Forecasting time series water levels on Mekong river using machine learning models

Thanh-Tung Nguyen
Faculty of Computer Science
and Engineering, Thuyloi University
Hanoi, Vietnam
Email: tungnt@tlu.edu.vn

Quynh Nguyen Huu
Information Technology Faculty,
Electric Power University
Hanoi, Vietnam
Email: quynhnh@epu.edu.vn

***Abstract*—Forecasting water levels on Mekong river is an important problem needed to be studied for flood warning. In this paper, we investigate the application to forecasting of daily water levels at Thakhek station on Mekong river using machine learning models such as LASSO, Random Forests and Support Vector Regression (SVR). Experimental results showed that SVR was able to achieve feasible results, the mean absolute error of SVR is $0.486(m)$ while the acceptable error of a flood forecast model required by the Mekong River Commission is between $0.5(m)$ and $0.75(m)$.**

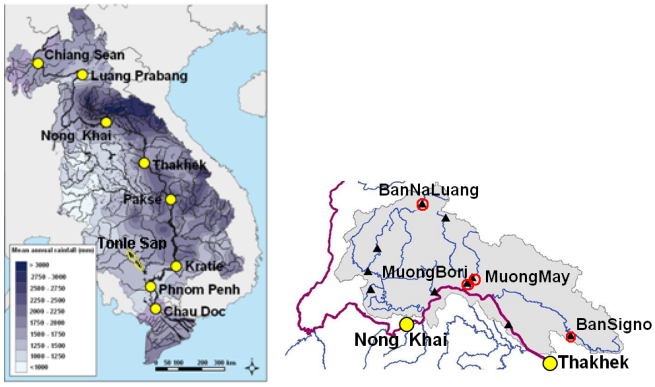
***Keywords*—Time series forecasting, LASSO, Random Forests, Support Vector Regression, Data mining**

I. Introduction

The Lower Mekong Basin lies within the countries of Thailand, Laos, Cambodia and Vietnam (Figure 1 (a)) and it is home for about 30 million people. It is both geographically and hydrologically variable with significant differences in topography, land cover, and climate. Elevation is highest in the north-east and reduces in the direction from north-east to south-west. The basin is generally flat in the floodplain of Cambodia and the Mekong Delta in Vietnam. So, the flood warning system is important to mitigate nature damages on floodplain of Mekong river. The accurate result of the water level forecast with lead days are effective information for flood warning.

There are two approaches to build the flood forecasting model generally, that are physically based and data-driven approaches. The goal of data-driven model (also called machine learning model) is to find the relationship between the input attributes and the water level while the physically based modelling aims to reproduce the hydrological process in a physically realistic. Physically based models are fully distributed models in increasing levels of complexity [1], a distributed model requires more time to develop the forecasting system and the sufficient expertise is needed to interpret the results. Data-driven model is quickly developed and easily implemented for building the forecasting model, it is useful for real-time river flow forecasting with accurate water level forecasts. Our approach based on data-driven model using machine learning method to forecast the water level for the flood warning system of the Mekong river.

Let $\mathcal{L} = \{(X_1, Y_1), .., (X_N, Y_N)\}$ be a input data set in the data-driven model, where Y is a continuous response *feature* or *attribute*, Y indicates the water level, X is a feature matrix



(a) (b)

Fig. 1. The Mekong River (research basin). (a) Rainfall distribution for the Lower Mekong Basin. (b) Sub-basin: 10,407 (km^2)

of N observed *samples* (also called *instances*) and M features. The task of a forecasting method in the data-driven model is to obtain an unknown function $f(X)$ from given input data set $\mathcal{L}$. The goal is to find a data-driven model such that its predictions forecasting by the function $f(\cdot)$, also denoted $\hat{Y}$, are as good as possible. Without loss of generality, the forecasting model is defined as a function $f : X \to Y$, where $X \in \mathbb{R}^M$ and $Y \in \mathbb{R}^1$. In the statistic sense, $X \in \mathbb{R}^M$ and $Y \in \mathbb{R}^1$ are *random features* with joint distribution $P(X, Y)$. That is, $P(X = x, Y = y)$ is the probability that random features X and Y take values x and y, respectively. We find a function $f(X)$ for forecasting Y given $X = x$ with the most used *loss function* $L(Y, f(X)) = (Y - f(X))^2$ where all errors are penalized in the forecast.

In this paper, we investigate data-driven models for time series data and apply the models to forecast the 5-lead-day water levels at Thakhek station on the Mekong River, where it shows the major contribution to the flows in the Lower Mekong River. Research locations are illustrated in the Figure 1. In the forecasting problem of 5 lead days, the relationship between the input-output features in the machine learning model is as follows:

$$H_{Thakhek}(t+5) = f(H_{Thakhek}(t), H_{Thakhek}(t-1), H_{Thakhek}(t-2), H_{up}(t), H_{up}(t-1), H_{up}(t-2)).$$

where the output feature $H_{Thakhek}(t+5)$ is the water level forecasted for the next 5 days at Thakhek gauging station.

$H_{Thakhek}(t)$, $H_{Thakhek}(t-1)$ and $H_{Thakhek}(t-2)$ are water levels measured in the current day and previous two days, respectively. $H_{up}(t)$, $H_{up}(t-1)$ and $H_{up}(t-2)$ are water levels measured in the current day and previous two days at $NongKhai$ gauging station, respectively. This is the upstream gauging station of Thakhek, with distance of 300 km upstream of the Mekong River.

Experimental results showed that our machine learning approaches including LASSO, Random Forests and Support Vector Regression, are efficient models to forecast accurate 5-lead-days based on the operational requirements of the Mekong River Commission. The Support Vector Regression model was able to achieve the lowest mean absolute error $(0.486(m))$ while the acceptable of a flood forecast model is in $[0.5(m), 0.75(m)]$. These data-driven models can be also applied to forecast 1, 2, 3, 4 lead days for the flood warning system.

II. Methodology - Data used

A. Data used and design of forecast evaluation

Continuous daily water level data for 1994 to 2003 were analyzed for the present study. In order to make forecasting in time series water levels, the data needs to be modified such that if return at time t needs to be forecasted then all the historical information water level until time t is used. By doing so, the latest information related to water levels is used in forecasting the machine learning model. The data from 1994 to 2000 (1071 data samples) were used for initial training and 2001 to 2003 (459 data samples) for testing. For each iteration, 1 sample from the testing data is added into the training data to build the forecasting model. The water level is linearly normalized by the formula:

$$X_{norm} = F_{min} + \frac{(X_i - X_{min})}{(X_{max} - X_{min})} \times (F_{max} - F_{min}),$$

where F_{max} and F_{min} are the smallest and largest values in the normalized domain respectively selected as 0.1 and 0.9; X_{norm} is normalized value, X_{min} and X_{max} are the smallest and largest value in the measurement data time series, X_i is the observed measurement value before normalization.

B. LASSO

We first introduce the multivariate regression model and it then is extended for LASSO model. Given a training data set $\mathcal{L}$, we consider the linear regression model which has the form

$$Y = E(Y|X) + \epsilon, \tag{1}$$

where the error ϵ is independent standard normal with expectation zero and variance σ^2, written $\epsilon \sim N(0, \sigma^2)$ and

$$E(Y|X) = \beta_0 + \sum_{i=1}^{N} X_i \beta_i. \tag{2}$$

The term β_0 is the intercept and β_i's are coefficients, our aim is to predict a real-value output Y, denote as $\hat{Y}$. The popular approaches based on the least mean square *(LMS)* method [2], in which we pick the coefficients $\beta = (\beta_0, \beta_1, .., \beta_M)^T$ to minimize the residual sum of squares (RSS)

$$\begin{aligned} RSS(\beta) &= \sum_{i=1}^{N} (Y_i - E(Y|X))^2 \\ &= \sum_{i=1}^{N} (Y_i - \beta_0 - \sum_{j=1}^{M} X_i \beta_i)^2. \end{aligned} \tag{3}$$

We determine the values $\hat{\beta}$ for the regression coefficients and error variance consistent, assumes that the linear model is a reasonable approximation, the error term is normally distributed $N(0, \sigma^2)$, and the matrix $X = N \times (M+1)$ has full column rank, see more details about its main assumptions in [2]. Equation (4) can be written as

$$RSS(\beta) = (Y - X\beta)^T (Y - X\beta). \tag{4}$$

If $X^T X$ is nonsingular, then the unique solution is given by

$$\hat{\beta} = (X^T X)^{-1} X^T Y, \tag{5}$$

the fitted values at the training inputs are

$$\hat{Y} = X\hat{\beta} = X(X^T X)^{-1} X^T Y. \tag{6}$$

Tibshirani proposed LASSO method (Least absolute shrinkage and selection operator) [3] which performs coefficient shrinkage and feature selection simultaneously. It minimizes the squared error $(Y - E(Y|X))^2$ subjected to the constraint that the sum of absolute values of coefficients should be less than a constant, denoted as s. We have $\sum_{j=1}^{M} |\beta_j| \leq s$, where s is known as a tuning parameter controlling the amount of shrinkage to be applied to the estimates. The β coefficients in Equation (5) can be estimated using LASSO as

$$\hat{\beta}_{LASSO} = argmin(Y - X\beta)^T (Y - X\beta) + \lambda \sum_{j=1}^{M} |\beta_j|. \tag{7}$$

The constraint will have no effect when s is large enough and the solution obtained will be just as with multiple linear least squares regression. We can use k-folds cross validation technique to estimate the best value of s.

C. Random Forests

Random forests (RF) had been proposed by Breiman [4], although many of the ideas had proposed earlier in the literature in different forms. The classification and regression tree (CART) was described as a non-parametric method for supervised learning in 1984 by Breiman et al. [5], who introduced bagging method to reduce prediction variance for CART in 1996 [6], a precursor to his version of RF. In 1995, Ho [7] introduced the *random decision forest* and used the same mind of trees grown in random subspaces of the features. In 1997 and 1998, Amit and Geman [8] and Ho [9] dependently proposed that trees grow only on a randomly chosen subspace, instead of whole feature space in bagging [6]. Dietterich [10] also proposed an improvement on bagging using additional randomization in 2000. Finally, Breiman reinforced, aggregated and extended these approaches and proposed a new framework in 2001, which is called *Random Forests*.

RF is an extension of bagging (bootstrap aggregation) [6] which involves the idea of a random subspace sampling to create dissimilar trees. Given a training data $\mathcal{L}$, a standard version of RF is a collection of CART built from bagged

samples. Each CART is grown from an independent bootstrap resample until all nodes contain observations less than or equal to the predefined minimum node size n_{min}. Each CART is grown nondeterministically, unlike that of a single tree. Each CART in the forest then provides a prediction of a Y value, and a single overall prediction is obtained by taking an average over the tree's prediction from the forest.

The number of candidate features that are selected for each iteration step is an additional tuning parameter, called $mtry$, $1 < mtry < M$. There is a special case in random forests when $mtry = M$, that is bagging. One side-effect of choosing $mtry < M$ is also a less time affecting algorithm. A popular choice is to randomly select $mtry = \sqrt{M}$ in classification and $mtry = M/3$ in regression as candidates for node splitting for data set of moderate size. For high dimensionality, RF is an efficient method to deal with this kind of data, especially the number of feature is greater than the number of samples [11], [12]. Breiman suggested in his work [4], the *minimum node size* $n_{min} = 1$ in the classification problem and $n_{min} = 5$ in the regression problem. This enables a more dissimilar subset of features to contribute to the joint prediction of a Random Forest, which results in an improved prediction accuracy.

Given an input $X = x$, the predicted value by the whole RF is obtained by aggregating the results given by individual trees. Denote K be the number of trees in the forest and $\hat{f}_k(x)$ be the prediction of unknown value y of input $x \in \mathbb{R}^M$ by kth tree ($k = 1..K$), respectively. The RF prediction is

$$\hat{Y} = \frac{1}{K}\sum_{k=1}^{K}\hat{f}_k(x). \tag{8}$$

D. Support Vector Regression

The Support Vector Regression (SVR) is an extension of the classic Support Vector Machines algorithm proposed by Vladimir Vapnik [13]. The basic idea is to find a linear function that has at most a predetermined deviation ε from the actual values of the input data $\mathcal{L}$. In other words, we do not care about the error of each forecast as long as it does not violate the predefined threshold, but we penalize any deviations higher than the threshold. In ε-SV, we aim to find a function $f(X)$ achieving the smallest ε:

$$f(X) = w^T\Psi(X) + b, \tag{9}$$

where $w \in \mathbb{R}^M$ and $b \in \mathbb{R}$, $\Psi(X)$ is a non-linear function to map the M-dimension in the input data into a higher dimensional feature space where linear regression is performed. The goal is to find the value of w, b such that values of X can be determined by minimizing the loss function $\min(\frac{1}{2}\|w\|^2)$ subject to $|Y - (w^T X + b)| \le \varepsilon$.

Expanding this initial framework, Vapnik and Cortes [13] proposed a soft margin model. The optimal regression model is given by the minimum of a convex quadratic optimization problem [14]. They added slack variables ξ and ξ^* to the loss function controlled through a cost parameter C, minimizing the loss function $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$ subject to

$$\begin{cases} Y - (wX + b) - \xi \le \varepsilon \\ (wX + b) - Y - \xi^* \le \varepsilon \\ \xi, \xi^* \ge 0. \end{cases}$$

where C is a constant which determines penalties to estimation errors. So, the solution is given to solve the primal problem that is:

$$\begin{aligned} L = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) - \sum_{i=1}^{N}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ - \sum_{i=1}^{N}\alpha_i(\varepsilon + \xi_i - Y_i + w^T X_i + b) \\ - \sum_{i=1}^{N}\alpha_i^*(\varepsilon + \xi_i^* + Y_i - w^T X_i - b) \end{aligned} \tag{10}$$

where $\eta_i, \eta_i^*, \alpha_i, \alpha_i^* (i = 1..N)$ are the Lagrange multipliers from the Lagrangian primal function (10).

We take the first partial derivative, equation (10) can be transformed to a dual optimization problem. The non-linear SVR solution, using an ε-insensitive loss function, is given by

$$\begin{aligned} max\big(\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\Psi(X_i, X_j) \\ -\varepsilon\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) + \sum_{i=1}^{N}Y_i(\alpha_i - \alpha_i^*)\big) \end{aligned} \tag{11}$$

subject to $\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0$ and $\alpha_i, \alpha_j \in [0, C], i, j = 1..N$ The solution of the equation (10) is

$$\hat{w} = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)X_i$$

and

$$\hat{b} = -\frac{1}{2}\hat{w}(X_j + X_k),$$

where X_j, X_k are support vectors, $\alpha_i \in (0, C)$ and $\alpha_i^* \in (0, C)$.

SVR uses the kernel function to map initial data into a higher dimensional space were such a linear function exists. There are four popular kernel functions, see [14] in details. In our experiment, we use the *radial basis function* (RBF) defined as $exp(-\sigma\|X - u\|^2)$, where σ is a scaling parameter.

III. RESULTS

A. Model selection and evaluation

The three models LASSO, RF and SVR in the latest R-packages *lars*, *randomForest* and *kernlab* from CRAN[1] were used in an R environment to conduct these experiments, respectively. Each model was evaluated using two repeats of 10-fold cross-validation to find the optimal parameter.

The Mekong River Commission currently adopts benchmarks to evaluate the accuracy of 5-lead-day water level forecasts at each station along the river (http://ffw.mrcmekong.org/accuracy.htm). The performance of a flood forecast model is considered acceptable if MAE is greater than 0.5(m) and less than 0.75(m) of forecasts, computed as

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(|H_i - \hat{H}_i|) \tag{12}$$

[1] http://cran.r-project.org

TABLE I. PREDICTIVE PERFORMANCE FOR FORECASTING 5-LEAD-DAY OF WATER LEVELS ON THE MEKONG RIVER.

Model used	Parameter	CE	RMSE	MAE
LASSO	default	0.911	0.761	0.604
RF	$mtry = 2, K = 1000$	**0.936**	0.649	0.491
SVR	$\varepsilon = 0.1, C = 8, \sigma = 0.235$	0.935	**0.646**	**0.486**

where N is the total number of data samples, H_i and $\hat{H}_i$ are the observed and predicted water levels at the i^{th} time interval, respectively. In addition, root mean squared error (RMSE) and coefficient of the efficiency (CE) [15] were used also to evaluate the forecast models, defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(H_i - \hat{H}_i)^2}. \quad (13)$$

and

$$CE = 1 - \frac{\sum_{i=1}^{N}(H_i - \hat{H}_i)^2}{\sum_{i=1}^{N}(H_i - \bar{H}_i)^2}. \quad (14)$$

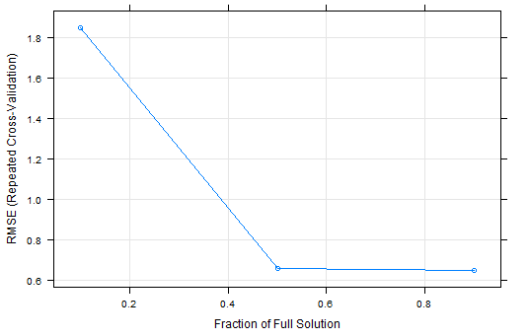
where $\bar{H}_i$ is the mean value of observed water levels.

Figure 2 (a) shows the RMSE profile of the LASSO model, a tuning parameter controlling the amount of shrinkage was chosen $s = 0.9$. Figure 2 (b) presents RMSE changed when the RF model performs with various $mtry$ values from 2 to 5 with 1000 trees in the forest, the optimal $mtry = 2$ was chosen using two repeats of 10-fold cross-validation. The SVR model was evaluated over cost values ranging from 2^{-2} to 2^{11} using RBF kernel function, figure 2 (c) shows the RMSE profile across the candidate values of the cost parameter, the optimal cost and kernel parameter were estimated to be $\varepsilon = 0.1$, $C = 2^0$ and $\sigma = 0.242$, respectively.

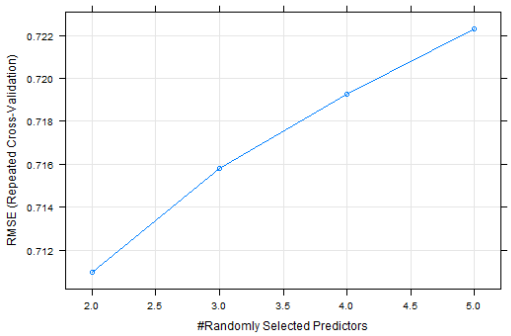
B. Experimental Results

Table I lists the results of three data-driven models to forecast the water level on the Mekong river, numbers in bold are the best results. It can be seen that the non-linear machine learning models outperform linear LASSO model when applied to time series water level data. The SVR model using RBF kernel function provides the lowest RMSE and MAE (meters), the RF model achieves the best result with CE measure but the difference is minor from the SVR model. This result indicates that the SVR model achieves a good performance when forecasted the water level at Thakhek station based on the requirement of Mekong River Commission. The acceptable of a flood forecast model is in $[0.5(m) \div 0.75(m)]$ (http://ffw.mrcmekong.org/accuracy.htm), it is worth noting that the SVR model produces MAE =$0.486(m)$ for 5-lead-day forecast.

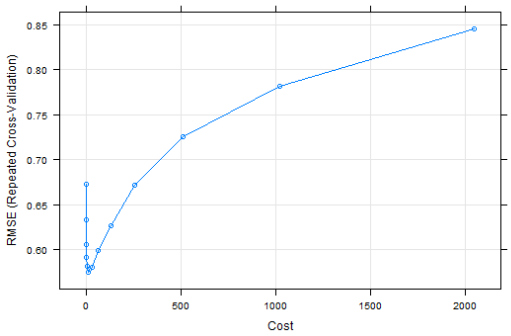
Figure 3, Figure 4 and Figure 5 plots results of time series from three models LASSO, RF and SVR, respectively. The observed water levels and forecast values from each model are showed to compare their forecast. The linear LASSO model is easy to implement and its computational time is fast, however its forecast water level at Thakhek station is far from the trend of observed water levels, shown in Figure 3. Significantly, two data-driven RF, SVR models have lower MAE and RMSE values compared to LASSO for 5-lead-day forecasting of water level at Thakhek gauging station, they follow the trend of observed water level closely.



(a) LASSO



(b) RF



(c) SVR

Fig. 2. The 10-folds cross-validation with 2 repetitions to estimate the optimal parameters for the three models. (a) LASSO. (b) Random Forests. (c) Support Vector Regression.

IV. CONCLUSION

We have presented some machine learning models to forecast time series water levels data for flood warning system. Our contribution is to investigate the application of LASSO, Random Forests and Support Vector Regression for flood forecasting in the mainstream of the Lower Mekong River. Model performance was assessed in terms of the inputs used to the machine learning models since the flow conditions at different locations along the mainstream of the station of interest. For Thakhek station, the experimental results achieve accurate forecast compared to the requirement of Mekong River Commission. The feasible MAE of a flood forecast model is in $[0.5(m), 0.75(m)]$, the forecast error for 5-lead-day in our experiment is MAE =$0.486(m)$.

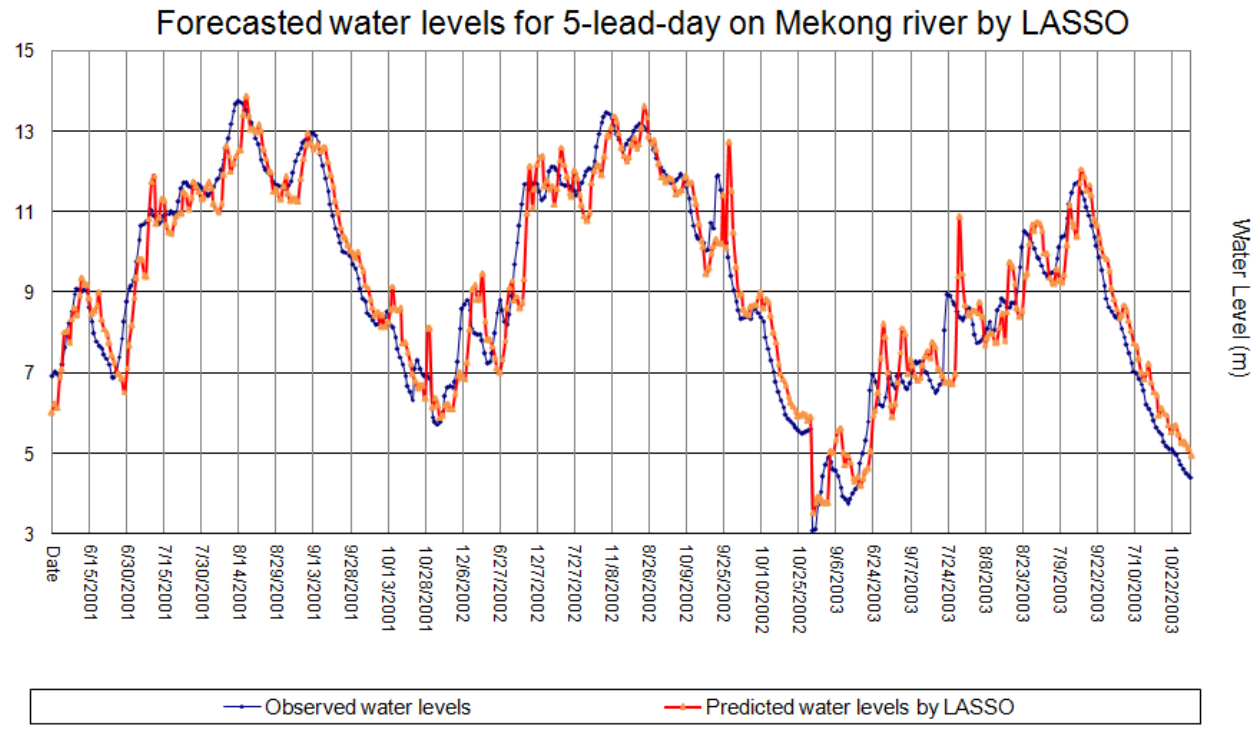


Fig. 3. The observed water levels and forecast values by the LASSO model.

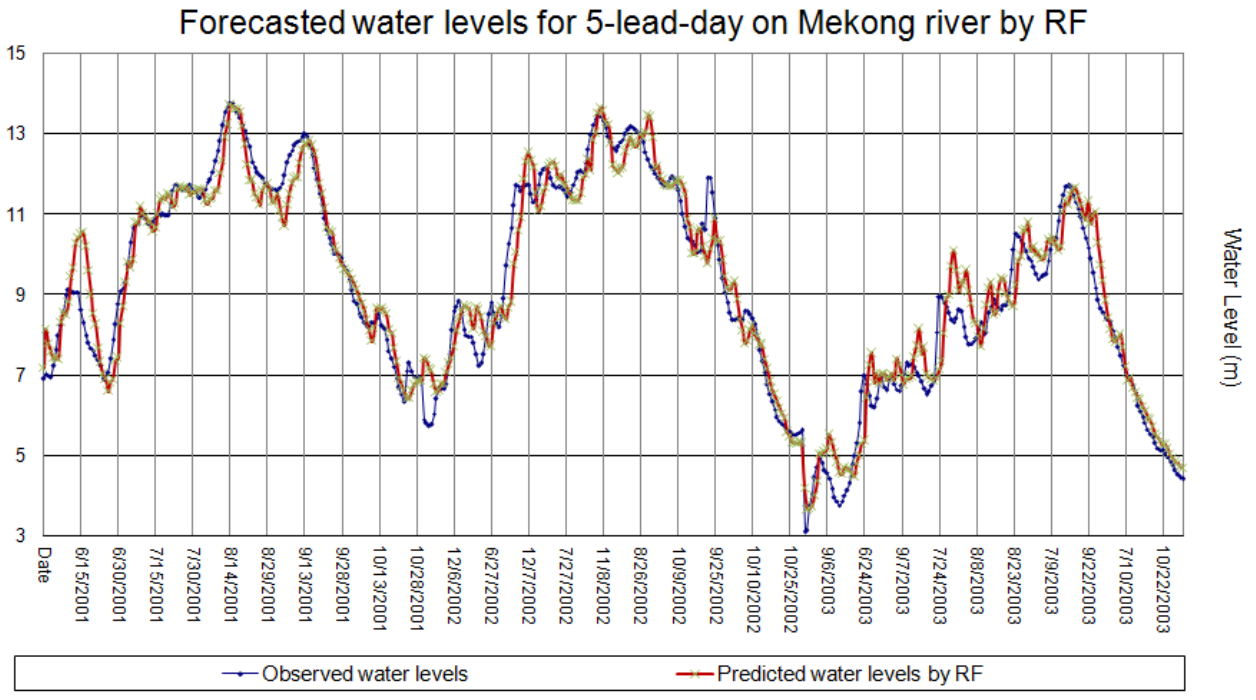


Fig. 4. The observed water levels and forecast values by the RF model.

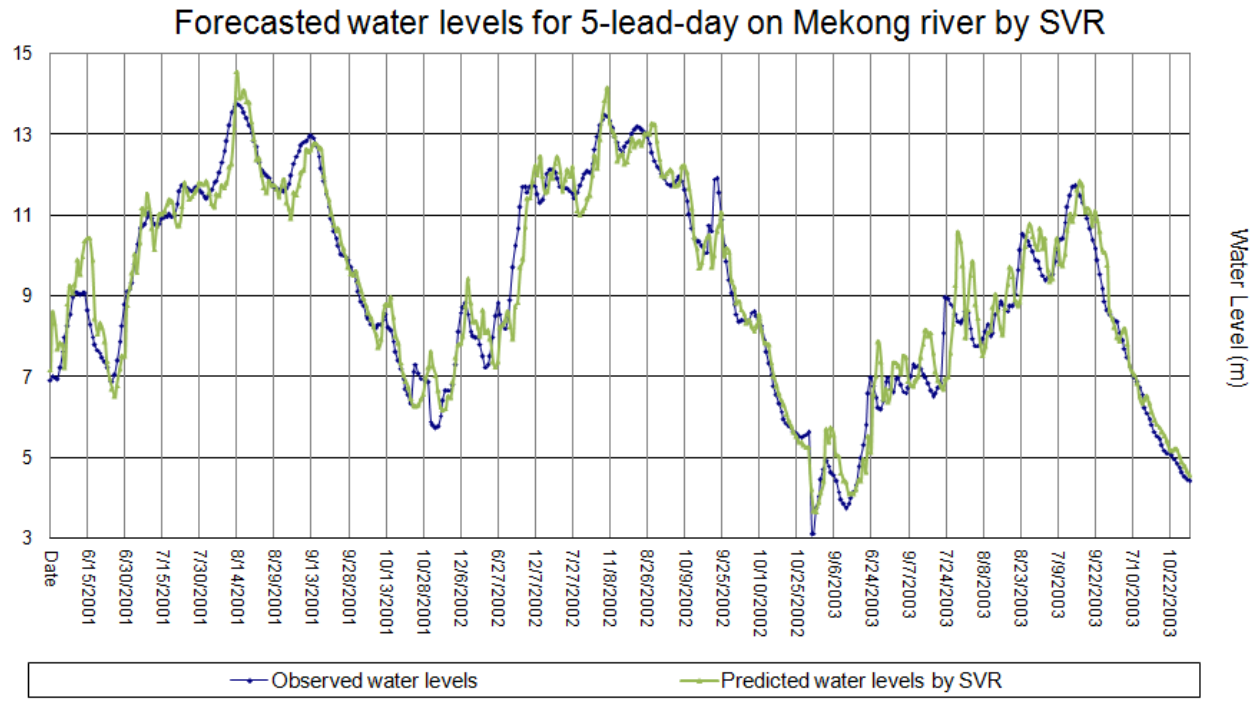


Fig. 5. The observed water levels and forecast values by the SVR model.

ACKNOWLEDGMENT

This research is supported in part by NSFC under Grant NO.61203294 and Natural Science Foundation of SZU (grant no. 201433), and the project "Advanced learning methods and their application in exploitation of electronic medical records" funded by the Vietnam Institute of Advanced Study of Mathematic (VIASM). The authors would like to thank Dr. Nguyen Khac Tien Phuoc who provides the data sets.

REFERENCES

[1] B. Calvo and F. Savi, "Real-time flood forecasting of the tiber river in rome," *Natural hazards*, vol. 50, no. 3, pp. 461–477, 2009.

[2] P. J. Huber, *Robust statistics*. Springer, 2011.

[3] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[4] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[6] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[7] T. K. Ho, "Random decision forests," in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 1. IEEE, 1995, pp. 278–282.

[8] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural computation*, vol. 9, no. 7, pp. 1545–1588, 1997.

[9] T. K. Ho, "The random subspace method for constructing decision forests," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 8, pp. 832–844, 1998.

[10] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine learning*, vol. 40, no. 2, pp. 139–157, 2000.

[11] T.-T. Nguyen, J. Huang, and T. Nguyen, "Two-level quantile regression forests for bias correction in range prediction," *Machine Learning*, pp. 1–19, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10994-014-5452-1

[12] T.-T. Nguyen, J. Z. Huang, Q. Wu, T. T. Nguyen, and M. J. Li, "Genome-wide association data classification and snps selection using two-stage quality-based random forests," *BMC Genomics*, vol. 16, no. Suppl 2, p. S5, 2015.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[14] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[15] J. Nash and J. V. Sutcliffe, "River flow forecasting through conceptual models part ia discussion of principles," *Journal of hydrology*, vol. 10, no. 3, pp. 282–290, 1970.