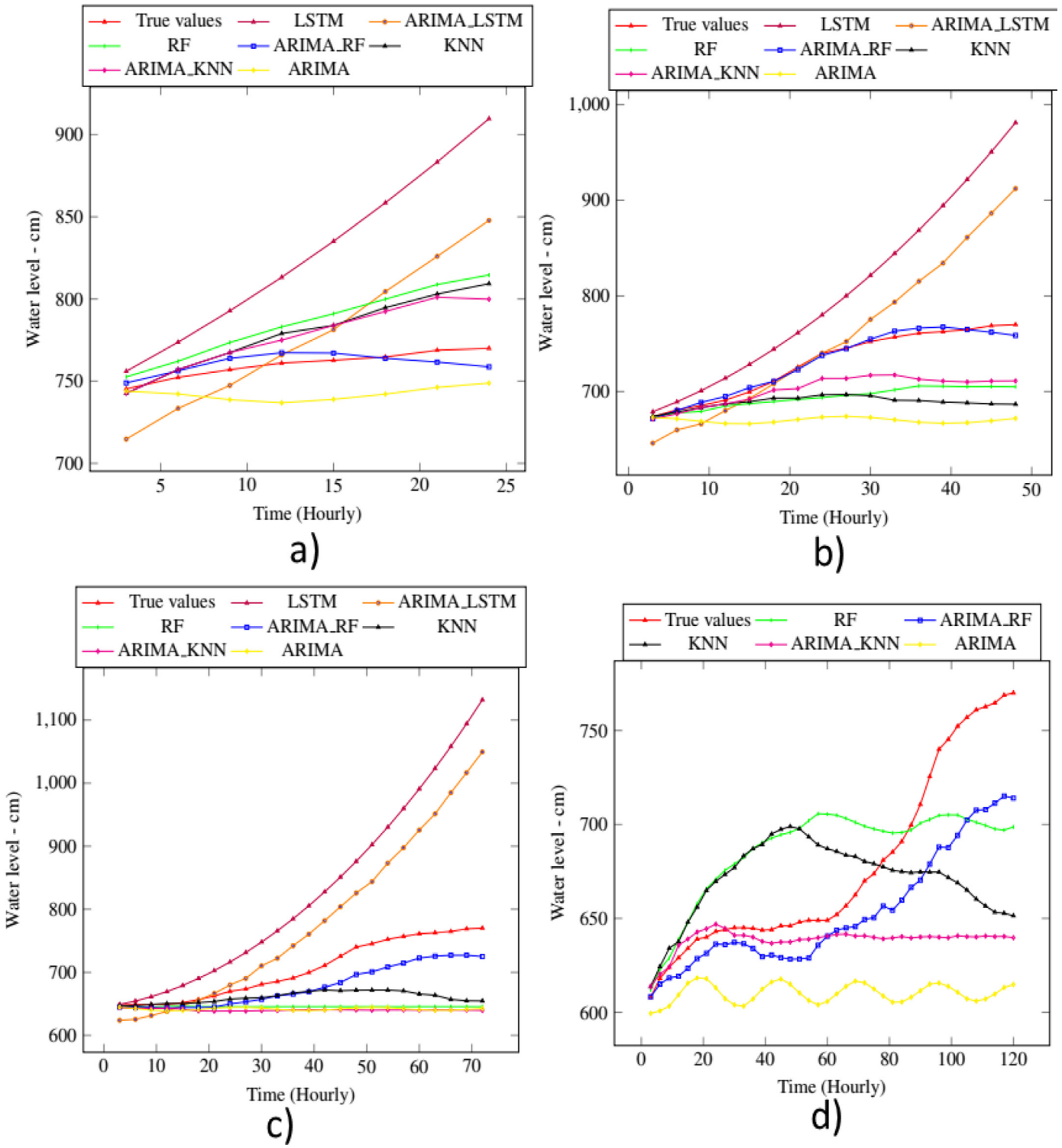

Fig. 10. Visual comparison of forecast values generating by different methods on 7/24/2017 at the Hanoi station a) 24h before the peak; b) 48h before the peak; c) 72h before the peak and d) 5 days before the peak with 3 hourly frequency.

methods using Model 1 (Eq. 6) such as ARIMA_RF and ARIMA_KNN most often produced better results than individual models (i.e. ARIMA, RF, KNN, SVM, or even LSTM) and other combined models (ARIMA_SVM, ARIMA_LSTM) as well as hybrid methods using Model 2 (Eq. 7, Table 6).

From Table 4, it can be seen that ARIMA_RF achieved the lowest MAE, RMSE, FSD and NSE, and highest Sim measures on most predicted periods. The second rank is ARIMA_KNN for these indicators (it also produced the best results on Vu Quang dataset for 48h forecasting of water level). This indicates ARIMA_RF and ARIMA_KNN (based on Eq. 6) are more accurate than other methods. Significantly, they could fully exploit separation of linear and nonlinear components. Looking at results of ARIMA_SVM, SVM, ARIMA_LSTM and LSTM methods, we see that ARIMA_SVM and ARIMA_LSTM do not improve the performance over SVM and LSTM, respectively. That is, when combining the nonlinear part with the original data, neither SVM nor LSTM could exploit and capture well the nonlinear component of the time series.

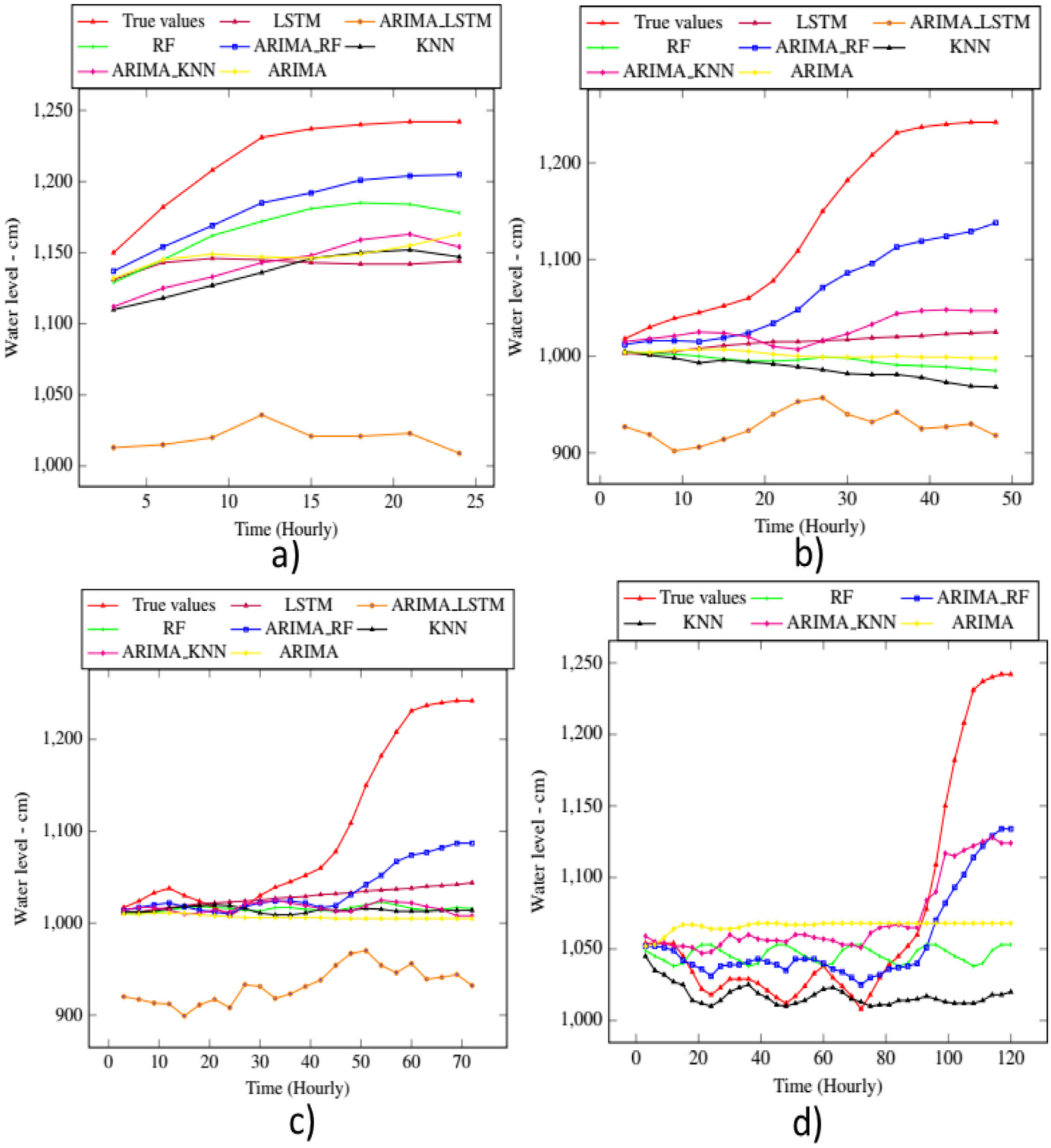

Fig. 11. Visual comparison of forecast values generating by different methods on 8/21/2016 at the Vu Quang station a) 12h before the peak; b) 24h before the peak; c) 72h before the peak and d) 5 days before the peak with 3 hourly frequency.

Table 5 presents the results of the different forecasting methods on the Hung Yen dataset. In this table, ARIMA_RF and ARIMA_KNN methods do not yield good results at all forecast horizons as in the Table 4 but the both methods still have performance improvements over single machine learning models (RF, KNN). Once again, ARIMA_SVM and ARIMA_LSTM methods failed to capture the non-linear part of the time series data. These methods yield results even worse than single machine learning models (SVM and LSTM).

Table 6 gives the average forecasting results using the proposed algorithms (Model 2 based on Eq. 7) on the Vu Quang dataset. The results indicate that the hybrid approaches based on Eq. 7 (Model 2) are no more effective than ARIMA. Overall results of the hybrid methods

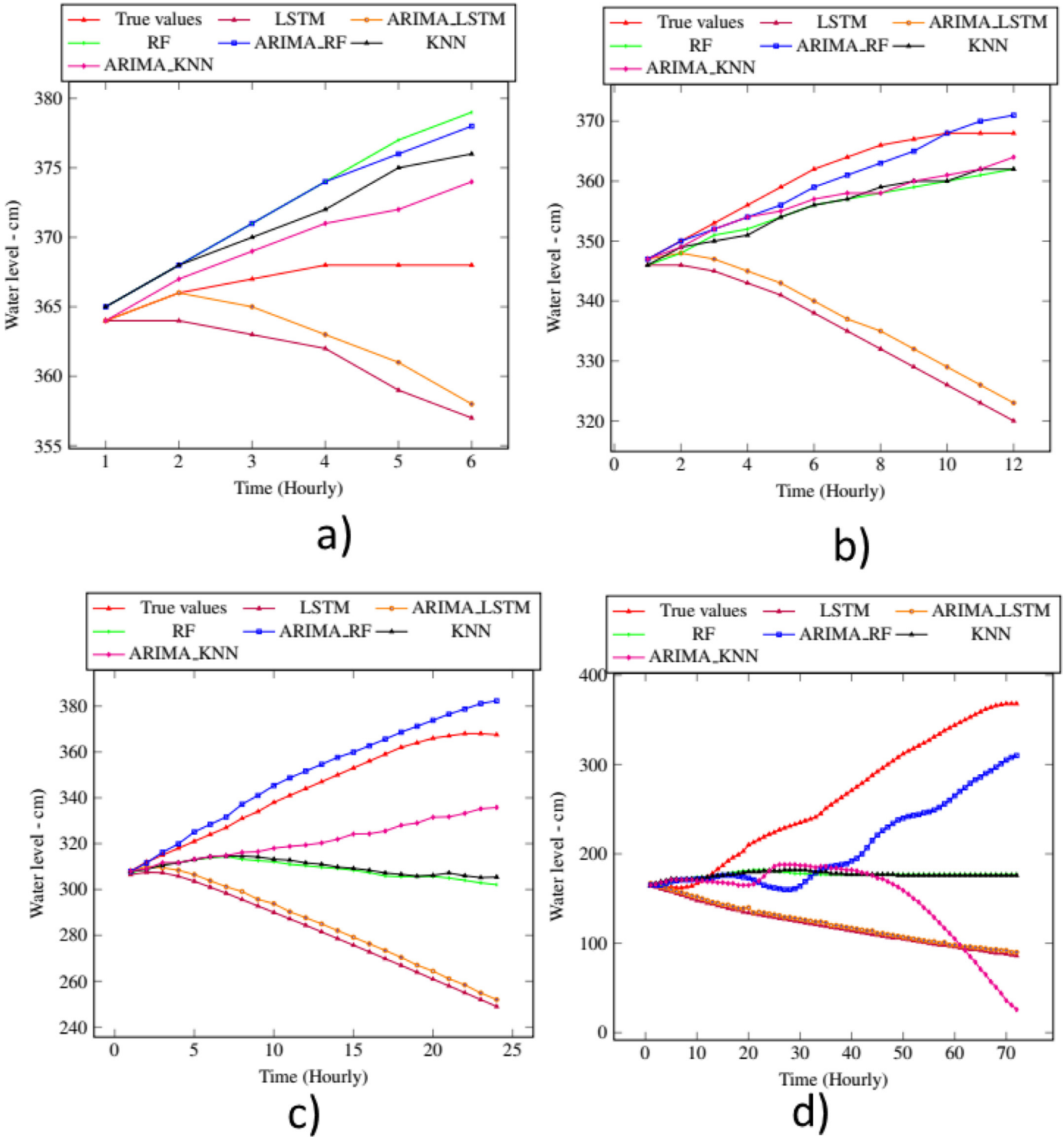

Fig. 12. Visual comparison of forecast values generating by different methods on 7/23/2014 at the Hung Yen station a) 6h before the peak; b) 12h before the peak; c) 24h before the peak and d) 3 days before the peak with hourly frequency.

(Model 1) on this dataset (Table 4) clearly shows that the methods of the first type of hybrid models (Model 1 using Eq. 6) outperforms significantly the methods of the second type of the hybrid approach (Model 2).

Figs. 7, 8, and 9 plot the results of time series forecasts from different algorithms on Hung Yen and Hanoi datasets. The observed water level values and predicted ones generated from each model are showed to compare their performances.

Fig. 7 compares the predicted water level of all models for 3-days ahead. In this Figure, we could see that ARIMA_RF yields the best results for the first day. Its forecast values are nearly identical to the actual ones. However, when predicting for the second and third days, the results are no longer as good as for the first day. It still captures rather well the trend of real data but the forecast errors are higher. On the contrary, ARIMA_KNN forecasts poorly for the first day but relatively well for the second and third days. RF, KNN, LSTM, ARIMA_LSTM produce good

Table 3
Selected parameters of various forecasting methods.

Dataset name	Method		
	ARIMA (p,d,q)	KNN	RF
Vu Quang	(5,1,1)	K=6	mtry=4, ntree=100
Hanoi	(5,1,5)	K=7	mtry=4,ntree=1000
Hung Yen	(5,1,3)	K=5	mtry=4, ntree=300
		ARIMA_KNN	ARIMA_RF
Vu Quang		K=6	mtry=6, ntree=200
Hanoi		K=5	mtry=7,ntree=1000
Hung Yen		K=6	mtry=6, ntree=100
		SVM	
Vu Quang		C=1, gamma= 0.001, kernel= rbf	
Hanoi		C=50, gamma= 0.001, kernel=rbf	
Hung Yen		C=50, gamma= 0.01, kernel=rbf	
		ARIMA_SVM	
Vu Quang		C=50, gamma= 0.001, kernel=rbf	
Hanoi		C=5, gamma=0.001, kernel=rbf	
Hung Yen		C=50, gamma= 0.001, kernel=rbf	
Parameters of LSTM & ARIMA_LSTM for all experiments			
	Library: keras	Metrics: Accuracy	Loss function: MSE
	Epochs=200	Validation_split=0.05	Batch size=3
	Optimizer function: Adam		

predicted values initially but then quickly fail to capture the trend of the real data.

It can be seen from Figs. 7 and 8 that in general ARIMA is not suitable for forecasting water level on Red river although initially it produces good results (Fig. 7). Fig. 8 demonstrates again the superiority of ARIMA_RF and ARIMA_KNN for forecasting water level on Hanoi dataset for the first 5 days. Prediction values of these two methods are closer to observed water levels than ARIMA, RF and KNN methods. For the last 5 days, only ARIMA_RF is able to catch approximately the trend of real data but the forecast error is large.

By observing the experimental results (Table 4, Fig 9), we have found that LSTM and ARIMA_LSTM produced worse prediction performance for the next 10 days.

Apart from normal forecasts, it is interesting and important to see if the hybrid approach is effective in critical situations, such as forecasting at a few days before the water level goes significantly high (suggesting a peak in a flood). Fig. 10, 11, and 12 present the visual comparisons of all methods for forecasting several periods of water level at Hanoi, Vu Quang and Hung Yen stations.

Fig. 10 shows the results of forecasting the water level before the peak (July 24, 2017 at 4h) at Hanoi station with 1 day (Fig 10a), 2

Table 4
Average results of various forecasting algorithms (hybrid approach based on function 6 - Model 1) on Vu Quang and Hanoi datasets.

Method	Size	Vu Quang - Model 1						Hanoi - Model 1					
		Sim	MAE	RMSE	FSD	R	NSE	Sim	MAE	RMSE	FSD	R	NSE
ARIMA	12h	0.67	10.0	11.63	1.15	0.92	-4.4	0.65	15.1	17.30	1.01	0.61	-3.08
LSTM		0.64	11.7	13.36	0.95	0.87	-5.4	0.63	17.0	20.10	0.62	**0.85**	-8.13
ARIMA_LSTM		0.39	42.1	43.08	**0.60**	0.72	-117	0.68	13.1	14.75	0.64	0.76	-3.16
RF		0.69	9.4	10.92	1.04	0.85	-2.4	0.72	11.3	13.16	0.70	0.84	-1.72
ARIMA_RF		**0.78**	**6.0**	**7.01**	0.64	**0.93**	**-0.2**	**0.81**	**7.0**	**8.26**	**0.37**	0.84	**-0.10**
KNN		0.71	8.7	9.95	0.88	0.84	-2.0	0.64	15.4	17.18	0.71	0.81	-4.10
ARIMA_KNN		0.71	7.1	7.96	0.74	0.91	-1.5	0.68	12.0	13.33	0.61	0.83	-2.13
SVM		0.62	12.8	14.76	0.95	0.92	-7.2	0.65	39.9	45.71	0.70	**0.85**	-28.1
ARIMA_SVM		0.61	20.6	25.50	0.84	0.93	-43	0.59	66.7	80.17	0.87	0.84	-72.3
ARIMA	24h	0.70	16.8	19.97	1.11	0.74	-3.2	0.69	26.3	30.63	1.06	0.63	-4.12
LSTM		0.67	18.9	21.81	0.88	0.66	-4.4	0.62	38.8	47.25	0.75	0.78	-27.2
ARIMA_LSTM		0.52	39.9	41.83	**0.48**	0.41	-43	0.70	25.0	30.95	0.76	0.71	-8.36
RF		0.71	15.7	18.89	0.93	0.57	-2.8	0.74	20.2	24.21	0.68	0.72	-3.38
ARIMA_RF		**0.77**	10.9	13.27	0.66	0.73	**-1.1**	**0.82**	**12.5**	**15.16**	**0.46**	0.82	**-0.34**
KNN		0.71	14.5	17.47	0.96	0.68	-2.6	0.71	23.7	27.50	0.81	0.74	-4.40
ARIMA_KNN		0.76	**10.5**	**12.30**	0.61	**0.87**	-1.2	0.75	18.2	20.98	0.61	**0.83**	-1.67
SVM		0.65	19.9	23.59	1.05	0.68	-5.8	0.68	58.5	66.25	0.80	0.80	-19.1
ARIMA_SVM		0.59	57.6	76.60	0.93	0.67	-90	0.57	113.1	132.92	0.84	0.78	-148
ARIMA	48h	0.74	29.7	36.93	1.42	0.59	-2.0	0.71	44.2	50.74	1.17	0.47	-2.88
LSTM		0.72	32.5	39.80	0.88	0.65	-3.2	0.58	92.9	113.27	0.74	0.70	-30.7
ARIMA_LSTM		0.65	42.3	48.31	0.73	0.44	-8.5	0.64	70.7	89.71	0.69	0.71	-16.8
RF		0.75	27.5	34.29	1.16	0.40	-1.5	0.74	35.9	42.42	0.65	0.68	-3.44
ARIMA_RF		0.80	18.1	22.21	0.70	0.74	-0.8	**0.82**	**20.0**	**23.62**	**0.42**	0.76	**-0.69**
KNN		0.75	27.9	35.61	1.05	0.53	-2.0	0.72	39.3	45.14	0.86	0.65	-3.12
ARIMA_KNN		**0.81**	**18.0**	**22.03**	**0.69**	**0.88**	**-0.6**	0.78	26.7	30.44	0.57	**0.81**	-0.86
SVM		0.71	35.7	43.78	1.11	0.61	-3.5	0.68	76.5	86.43	0.75	0.67	-28.3
ARIMA_SVM		0.55	115.2	144.50	0.97	0.59	-161	0.57	138.0	151.45	0.73	0.64	-53.2
ARIMA	72h	0.76	42.4	53.03	1.55	0.44	-1.7	0.72	55.3	62.61	1.25	0.35	-2.31
LSTM		0.73	45.2	55.50	0.94	0.62	-2.6	0.55	159.8	197.02	0.82	0.70	-42.1
ARIMA_LSTM		0.70	49.1	57.66	0.80	0.48	-3.3	0.60	130.3	165.92	0.76	0.72	-28.6
RF		0.76	40.2	51.00	1.26	0.36	-1.2	0.75	45.9	53.49	0.60	0.63	-2.57
ARIMA_RF		**0.82**	**25.5**	**31.47**	**0.72**	0.78	**-0.8**	**0.83**	**25.1**	**29.06**	**0.40**	0.78	**-0.21**
KNN		0.77	40.2	50.35	1.11	0.49	-1.2	0.73	49.6	56.79	0.86	0.56	-1.98
ARIMA_KNN		**0.82**	25.7	32.41	0.78	**0.86**	-1.3	0.80	30.8	34.72	0.50	**0.82**	-0.36
SVM		0.72	49.3	60.75	1.17	0.58	-3.9	0.70	80.9	89.82	0.72	0.60	-7.47
ARIMA_SVM		0.54	147.8	176.32	0.98	0.58	-105	0.59	140.9	152.66	0.59	0.57	-17.9
ARIMA	5 days	0.74	62.1	74.51	1.66	0.43	-2.1	0.73	70.5	79.36	1.40	0.28	-2.51
LSTM		0.71	68.2	82.20	0.78	0.66	-4.2	0.48	356.9	454.77	1.15	0.67	-130
ARIMA_LSTM		0.73	58.1	68.99	0.80	0.50	-1.9	0.52	309.6	402.00	1.11	0.67	-101
RF		0.75	58.4	70.43	1.35	0.34	-1.5	0.76	56.9	64.73	0.54	0.62	-2.26
ARIMA_RF		0.80	**37.41**	**44.38**	**0.77**	**0.79**	**-1.0**	**0.82**	**30.8**	**34.82**	**0.34**	**0.78**	**-0.32**
KNN		0.76	54.5	65.42	1.18	0.40	-1.4	0.73	64.4	73.17	0.94	0.56	-2.54
ARIMA_KNN		**0.81**	37.43	45.59	0.82	0.78	-1.9	**0.82**	34.7	39.15	0.54	0.77	-0.45
SVM		0.72	64.9	76.99	1.08	0.54	-3.4	0.72	81.8	91.08	0.69	0.59	-4.19
ARIMA_SVM		0.51	182.3	206.60	0.91	0.57	-79	0.61	138.4	149.74	0.53	0.58	-12.4

Table 5
Average results of various forecasting algorithms (hybrid approaches based on function 6 - Model 1) on Hung Yen time series.

Method	Sim	MAE	RMSE	FSD	R	NSE	Sim	MAE	RMSE	FSD	R	NSE
	6h						48h					
ARIMA	0.78	12.95	14.66	0.6	0.93	-0.71	0.81	24.2	29.34	0.93	0.6	-0.38
LSTM	0.85	8.92	10.16	0.3	0.97	0.42	0.79	28.45	36.06	0.52	0.52	-0.98
ARIMA_LSTM	0.82	9.87	11.54	0.31	0.93	0.32	0.8	27.96	35.64	0.48	0.5	-0.91
RF	0.77	13.16	15.18	0.46	0.92	-2.35	0.76	36.05	45.18	**0.41**	0.36	-2.14
ARIMA_RF	**0.87**	**6.43**	**7.37**	**0.28**	**0.98**	**0.44**	0.78	30.92	38.19	0.42	0.48	-1.4
KNN	0.8	11.41	13.1	0.37	0.93	-1.79	0.79	29.27	36.7	0.45	0.32	-1.1
ARIMA_KNN	0.82	9.79	11.28	0.39	0.97	-0.25	**0.82**	**23.15**	**29.01**	0.48	**0.67**	**-0.35**
SVM	0.79	16.18	19.65	0.52	0.86	-2.12	0.76	36.75	44.19	0.49	0.38	-3.06
ARIMA_SVM	0.73	25.34	27.74	0.41	0.93	-6.08	0.6	77.71	86.65	0.56	0.46	-15.62
	12h						72h					
ARIMA	0.79	17.25	19.92	0.36	0.86	-1.07	0.8	28.9	34.87	1.1	0.48	-0.62
LSTM	0.83	13.66	16.24	**0.33**	0.86	0.05	0.79	33.7	41.51	0.66	0.45	-1.22
ARIMA_LSTM	0.82	14.23	16.75	**0.33**	0.85	0.04	0.79	33.04	40.94	0.61	0.43	-1.14
RF	0.77	20.95	25.37	0.35	0.81	-1.63	0.74	42.52	52.43	0.39	0.33	-3.17
ARIMA_RF	**0.86**	**10.42**	**12.84**	0.41	0.88	**0.36**	0.76	38.95	48.15	**0.35**	0.32	-2.44
KNN	0.81	16.86	20.66	0.34	0.81	-0.91	0.8	30.59	37.91	0.46	0.32	-0.92
ARIMA_KNN	0.82	14.52	17.16	0.58	**0.9**	-0.39	**0.82**	**28.05**	**34.72**	0.4	**0.64**	**-0.61**
SVM	0.78	24.22	27.65	0.44	0.83	-2.54	0.76	38.53	46.53	0.5	0.25	-2.22
ARIMA_SVM	0.7	36.92	42.9	0.5	0.79	-12.52	0.59	85.69	93.73	0.47	0.44	-17.58
	24h						5 days					
ARIMA	**0.82**	19.81	24.16	0.6	**0.75**	-0.62	0.80	**33.78**	**40.35**	1.35	0.31	-0.90
LSTM	**0.82**	21.48	27.22	0.42	0.67	-0.58	0.79	40.49	48.30	0.84	0.38	-1.31
ARIMA_LSTM	**0.82**	19.94	25.29	**0.41**	0.66	-0.44	0.79	39.47	47.33	0.80	0.37	-1.21
RF	0.79	26.11	33.26	0.52	0.6	-1.22	0.75	49.86	60.77	**0.34**	0.24	-2.97
ARIMA_RF	**0.82**	19.92	25.42	0.46	0.59	-0.27	0.75	47.87	58.74	0.36	0.35	-3.09
KNN	0.79	24.78	31.94	0.47	0.56	-0.99	0.80	36.17	44.50	0.47	0.29	**-0.85**
ARIMA_KNN	0.81	**19.78**	**23.98**	0.67	**0.75**	**-0.26**	**0.81**	37.09	45.00	0.43	**0.53**	-1.20
SVM	0.77	32.14	38.01	0.54	0.67	-3.48	0.77	42.95	51.50	0.58	0.20	-1.67
ARIMA_SVM	0.66	58.15	67.13	0.68	0.64	-11	0.56	113.02	122.27	0.45	0.32	-17.11

Table 6
Average results of various forecasting algorithms (hybrid approaches based on function 7 - Model 2) on Vu Quang dataset.

Method	Sim	MEA	RMSE	FSD	R	NSE	Sim	MEA	RMSE	FSD	R	NSE
	12h						72h					
ARIMA	0.67	10.04	11.63	1.15	0.92	-4.38	0.756	42.35	53.03	1.55	0.44	-1.65
ARIMA_LSTM2	0.671	10.26	11.835	1.18	0.89	-4.45	0.758	42.212	52.868	1.557	0.454	-1.65
ARIMA_RF2	0.666	10.31	11.86	1.14	0.78	-4.48	0.758	42.191	52.832	1.552	0.422	-1.64
ARIMA_KNN2	0.669	10.37	11.889	1.13	0.84	-4.71	0.758	42.212	52.853	1.561	0.44	-1.64
ARIMA_SVM2	0.665	10.5	12.004	1.16	0.81	-4.91	0.757	42.26	52.887	1.568	0.45	-1.65
	24h						5 days					
ARIMA	0.7	16.82	19.97	1.11	0.74	-3.18	0.74	62.12	74.51	1.66	0.43	-2.06
ARIMA_LSTM2	0.7	16.6	19.77	1.11	0.74	-3.17	0.737	62.7	75.277	1.667	0.426	-2.05
ARIMA_RF2	0.7	16.61	19.75	1.11	0.73	-3.14	0.737	62.67	75.255	1.664	0.421	-2.04
ARIMA_KNN2	0.7	16.64	19.75	1.11	0.75	-3.19	0.737	62.69	75.263	1.669	0.42	-2.04
ARIMA_SVM2	0.69	16.77	19.88	1.13	0.72	-3.25	0.737	62.71	75.273	1.675	0.427	-2.04
	48h											
ARIMA	0.74	29.69	36.93	1.42	0.59	-2.04						
ARIMA_LSTM2	0.75	29.37	36.71	1.44	0.56	-2.04						
ARIMA_RF2	0.75	29.36	36.68	1.42	0.57	-2.04						
ARIMA_KNN2	0.75	29.37	36.7	1.44	0.56	-2.04						
ARIMA_SVM2	0.75	29.45	36.76	1.45	0.56	-2.05						

days (Fig 10b), 3 days (Fig 10c) and 5 days (Fig 10d) ahead. Again, it is evidenced from the Figures that ARIMA_RF yields the best results: 1 day-ahead and 2 days-ahead forecast values are close to the real ones. When forecasting 3 days and 5 days in advance, the ARIMA_RF could capture quite well the trend of the actual data, but the forecasting errors are rather high. Other methods do not work well for all these cases.

Fig. 11 demonstrates the results of forecasting the water level before the peak (August 21, 2016) at Vu Quang station with 1 day (Fig 11a), 2 days (Fig 11b), 3 days (Fig 11c) and 5 days (Fig 11d) ahead. The Figure shows a similar results to the case of Hanoi station in that ARIMA_RF could forecast quite accurately the peak with 1 and 2 days ahead. It still captures rather well the trend of the data in the 3 and 5 day ahead forecasts but the errors are high. Meanwhile, all of other models failed to forecast the peak and could not capture the data trend.

For Hung Yen data with hourly sampling, the highest water level was on July 23, 2014 at 22h (368 cm). We forecast the water levels before the peak 6h (Fig 12a), 12h (Fig 12b), 24h days (Fig 12c) and 3 days (Fig 12d). Fig. 12 demonstrates once again that the hybrid approach is superior to the single component methods. When predicting water levels 6h-ahead, ARIMA_KNN produces the closest values to the real ones (Fig 12a). However, when forecasting water levels with 12h and 24h in advance, ARIMA_KNN no longer gives good results as in the case of 6h. It is always that ARIMA_RF produces predicted values, which are more similar to the true ones than other methods (Fig 12b, 12 c). Although ARIMA_KNN is second behind ARIMA_RF, the gaps between the forecast errors of the two methods are rather wide. For HungYen dataset, when predicting 3 day ahead of the peak, only ARIMA_RF could capture the trend of data, but the forecast error is relatively large.

5. Conclusion

Forecasting accurately time series, especially water level for flood warning systems, is a very important but challenging task. ARIMA, KNN, RF, SVM and LSTM are widely popular and effective forecasting models proposed and tested on hydrological time series in the literature. ARIMA can model well linear, whereas the other statistical machine learning models are most suitable for nonlinear time series. However, in reality, a (hydrological) time series often includes both linear and nonlinear correlation structures. Therefore, in this paper, we have proposed two types of hybrid approaches in order to improve the forecasting performance. They have taken the advantages of each individual model type (i.e. linear and non-linear) and complexity levels (i.e. single or hybrid) in time series forecasting. The first type of hybrid approach (Model 1) is to combine the original data and the residuals obtained from ARIMA to build the predictive models, namely ARIMA_KNN, ARIMA_RF, ARIMA_SVR, and ARIMA_LSTM. The second hybrid type is to perform predictions on the original data and ARIMA's residuals then to aggregate the forecast results (Model 2). These models have been tested on three real big datasets on the water level time series of the Red river and compared with each single component model. The experimental results showed that the combined methods ARIMA_RF and ARIMA_KNN of Model 1 yield superior and more reliable results than other hybrid models (e.g. ARIMA_SVM and ARIMA_LSTM), as well as better than the traditional single component methods - ARIMA, KNN, RF, SVM and LSTM. In future, we are planning to apply our novel methods (ARIMA_RF and ARIMA_KNN) for forecasting water level on other horological stations of the Red river as well as for other rivers.

Declaration of Competing Interest

All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version. This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue. The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.

CRediT authorship contribution statement

Thi-Thu-Hong Phan: Conceptualization, Methodology, Formal analysis, Writing - original draft, Writing - review & editing, Software. **Xuan Hoai Nguyen:** Conceptualization, Methodology, Formal analysis, Writing - original draft, Writing - review & editing.

Acknowledgments

The research presented in this paper was funded by National Science and Technology Program to respond to climate change, manage natural resources and the environment in the period 2016-2020 in the project titled "Applications of Artificial Intelligence for Forecasting Hydro-Meteorological Anomalies in Vietnam under the Context of Climate Change", Grant Number: BĐKH.34/16-20.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.advwatres.2020.103656

References

Adnan, R.M., Yuan, X., Kisi, O., Adnan, M., Mehmood, A., 2018. Stream flow forecasting of poorly gauged mountainous watershed by least square support vector machine, fuzzy genetic algorithm and M5 model tree using climatic data from nearby station. Water Resour. Manage. 32 (14), 4469–4486.

Aho, K., Derryberry, D., Peterson, T., 2014. Model selection for ecologists: the worldviews of AIC and BIC. Ecology 95 (3), 631–636.

Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. 46, 175–185.

Anderson, P.L., Meerschaert, M.M., Zhang, K., 2013. Forecasting with prediction intervals for periodic autoregressive moving average models: FORECASTING WITH PREDICTION INTERVALS FOR PARMA MODELS. J. Time Ser. Anal. 34 (2), 187–193.

Bafitlhile, T.M., Li, Z., 2019. Applicability of -support vector machine and artificial neural network for flood forecasting in humid, semi-humid and semi-arid basins in china. Water (Basel) 11 (1), 85.

Birylo, M., Rzepecka, Z., Kuczynska-Siehien, J., Nastula, J., 2018. Analysis of water budget prediction accuracy using ARIMA models. Water Sci. Technol. 18 (3), 819–830.

Box, G., Jenkins, G.M., 1976. Time series analysis: forecasting and control. Holden-Day.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

, 2015. Cyclostationarity: theory and methods - II: contributions to the 7th workshop on cyclostationary systems and their applications, grodek, poland, 2014. In: Chaari, F., Leskow, J., Napolitano, A., Zimroz, R., Wylomanska, A., Dudek, A. (Eds.). Applied Condition Monitoring, 3. Springer International Publishing, Cham.

Chen, K.-Y., Wang, C.-H., 2007. A hybrid SARIMA and support vector machines in forecasting the production values of the machinery industry in taiwan. Expert Syst. Appl. 32 (1), 254–264.

Chen, L., Sun, N., Zhou, C., Zhou, J., Zhou, Y., Zhang, J., Zhou, Q., 2018. Flood forecasting based on an improved extreme learning machine model combined with the backtracking search optimization algorithm. Water (Basel) 10 (10), 1362.

Choi, C., Kim, J., Han, H., Han, D., Kim, H.S., 2019. Development of water level prediction models using machine learning in wetlands: a case study of upo wetland in south korea. Water (Basel) 12 (1), 93.

Christopher, O., 2015. Understanding LSTM Networks – colah's blog.

Dang, M., De Smedt, F., 2017. A combined hydrological and hydraulic model for flood prediction in vietnam applied to the huong river basin as a test case study. Water (Basel) 9 (11), 879.

Di, C., Yang, X., Wang, X., 2014. A four-stage hybrid model for hydrological time series forecasting. PLoS ONE 9 (8), e104663.

Ömer Faruk, D., 2010. A hybrid neural network and arima model for water quality time series prediction. Eng. Appl. Artif. Intell. 23 (4), 586–594.

Fatichi, S., Vivoni, E.R., Ogden, F.L., Ivanov, V.Y., Mirus, B., Gochis, D., Downer, C.W., Camporese, M., Davison, J.H., Ebel, B., Jones, N., Kim, J., Mascaro, G., Niswonger, R., Restrepo, P., Rigon, R., Shen, C., Sulis, M., Tarboton, D., 2016. An overview of current applications, challenges, and future trends in distributed process-based models in hydrology. J. Hydrol. (Amst.) 537, 45–60.

Fernandez-Delgado, M., Cernadas, E., Barro, S., Amorim, D., 2014. Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. 3133–3181.

Garcia, F.C.C., Retamar, A.E., Javier, J.C., 2016. Development of a predictive model for on-demand remote river level nowcasting: Case study in Cagayan River Basin, Philippines. In: 2016 IEEE Region 10 Conference (TENCON). IEEE, Singapore, pp. 3275–3279.

Garsole, P., Rajurkar, M.P., 2015. Streamflow forecasting by using support vector regression. In: 20th International Conference on Hydraulics, Water Resources and River Engineering India, p. 8.

Ghimire, B.N., 2017. Application of ARIMA model for river discharges analysis. J. Nepal Phys. Soc. 4 (1), 27.

Ghumman, A., Ghazaw, Y.M., Sohail, A., Watanabe, K., 2011. Runoff forecasting by artificial neural network and conventional model. Alexandria Eng. J. 50 (4), 345–350.

Gjika, E., Ferrja, A., Kamberi, A., 2019. A study on the efficiency of hybrid models in forecasting precipitations and water inflow albania case study. Adv. Sci. Technol. Eng. Syst. J. 4 (1), 302–310.

Hamid, N., Alireza, I., Mahdi, S., Mahdi, A., 2019. Comparing three main methods of artificial intelligence in flood estimation in yalphan catchment. J. Geograph. Environ. Plan. 29 (4).

Haseyama, M., Kitajima, H., 2001. An arma order selection method with fuzzy reasoning. Signal Process. 81, 1331–1335.

Hipni, A., El-shafie, A., Najah, A., Karim, O.A., Hussain, A., Mukhlisin, M., 2013. Daily forecasting of dam water levels: comparing a support vector machine (SVM) model with adaptive neuro fuzzy inference system (ANFIS). Water Resour. Manage. 27 (10), 3803–3823.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Hrnjica, B., Bonacci, O., 2019. Lake level prediction using feed forward and recurrent neural networks. Water Resour. Manag.: Int. J. Publ. Eur. Water Resour. Assoc. (EWRA) 33(7), 2471–2484.

Hurvich, C.M., Tsai, C.-L., 1989. Regression and time series model selection in small samples. Biometrika 76 (2), 297–307.

Hyndman, R.J., Khandakar, Y., 2008. Automatic time series forecasting: the forecast package for R. J. Stat. Softw. 26 (3), 1–22.

Kasiviswanathan, K., He, J., Sudheer, K., Tay, J.-H., 2016. Potential application of wavelet neural network ensemble to forecast streamflow for flood management. J. Hydrol. (Amst.) 536, 161–173.

Khan, M.S., Coulibaly, P., 2006. Application of support vector machine in lake water level prediction. J. Hydrol. Eng. 11 (3), 199–205.

Khashei, M., Bijari, M., 2010. An artificial neural network (p,d,q) model for timeseries forecasting. Expert Syst. Appl. 37 (1), 479–489.

Kim, S.E., Seo, I.W., 2015. Artificial neural network ensemble modeling with conjunctive data clustering for water quality prediction in rivers. J. Hydro-Environ. Res. 9 (3), 325–339.

Kumar, P., Lohani, A., Nema, A., 2019. Rainfall runoff modeling using MIKE 11 nam model. Curr. World Environ. 14 (1), 27–36.

Le, Ho, Lee, Jung, 2019. Application of long short-term memory (LSTM) neural network for flood forecasting. Water (Basel) 11 (7), 1387.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.
Ljung, L., 1986. System identification: Theory for the user. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
Mirzavand, M., Ghazavi, R., 2015. A stochastic modelling technique for groundwater level forecasting in an arid environment using time series methods. Water Resour. Manage. 29, 1315–1328.
Mohan, B.R., Reddy, G.R.M., 2018. A hybrid ARIMA-ANN model for resource usage prediction. Int. J. Pure Appl. Math. 119, 10.
Mousavi-Mirkalaei, P., Banihabib, M.E., 2019. An ARIMA-NARX hybrid model for forecasting urban water consumption (case study: tehran metropolis). Urban Water J. 16 (5), 365–376.
Nash, J.E., Sutcliffe, J.V., 1970. River flow forecasting through conceptual models part I A discussion of principles. J. Hydrol. (Amst) 10 (3), 282–290.
Nazir, H.M., Hussain, I., Faisal, M., Elashkar, E.E., Shoukry, A.M., 2019. Improving the prediction accuracy of river inflow using two data pre-processing techniques coupled with data-driven model. PeerJ 7.
Nguyen, T.-T., Huu, Q.N., Li, M.J., 2015. Forecasting Time Series Water Levels on Mekong River Using Machine Learning Models. In: 2015 Seventh International Conference on Knowledge and Systems Engineering (KSE). IEEE, Ho Chi Minh City, Vietnam, pp. 292–297.
Pannakkong, W., Pham, V.-H., Huynh, V.-N., 2017. A novel hybridization of ARIMA, ANN, and k-means for time series forecasting:. Int. J. Know. Syst. Sci. 8 (4), 30–53.
Pasupa, K., Jungjareantrat, S., 2016. Water levels forecast in Thailand: a case study of Chao Phraya river. In: 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV). IEEE, Phuket, Thailand, pp. 1–6.
Patro, S., Chatterjee, C., Mohanty, S., Singh, R., Raghuwanshi, N.S., 2009. Flood inundation modeling using MIKE FLOOD and remote sensing data. J. Indian Soc. Remote Sens. 37 (1), 107–118.
Peng, T., Zhou, J., Zhang, C., Fu, W., 2017. Streamflow forecasting using empirical wavelet transform and artificial neural networks. Water (Basel) 9 (6), 406.
Phan, T.-T.-H., Poisson Caillault, EP., Lefebvre, A., Bigand, A., 2017. Dynamic time warping-based imputation for univariate time series data. Pattern Recognit. Lett..
Rezaie-Balf, M., Kim, S., Fallah, H., Alaghmand, S., 2019. Daily river flow forecasting using ensemble empirical mode decomposition based heuristic regression models: application on the perennial rivers in iran and south korea. J. Hydrol. (Amst.) 572, 470–485.
Riad, S., Mania, J., Bouchaou, L., Najjar, Y., 2004. Rainfall-runoff model usingan artificial neural network approach. Math. Comput. Model. 40 (7–8), 839–846.
Seo, Y., Kim, S., Kisi, O., Singh, V.P., 2015. Daily water level forecasting using wavelet decomposition and artificial intelligence techniques. J. Hydrol. (Amst.) 520, 224–243.
Shibata, R., 1976. Selection of the order of an autoregressive model by akaike's information criterion. Biometrika 63 (1), 117–126.
Temr, A.S., Akgn, M., Temr, G., 2019. Predicting housing sales in turkey using arima, LSTM and hybrid models. J. Bus. Econ. Manag. 20 (5), 920–938.
Toro, C.H.F., Gmez Meire, S., Glvez, J.F., Fdez-Riverola, F., 2013. A hybrid artificial intelligence model for river flow forecasting. Appl. Soft. Comput. 13 (8), 3449–3458.
Tran Anh, D., Hoang, L.P., Bui, M.D., Rutschmann, P., 2018. Simulating future flows and salinity intrusion using combined one- and two-dimensional hydrodynamic modelling the case of hau river, vietnamese mekong delta. Water (Basel) 10 (7), 897.
Valenzuela, O., Rojas, I., Rojas, F., Pomares, H., Herrera, L., Guillen, A., Marquez, L., Pasadas, M., 2008. Hybridization of intelligent techniques and ARIMA models for time series prediction. Fuzzy Sets Syst. 159 (7), 821–845.
Valipour, M., Banihabib, M.E., Behbahani, S.M.R., 2013. Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of dez dam reservoir. J. Hydrol. (Amst). 476, 433–441.
Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, Berlin, Heidelberg.
Wang, X., Liu, T., Zheng, X., Peng, H., Xin, J., Zhang, B., 2018. Short-term prediction of groundwater level using improved random forest regression with a combination of random features. Appl. Water Sci. 8 (5), 125.
Wongsathan, R., Jaroenwiriyapap, W., 2016. A hybrid ARIMA and RBF neural network model for tourist quantity forecasting: a case study for chiangmai province. KKU Res. J. 21(1), 37–54.
Woodward, W.A., Gray, H.L., Elliott, A.C., 2017. Applied time series analysis, with r, second edition CRC Press, Taylor & Francis Group, Boca Raton.
Wu, C., Chau, K., Li, Y., 2009. Methods to improve neural network performance in daily flows prediction. J. Hydrol. (Amst.) 372 (1–4), 80–93.
Xie, Y., Lou, Y., 2019. Hydrological Time Series Prediction by ARIMA-SVR Combined Model based on Wavelet Transform. In: Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence - ICIAI 2019. ACM Press, Suzhou, China, pp. 243–247.
Xu, G., Cheng, Y., Liu, F., Ping, P., Sun, J., 2019. A Water Level Prediction Model Based on ARIMA-RNN. In: 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService). IEEE, Newark, CA, USA, pp. 221–226.
Yan, H., Zou, Z., 2013. Application of a hybrid ARIMA and neural network model to water quality time series forecasting. J. Converg. Inf. Technol.(JCIT) 8.
Yang, J.-H., Cheng, C.-H., Chan, C.-P., 2017. A time-series water level forecasting model based on imputation and variable selection method. Comput. Intell. Neurosci..
Yaseen, Z.M., Fu, M., Wang, C., Mohtar, W.H.M.W., Deo, R.C., El-shafie, A., 2018. Application of the hybrid artificial neural network coupled with rolling mechanism and grey model algorithms for streamflow forecasting over multiple time horizons. Water Resour. Manage. 32 (5), 1883–1899.
Yu, Z., Lei, G., Jiang, Z., Liu, F., 2017. ARIMA modelling and forecasting of water level in the middle reach of the Yangtze River. In: 2017 4th International Conference on Transportation Information and Safety (ICTIS). IEEE, Banff, AB, Canada, pp. 172–177.
Zhang, G.P., 2003. Time series forecasting using a hybrid arima and neural network model.. Neurocomputing 50, 159–175.
Zhong, C., Guo, T., Jiang, Z., Liu, X., Chu, X., 2017. A hybrid model for water level forecasting: a case study of Wuhan station. In: 2017 4th International Conference on Transportation Information and Safety (ICTIS). IEEE, Banff, AB, Canada, pp. 247–251.
Zhong, C., Jiang, Z., Chu, X., Guo, T., Wen, Q., 2017. Water level forecasting using a hybrid algorithm of artificial neural networks and local kalman filtering. Proc. Inst. Mech. Eng. Part M 233 (1), 174–185.
Zhu, C., Liang, Q., Yan, F., Hao, W., 2013. Reduction of waste water in erhai lake based on MIKE21 hydrodynamic and water quality model. Sci. World J. 2013, 1–9.