

# Prediction of hydrological time-series using extreme learning machine

Md Atiquzzaman and Jaya Kandasamy

## ABSTRACT

Applying feed-forward neural networks has been limited due to the use of conventional gradient-based slow learning algorithms in training and iterative determination of network parameters. This paper demonstrates a method that partly overcomes these problems by using an extreme learning machine (ELM) which predicts the hydrological time-series very quickly. ELMs, also called single-hidden layer feed-forward neural networks (SLFNs), are able to well generalize the performance for extremely complex problems. ELM randomly chooses a single hidden layer and analytically determines the weights to predict the output. The ELM method was applied to predict hydrological flow series for the Tryggevælde Catchment, Denmark and for the Mississippi River at Vicksburg, USA. The results confirmed that ELM's performance was similar or better in terms of root mean square error (RMSE) and normalized root mean square error (NRMSE) compared to ANN and other previously published techniques, namely evolutionary computation based support vector machine (EC-SVM), standard chaotic approach and inverse approach.

**Key words** | extreme learning machine, flows, forecasting, hydrology, time-series

**Md Atiquzzaman** (corresponding author)  
**Jaya Kandasamy**  
 School of Civil and Environmental Engineering,  
 University of Technology Sydney,  
 PO Box 123,  
 Broadway,  
 NSW 2007,  
 Australia  
 E-mail: md.atiquzzaman@student.uts.edu.au

## NOTATION

$Q_t$	Flow at time $t$
$(Q_m)_t$	Modelled flow at time $t$
$(Q_o)_t$	Observed flow at time $t$
$H$	Hidden layer output matrix
$H'$	Moore–Penrose generalized inverse of hidden layer output matrix
$a_i$	Weight vector connecting the $i^{\text{th}}$ hidden node and the input variables
$b_i$	Bias of the $i^{\text{th}}$ hidden node
$L$	Number of random hidden nodes
$\beta_i$	Weight connecting the hidden node and the output node
$g(x)$	Activation function (example, sigmoidal function)
$t_j$	Target at time $j$
$y_j$	Output at time $j$

## INTRODUCTION

The application of data-driven modelling approaches including artificial neural networks (ANN) and support vector machines (SVM) has been widespread in the water resource engineering field, especially for predicting hydrological time-series. This is because they can establish complex non-linear relationships between input and output variables (Tokar & Johnson 1999). The main advantage of these techniques is that they do not require information about the complex nature of the underlying hydrological process. When data-driven modeling is applied, input variables including precipitation, lagged precipitation, and lagged discharges are normally employed to forecast the discharges (Akhtar *et al.* 2009). ANN with multi-layer perceptron (MLP) networks trained with gradient-based methods has been used in many applications. Traditionally, the weight vectors in

ANN models is determined using the back-propagation (BP) algorithm by minimizing the mean square error between the measured and forecasted discharges of the hydrological process. However, the performance of ANN depends on network architecture (e.g., number of hidden layers, the number of neurons, activation functions, etc.), performance criteria, division and pre-processing of data, and determining appropriate model inputs (Maier & Dandy 2000). According to Chen & Chang (2009), a very simple ANN network architecture may not accurately predict while too complex architecture may reduce its generalization ability due to over-fitting. In order to overcome this problem, Chen & Chang (2009) proposed an evolutionary algorithm (EA)-based ANN (EANN) to: first, define the optimal network architecture and second, generate a model to accurately predict a hydrological system. They applied EANN to predict real-time inflows to Shihmen Reservoir in Taiwan. It performed better than autoregressive moving average with the exogenous (ARMAX) input time-series approach.

Sing *et al.* (2015) applied ANN to establish relationships between rainfall and temperature data with runoff from an agricultural catchment (973 ha) in Kapgari, India. Several resampling of short length training data sets using bootstrap resampling-based ANN (BANN) found solutions without over-fitting. A 10-fold cross-validation (CV) technique-based ANN was also applied to obtain unbiased reliable testing results. Sing *et al.* (2015) demonstrated that BANN provides more stable solutions and was able to solve problems of over-fitting and under-fitting than 10-fold CV-based ANN.

Hsu *et al.* (1995) demonstrated that non-linear ANN models provided a more representative rainfall-runoff relationship. They compared ANN results with ARMAX and the conceptual Sacramento soil moisture accounting model. Fernando & Jayawardena (1998) also modelled this by using radial basis function neural network (RBF-NN). They showed it performed better than the ARMAX. Tayfur & Sing (2006) applied ANN and fuzzy logic (FL) for predicting event-based runoff and the results were compared against the kinematic wave approximation (KWA). A three-layered feed-forward ANN was developed using the sigmoid function and the BP algorithm. The FL model was developed employing the triangular fuzzy membership

functions for the input and output variables. Adaptive neuro-fuzzy inference system (ANFIS) and ANN were applied by Tingsanchali & Quang (2004) to forecast the daily flood flow for the Yom River Basin in Thailand. ANFIS performed better.

Liong *et al.* (2002a) applied genetic programming (GP) to real-time runoff forecasting. GP was used as an error updating scheme to complement a traditional hydrological model (MIKE11-NAM). The prediction accuracy was enhanced by non-dimensionalizing the variables. The functional relationship between rainfall and runoff derived by GP showed good prediction accuracy. Rodriguez-Vazquez *et al.* (2012) proposed GP and genetic algorithm (GA) for rainfall-runoff modelling of a sub-basin located near Mexico City. They developed two different models for the analysis. The first was a multi-objective optimization-based GP model for determining the structures and parameters of non-linear auto-regressive models (NARMAX). The second was a GA-based model that optimized the parameters of a non-conventional rainfall-runoff model. Their analysis concluded that the multi-objective optimization-based GP model best fitted the analyzed storms of interest. Nourani *et al.* (2013) applied GP for rainfall-runoff modelling and included watershed geomorphological features as spatial data together with temporal data. Two separate models, namely separated geomorphological GP (SGGP) and integrated geomorphological GP (IGGP) models and their application were described. These models applied to the Eel River Watershed in California, USA, could compensate for a lack of temporal data. Specifically, the SGGP model could distinguish the dominant variables of the runoff process in the sub-basins and IGGP was a reliable tool for spatial and temporal interpolation of runoff through the watershed.

Support vector machine (SVM) based on the structural risk minimization approach has proved a promising learning machine for forecasting problems. It was successfully applied to regression problems including flow forecasting and rainfall-runoff modelling. SVM was used to predict the stage in Dhaka, Bangladesh using daily water level data from five gauging stations (Liong & Sivapragasam 2002). Results showed that SVM performed better than ANN. Sivapragasam (2002) applied SVM to rainfall-runoff modelling using six storm events that occurred in Upper

Bukit Timah Catchment, Singapore. The results showed the robustness of SVM compared to multi-layered feed-forward ANN. [Lin et al. \(2006\)](#) reported SVM as a powerful tool that could overcome some of the drawbacks that were evident in ANN: (1) finding global solutions, (2) over-fitting unlikely, (3) generating non-linear solutions using the Kernel Function, and (4) obtaining optimized solutions using a limited training data set. However, SVM requires a long simulation time for large complex problems and it also requires the selection of an appropriate kernel function and its specific parameters ( $C$  and  $\epsilon$ ). [Yu et al. \(2004\)](#) presented a combined application of chaos theory and SVM where the parameters were optimized with an EA to reduce prediction error. In SVM, Gaussian kernel function, being more suitable, was applied to hydrological time-series application ([Liong & Sivapragasm 2002](#)). An EA engine, called shuffled complex evolution (SCE), was applied to determine five parameters, i.e., time delay, embedding dimension and three SVM parameters (tradeoff between empirical error and model complexity, insensitive loss function and width of Gaussian kernel function). EA-based SVM (EC-SVM) was used to predict runoff time-series for catchments including the Tryggevælde Catchment, Denmark and the Mississippi River, USA. The results showed that EC-SVM improved the prediction accuracy compared to standard chaos technique, Naïve, ARIMA and inverse approach. [Wang et al. \(2013\)](#) applied an EA, called particle swarm optimization (PSO), to determine SVM parameters. They further proposed ensemble empirical mode decomposition (EEMD) for decomposing annual rainfall series in SVM to avoid model over-fitting or under-fitting. The proposed model (PSO-SVM-EEMD) improved the rainfall-runoff forecasting significantly compared to ordinary least-square regression model and ANN. [Sivapragasam et al. \(2001\)](#) enhanced the performance of SVM by pre-processing the input data using a noise-reduction algorithm, called singular spectrum analysis (SSA). SSA was coupled with SVM and used to predict the flows from the Tryggevælde Catchment. It improved the prediction accuracy compared to the non-linear prediction (NLP) method.

Application of data-driven modelling methods has been made to quantify the uncertainty associated with the prediction. [Kingston et al. \(2005\)](#) highlighted ANN's failure to account for prediction uncertainty as the quantification of

uncertainty associated with the ANN parameter, (e.g. weights) is complex and difficult. They proposed a Bayesian training method to assess weight uncertainty in ANN. [Cui et al. \(2014\)](#) examined the impact of topographic uncertainty in their rainfall-runoff model (TOPMODEL). The performance of TOPMODEL is influenced by the grid size of the digital elevation model (DEM) that defines the topography. The relationship between DEM resolution and TOPMODEL performance was investigated using fuzzy analysis technique. Different grid sizes of the DEM ranging from 30 m to 200 m were used in TOPMODEL. It revealed that the best results were produced with the 30 m resolution. Uncertainty in streamflow prediction was assessed by [Boucher et al. \(2009\)](#) based on ensemble forecasts using stacked neural network. Instead of forecasting a single value (e.g., 1-day lead prediction), they predicted an ensemble of streamflows which was then used to fit a probability density function to assess the confidence interval as well as other measures of forecast uncertainty. The uncertainty associated with prediction of water levels (or discharges) was analyzed by [Alvisi & Franchini \(2011\)](#). They introduced fuzzy numbers to determine the weights and biases of neural networks to estimate prediction uncertainty of water levels and discharges. The comparison of this fuzzy neural network method with Bayesian neural network and the local uncertainty estimation model demonstrated the effectiveness of the proposed method where the uncertainty bands had slightly smaller widths than other data-driven models. [Alvisi & Franchini \(2012\)](#) found better accuracy in forecasting water levels and narrower uncertainty band width compared to Bayesian neural network using grey neural network (GNN). Here the parameters are represented by unknown grey numbers that lie within known upper and lower limits. The grey parameters are searched in such a way that the grey forecasted river stages include at least a preselected percentage of observed river stages.

Many of these data-driven modelling methods including traditional ANN learning algorithms are slow, requiring numerous iterations to generate optimal solutions ([Zhang et al. 2007](#); [Ding et al. 2015](#)), and may not be suitable for real-time prediction. To overcome this, [Huang et al. \(2006\)](#) proposed a learning algorithm based on SLFNs, called 'extreme learning machine' (ELM) which analytically determines weights related to the output. The performance of

ELM was compared with conventional NN and SVM on benchmarking problems in the function approximation and classification areas. Huang *et al.* (2006) found that ELM approximates any continuous function and implements any classification. ELM may need more hidden nodes but learns faster than SVM. The generalization performance of ELM is stable with a wide range of numbers of hidden nodes. Ding *et al.* (2015) stated that ELM, which requires a single iteration, overcomes the slow training speed and over-fitting problems unlike other conventional ANN learning algorithms. ELM's robustness and fast learning rate was proved in different fields including real data set classification and regression (Huang *et al.* 2012). Zhang *et al.* (2007) applied ELM to multi-category classification problems in cancer diagnosis and found that ELM did not have problems like falling in local minima and over-fitting which are commonly experienced by iteration-based learning methods.

This paper presents the application of ELM (a MATLAB program developed by Huang *et al.* 2006) for predicting hydrological flow time-series. Predictions were made for the Tryggevælde Catchment, Denmark and Mississippi River at Vicksburg, USA to demonstrate the application of the scheme.

## EXTREME LEARNING MACHINE (ELM)

ELM is an emerging learning technique that provides efficient unified solutions to generalized feed-forward network with hidden neurons generated randomly. ELM, a biologically inspired neural network, chooses input weights randomly and analytically determines the output weights. Here, the input weights and biases are not tuned and the hidden layer output matrix remains unchanged. This means that ELM hidden node parameters are independent between the hidden layer and the training data. It generates the hidden node parameters without depending on training data.

The advantages of ELM are: (1) faster learning speed than conventional methods; (2) learns without iteration; (3) better generalization performance; (4) automatically determines all the network parameters analytically; (5) suitable for many non-linear activation function and kernel functions; (6) efficient for online and real-time applications; and (7) viable alternative technique for large-scale computing and machine learning.

The mathematical equation for SLFNs can be formulated as (Figure 1) (Huang *et al.* 2006):

$$f_L(x) = \sum_{i=1}^L \beta_i G(a_i, b_i, x_j) = \sum_{i=1}^L \beta_i g(a_i \cdot x_j + b_i) \quad (1)$$

where  $a_i$  is the weight vector connecting the  $i^{\text{th}}$  hidden node and the input variables and  $b_i$  is the bias of the  $i^{\text{th}}$  hidden node,  $L$  is random hidden nodes;  $\beta_i$  is the weight connecting the hidden node and the output node and  $g(x)$  is an activation function (e.g., sigmoidal function:  $g(x) = 1/(1 + \exp(-x))$ );  $x \in R^n$  and  $a_i \in R^n$ .

When the difference between the target ( $t_j$ ) and the model  $\{f_L(x) = y_j\}$  is zero for a time-series of  $N$  samples,

$$\sum_{j=1}^L ||y_j - t_j|| = 0 \quad (2)$$

where  $J = 1, \dots, N$ .

This means:

$$\sum_{i=1}^L \beta_i g(a_i \cdot x_j + b_i) = t_j \quad (3)$$

Equation (3) can be written as:

$$H\beta = T \quad (4)$$

where  $H$  is called hidden layer output matrix of SLFNs.

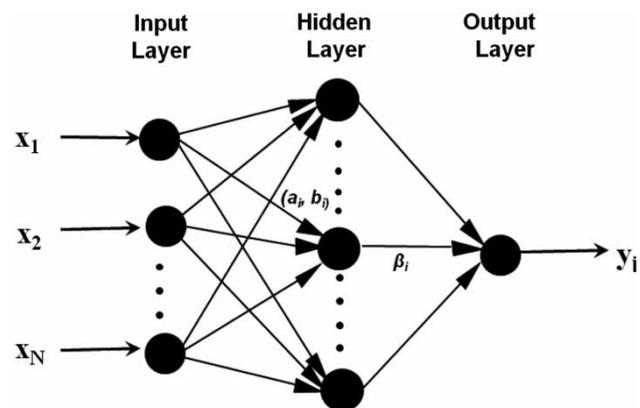


Figure 1 | Structure of neural network.

For the fixed input weight and input biases, training of SLFN finds a least squares solution  $\hat{\beta}$  of the above equation.

$$\|H(\tilde{a}, \tilde{b}, \tilde{x})\hat{\beta} - T\| = \min_{\beta} \|H(\tilde{a}, \tilde{b}, \tilde{x})\beta - T\| \quad (5)$$

where  $\tilde{a} = a_1, \dots, a_L$ ;  $\tilde{b} = b_1, \dots, b_L$ ;  $\tilde{x} = x_1, \dots, x_N$

$$H(\tilde{a}, \tilde{b}, \tilde{x}) = \begin{bmatrix} g(a_1.x_1 + b_1) & \dots & g(a_L.x_1 + b_L) \\ \vdots & \dots & \vdots \\ g(a_1.x_N + b_1) & \dots & g(a_L.x_N + b_L) \end{bmatrix}_{N \times L}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times k} \quad \text{and} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m}$$

$k$  is the number of targets.

Equations (4) or (5) is solved using the smallest norm least-squares solution method, where:

$$\hat{\beta} = H^T T \quad (6)$$

where  $H^T$  is called Moore–Penrose generalized inverse of matrix  $H$ . If the number of hidden neurons is equal to the number of training samples, SLFNs can approximate the training samples with zero error.  $H^T$  may be calculated using several methods including orthogonal projection method, orthogonalization method, iterative method, singular value decomposition (SVD), etc. In ELM, the SVD method is used to calculate  $H^T$ . It was shown that SLFNs with randomly generated hidden nodes and with a widespread piecewise continuous activation function can universally approximate any continuous target function. Further details are available elsewhere (Huang *et al.* 2006).

## APPLICATION

In this study ELM was applied to estimate the catchment runoff (one-step-ahead prediction) using the past and current information of hydrological flow as input data. Mathematically, the relationship can be expressed as:

$$Q_{t+\Delta t} = f(Q_t, Q_{t-\Delta t}, \dots, Q_{t-m\Delta t}) \quad (7)$$

if past historical flow series is considered.

Or,

$$Q_{t+\Delta t} = f(Q_t, Q_{t-\Delta t}, \dots, Q_{t-m\Delta t}, dQ_t, dQ_{t-\Delta t}, \dots, dQ_{t-m\Delta t}) \quad (8)$$

if past historical flow and flow difference data series are considered.

$$\text{Or, } dQ_{t+\Delta t} = f(dQ_t, dQ_{t-\Delta t}, \dots, dQ_{t-m\Delta t}) \quad (9)$$

if past historical flow difference series is considered.

In Equations (7)–(9),  $Q$  is the flow ( $\text{m}^3/\text{s}$ ),  $dQ_{t+\Delta t}$  is error predictor,  $dQ_t = Q_{t+\Delta t} - Q_t$ ,  $m$  represents how far back the recorded data of the time-series affects the flow prediction and  $\Delta t$  is time interval.

Once the error predictor is determined from the model in Equation (9), the predicted flow is estimated as:

$$Q_{t+\Delta t} = Q_t + dQ_t \quad (10)$$

The performance of trained ELM was evaluated with standard goodness-of-fit measures such as root mean square error (RMSE) and normalized root mean square error (NRMSE).

The RMSE and NRMSE are defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{t=1}^N [(Q_m)_t - (Q_o)_t]^2} \quad (11)$$

$$\text{NRMSE} = \sqrt{\frac{\sum_{t=1}^N [(Q_o)_t - (Q_o)]^2}{\sum_{t=1}^N [(Q_o)_t - \bar{Q}_o]^2}} \quad (12)$$

where  $(Q_m)_t$  and  $(Q_o)_t$  are the predicted and observed values at time  $t$ ;  $N$  is the number of observations and  $\bar{Q}_o$  is the mean observed flow.

RMSE represents the forecasting error and estimates the sample standard deviation of the differences between predicted values and observed values. It is a good measure when large model errors are not desirable. The NRMSE normalizes the RMSE and facilitates comparison between data sets. NRMSE close to zero indicates a perfect match between the observed and predicted values and greater

than 1 means predictions are inferior to the constant mean value (Liong *et al.* 2002b).

ELM was used to estimate the 1-day lead prediction of flows for Tryggevælde Catchment and Mississippi River at Vicksburg. The ELM was trained with the same data as used in Liong *et al.* (2002b) and Yu *et al.* (2004) and the results were compared with standard chaos technique, inverse approach, ANN and EC-SVM. In Yu *et al.* (2004), data of 1975–1991 was used for training and 1992–1993 for validation in standard chaos technique. Phoon *et al.* (2002) used 1975–1989 for training, 1990–1991 for testing and 1992–1993 for validation in their application of the inverse approach.

In this study, data of 1975–1991 was used for training and 1992–1993 for validation, similar to the standard chaos technique. The number of hidden nodes in ELM was selected as the number of training samples (6,204). This is because ELM can generate zero error when the number of hidden neurons learns the same number of distinct observations (Huang *et al.* 2006). For the output node, the widely used sigmoid activation function was chosen.

## TRYGGEVÆLDE CATCHMENT

The Tryggevælde Catchment (130.5 km<sup>2</sup>) is located in the eastern part of Sealand, north of Karise. The daily measured flows are available for the period 1 January 1975 to 31 December 1993. The statistics of the flow series are: mean flow = 0.977 m<sup>3</sup>/s; standard deviation = 1.367 m<sup>3</sup>/s; maximum flow = 11.068 m<sup>3</sup>/s; and minimum flow = 0.014 m<sup>3</sup>/s. The statistics of flows indicate that there are distinct wet and dry periods in the time-series.

The training and validation accuracies are presented in Table 1 in terms of RMSE and NRMSE for ELMI (lagged

flow,  $Q$ ), ELMII (lagged flow difference,  $dQ$ ) and ELMIII ( $Q$ ,  $dQ$ ). The validation accuracies from all three ELM models were between 0.491–0.504 (RMSE) and 0.337–0.347 (NRMSE), respectively. The ELMIII performed the best among the three models. The time required to train ELM models was about 100 sec on a Windows-based machine (Intel i7 CPU at 2.67 GHz). The corresponding validation times were between 0.47 and 0.49 sec. An ANN model, which estimates the default number of hidden nodes (81) based on number of input variables, output variables and training samples, was trained with the same data set as ELM (hidden nodes = 6,204) for approximately 100 sec (training time of ELM) for comparison with ELM. The prediction accuracies of ANN obtained from 633 iterations were 0.588 (RMSE) and 0.403 (NRMSE) (Table 2).

Table 2 compares the ELM results with other available techniques (Liong *et al.* 2002b; Yu *et al.* 2004). Yu *et al.* (2004) used two types of input time-series, namely daily flow series ( $Q$ ) and flow difference series ( $dQ$ ) separately in EC-SVM. The use of  $dQ$ -series in EC-SVM provided better results than the  $Q$ -series. The number of iterations required was 151,668 for EC-SVM( $Q$ ) and 11,800 for EC-SVM( $dQ$ ). All ELM techniques were faster as no additional iteration was required and produced better results (Table 2). The ELMIII model improved the prediction accuracy in terms of RMSE by 24% over the standard chaotic approach, 7% over the inverse approach and 4% over the EC-SVM( $Q$ ). ELMII performed similarly.

## MISSISSIPPI RIVER AT VICKSBURG FLOW

A similar approach was also applied to predict flows in the Mississippi River at Vicksburg, and used the same daily flows documented in Yu *et al.* (2004). The daily measured flows cover the period 1 January 1975 to 31 December 1993. The

Table 1 | ELM prediction accuracy for both training and validation data set for Tryggevælde Catchment

Model	No. of iterations	Training			Validation		
		RMSE	NRMSE	Time (s)	RMSE	NRMSE	Time (s)
ELMI( $Q$ )	Single	0.495	0.362	99.28	0.497	0.341	0.49
ELMII( $dQ$ )	Single	0.508	0.372	108.09	0.504	0.347	0.47
ELMIII( $Q$ , $dQ$ )	Single	0.488	0.357	100.88	0.491	0.337	0.48

**Table 2** | Prediction accuracy for validation data set, the number of iterations and training time from various techniques for Tryggevælde Catchment

Model	RMSE	NRMSE	No. of iterations (training)	Training time (sec)
Standard chaos technique*	0.647	0.444	–	–
Inverse approach*	0.527	0.361	–	–
EC-SVM(Q)*	0.514	0.352	151,668	–
EC-SVM(dQ)*	0.504	0.347	11,800	–
ANN	0.588	0.403	633	100**
ELMI(Q)	0.497	0.341	Single	99.28**
ELMII(dQ)	0.504	0.347	Single	108.09**
ELMIII(Q, dQ)	0.491	0.337	Single	100.88**

\*Liong *et al.* (2002b) and Yu *et al.* (2004).

\*\*Windows Intel i7 at 2.67 GHz.

statistics of flow series are: mean flow = 18,456.54 m<sup>3</sup>/s; standard deviation = 9,727.27 m<sup>3</sup>/s; maximum flow = 52,103.00 m<sup>3</sup>/s; and minimum flow = 3,907.72 m<sup>3</sup>/s.

Table 3 shows the results of training and validation obtained from ELMI(Q), ELMII(dQ) and ELMIII(Q, dQ). The time required to train ELM models was about 97 sec on the same computer (Windows Intel i7 CPU at 2.67 GHz). The corresponding validation times were between 0.45 and 0.47 sec.

The results (Table 3) showed that ELMIII performed best with RMSE and NRMSE of 308.66 and 0.0391, respectively. These values were slightly higher for the other two ELM models (ELMI and ELMII). ELMIII predicted better results (Table 4) than the standard chaos technique (RMSE = 1,738.95 and NRMSE = 0.2064) and inverse approach (RMSE = 356.89 and NRMSE = 0.0452). An ANN model (default hidden nodes = 81) trained with the same data set as ELM (hidden nodes = 6,204) was run for approximately 97 sec (training time of ELM) for comparison with ELM.

ANN prediction accuracies obtained from 794 iterations were 549.70 (RMSE) and 0.0696 (NRMSE) (Table 4).

Table 4 shows that the RMSE and NRMSE of ELM models were slightly higher compared to EC-SVM(Q) and EC-SVM(Q, dQ) models. However, ELM predicted these solutions quickly (single run) compared to EC-SVM(Q) and EC-SVM(Q, dQ), where 1,732,579 and 57,590 iterations, respectively, were required. This demonstrates that ELM can be efficient for online and real-time applications.

## DISCUSSION

This study demonstrates the application of ELM for predicting hydrological flow time-series. The results show that the prediction accuracies of ELM were similar or better than ANN and other previously published techniques including EC-SVM, standard chaotic approach and inverse approach. More specifically, ELM improved the flow prediction accuracy in terms of RMSE by 24% over standard chaos technique, 7% over the inverse approach and 4% over the EC-SVM for the Tryggevælde Catchment. ELM provided solutions of similar accuracy to EC-SVM when predicting Mississippi River flows.

ELM's real strength is that it reached the solutions quickly compared to other techniques including EC-SVM. This is because no additional iteration is required in ELM whereas other techniques require thousands of iterations to predict the same flow time-series although most do so with less accuracy. Such runs typically have a much longer processing time. Importantly, this processing time will significantly increase for more complex scenarios where many more iterations are required to obtain an optimal solution. This longer processing time may be a limiting factor for real-time application. ELM's fast learning capability

**Table 3** | ELM prediction accuracy for both training and validation data set for Mississippi River flow, Vicksburg

Model	No. of iterations	Training			Validation		
		RMSE	NRMSE	Time (s)	RMSE	NRMSE	Time (s)
ELMI(Q)	Single	396.58	0.040	97.92	320.00	0.040	0.47
ELMII(dQ)	Single	389.37	0.0394	96.85	312.8	0.0397	0.47
ELMIII(Q, dQ)	Single	382.49	0.0387	96.84	308.66	0.0391	0.45

**Table 4** | Prediction accuracy for validation data set, the number of iterations and training time from various techniques for Mississippi River flow, Vicksburg

Model	RMSE	NRMSE	No. of iterations (training)	Training time (sec)
Standard chaos technique*	1,738.95	0.2064	–	–
Inverse approach*	356.89	0.0452	–	–
EC-SVM(Q)*	306.58	0.0387	1,732.579	–
EC-SVM(dQ)*	304.26	0.0385	47, 590	–
ANN	549.70	0.0696	794	97**
ELMI(Q)	320.00	0.04	Single	97.92**
ELMII(dQ)	312.8	0.0397	Single	96.85**
ELMIII(Q, dQ)	308.66	0.0391	Single	96.84**

\*Liong *et al.* (2002b) and Yu *et al.* (2004).

\*\*Windows Intel i7 at 2.67 GHz.

from a training data set means that it would be more suitable for online and real-time applications where quick processing is important or vital. These include flood forecasting, the prediction of inflows for reservoir operations, supply of water to meet irrigation demand, real time control of water systems and sewer systems, etc. Furthermore, by having improved or at least comparable prediction accuracy to other available methods, ELM is no less capable for use in water resource management and decision-making.

This paper demonstrates the potential of ELM for predicting hydrological flow time-series. The data obtained from a relatively small (Tryggevælde Catchment, Denmark) and large catchment (the Mississippi River Catchment at Vicksburg) covered both dry and wet periods. In both cases, ELM's performance is comparable for 1-day lead prediction which means that the potential issue of extrapolating flows beyond the range of training data set is unlikely to be significant. The speed, predicting capability and accuracy of ELM can also depend on the number of hidden nodes, activation functions and different sampling of input data. These aspects require more investigation.

## CONCLUSION

This paper presents the application of ELM for predicting hydrological flow time-series for the Tryggevælde

Catchment and Mississippi River at Vicksburg. The results show that the prediction accuracies of ELM are similar or better than ANN and other previously published techniques. The real strength of ELM is the short computational run-time to reach solutions comparable with other techniques including EC-SVM. This is because ELM does not require iteration whereas other techniques (e.g., EC-SVM) may require thousands of iterations to predict the same flow time-series and yet with less accuracy. Such runs typically take a much longer processing time. ELM's fast learning capability from a training data set means that it would be more suitable for online and real-time applications where quick processing is important or vital.

## REFERENCES

- Akhtar, M. K., Corzo, G. A., Andel, S. J. & Jonoski, A. 2009 *River flow forecasting with artificial neural networks using satellite observed precipitation pre-processed with flow length and travel time information: case study of the Ganges river basin. Hydrol. Earth Syst. Sci.* **13**, 1607–1618.
- Alvisi, S. & Franchini, M. 2011 *Fuzzy neural networks for water level and discharge forecasting with uncertainty. Environ. Model. Softw.* **26** (4), 523–537.
- Alvisi, S. & Franchini, M. 2012 *Grey neural networks for river stage forecasting with uncertainty. Phys. Chem. Earth* **42–44**, 108–118.
- Boucher, M. A., Perreault, L. & Anctil, F. 2009 *Tools for the assessment of hydrological ensemble forecasts obtained by neural networks. J. Hydroinform.* **11** (3), 297–306.
- Chen, Y. H. & Chang, F. J. 2009 *Evolutionary artificial networks for hydrological systems forecasting. J. Hydrol.* **367**, 125–137.
- Cui, L., Li, Y. P., Huang, G. H. & Huang, Y. 2014 *Effects of digital elevation model resolution on topography-based runoff simulation under uncertainty. J. Hydroinform.* **16** (6), 1343–1358.
- Ding, S., Zhao, H., Zhang, Y., Xu, X. & Nie, R. 2015 *Extreme learning machine: algorithm, theory and application. Artif. Intell. Rev.* **44**, 103–115.
- Fernando, D. A. K. & Jayawardena, A. W. 1998 *Runoff forecasting using RBF networks with OLS algorithm. J. Hydrol. Engng.* **3** (3), 203–209.
- Hsu, K. L., Gupta, H. V. & Sorooshian, S. 1995 *Artificial neural network modeling of the rainfall-runoff process. Water Resour. Res.* **31** (10), 2517–2530.
- Huang, G.-B., Zhu, Q.-Y. & Siew, C.-K. 2006 *Extreme learning machine: theory and applications. Neurocomput.* **70**, 489–501.
- Huang, G. B., Zhou, H., Ding, X. & Zhang, R. 2012 *Extreme learning machine for regression and multiclass classification. IEEE Trans Sys. Man. Cyber-Part B: Cybernetics* **42** (2), 513–529.

- Kingston, G. B., Lambert, M. F. & Maier, H. R. 2005 Bayesian training of artificial neural networks used for water resources modeling. *Water Resour. Res.* **41** (12), W12409.
- Lin, J. Y., Cheng, C. T. & Chau, K. W. 2006 Using support vector machine for long-term discharge prediction. *Hydrol. Sci.* **51** (4), 599–612.
- Liong, S. Y. & Sivapragasam, C. 2002 Flood stage forecasting with SVM. *J. Am. Water Res. Assoc.* **38** (1), 173–186.
- Liong, S. Y., Gautam, T. R., Khu, S. T., Babovic, V. & Muttill, N. 2002a Genetic programming: a new paradigm in rainfall-runoff modelling. *J. Am. Water Res. Assoc.* **38** (3), 705–718.
- Liong, S. Y., Phoon, K. K., Pasha, M. F. K. & Doan, C. D. 2002b A robust and efficient scheme in search for optimal prediction parameters set in chaotic time-series. In: *First Asia Pacific DHI Software Conference, 17–18 June*, Bangkok.
- Maier, H. R. & Dandy, G. C. 2000 Neural networks for prediction and forecasting of water resources variables: a review of modelling issue and applications. *Environ. Modell. Softw.* **15**, 101–124.
- Nourani, V., Komasi, M. & Alami, T. 2013 Geomorphology-based genetic programming approach for rainfall-runoff modelling. *J. Hydroinform.* **15** (2), 427–445.
- Phoon, K. K., Islam, M. N., Liaw, C. Y. & Liong, S. Y. 2002 Practical inverse approach for forecasting nonlinear hydrological time-series. *J. Hydrol. Engng.* **7** (2), 116–128.
- Rodriguez-Vazquez, K., Arganis-Juarez, M. L., Cruickshank-Villanueva, C. & Dominguez-Mora, R. 2012 Rainfall-runoff modelling using genetic programming. *J. Hydroinform.* **14** (1), 108–121.
- Sing, G., Panda, R. K. & Lamers, M. 2015 Modelling of daily runoff from a small agricultural watershed using artificial neural network with resampling techniques. *J. Hydroinform.* **17** (1), 56–74.
- Sivapragasam, C. 2002 Multi-objective evolutionary techniques in defining optimal policies for real time operation of reservoir systems. PhD thesis, National University of Singapore, Singapore.
- Sivapragasam, C., Liong, S. Y. & Pasha, F. K. 2001 Rainfall and runoff forecasting with SSA-SVM approach. *J. Hydroinform.* **3** (3), 141–152.
- Tayfur, G. & Sing, V. P. 2006 ANN and fuzzy logic models for simulating event-based rainfall-runoff. *J. Hydrol. Engng.* **132** (12), 1321–1330.
- Tingsanchali, T. & Quang, C. N. X. 2004 Application of neuro-fuzzy technique in flood forecasting. In: *6th International Conference on Hydroinformatics*, Singapore.
- Tokar, A. S. & Johnson, P. A. 1999 Rainfall-runoff modelling using artificial neural network. *J. Hydrol. Engng.* **4** (3), 232–239.
- Wang, W.-C., Xu, D.-M., Chau, K.-W. & Chen, S. 2013 Improved annual rainfall-runoff forecasting using PSO-SVM model based on EEMD. *J. Hydroinform.* **15** (4), 1377–1390.
- Yu, X., Liong, S. Y. & Babovic, V. 2004 EC-SVM approach for real-time hydrologic forecasting. *J. Hydroinform.* **6** (3), 209–223.
- Zhang, R., Huang, G.-B., Sundararajan, N. & Saratchandran, P. 2007 Multi-category classification using an extreme learning machine for microarray gene expression cancer diagnosis. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **4** (3), 485–495.

First received 2 February 2015; accepted in revised form 27 August 2015. Available online 7 October 2015