Chapter 6
Hybrid Data Intelligent Models and Applications for Water Level Prediction

Zaher Mundher Yaseen
The National University of Malaysia, Malaysia

Ravinesh C. Deo
University of Southern Queensland, Australia

Isa Ebtehaj
Razi University, Iran

Hossein Bonakdari
Razi University, Iran

ABSTRACT

Artificial intelligence (AI) models have been successfully applied in modeling engineering problems, including civil, water resources, electrical, and structure. The originality of the presented chapter is to investigate a non-tuned machine learning algorithm, called self-adaptive evolutionary extreme learning machine (SaE-ELM), to formulate an expert prediction model. The targeted application of the SaE-ELM is the prediction of river water level. Developing such water level prediction and monitoring models are crucial optimization tasks in water resources management and flood prediction. The aims of this chapter are (1) to conduct a comprehensive survey for AI models in water level modeling, (2) to apply a relatively new ML algorithm (i.e., SaE-ELM) for modeling water level, (3) to examine two different time scales (e.g., daily and monthly), and (4) to compare the inspected model with the extreme learning machine (ELM) model for validation. In conclusion, the contribution of the current chapter produced an expert and highly optimized predictive model that can yield a high-performance accuracy.

DOI: 10.4018/978-1-5225-4766-2.ch006

1. INTRODUCTION

In this chapter, we develop highly novel machine learning (ML) models as optimal tools for water resource forecasting. ML approaches have been broadly researched using soft computing engineering principles (Coppin, 2004; Ford, 1987; Gevarter, 1987; Noureldin, El-Shafie, & Bayoumi, 2011; Russell & Norvig, 2010). The most common ML models are designed by artificial intelligence (AI) techniques identify the nonlinear, dynamic, nonstationary relationships min predictor data for both regression and classification problems (Nourani, Hosseini Baghanam, Adamowski, & Kisi, 2014; Yaseen, El-shafie, Jaafar, Afan, & Sayl, 2015). Advantageous mechanism of soft computing-based AI techniques have led to several applications of science and engineering optimization problems (e.g., sediment transport, evaporation rate, stream-flow, river water quality, ground water and etc.) (Nourani, Hosseini Baghanam, Adamowski, & Kisi, 2014; Yaseen, El-shafie, Jaafar, Afan, & Sayl, 2015). Generally, machine learning utilizing data-driven methodologies are designed with artificial neural network, fuzzy set, evolutionary computing to solve complex computational problems in science and engineering. In 1994, Zadeh coined the term soft computing and defined it as:

collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness, and low solution cost.

Based on Zadeh's opinion, that we live in a pervasively imprecise and uncertain world and that precision and certainty carry a cost. Therefore, soft computing should be seen as a partnership of distinct optimization methods, rather than as a homogeneous body of concepts and techniques.

The water level sector has been explored by several AI techniques (Alvisi & Franchini, 2011; F. J. Chang & Chang, 2006; Elhatip & Kömür, 2008; Fallah-Mehdipour, Bozorg Haddad, & Mariño, 2013; Wei, 2012). Its history for water level modeling back to 1998 with the earliest research accomplished in water level forecasting (Thirumalaiah & Deo, 1998), who conducted real-time streamflow stage using artificial neural network. The results showed that modelling low flow superior the high value of forecasting. Campolo et al. (1999) studied river flood forecasting based on rainfall and water level variables using ANN model. The main goal of their research was to build a predictive model by including climatological information. The modelling of hourly basis showed remarkable results of accuracies. Liong et al. (2000) demonstrated the application of ANN in forecasting water level stage at Dhaka, Bangladesh. The input variables were investigated up to seven days' lead time to forecast on step ahead water level. The authors revealed that the results made a desirable advanced warning forecasting tool. In 2002, the application of Support Vector Machines have been implemented for forecasting flood stage by (S. Y. Liong & Sivapragasam, 2002). The results of SVM were compared with that of ANN based on the antecedent records of water level "one-lead day to seven-lead day forecasting". The improvements in maximum predicted water level errors by SVM over ANN for four-lead day to seven-lead day are 9.6 cm, 22.6 cm, 4.9 cm and 15.7 cm, respectively. Clearly, these studies showed the relevance of AI techniques for optimization of forecasting methodologies in area of water resources.

Since the span time scale of water resource variables is significantly important for modelling, the input data often reflect stochastic nature which tends to complicate the regression problem. Addressing this issue with optimisation techniques, short-term time scale water level prediction have been researched by (Bazartseren, Hildebrandt, & Holz, 2003). They have optimized their models with the neuro-fuzzy (NF) system model and compared its proficiency with an ANN model. The outcome indicated the possibility to predict water level at a certain location for a short time span with an adequate accuracy using NF model. Solomatine and Xue (2004) investigated the proficiency of M5 trees model in modelling flood forecasting problem. The modelling results was compared with ANN model. It was observed that M5 trees model being similar to the linear functions model, in which have a certain advantage compared to ANN. They are more transparent and hence acceptable by decision makers, are very fast in training and always converge. In addition, the improved accuracy of predicting high floods was achieved by building a mixture of models.

Further optimisation of forecasted results were undertaken by other researchers, such as Chang and Chang (2006). They proposed a forecasting model known as the adaptive network-based fuzzy inference system (ANFIS) based on fuzzy set theory (Zadeh, 1994). The application was made to data sets in the Shihmen reservoir, Taiwan to forecast 1-3 hours-ahead water level for the purpose of optimisation of reservoir safety and minimizing the damage resulting from a natural disaster. The ANFIS model provided accurate and reliable water level prediction for next three-time steps. Their study found that the construction of ANFIS model, through the subtractive fuzzy clustering, can efficiently deal with vast and complex input-output patterns, and has a great ability to learn and build up a neuro-fuzzy inference system for prediction. Another research based on integrating the ANN model with *add-in* optimization algorithm (i.e., particle swarm optimization (PSO)) (Chau, 2006) found improved forecasted results attained by the use of the optimization algorithm.

In 2007, Han and his contributors conducted a flood forecasting application using SVM model using linear and nonlinear kernel functions (i.e. radial basis function) This study showed an SVM response to different rainfall inputs was good, where lighter rainfalls would generate very different responses than heavier ones, and this would a very useful way to reveal the behaviour and shortcomings of n SVM. In a study that investigated how effective the steepness coefficient for the sigmoid function of ANN was, researchers found good accuracy of the 1-day forecasts of water level (Sulaiman, El-Shafie, Karim, & Basri, 2011). In 2012, researchers attempted to improve the water level forecasting using SVM and wavelet decomposition approach (W-SVM) (Wei, 2012). The research was conducted using short time scale water level (i.e., hourly time horizon). What make the case study extra ordinary, the catchment river was influenced by a strong precipitation events and affected by tidal effects during typhoons. Here, the wavelet is a function with many dimensions was able to approximate the functions that were arbitrarily nonlinear, and showed that the prediction of the optimized wavelet SVM models were more accurate than those of the Gaussian SVM models.

Furquim et al. (2014) used wireless sensor network to collect the natural behaviour of water in rivers in Brazil to create a prediction flood model using artificial neural network. The established flood prediction model was a convinced methodology to forecast flood events. In developing countries, a lack of adequate and good quality data for hydrological modelling for flood forecasting poses a significant challenge. Thus, it is important to explore machine learning optimizer algorithm. For instance, Mwale et al. (2014) integrated the self-organizing maps (SOM) with the multi-layer perceptron artificial neural networks (MLP-ANN) to forecast flow and water level in Malawi. A very satisfactory forecasted result were obtained with the latter for up to 2-day lead time. However, when SOM features were used, the lead time for very satisfactory forecasts increased to 5 days, due to the more comprehensive information of the time series data was extracted, revealing the important role of optimizer algorithms for better accuracy.

Chang et al. (2014) developed two neural networks including static and dynamic. In the first stage of the study, the historical hydrologic data are fully explored by statistical techniques to identify the time span of rainfall affecting the rise of the water level in the flood water storage pond (FSP) at the gauging station. In the second stage, an effective factor (i.e., rainfall stations) that significantly affect the FSP water level were extracted by the Gamma test (GT). At the final stage, the prediction models constructed a multi-step-ahead FSP water level forecast models through two scenarios, in which scenario I adopted rainfall and FSP water level data as model inputs while scenario II adopted only rainfall data as model inputs. The results demonstrated that the GT can efficiently identify the effective rainfall stations as important inputs to the static and dynamic neural network. The yielded results showed a positive high accuracy belonging to the dynamic network in comparison with the static network.

Ouyang (2016) developed a forecasting model of water level during typhoons events. The author anticipated a forecasting model for river water level real time using online data base. The mathematical model was built using autoregressive moving average with exogenous model. The modelling inputs were used in this research including rainfall and water level from the target and nearby stations. Here the analysis of the cumulative rainfall in addition to the water level were obtained to approximate the possibility of the cumulative rainfall duration and lag times associated with each gaging station. The results demonstrate the perfect model can yield a significant time shifting.

Water level prediction model has been conducted through integrating ANN model with evolutionary optimization algorithm (i.e., firefly algorithm (FFA)) by Soleymani et al., (2016). The research was conducted in in tropical environment, Malaysia. The main contribution of this research is to determine the internal parameter of the radial basis function via the features of FFA. The results of the hybrid model exhibited very excellent performance over the ANN-based model.

Comprehending the complexity phenomena of hydrology cycle, researchers are passionate to solve this scientific problem using sophistication of standalone AI models and their optimized versions. Throughout the narrative two decades studies, water level modelling has received a massive attention. This is due to the fact that, the global climate changes have been influenced the hydrologic cycle that caused numerous of flood and drought events. According to the literature, water level modelling has been undertaken based on two main methodologies, physical based models and conceptual based models "i.e., AI model". Physical models usually required more effort and various hydrological variables to simulate the elemental physical processes of the watershed (Yaseen, Kisi, & Demir, 2016). Whereas, soft computing approaches have shown the capability to capture the non-linearity relationship between the predictors and predicted without advance knowledge with less inputs hydrological parameters (Afan, El-Shafie, Yaseen, Hameed, Wan Mohtar, & Hussain, 2014; Deo & Şahin, 2015; Deo, Samui, & Kim, 2015; Fahimi, Yaseen, & El-shafie, 2016).

In optimisation problems, the ANN model has been the predominant AI method where the algorithms include radial basis function, multi-layer perceptron, Back propagation neural network and generalized regression neural network. However, an ANN model can have issues such as slow convergence rate, inability to attain a global solution, iterative tuning o model parameters and lower accuracy compared to the more advanced versions of neural networks. Most recently, a new AI model called extreme learning machine was proposed by (G.-B. Huang, Zhu, & Siew, 2006). The merit of this model is required no tuning during the training processes of the constructed predictive model (I Ebtehaj & Bonakdari, 2016; G. Huang, Huang, Song, & You, 2015). A basic version of ELM model had been modified through the utilization of evolutionary optimization algorithm and produce new version called self-adaptive evolutionary extreme learning machine (SaE-ELM) as a new evolutionary case of ELM (Cao, Lin, Huang, & Bin, 2012).

In this chapter, we propose to utilize the SaE-ELM algorithm as an optimized model in a problem of water resources modelling, following its diverse range of applications. Sa-E-ELM has been used due to its high-performance and innovative design features (i.e., random generation of the parameters of hidden nodes without the need for iteratively tuning the algorithm, determining the output weights analytically by solving a least squares problem and yielding significantly faster solutions compared to traditional neural network models (e.g. FFBP-ANN)) (Azimi, Bonakdari, & Ebtehaj, 2017; Isa Ebtehaj, Sattar, Bonakdari, & Zaji, 2017). In addition, the SaE-ELM can be automatically updated as new data arrive, either with a single datum or a chunk of data. Hence and for the best knowledge of the authors, SaE-ELM is developed to forecast river water level application. The modeling conducted over two-time horizons daily and monthly time scales. The input variables were initiated based on the antecedent values "i.e., correlated lag times". The implemented evolved version of SaE-ELM is validated against the ELM model.

2. DATA DESCRIPTION

The investigated case study located in tropical environment which is Linggi River located in Negeri Sembilan state in peninsular Malaysia (see Figure 1). The selected case study experiences a tropical environment characteristic with high humidity and uniform temperature almost all along the year. The streams of the catchment influenced highly by the monsoon rainfall starting November to March with total annual rainfall records range between 2100 to 2250 mm per year (Tan, Ibrahim, Yusop, Chua, & Chan, 2017). The drainage area of Linggi River is 1320 km^2. The collected metrological information belongs to 1965-2015-time period. Those data were collected from the Department of Irrigation and Drainage organization. These data modeled based on univariate problem which is called forecasting (only river water stage is included). It is even significant to mention, the selection of this case study due to it's potential location and efficient supplement to the water treatment plant as well as the it natural river system.

3. THEORETICAL OVERVIEW

In this section, the theory useful in in the development of the SaE-ELM to forecast water level is provided. Firstly, the extreme learning machine (ELM) approach is presented, differential evolution (DE) method is described and finally the SaE-ELM mathematical principle is defined.

3.1. Extreme Learning Machine (ELM)

Extreme learning machine (ELM) is a single layer feed-forward neural network (SLFFNN) which is originally introduced by Huang et al. (2006a). This technique overcomes the shortcomings of traditional backpropagation (BP) algorithm which is used gradient descent-based learning algorithm. The main problems in BP algorithm are trapping in local minima and specifying weight or bias by tuning which is lead to the reduction of learning speed (Faruk Ertuğrul & Kaya, 2014).

One of two main concerns in designing a learning algorithm is learning speed and generalization ability. Training a single layer feed-forward neural network (SLFFNN) with a completely differentiable activation function by ELM algorithm that hidden biases and weights randomly tuned and the learning speed is extremely fast results in good generalization performance and have the ability to achieve global minima. Moreover, the theoretical results of Huang et al. (2006b) represent the universal estimation

Figure 1. Map displayed the located on the investigated river water level, Sembilan state, Malaysia

aptitude of SLFFNN trained with ELM for continuous activation functions. Thus, meticulous results of ELM performance on universal estimation potency firmly support real-large and artificial applications of SLFFNN. A basic structure of ELM which has three layers; input, hidden and output, has been presented in Figure 2.

Consider N samples of a training data set as

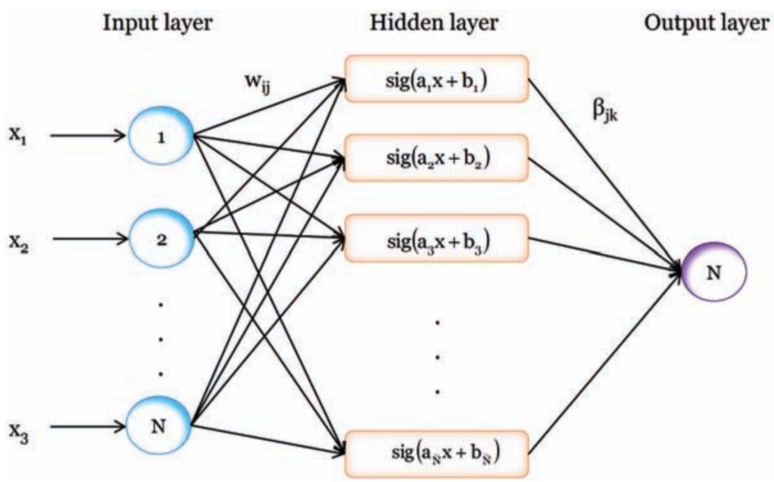
$$D = \left\{ \left(x_i, t_i \right) \mid x_i \in R^n, t_i \in R^m, i = 1, 2, \ldots, N \right\}$$

where $x_i = \left[x_{i1}, x_{i2}, \ldots, x_{in} \right]^T$ and $t_i = \left[t_{i1}, t_{i2}, \ldots, t_{im} \right]^T$ are input and output vectors, respectively. A SLFFNN with g(x) as activation function and M hidden nodes is defined as:

$$f\left(x_j\right) = \sum_{i=1}^{M} \beta_i g_i \left(x_j\right) = \sum_{i=1}^{M} \beta_i \left(\omega_i^T x_j + b_i \right) = O_j \left(j = 1, 2, \ldots, N \right) \tag{1}$$

here, $\omega_i = \left[\omega_{i1}, \omega_{i2}, \ldots, \omega_{in} \right]^T$ and $\beta_i = \left[\beta_{i1}, \beta_{i2}, \ldots, \beta_{im} \right]^T$ are the weight vector linking i^{th} hidden node with inputs and output (respectively) nodes. Also, the threshold of i^{th} hidden node is considered as b_i.

Figure 2. The basic structure of ELM network



The main objective of SLFFNN is the approximation of all samples with least errors as far as possible. Thus, the equation (1) is rewritten as following matrix form:

$$H\beta = T \tag{2}$$

where H is the output matrix of the hidden layer. The β, T and H are defined as follows:

$$\beta = \left[\beta_1^T \cdots \beta_M^T\right]_{M\times m}^T \tag{3}$$

$$T = \left[t_1^T \cdots t_M^T\right]_{N\times m}^T \tag{4}$$

$$H = \begin{bmatrix} g\left(\omega_1^T x_1 + b_1\right) & \cdots & g\left(\omega_M^T x_1 + b_M\right) \\ \vdots & \ddots & \vdots \\ g\left(\omega_1^T x_N + b_1\right) & \cdots & g\left(\omega_M^T x_N + b_M\right) \end{bmatrix}_{N\times M} \tag{5}$$

The results of Huang et al. (2006a) indicate that the number of training samples is larger than the number of hidden nodes $\left(i.e. M \leq N\right)$ if the activation function is infinitely differentiable such as sigmoid. Also, Randomly assigned of b_i (i=1, 2, …, M) and ω_i to find a least-square solution (β*) which is mathematically formulated as follows is done during SLFFNN training procedure:

$$H\beta^{*} - T = \min_{\beta} H\beta - T \tag{6}$$

The β* is calculated in following from due to definition of Moore-Penrose generalized inverse (MPGI) (Huang, Zhu, & Siew, 2006):

$$\beta^{*} = H^{+}T \tag{7}$$

where H^+ is the MPGI of H.

The use of active ELM as training algorithm does not require additional parameters such as stopping criterion and learning rate. This method optimizes hidden biases nodes and input weights of the network and is able to obtain output weights of the network rapidly. In a SLFFNN with random hidden nodes, at first the input dataset and real actual output of the model $\left[\left(X\right),\left(Y\right)\right]$ is determined. Subsequently, the number of hidden nodes [M] and the type of activation function [g(x)], is determined. Then, its weight and bias values are presented in random order [(W), (b)]. Then, hidden layers' matrix [H] is determined and then the weight of output as analytic is calculated [β].

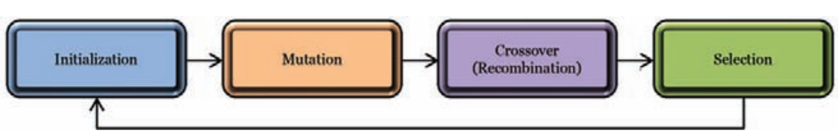
3.2. Differential Evolution (DE)

The DE algorithm is capable to find a solution for nonlinear complex problems with lest error. This algorithm was introduced by (Storn & Price, 1995) to overcome the genetic algorithm (GA) shortcoming in lack of local search. The difference of DE and GA is in the order of mutation and crossover and operation of selection operator (Figure 3).

This algorithm employed an evolution operator to produce new population so that this operator exchanges the information between the population members. One of the advantages of this algorithm is having a memory that keeps the information up-to-date in the current population. Another advantage of this algorithm is its selection operator. In this algorithm, all members of a population have the same chance of being elected as one of the parents. In this way, the generation of the infant with the parent generation is compared in terms of fitness value that is measured by the objective function. Then the best members will enter the next step as the next generation.

The most important features of the DE algorithm are its high speed, robustness and simplicity. This method only works by adjustment three parameters; the number of population (NP), mutation factor (F) and crossover (C_r). Based on (Storn & Price, 1995) suggestion, F and C_r are in the range of [0 2] and [0 1], respectively. In general, this algorithm has four basic steps: 1) initialization; 2) mutation; 3) crossover and; 4) selection.

Figure 3. The task process of operators of differential evolution algorithm



1. Initialization

This algorithm initially produces a random population (NP) in the range of the corresponding problem quantities. At this stage, the boundaries of the values are determined. The Population vector contains N parameters are produced as follows:

$$X_{i,j} = rand_j\left[0,1\right]\cdot\left(X_{i,\max} - X_{j,\min}\right) + X_{j,\min}(j = 1,2,\ldots,N)(i = 1,2,\ldots,NP) \tag{8}$$

where j is the number of parameters of a vector which are optimized through training process, i is the number of population parameter vector.

2. Mutation

The mutated vector which are defined in different forms are generated are presented to produced new population from initial population which are defined randomly. Different mutation vector are presented as follows:

$$V_{i,G+1} = X_{r1,G} + F\left(X_{r2,G} - X_{r3,G}\right) \tag{9}$$

$$V_{i,G+1} = X_{best,G} + F\left(X_{r1,G} - X_{r2,G}\right) \tag{10}$$

$$V_{i,G+1} = X_{i,G} + F\left(X_{best,G} - X_{i,G}\right) + F\left(X_{r1,G} - X_{r2,G}\right) \tag{11}$$

$$V_{i,G+1} = X_{r1,G} + F\left(X_{r2,G} - X_{r3,G}\right) + F\left(X_{r4,G} - X_{r5,G}\right) \tag{12}$$

where $X_{r1,G}$, $X_{r2,G}$, $X_{r3,G}$, $X_{r4,G}$ and $X_{r5,G}$ which are selected randomly are the results' vector related to rith vector and Gth generation, X_{best}^{G} is the best vector in Gth generation nad F is scale factor which are defined to control the convergence speed.

3. Crossover

In this operator, the trial vector is generated using a combination of a mutated vector and a target vector which are selected in the first stage. The basis of this combination is based on the crossover coefficient (C_r). So that, each of the mutated vector components is transmitted to the candidate vector with a Cr probability. Otherwise, the equivalent component is replaced in the original vector.

$$u_{ji,G+1} = \begin{Bmatrix} V_{ji,G+1} & If\ rand(j) \le Cr\ or\ j = randb() \\ x_{ji,G} & Otherwise \end{Bmatrix} \tag{13}$$

4. Selection

At this stage, the trial vector obtained from the previous stage and target vector, which was selected in the first stage, is evaluated according to the objective function. If the trial vector is worth more than the target vector, it will be one of the next generation members. Otherwise, the target vector becomes one of the next generation population.

$$X_{ji,G+1} = \begin{Bmatrix} u_{ji,G+1} & If\ f(u_{i,G+1}) \le f(x_{i,G}) \\ x_{ji,G} & Otherwise \end{Bmatrix} \tag{14}$$

3.3. SaE-ELM Model

The SaE-ELM is a novel learning algorithm named self-adaptive evolutionary (SaE) extreme learning machine (ELM) which is proposed by (Cao, Lin, & Huang, 2012) to overcome the limitation manually choosing control parameters and selectiob of trial vector generation strategies in (Zhu et al. 2005; Subudhi & Jena, 2008). The SaE-ELM algorithm for SLFFNN is a combination of self-adaptive differential evolution algorithm (SaDE) (Qin, Huang, & Suganthan, 2009) and ELM (Guang-Bin Huang & Chee-Kheong Siew, 2004). The hidden node biases and network input weights are optimized by SaDE and the network output weights are derived by ELM.

Consider a training data set (D) with an activation function (g(x)) and M hidden nodes, the SaE-ELM algorithm are summarized in the four steps: 1) Initialization; 2) Calculations of output weights and RMSE; 3) Mutation and Crossover; and 4) Evaluation.

1. Initialization

The first generation population is initialized as a NP vectors so that each one containing the parameters of hidden node.

$$X_{k,G} = \left[a^T_{1,(k,G)}, a^T_{2,(k,G)}, \ldots, a^T_{M,(k,G)}, b_{1,(k,G)}, b_{2,(k,G)}, \ldots, b_{M,(k,G)}\right] \tag{15}$$

where G is the generation number, k=1, 2, …, NP; and a_i and b_i are randomly produced (i=1, 2, …, M).

2. Calculations of Output Weights and RMSE

The root mean square error (RMSE) and the matrix of output weight du to each population vector are calculated as follow, respectively:

$$\mathrm{RMSE}_{k,G} = \sqrt{\frac{\sum_{i=1}^{N}\left\|\sum_{j=1}^{M}\beta_j g\left(a_{j,(k,G)}, b_{j,(k,G)}, x_i\right) - t_i\right\|}{m \times N}} \tag{16}$$

$$\beta_{k,G} = H_{k,G}^{+} T \tag{17}$$

where $H^{+}_{k,G}$ is the is the MPGI verse of $H_{k,G}$. The $H_{k,G}$ is defined as follows:

$$\mathrm{H}_{k,G} = \begin{bmatrix} g\left(a_{1,(k,G)}, b_{1,(k,G)}, x_1\right) & \cdots & g\left(a_{M,(k,G)}, b_{M,(k,G)}, x_1\right) \\ \vdots & \ddots & \vdots \\ g\left(a_{1,(k,G)}, b_{1,(k,G)}, x_N\right) & \cdots & g\left(a_{M,(k,G)}, b_{M,(k,G)}, x_N\right) \end{bmatrix} \tag{18}$$

$$X_{k,G+1} = \begin{bmatrix} u_{k,G+1} & if\ RMSE_{X_{k,G}} - RMSE_{X_{K,G+1}} > \varepsilon \cdot RMSE_{X_{k,G}} \\ u_{k,G+1} & if\ \left|RMSE_{X_{k,G}} - RMSE_{u_{K,G+1}}\right| < \varepsilon \cdot RMSE_{X_{k,G}}\ and\ \left\|\beta_{u_{K,G+1}}\right\| < \left\|\beta_{X_{K,G}}\right\| \\ X_{k,G} & otherwise \end{bmatrix} \tag{19}$$

The population vector with the lowest RMSE in the first generation is known as $RMSE_{\theta_{best,1}}$ and saved in $\theta_{best,1}$.

3. Mutation and Crossover

For each target vector correspond to current generation, the trial vector production strategy (Equations 9-12) is selected from a candidate pool constructed du to probability $p_{l,G}$ (Qin, Huang, & Suganthan, 2009) where $p_{l,G}$ is the selection probability of l^{th} strategy at the G^{th} generation. By definition of generation number as learning period (LP), the probability is updated based on comparison of current generation number (G) and LP. If the current generation number is lower than generation number (G≤LP), the probability of each strategy is equal (i.e. $p_{l,G}$ =0.25). Otherwise (G>LP), the probability og each one are computed as follows:

$$P_{l,G} = \frac{\dfrac{\sum\limits_{g=G-LP}^{G-1} ns_{l,g}}{\sum\limits_{g=G-LP}^{G-1} ns_{l,g} + \sum\limits_{g=G-LP}^{G-1} nf_{l,g}} + \varepsilon}{\sum\limits_{l=1}^{4} S_{l,G}}, \left(l = 1,2,3,4\right) \tag{20}$$

where $ns_{l,g}$ and $nf_{l,g}$ are the number of generated trial vector by l^{th} strategy at g^{th} iterations that can enter or discard the next iteration, respectively. The number of failure and success trial vectors are saved. Once the generations go beyond the initial iterations, the new records are replaced with earliest ones. The ε is a positive constant value to prevent the possible null success rate. Moreover, a set oc control parameters namely as crossover constant (C_r) and scale factor (F) are randomly produced for each target vector due to the normal distribution $N(0.5,0.1)$ and $N(0.5,0.3)$ (respectively).

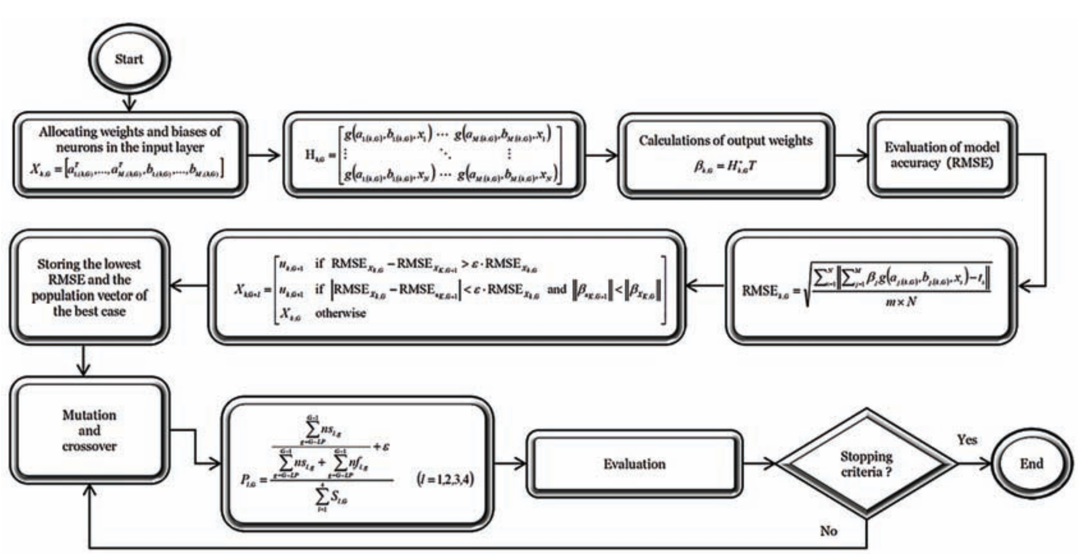
4. Evaluation

All generated trial vectors at $(G+1)^{th}$ iterations are assessment using Equation (19). Due to (Bartlett, 1998) underscore, the norm of output weight $\left(\|\beta\|\right)$ is also considered as additional criteria in selection of trial vector to reach better generalization performance with small weights. Steps *mutation and cross-over* and "evaluation" are repeated until the LP is completed or expected goal is achieved. The flowchart of SaE-ELM is presented in Figure 4.

5. DISCUSSION OF THE APPLICATION

The effectiveness of the proposed techniques is examined upon real historical water level data obtained From the Department of Irrigation and Drainage (DID), Malaysia. In this section, a detailed description and analysis of the proposed and the comparable predictive methods are debated. It should be noticed that the utilized data is continuous and not experience any missing monitoring events data during the period under the study. The forecasting skill was examined using multiple statistical index including regression coefficient R^2, coefficient of determination R, variance (VAF), root mean square error (RMSE), scat-

Figure 4. The flowchart of SaE-ELM model



ter index (SI), mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square relative error (RMSRE), mean relative error (MRE), BIAS and Nash-Sutcliffe coefficient (*NS*).

Table 1 and 2 displayed the performance indicators of daily time scale modelling for ELM and SaE-ELM, respectively. As visualized in these two tables that nine antecedent values of water level were used to construct the forecasting models. The best performance accuracies were performed for three previous records to forecast one step ahead consistently for both model. The results of the statistical metrics showed a harmony the in the performance in term of minimum absolute error measures and maximum best fit of goodness measurements.

On the other hand, Table 3 and 4 tabulated the forecasting skills indicators of the long-term (e.g., monthly time scale) for ELM and SaE-ELM, respectively. In this case, the constructed input variables were inspected up to eleven months lag time. This is owing to the results of the correlation statistics. The best lag time performed for ELM model is nine antecedents' values. Whereas, SaE-ELM model achieved its best forecasting accuracies using five lag times attributes.

Scatter plots graphical inspection generated between the observed and forecasted values of river water level over the testing period (See Figure 5). The figure revealed an excellent agreement between

Table 1. Forecasting performance of daily time scale water level using ELM model

Models	R^2	R	VAF	RMSE	SI	MAE	MAPE	RMSRE	MRE	BIAS	NS
Model 1	0.925	0.962	92.361	0.294	0.065	0.179	0.038	0.058	-0.008	-0.027	0.922
Model 2	0.930	0.965	93.021	0.280	0.062	0.172	0.036	0.053	-0.005	-0.014	0.927
Model 3*	0.933	0.966	93.256	0.275	0.061	0.170	0.036	0.053	-0.005	-0.014	0.930
Model 4	0.925	0.962	92.523	0.289	0.064	0.173	0.037	0.056	-0.003	-0.003	0.922
Model 5	0.915	0.956	91.376	0.311	0.069	0.199	0.043	0.064	0.001	0.017	0.911
Model 6	0.926	0.962	92.517	0.290	0.064	0.179	0.038	0.057	-0.002	0.000	0.923
Model 7	0.908	0.953	90.595	0.325	0.072	0.213	0.046	0.064	0.000	0.007	0.906
Model 8	0.921	0.960	92.020	0.299	0.066	0.185	0.039	0.058	-0.002	0.003	0.917
Model 9	0.905	0.951	90.247	0.331	0.073	0.217	0.047	0.069	0.000	0.012	0.902

Table 2. Forecasting performance of time scale water level using SaE-ELM model

Models	R^2	R	VAF	RMSE	SI	MAE	MAPE	RMSRE	MRE	BIAS	NS
Model 1	0.925	0.962	92.347	0.294	0.065	0.179	0.038	0.058	-0.008	-0.027	0.922
Model 2	0.931	0.965	93.045	0.280	0.062	0.170	0.036	0.053	-0.005	-0.014	0.928
Model 3*	0.934	0.966	93.354	0.273	0.060	0.166	0.035	0.052	-0.005	-0.010	0.931
Model 4	0.934	0.967	93.427	0.272	0.060	0.167	0.035	0.051	-0.005	-0.010	0.931
Model 5	0.927	0.963	92.680	0.287	0.063	0.179	0.038	0.056	-0.006	-0.013	0.924
Model 6	0.928	0.963	92.807	0.284	0.063	0.175	0.037	0.055	-0.006	-0.018	0.924
Model 7	0.927	0.963	92.212	0.295	0.065	0.200	0.043	0.059	0.000	-0.004	0.927
Model 8	0.931	0.965	93.021	0.280	0.062	0.174	0.037	0.053	-0.005	-0.012	0.928
Model 9	0.925	0.962	92.111	0.297	0.066	0.203	0.044	0.060	-0.001	-0.007	0.925

Table 3. Forecasting performance of monthly time scale water level using ELM model

Models	R^2	R	VAF	RMSE	SI	MAE	MAPE	RMSRE	MRE	BIAS	NS
Model 1	0.853	0.923	84.457	0.405	0.090	0.280	0.060	0.082	-0.017	-0.067	0.845
Model 2	0.868	0.931	86.748	0.368	0.082	0.242	0.051	0.070	-0.005	0.002	0.843
Model 3	0.881	0.939	88.080	0.350	0.077	0.241	0.051	0.068	-0.008	-0.014	0.864
Model 4	0.889	0.943	88.781	0.340	0.075	0.237	0.050	0.066	-0.010	-0.032	0.881
Model 5	0.873	0.934	87.143	0.364	0.081	0.242	0.051	0.071	-0.009	-0.026	0.864
Model 6	0.853	0.923	85.161	0.391	0.087	0.261	0.054	0.075	-0.010	-0.028	0.837
Model 7	0.844	0.919	84.384	0.400	0.089	0.270	0.057	0.078	-0.009	-0.020	0.824
Model 8	0.852	0.923	84.912	0.395	0.087	0.261	0.056	0.078	-0.011	-0.033	0.840
Model 9*	0.890	0.943	88.906	0.339	0.075	0.241	0.051	0.066	-0.011	-0.036	0.882
Model 10	0.878	0.937	87.750	0.356	0.079	0.246	0.052	0.068	-0.010	-0.033	0.867
Model 11	0.878	0.937	87.196	0.368	0.081	0.276	0.059	0.075	-0.016	-0.064	0.872

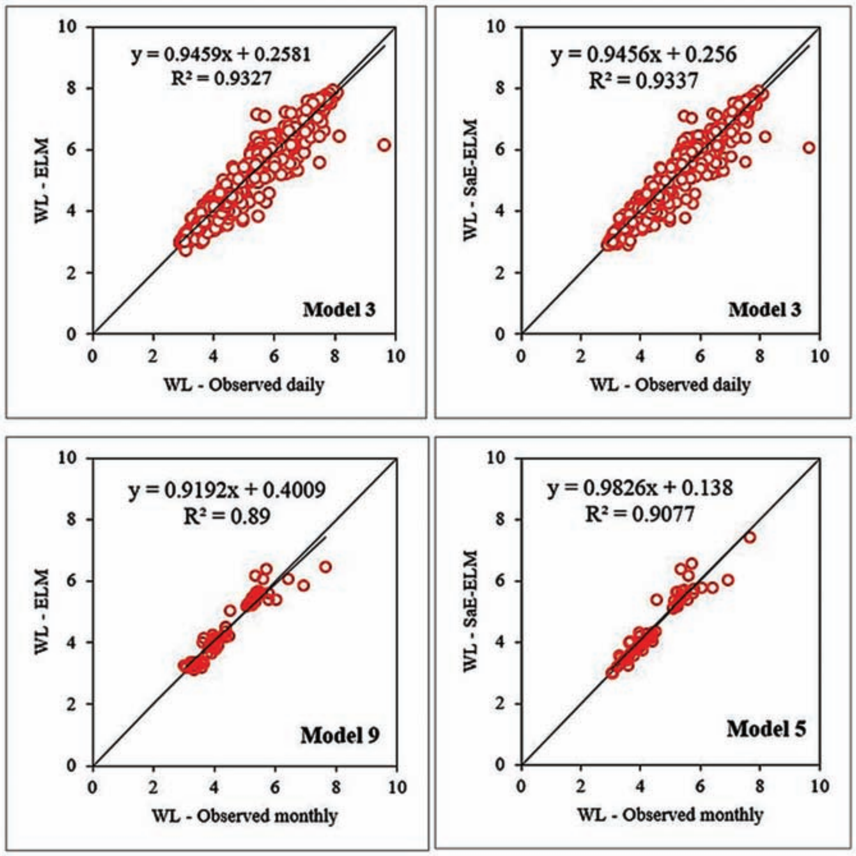
Table 4. Forecasting performance of monthly time scale water level using SaE-ELM model

Models	R^2	R	VAF	RMSE	SI	MAE	MAPE	RMSRE	MRE	BIAS	NS
Model 1	0.865	0.930	85.932	0.384	0.085	0.265	0.057	0.079	-0.016	-0.059	0.858
Model 2	0.877	0.936	87.610	0.356	0.079	0.229	0.048	0.067	-0.004	0.005	0.852
Model 3	0.881	0.939	88.080	0.350	0.077	0.241	0.051	0.068	-0.008	-0.014	0.864
Model 4	0.895	0.946	89.340	0.333	0.074	0.237	0.050	0.064	-0.012	-0.044	0.889
Model 5*	0.908	0.953	90.156	0.323	0.072	0.214	0.045	0.065	-0.014	-0.059	0.905
Model 6	0.875	0.935	87.363	0.361	0.080	0.235	0.049	0.071	-0.009	-0.027	0.865
Model 7	0.848	0.921	84.750	0.395	0.088	0.259	0.055	0.076	-0.007	-0.010	0.825
Model 8	0.876	0.936	87.484	0.359	0.080	0.248	0.053	0.072	-0.010	-0.030	0.864
Model 9	0.896	0.946	89.384	0.333	0.074	0.233	0.049	0.066	-0.012	-0.046	0.890
Model 10	0.890	0.943	88.936	0.337	0.075	0.226	0.047	0.065	-0.008	-0.024	0.879
Model 11	0.882	0.939	88.161	0.349	0.077	0.236	0.048	0.066	-0.009	-0.024	0.872

the observed and forecasted values for both time scales. However, SaE-ELM model outperformed the ELM model for both investigated time scales.

Evidently, implementing only historical water level information is not sufficiently provide outstanding forecasting skills. Thus, it is necessary to consider other model input parameters which are dependent such as other related metrological, physical or even climatological. Both models have been developed performed almost the same level of accuracy; apparently, the predictive models were not the source the low prediction skills yet the nature of the water level pattern is highly complicated. Based on the authors knowledge, the perception is the rainfall is the source of the high fluctuation of the time series data. Therefore, reason that there could be an improvement in forecasts only by using climatic parameters to support the historical datasets in order to develop a good and successful model for forecasting. It may

Figure 5. Scatter plot graphical presentation over the testing phase using ELM and SaE-ELM models



be expected that the availability of parameters for rainfall parameters and their use as input variables during modelling can improve the level of accuracy.

6. CONCLUSION AND REMARKS

In this chapter, a novel predictive model called (SaE-ELM) is proposed to forecast one-step-ahead short- and long-term river water level. To achieve this objective, a highly stochastic case study is examined positioned in tropical environment, Linggi River, Negeri Sembilan state in peninsular Malaysia. The forecasting modelling has been undertaken based on the correlated antecedent values of the time series. The results of the proposed approach compared and assessed with ELM model using several performance indicators. The results demonstrated that SaE-ELM model can be applied efficiently to establish accurate and reliable daily and monthly water level at this particular case study. This research can be expended for future investigation which might be enhance the forecasting modelling. For instance, involve some other hydrological parameters “e.g., rainfall, streamflow or other catchment physical properties” that may provide more knowledge about the water level phenomena.

REFERENCES

Adhikari, P., Hong, Y., Douglas, K. R., Kirschbaum, D. B., Gourley, J., Adler, R., & Brakenridge, G. R. (2010). A digitized global flood inventory (1998-2008): Compilation and preliminary results. *Natural Hazards*, *55*(2), 405–422. doi:10.100711069-010-9537-2

Afan, H. A., El-Shafie, A., Yaseen, Z. M., Hameed, M. M., Wan Mohtar, W. H. M., & Hussain, A. (2014). ANN Based Sediment Prediction Model Utilizing Different Input Scenarios. *Water Resources Management*, *29*(4), 1231–1245. doi:10.100711269-014-0870-1

Allenby, B., & Fink, J. (2005). Toward inherently secure and resilient societies. *Science*, *309*(5737), 1034–1036. doi:10.1126cience.1111534 PMID:16099973

Alvisi, S., & Franchini, M. (2011). Fuzzy neural networks for water level and discharge forecasting with uncertainty. *Environmental Modelling & Software*, *26*(4), 523–537. doi:10.1016/j.envsoft.2010.10.016

Azimi, H., Bonakdari, H., & Ebtehaj, I. (2017). Sensitivity analysis of the factors affecting the discharge capacity of side weirs in trapezoidal channels using extreme learning machines. *Flow Measurement and Instrumentation*, *54*, 216–223. doi:10.1016/j.flowmeasinst.2017.02.005

Bartlett, P. L. (1998). The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, *44*(2), 525–536. doi:10.1109/18.661502

Bazartseren, B., Hildebrandt, G., & Holz, K.-P. (2003). Short-term water level prediction using neural networks and neuro-fuzzy approach. *Neurocomputing*, *55*(3–4), 439–450. doi:10.1016/S0925-2312(03)00388-6

Campolo, M., Andreussi, P., & Soldati, A. (1999). River flood forecasting with a neural network model. *Water Resources Research*, *35*(4), 1191–1197. doi:10.1029/1998WR900086

Cao, J., Lin, Z., & Huang, G. (2012). Self-adaptive evolutionary extreme learning machine. *Neural Processing Letters*, *36*(3), 285–305. doi:10.100711063-012-9236-y

Chang, F. J., & Chang, Y. T. (2006). Adaptive neuro-fuzzy inference system for prediction of water level in reservoir. *Advances in Water Resources*, *29*(1), 1–10. doi:10.1016/j.advwatres.2005.04.015

Chang, F.-J., Chen, P.-A., Lu, Y.-R., Huang, E., & Chang, K.-Y. (2014). Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. *Journal of Hydrology (Amsterdam)*, *517*, 836–846. doi:10.1016/j.jhydrol.2014.06.013

Chau, K. W. (2006). Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. *Journal of Hydrology (Amsterdam)*, *329*(3–4), 363–367. doi:10.1016/j.jhydrol.2006.02.025

Coppin, B. (2004). Artificial Intelligence Illuminated. *Expert Systems: International Journal of Knowledge Engineering and Neural Networks*. doi:10.1049/esn.1987.0009

Deo, R. C., & Şahin, M. (2015). Application of the Artificial Neural Network model for prediction of monthly Standardized Precipitation and Evapotranspiration Index using hydrometeorological parameters and climate indices in eastern Australia. *Atmospheric Research*, *161–162*, 65–81. doi:10.1016/j.atmosres.2015.03.018

Deo, R. C., Samui, P., & Kim, D. (2015). Estimation of monthly evaporative loss using relevance vector machine, extreme learning machine and multivariate adaptive regression spline models. *Stochastic Environmental Research and Risk Assessment*. doi:10.100700477-015-1153-y

Ebtehaj, I., & Bonakdari, H. (2016). A Comparative Study of Extreme Learning Machines and Support Vector Machines in Prediction of Sediment Transport in Open Channels. *International Journal of Engineering Transactions B: Applications International Journal of Engineering J, 29*(11). Retrieved from https://www.researchgate.net/profile/Isa_Ebtehaj2/publication/308796822_A_Comparative_Study_of_Extreme_Learning_Machines_and_Support_Vector_Machines_in_Prediction_of_Sediment_Transport_in_Open_Channels/links/5836f88c08ae3d91723ba832.pdf

Ebtehaj, I., Sattar, A. M. A., Bonakdari, H., & Zaji, A. H. (2017). Prediction of scour depth around bridge piers using self-adaptive extreme learning machine. *Journal of Hydroinformatics*, *19*(2), 207–224. doi:10.2166/hydro.2016.025

Elhatip, H., & Kömür, M. A. (2008). Evaluation of water quality parameters for the Mamasin dam in Aksaray City in the central Anatolian part of Turkey by means of artificial neural networks. *Environmental Geology*, *53*(6), 1157–1164. doi:10.100700254-007-0705-y

Fahimi, F., Yaseen, Z. M., & El-shafie, A. (2016). Application of soft computing based hybrid models in hydrological variables modeling: A comprehensive review. *Theoretical and Applied Climatology*, 1–29. doi:10.100700704-016-1735-8

Fallah-Mehdipour, E., Bozorg Haddad, O., & Mariño, M. A. (2013). Prediction and simulation of monthly groundwater levels by genetic programming. *Journal of Hydro-environment Research*, *7*(4), 253–260. doi:10.1016/j.jher.2013.03.005

Faruk Ertuğrul, Ö., & Kaya, Y. (2014). A detailed analysis on extreme learning machine and novel approaches based on ELM. *American Journal of Computer Science and Engineering*, *1*(5), 43–50. Retrieved from http://www.openscienceonline.com/journal/ajcse

Ford, N. J. (1987). Artificial intelligence: The very idea. *International Journal of Information Management*, *7*(1), 59–60. doi:10.1016/0268-4012(87)90008-9

Furquim, G., Mello, R., & Pessin, G. (2014). An Accurate Flood Forecasting Model Using Wireless Sensor Networks and Chaos Theory: A Case Study with Real WSN Deployment in Brazil. In *International Conference on Engineering Applications of Neural Networks* (pp. 92–102). Retrieved from http://link.springer.com/chapter/10.1007/978-3-319-11071-4_9

Gevarter, W. B. (1987). Introduction to artificial intelligence. *Chemical Engineering Progress*, *83*(9), 21–37. doi:10.2207/qjjws1943.57.490

Han, D., Chan, L., & Zhu, N. (2007). Flood forecasting using support vector machines. *Journal of Hydroinformatics*, *9*(4), 267. doi:10.2166/hydro.2007.027

Huang, G., Huang, G., Song, S., & You, K. (2015). Trends in extreme learning machines: A review. *Neural Networks*, *61*, 32–48. doi:10.1016/j.neunet.2014.10.001 PMID:25462632

Huang, G.-B., & Siew, C-K. (2004). Extreme learning machine: RBF network case. In *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference* (*Vol. 2*, pp. 1029–1036). Academic Press. 10.1109/ICARCV.2004.1468985

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, *70*(1-3), 489–501. doi:10.1016/j.neucom.2005.12.126

Jonkman, S. N. (2005). Global perspectives on loss of human life caused by floods. *Natural Hazards*, *34*(2), 151–175. doi:10.100711069-004-8891-3

Liang, N.-Y., Huang, G.-B., Saratchandran, P., & Sundararajan, N. (2006). A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on Neural Networks, 17*(6), 1411–1423. 10.1109/TNN.2006.880583

Liong, S.-Y., Lim, W.-H., & Paudyal, G. N. (2000). River Stage Forecasting in Bangladesh: Neural Network Approach. *Journal of Computing in Civil Engineering*, *14*(1), 1–8. doi:10.1061/(ASCE)0887-3801(2000)14:1(1)

Liong, S. Y., & Sivapragasam, C. (2002). Flood stage forecasting with support vector machines. *Journal of the American Water Resources Association*, *38*(1), 173–186. doi:10.1111/j.1752-1688.2002.tb01544.x

Mwale, F. D., Adeloye, A. J., & Rustum, R. (2014). Application of self-organising maps and multi-layer perceptron-artificial neural networks for streamflow and water level forecasting in data-poor catchments: The case of the Lower Shire floodplain, Malawi. *Hydrology Research*, *45*(6), 838–854. doi:10.2166/nh.2014.168

Nourani, V., Hosseini Baghanam, A., Adamowski, J., & Kisi, O. (2014). Applications of hybrid wavelet–Artificial Intelligence models in hydrology: A review. *Journal of Hydrology (Amsterdam)*, *514*, 358–377. doi:10.1016/j.jhydrol.2014.03.057

Noureldin, A., El-Shafie, A., & Bayoumi, M. (2011). GPS/INS Integration Utilizing Dynamic Neural Networks for Vehicular Navigation. *Information Fusion*, *12*(1), 48–57. doi:10.1016/j.inffus.2010.01.003

Ouyang, H.-T. (2016, February). Multi-objective optimization of typhoon inundation forecast models for water-level gauging network by integration of ARMAX with a Genetic Algorithm. *Natural Hazards and Earth System Sciences Discussions*, 1–22. 10.5194/nhess-2016-1

Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, *13*(2), 398–417. doi:10.1109/TEVC.2008.927706

Russell, S. J., & Norvig, P. (2010). Artificial Intelligence: A Modern Approach. *Artificial Intelligence*. doi:10.1017/S0269888900007724

Smith, K., & Petley, D. N. (2008). *Environmental Hazards: Assessing risk and reducing disaster*. Routledge Taylor & Francis Group. 10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C

Smith, K., & Ward, R. (1998). *Floods: Physical processes and human impacts*. New York: John Wiley and Sons Ltd.

Soleymani, S. A., Goudarzi, S., Anisi, M. H., Hassan, W. H., Idris, M. Y. I., Shamshirband, S., ... Ahmedy, I. (2016). A Novel Method to Water Level Prediction using RBF and FFA. *Water Resources Management*, 1–19. doi:10.100711269-016-1347-1

Solomatine, D. P., & Xue, Y. (2004). M5 Model Trees and Neural Networks: Application to Flood Forecasting in the Upper Reach of the Huai River in China. *Journal of Hydrologic Engineering*, *9*(6), 491–501. doi:10.1061/(ASCE)1084-0699(2004)9:6(491)

Storn, R., & Price, K. (1995). *Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces*. International Computer Science Institute-Publications-Tr. doi:10.1023/A:1008202821328

Sulaiman, M., El-Shafie, A., Karim, O., & Basri, H. (2011). Improved water level forecasting performance by using optimal steepness coefficients in an artificial neural network. *Water Resources Management*, *25*(10), 2525–2541. doi:10.100711269-011-9824-z

Tan, M. L., Ibrahim, A. L., Yusop, Z., Chua, V. P., & Chan, N. W. (2017). Climate change impacts under CMIP5 RCP scenarios on water resources of the Kelantan River Basin, Malaysia. *Atmospheric Research*, *189*, 1–10. doi:10.1016/j.atmosres.2017.01.008

Thirumalaiah, K., & Deo, M. C. (1998). River Stage Forecasting Using Artificial Neural Networks. *Journal of Hydrologic Engineering*, *3*(1), 26–32. doi:10.1061/(ASCE)1084-0699(1998)3:1(26)

Wei, C. C. (2012). Wavelet kernel support vector machines forecasting techniques: Case study on water-level predictions during typhoons. *Expert Systems with Applications*, *39*(5), 5189–5199. doi:10.1016/j.eswa.2011.11.020

Yaseen, Z. M., El-shafie, A., Jaafar, O., Afan, H. A., & Sayl, K. N. (2015). Artificial intelligence based models for stream-flow forecasting: 2000–2015. *Journal of Hydrology (Amsterdam)*, *530*, 829–844. doi:10.1016/j.jhydrol.2015.10.038

Yaseen, Z. M., Kisi, O., & Demir, V. (2016). Enhancing Long-Term Streamflow Forecasting and Predicting using Periodicity Data Component: Application of Artificial Intelligence. *Water Resources Management*, *30*(12), 4125–4151. doi:10.100711269-016-1408-5

Zadeh, L. A. (1994). Soft computing and fuzzy logic. *IEEE Software*, *11*(6), 48–56. doi:10.1109/52.329401