Enhancing accuracy of autoregressive time series forecasting with input selection and wavelet transformation

H. D. Tran, N. Muttil and B. J. C. Perera

ABSTRACT

Autoregressive time series forecasting is common in different areas within water resources, which include hydrology, ecology, and the environment. Simple forecasting models such as linear regression have the advantage of fast runtime, which is attractive for real-time forecasting. However, their forecasting performance might not be acceptable when a non-linear relationship exists between model inputs and output, which necessitates the use of more sophisticated forecasting models such as artificial neural networks. This study investigates the performance and potential of a hybrid pre-processing technique to enhance the forecasting accuracy of two commonly used neural network models (feed-forward and layered recurrent neural network models) and a multiple linear regression model. The hybrid technique is a combination of significant input variable selection (using partial linear correlation) for reducing the dimensionality of the input data and input data transformation using discrete wavelet transform for decomposing the input time series into low and high frequency components. Two case study forecasting applications, namely, monthly inflow forecasting for a lake in Victoria (Australia) and weekly algal bloom predictions at a bay in Hong Kong were used to assess the forecasting ability of the models when used in conjunction with the hybrid technique. Results demonstrated that the hybrid technique can significantly improve the forecasting performance of all the models used.

Key words | artificial neural networks, autoregressive forecast, input selection, multiple linear regression, wavelet decomposition

H. D. Tran (corresponding author)
School of Civil, Environmental and Chemical Engineering,
RMIT University,
Melbourne,
Victoria 3000,
Australia
E-mail: dungplus@yahoo.com

N. Muttil
B. J. C. Perera
College of Engineering and Science,
Victoria University,
P.O. Box 14428,
Melbourne,
Victoria 8001,
Australia

N. Muttil
B. J. C. Perera
Institute for Sustainability and Innovation,
Victoria University,
P.O. Box 14428,
Melbourne,
Victoria 8001,
Australia

INTRODUCTION

Autoregressive time series forecasting is common in many different disciplines such as the financial sector and in water resources. The important feature of the autoregressive time series forecasting is the use of a model to forecast future values of a time series variable of interest in a process or a system through the use of its known past values. The goal of autoregressive time series forecasting is to provide accurate forecasts of the variable of interest, such as the foreign exchange rates (Vojinovic *et al.* 2001), streamflow (Wei *et al.* 2012), salinity (Maier *et al.* 2010), and chlorophyll (Cheng & Wei 2009). Such forecast information can be utilized in future operational planning of the system at hand.

The general difficulty associated with autoregressive time series forecasting is the non-linear and non-stationary characteristics which are often encountered in most water-related time series data (Coulibaly *et al.* 2001). The non-linear characteristic of the time series data shows the dynamic relationship between past values (i.e., model inputs) and future values (i.e., model outputs) in a way that a small change in model inputs will affect model outputs significantly. Exponentially increasing the magnitude of model outputs due to a small change in model inputs is one example of non-linear behavior, while the effect is ignorable in the linear behavior (Makridakis *et al.* 2003). The

non-linear characteristic is commonly modeled as a non-linear combination of model inputs to predict the model output.

The non-stationary characteristic of the time series data is realistically considered as not having constant mean and/or constant variation within the time series (Tsay 2005). This characteristic can be observed through the seasonality and/or the trend in the time series data. However, the key consideration is not the time variation of data but whether the underlying process that generates the data is itself evolving (Cannas *et al.* 2006). This consideration is commonly taken into account by the use of a 'random walk' model, which assumes that the future value depends on the previous value plus a random noise in order to capture the non-stationary characteristic (Tsay 2005).

Many forecasting models have been developed in the past. The multiple linear regression (MLR) model is among the simplest, which basically performs forecasts by fitting a linear relationship between inputs and outputs. Although the MLR model might be very attractive for real-time forecasting due to its simplicity, the MLR model is considered unsuitable for non-linear data, which necessitate the use of more sophisticated models (Elshorbagy *et al.* 2010; Jung *et al.* 2010). To overcome this limitation of linear models, various data-driven modeling techniques like artificial neural networks (ANN), support vector machines (SVM), and genetic programming (GP) have been proposed (Minns & Hall 1996; Babovic & Keijzer 2002; Liong *et al.* 2002; Yu *et al.* 2004). Among them, ANN models have been widely used for autoregressive time series forecasting over the last 15 years (Islam 2010; Maier *et al.* 2010; Arunkumar & Jothiprakash 2013). Several distinguishing features of ANN models make them valuable and attractive for forecasting tasks (Samarasinghe 2006; Maier *et al.* 2010). First, there are few *a priori* assumptions about the ANN models, and they learn from examples and capture the functional relationship in the data even if the underlying relationships are too complex to specify. Second, ANN models can generalize after learning from the sample data presented to them. Third, ANN models are universal functional approximators for any continuous function to the desired accuracy.

There are many different types of ANN models, and among them, feed-forward neural networks (FFNN) and layered recurrent neural networks (LRNN) have gained attention in the literature for autoregressive time series forecasting (Güldal & Tongal 2010; Wei *et al.* 2012). FFNN are static and are the most commonly used architecture due to their simple framework (Güldal & Tongal 2010). The LRNN provide a representation of dynamic internal feedback loops to store information for later use and to enhance the efficiency of learning. The LRNN is more suited for mapping problems that involve auto-correlated samples (Parlos *et al.* 2000). However, the forecasting accuracy of ANN models has not always been high. For example, in a recent study by Wang *et al.* (2009) involving ANN models for forecasting monthly streamflow time series at two different case study locations, one demonstrated a high Nash–Sutcliffe coefficient (E) value of 0.87 while the other had an E value of 0.61 between observed and forecasted streamflow values for the validation dataset. Furthermore, there is limited information on comparative performances between feed-forward and RNN on the autoregressive time series forecasting.

Recently, input selection techniques have been used for reducing the number of model inputs, and can improve forecasting performance in some cases (May *et al.* 2008; Tran *et al.* 2015). Input selection techniques can be basically divided into two approaches, namely, model-free and model-based approaches (Fernando *et al.* 2009). Model-free approaches use statistical measures of dependence to determine the strength of the relationship between candidate model inputs and the model output. The partial linear correlation (PLC) technique is the simplest model-free technique, which can select inputs that are linearly correlated with outputs. The PLC technique has been used in several studies (Tiwari *et al.* 2013; Valipour *et al.* 2013; Liu *et al.* 2014). The partial mutual information (PMI) technique (Fernando *et al.* 2009) is another model-free technique which can select inputs that are non-linearly related with outputs. On the other hand, model-based approaches use the performance of calibrated models with different inputs as the basis for choosing the most appropriate inputs. ANN and GP based techniques are examples of the model-based approach. Muttil & Chau (2007) compared two model-based approaches (ANN and GP) for selecting ecologically significant inputs for algal bloom prediction. They showed that both these methods produced similar results. Tran *et al.* (2015) conducted a comparative study between PLC,

PMI, and GP techniques on four hypothetical and two real datasets. They found that there are differences in significant inputs selected by these techniques. However, in terms of forecasting performance, inputs selected by the PLC technique provided similar forecasting performance with inputs selected by the PMI and GP techniques when tested using ANN models.

Another approach that has also been recently used to improve the forecasting performance of ANN models is input preprocessing using wavelet transformation (Adamowski & Sun 2010; Wei *et al.* 2012; Tiwari *et al.* 2013). The wavelet transformation is used to decompose an input variable into a set of linear components with high and low frequency. Such decomposition can be considered to separate the non-stationary and stationary parts, which becomes particularly useful for autoregressive time series forecasting. However, in their study, the effect of different wavelet parameters including wavelet functions and decomposition levels have not been investigated.

In this study, the overall objective was to improve the performance of autoregressive time series forecasting by using a hybrid combination of significant input selection and input preprocessing using wavelet transformation. Several studies have used the hybridization of various techniques to enhance the forecasting accuracy (Kisk 2010; Cao *et al.* 2013; Sun *et al.* 2014). However, as far as the authors are aware, the hybridization of the PLC technique for input selection (used in this study because of its simplicity) with wavelet decomposition to improve the forecasting performance of MLR and ANN models has not been undertaken in the past. The major contributions of this paper are as follows:

(i) A hybrid technique that combines input selection using the PLC technique with wavelet decomposition for MLR and ANN models is investigated for enhancing autoregressive time series forecasting.

(ii) Wavelet transformation has been demonstrated to be useful for improving time series forecasting. This study analyzes the sensitivity of different decomposition levels and different wavelet functions on the forecasting performance.

(iii) The improvement in performance of two popularly used ANN models (FFNN and LRNN) when used in conjunction with input selection and wavelet transformation is analyzed.

The remainder of this paper is structured as follows. Brief descriptions of feed-forward and layered recurrent ANN models, and then PLC and wavelet decomposition techniques are presented in the next section. A description of the performance indicators used in this study to evaluate the performance of the models is also included in the next section. This is followed by a description of the two case study datasets. Results and discussion are then presented, which is followed by the conclusions drawn from this study.

METHODS AND TECHNIQUES

Figure 1 presents a flow chart of the input processing techniques used and the ANN models that are compared in this study. The first set of inputs is all probable inputs (called Original inputs). The PLC technique is then applied to the Original inputs to produce the second and reduced set of inputs (called PLC-selected inputs). The discrete wavelet decomposition (DWD) technique is then applied to the PLC-selected inputs to create the third set of inputs (called PLC-Wavelet inputs). These three sets of inputs are used as inputs for MLR, FFNN, and LRNN models and the predicted outputs are compared using various performance indicators. Brief descriptions of feed-forward and layered recurrent ANN models are also provided in this section. The MLR model is well known and hence is not presented. This is followed by descriptions of the PLC input selection technique, wavelet decomposition technique, and the performance indicators used to evaluate the models. Detailed explanations of the techniques used in this study
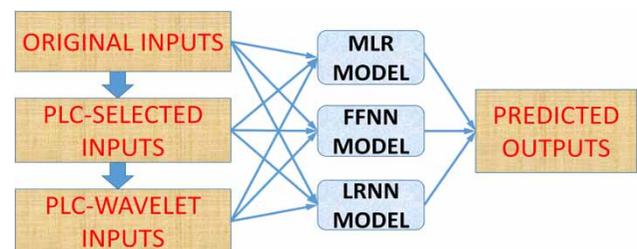

Figure 1 | Flow chart of input processing techniques and the models compared in this study.

can be found in the relevant references presented in each of the following sub-sections.

ANN models

Many types of ANN have been developed in the last few decades, starting with the well-known multilayer perceptron structure (Samarasinghe 2006). Each perceptron is a signal processing unit, and a network of connected perceptrons is capable of performing classifying and forecasting tasks. The classifying and forecasting tasks are done through two-step processes of training (i.e., learning from sample data) and simulation (i.e., generalizing to classify or forecast). Multilayer perceptron networks are further classified into static FFNN and dynamic RNN with regards to how signals are transferred between layers of perceptron. In the static FFNN, signals essentially propagate in a forward direction, while a backward propagation of so-called feedback signals can occur in the RNN (Güldal & Tongal 2010). The static FFNN with a typical layout, as shown in Figure 2(a), can achieve satisfactory analytical outcomes if sufficient data are available. Nevertheless, the static neural networks may have difficulty in modeling the temporal or short-term dynamic properties. The FFNN was investigated in this study.

The RNN, on the other hand, has some feedback loops from the output layer or hidden layer to the input layer, as shown in Figure 2(b). The RNN allows signals to propagate in both forward and backward directions, which offers the network dynamic memories (Samarasinghe 2006). In other words, RNN has the capacity associated with the feedback loops to delay and store information from the previous time step that is not present in the architecture of feed-forward networks. The presence of these feedback loops
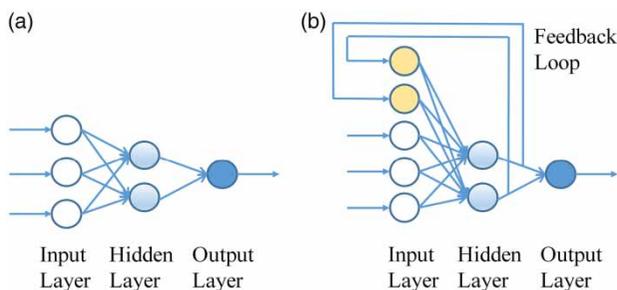
Figure 2 │ A typical layout of (a) a feed-forward neural network and (b) LRNN (adapted from Güldal & Tongal 2010).

has a profound impact on the learning capability and performance of the neural network, which allows the network to capture the true hidden dynamic memories or autoregressive components of nonlinear time series systems (Carcano *et al.* 2008). The RNN has been proved to be a powerful method for handling complex systems such as nonlinear time-varying systems. A LRNN, which is a form of RNN, is also investigated and compared with the FFNN in this study.

PLC

The linear correlation function $\rho_{X,Y}$ is a bivariate statistic that measures how two variables X and Y co-vary in a linear manner which also can be visually examined using a scatter plot (Makridakis *et al.* 2003). In time series data, when Y_t has a correlation with its past data Y_{t-h} (h being the number of time lag), the $\rho_{X,Y}$ is known as autocorrelation (Tsay 2005). On the other hand, if Y_t has a linear correlation with X_{t-h} of a different time series, the $\rho_{X,Y}$ is known as lag-h cross-correlation. The $\rho_{X,Y}$ can be estimated from the sample data of X and Y using Equation (1) below.

$$\rho_{X,Y} = \frac{\sum_{n=1}^{N} (x_n - \bar{x})(y_n - \bar{y})}{\sqrt{\sum_{n=1}^{N} (x_n - \bar{x})^2 \sum_{n=1}^{N} (y_n - \bar{y})^2}} \tag{1}$$

where x and y are sample values of X and Y, N is the sample size and $\bar{x}$ and $\bar{y}$ are the sample means.

However, the linear correlation function $\rho_{X,Y}$ alone is unable to detect the so-called 'bridge correlation'. For example, Y_t has auto-correlation with Y_{t-1}. This means that Y_{t-1} will automatically have correlation with Y_{t-2} due to the nature of time series data, and thus Y_{t-2} is also automatically correlated with Y_t (i.e., bridge correlation) through Y_{t-1}. The PLC is often used to detect the true linear relationship between Y_t and Y_{t-2} by removing the effect of Y_{t-1}. The partial correlation coefficients which represent the true level of linear relationship between Y_t and Y_{t-2} controlling on Y_{t-1} and Y_{t-3} (for example) can be found by computing the correlation (using Equation (1)) between two residuals from two MLR models as shown in Equations (2)–(5) (Tsay 2005):

$$Y_{t-2} = a_{2,0} + a_{2,1} Y_{t-1} + a_{2,3} Y_{t-3} + \varepsilon_t \tag{2}$$

$$r_{t-2} = Y_{t-2} - \left[a_{2,0} + a_{2,1} Y_{t-1} + a_{2,3} Y_{t-3} \right] \qquad (3)$$

$$Y_t = b_{2,0} + b_{2,1} Y_{t-2} + b_{2,3} Y_{t-3} + \varepsilon_t \qquad (4)$$

$$r_{t,2} = Y_t - \left[b_2 + b_{2,1} Y_{t-2} + b_{2,3} Y_{t-3} \right] \qquad (5)$$

where $a_{i,j}$ and $b_{i,j}$ are regression coefficients, r_{t-2} and $r_{t,2}$ are the residuals in MLR models for Y_{t-2} and Y_t, respectively, on the remaining variables, and ϵ_t is the error term. The same procedure can be applied when Y_t has cross-correlation with X_{t-2} through X_{t-1} of a different time series. The partial correlation coefficients are often statistically tested using the significant *t*-test with 95% confidence in the MLR analysis, in order to avoid selection of significant inputs by chance. In this study, a variable is considered statistically significant if its partial correlation coefficients pass the *t*-test and have a value larger than 0.1. The MLR analysis of the statistical software package IBM SPSS was used to implement this technique.

DWD

Preprocessing of input data has been recently used to enhance the performance of ANN models (Wu *et al.* 2009). The goal of data preprocessing is to identify important characteristics such as trend, oscillation, and noise in order to attain better predictability, because the time series data have time varying signals with static and dynamic components. Among the available data preprocessing techniques, wavelet decomposition has demonstrated good performance in several forecasting studies (Wu *et al.* 2009; Kisl & Cimen 2011; Wei *et al.* 2012). The underlying principle of this data preprocessing technique is to decompose the time series data of interest into several time series components, which are then used as model inputs to forecast the future value of the time series data.

Wavelet analysis is capable of revealing aspects of data such as trend and discontinuities (Catalao *et al.* 2011). It is performed based on wavelet transformation or decomposition that can decompose an original time series into a set of component time series. The sum of these component time series becomes the original time series. In general, wavelet decomposition is mathematically expressed as

follows (Cannas *et al.* 2006):

$$CWT(\tau,\ s,\ \psi) = \frac{1}{\sqrt{|s|}} \int\limits_{-\infty}^{+\infty} Y(t) * \psi \left(\frac{t-\tau}{s} \right) dt \qquad (6)$$

where $CWT(\tau,s,\psi)$ are the continuous transform coefficients of a continuous wavelet transformation of a time series data $Y(t)$; τ is the scale or frequency parameter; s is the position or shifting parameter; and ψ is the transforming function or wavelet. A wavelet is a waveform of effectively limited duration that has an average value of zero. In comparison, sine waves, which are the basis of Fourier analysis, do not have limited duration and extend from minus to plus infinity. While sinusoids are smooth and predictable, wavelets tend to be irregular and asymmetric. The $CWT(\tau,s,\psi)$ provides the local similarity between the time series data and the wavelet at a particular τ and s. However, the shortfall of the continuous wavelet transformation is that there are numerous pairs $(\tau,\ s)$ that can be used, thus resulting in redundant information and enormous computational effort in such a continuous domain (Kisl & Cimen 2011). Furthermore, many practical time series data (except electrical signals) are discrete in nature.

DWD was introduced to overcome the mentioned shortfall associated with continuous wavelet decomposition (Kisl & Cimen 2011). In the discrete domain, values of pair $(\tau,\ s)$ are limited to powers of two and are dependent on two integer parameters m and n (Kisl & Cimen 2011):

$$DWT(m,\ n,\ \psi) = 2^{-m/2} \sum_{t=0}^{N-1} Y(t) * \psi \left(\frac{t - n2^m}{2^m} \right) \qquad (7)$$

where $DWT(m,n,\psi)$ are the discrete transform coefficients of discrete transformation of time series data $Y(t)$ with length N; m indicates the level of decomposition ($m < M$ with $N = 2^M$) and the value of n is dependent on m. As a result of discrete wavelet transformation, a given time series data $Y(t)$ can be decomposed as:

$$Y(t) = A_1(t) + D_1(t) = A_2(t) + D_1(t) + D_2(t) = \ldots$$
$$= A_m(t) + D_1(t) + D_2(t) + \ldots + D_m(t) \qquad (8)$$

where $A_m(t)$ is called the approximation component time series and $D_m(t)$ is called the detail component time series.

From Equation (8), it can be seen that starting from the left side, $A_1(t)$ and $D_1(t)$ are obtained by discrete wavelet transformation of $Y(t)$ and then $A_2(t)$ and $D_2(t)$ are obtained by discrete wavelet transformation of $A_1(t)$, and so on. In theory, $A_m(t)$ represents the low frequency part of the signal, which is smooth and has a trend, while $D_m(t)$ represents the high frequency part, which has fluctuation or noise (Catalao *et al.* 2011). The performance of DWD is dependent on the choice of wavelet function ψ and decomposition levels. However, there are no guidelines for selecting the wavelet function and decomposition levels for a particular case. The 'Haar' wavelet function, which is the simplest wavelet and resembles a step function (Taylor 2009), is used for DWD in this study. Furthermore, three levels of decomposition, as found in several other studies (Choi *et al.* 2011; Kisl & Cimen 2011) are chosen for this study. Finally, an analysis of the forecasting performance of other wavelet functions and decomposition levels is also conducted in this study.

Performance indicators

Model performance in this study was evaluated using three performance indicators, the mean square error (MSE), Nash–Sutcliffe coefficient of efficiency (E), and the mean absolute percentage error (MAPE), as shown in Equations (9)–(11) (Carcano *et al.* 2008). The MSE quantifies the differences between observed and predicted values along the time series axis, and penalizes the large differences because of its square power. Minimizing of MSE is commonly used as the training objective in ANN models (Samarasinghe 2006). This allows the ANN models to better capture peak values of time series data. E captures the goodness of fit of the model of interest by comparing it with a naïve model in which the mean value is used as the predicted values. The value of E lies between one and minus infinity. A value of unity implies that the model exactly matches observations, zero implies that the model is no better than assuming the mean value, and a negative value indicates that the mean values of the observed time series is a better predictor than the model. However, the limitations of MSE and E are that they are unable to provide the practical assessment of difference between predicted and observed values. For example, MSE provides the same value of 4 between

observation of 1,000 and prediction of 1,002, and observation of 1 and prediction of 3. However, it is obvious that the former prediction is a much better prediction. The MAPE was therefore used in this study in addition to the well-known MSE and E, to provide a scale-free percentage difference between observed and predicted values (Coulibaly *et al.* 2001).

$$MSE = \frac{\sum_{i=1}^{n}(Q_{observed}(i) - Q_{predicted}(i))^2}{n} \qquad (9)$$

$$E = 1 - \frac{\sum_{i=1}^{n}(Q_{observed}(i) - Q_{predicted}(i))^2}{\sum_{i=1}^{n}(Q_{observed}(i) - \overline{Q_{observed}})^2} \qquad (10)$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Q_{observed}(i) - Q_{predicted}(i)}{Q_{observed}(i)}\right| \qquad (11)$$

where n is the number of data points.

CASE STUDIES

In this study, two real-world case study datasets have been used. Data #1 is based on a monthly time series of inflow (m³) to Lake Eildon in Victoria, Australia. The monthly inflow data used is from January 1891 to June 2007. Lake Eildon was constructed to protect farmers during drought years and to provide irrigation water for what was a vast uncultivated area in the northern plains of Victoria. This region has since become the largest area of irrigated farmland in Australia, which is known as the Goulburn-Murray Irrigation District. Construction of the lake took place in several development stages. The last development stage, completed in 1955, enlarged the storage to its present capacity of 3,334,158 ML. The enlargement plans also considered Victoria's electricity needs. The original 15 MW hydro-electric generation capacity was increased to 120 MW. The oldest turbines were renovated in 2001 to provide a generation capacity of 135 MW. Figure 3 shows the monthly time series plot of Data #1. The probable inputs for the one-month ahead forecast of the inflow Y_t are assumed to depend on the time lagged values of Y_t until Y_{t-15} (i.e., from Y_{t-1} to Y_{t-15}).
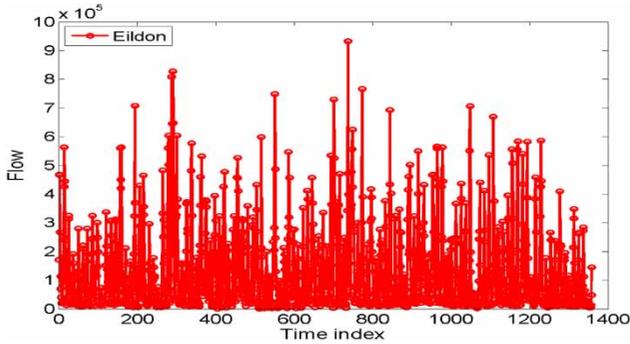
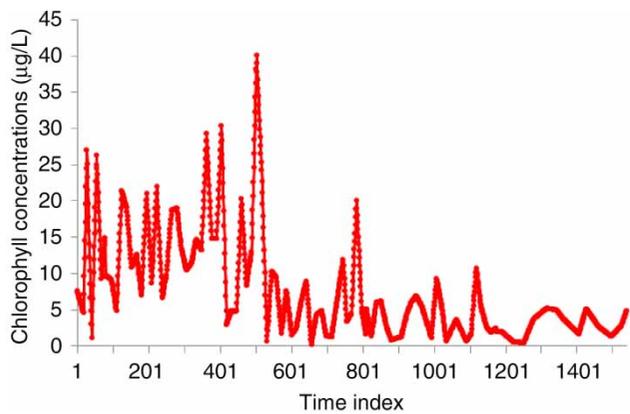**Figure 3** | Time series plot of monthly streamflow (m^3) (Data #1).

Figure 4 | Time series plot of daily chlorophyll concentrations (µg/L) (Data #2).

Data #2 is a time series of daily chlorophyll-a values (an indicator of algal biomass), which has been used for predicting chlorophyll concentrations (µg/L) in Tolo Harbor in Hong Kong (Muttil & Chau 2006, 2007). The 7-day ahead

chlorophyll concentration Y_t depends on its own time lagged variables (i.e., from Y_{t-7} to Y_{t-13}). The following eight variables were also assumed to be influential for the prediction of chlorophyll concentrations at Tolo Harbour: total inorganic nitrogen, $X1$ (mg/L); phosphorus, $X2$ (mg/L); dissolved oxygen, $X3$ (mg/L); secchi-disc depth, $X4$ (m); water temperature, $X5$ (°C); daily rainfall, $X6$ (mm); daily solar radiation, $X7$ (MJ/m^2), and daily average wind speed, $X8$ (m/s). Figure 4 shows the time series plot of daily chlorophyll concentrations used in Data #2.

RESULTS

Input preprocessing

Three types of model inputs (as summarized in Table 1) were used in this study for the two case studies to investigate the effectiveness of input selection and wavelet transformation on the autoregressive forecasting. The first type of input is called 'Original inputs' and these are created by selecting all probable inputs for forecasting. These Original inputs are not subjected to either of the 2 input preprocessing techniques. As shown in Table 1, for Data #1, a set of 15 time lagged inputs (i.e., Y_{t-1} to Y_{t-15}) is used to forecast the current value, Y_t. For Data #2, nine variables are used as input, which include the eight variables presented earlier along with chlorophyll itself. For each of these nine variables, a time lag of 7–13 days is used (giving a total of 63 inputs) for the 7-day ahead forecasting of chlorophyll concentrations.

Table 1 | Types of model inputs applied to the two datasets

Type of model inputs	Description	Data #1	Data #2
Original inputs	All probable inputs to forecast the current value Y_t	$Y_{t-1}, Y_{t-2}, …, Y_{t-15}$ (15 inputs)	$Y_{t-7}, Y_{t-8}, …, Y_{t-13} X_{(1)t-7}, X_{(1)t-8}, …, X_{(1)t-13}; X_{(2)t-7}, X_{(2)t-8}, …, X_{(2)t-13}…; X_{(8)t-7}, X_{(8)t-8}…, X_{(8)t-13}$ (63 inputs)
PLC-selected inputs	Inputs selected by PLC technique from the Original inputs	$Y_{t-1}, Y_{t-11}, Y_{t-13}, Y_{t-15}$ (4 inputs)	$Y_{t-7}, Y_{t-8}, Y_{t-13}$ (3 inputs)
PLC-Wavelet inputs	Inputs created by which wavelet transformation is applied on PLC inputs	$D1_{t-1}, D1_{t-11}, D1_{t-13}, D1_{t-15}; D2_{t-1}, D2_{t-11}, D2_{t-13}, D2_{t-15}; D3_{t-1}, D3_{t-11}, D3_{t-13}, D3_{t-15}; A3_{t-1}, A3_{t-11}, A3_{t-13}, A3_{t-15};$ (16 wavelet inputs)	$D1_{t-7}, D1_{t-8}, D1_{t-13}; D2_{t-7}, D2_{t-8}, D2_{t-13}; D3_{t-7}, D3_{t-8}, D3_{t-13}; A3_{t-7}, A3_{t-8}, A3_{t-13};$ (12 wavelet inputs)

The second type of input is called PLC-selected inputs, which are identified by applying the PLC technique on the Original inputs. As can be seen from Table 1, the PLC technique selected 4 inputs out of 15 probable inputs for Data #1 while only 3 inputs are selected out of 63 inputs for Data #2.

The third type of inputs are called the PLC-Wavelet inputs, which are created by applying the DWD technique on the PLC inputs. For each input variable (e.g., $Y_{t-1)}$, four components are created (e.g., $D1_{t-1}$, $D2_{t-1},...,D3_{t-1}$, $A3_{t-1}$) for three decomposition levels as per Equation (8). As a result, for Data #1, a total of 16 wavelet inputs are created while a total of 12 inputs are made for Data #2, as shown in Table 1. In this study, the PLC technique was implemented using the SPSS software package and the DWD technique was implemented using the Wavelet Toolbox of the Matlab software package.

Forecasting performance

The FFNN and LRNN were implemented using the Neural Network Toolbox of Matlab. Both FFNN and LRNN models used only one hidden layer for ease of comparison. The MLR models were also implemented using Matlab.

All datasets were divided into training (60%), validating (20%), and testing (20%) datasets. The testing dataset was the last 20% of the time series data. The training and validation datasets are randomly selected to have a better chance of representative learning data for FFNN and LRNN models. As indicated in the flow chart of the methodology (presented in Figure 1), the original inputs (made up of the training dataset) are used as input for the PLC to select the significant inputs, called the PLC-selected inputs. The PLC-selected inputs are then decomposed by the DWD to generate the PLC-Wavelet inputs. For the purpose of comparison, the original inputs (containing all probable inputs), the PLC-selected inputs and the PLC-Wavelet inputs are used as inputs for all three forecasting models. The training and validation datasets were used in the training process in FFNN and LRNN models in which the training dataset was used to provide 'learning knowledge', while the validation dataset was used to avoid over-fitting. The MLR model used the initial 80% of the dataset for calibrating without splitting into a validation dataset. The testing dataset was used only for assessing the forecasting performance of the MLR, FFNN, and LRNN models. Both two datasets were scaled between [−1, 1] and were then used for training and testing.

The training process for FFNN and LRNN models was carried out using the Levenberg–Marquardt algorithm (Samarasinghe 2006). The least square method (Tsay 2005) was used for the calibration of the MLR model. The single objective of minimizing MSE was used in the training process of all models. The remaining two performance indicators (i.e., E and MAPE) were then computed using the training (i.e., combination of training and validation) and testing outcomes. A suitable number of hidden neurons for FFNN and LRNN models was selected using trial and error. The results are presented in Table 2 only for the testing performances.

The difficulty of forecasting for both the datasets was demonstrated through the E values of the MLR model

Table 2 │ Performances of the MLR and ANN models for the testing dataset with the different inputs

Models	Inputs	Data #1				Data #2			
		hn	MSE	MAPE	E	hn	MSE	MAPE	E
MLR	Original inputs	–	0.052	119.3	0.567	–	0.030	26.2	0.769
	PLC-selected inputs	–	0.047	106.7	0.570	–	0.024	25.5	0.770
	PLC-Wavelet inputs	–	0.026	85.1	0.702	–	0.011	16.7	0.914
FFNN	Original inputs	12	0.034	86.3	0.589	2	0.020	22.3	0.873
	PLC-selected inputs	10	0.030	79.0	0.632	6	0.018	20.8	0.884
	PLC-Wavelet inputs	16	0.022	77.8	0.732	6	0.009	16.8	0.944
LRNN	Original inputs	12	0.035	104.2	0.580	2	0.021	24.1	0.867
	PLC-selected inputs	12	0.033	65.1	0.604	2	0.018	21.9	0.882
	PLC-Wavelet inputs	16	0.018	71.6	0.787	6	0.009	16.7	0.940

hn, number of hidden neurons.

using the Original inputs, as shown in Table 2. A higher level of difficulty was found with Data #1, with an E value of 0.567, as compared to that with Data #2 (with an E value of 0.769). The FFNN and LRNN produced slightly better performance in terms of MAPE and E than the MLR for Data #1 but their performances are much better than that of MLR for Data #2. The reason could be due to the non-linearity in Data #2 as compared to Data #1. Figures 5 and 6 show the time series plot for observed and predicted values of testing data of Data #1 and Data #2 by the different models. As can be seen in Figure 5, all models produced an underestimated forecast for Data #1 for most peak events. On the other hand, all models showed a better forecast for Data #2, as shown in Figure 6.

With regards to the effect of input selection, the forecasting performances of MLR, FFNN, and LRNN models showed slight improvement when the PLC-selected inputs were used. As can be seen from Table 2, the use of PLC-selected inputs slightly improved the performance of ANN models in all cases. Although the use of input selection provided a marginal improvement in forecasting performance, a secondary benefit is the reduction in computing time when only significant inputs are used.

On the other hand, the use of the Wavelet inputs produced significant improvement in the testing performances for all models (MLR, FFNN, and LRNN), as demonstrated by the three performance indicators in Table 2 for both the datasets. Furthermore, as can be seen by comparing the time series plots in Figures 7 and 8 (that used the PLC-Wavelet inputs) with those in Figures 5 and 6 (for the Original inputs), the capture of peak values improved with PLC-Wavelet inputs for both the datasets.
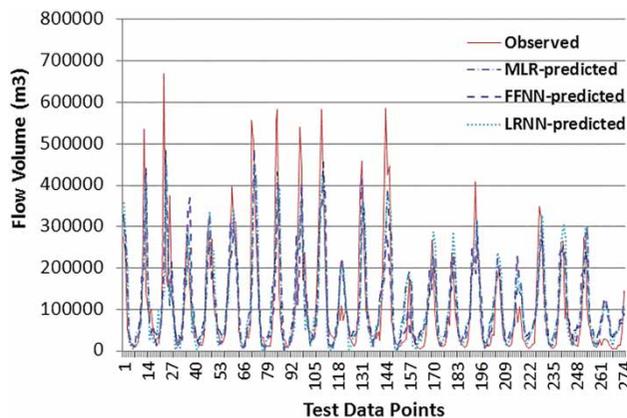

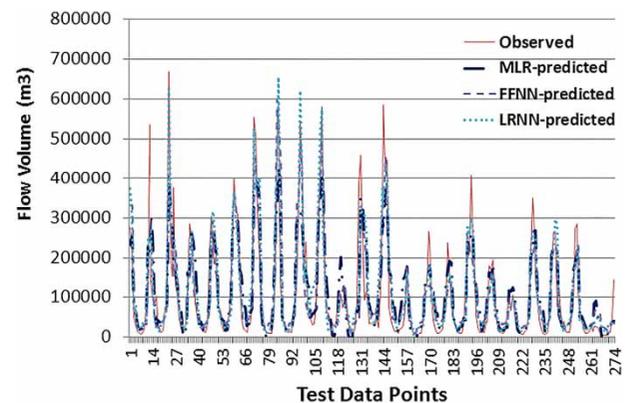**Figure 5** │ Time series plot of monthly flow prediction using Original inputs for Data #1.

Figure 7 │ Time series plot of monthly flow prediction using PLC-Wavelet inputs for Data #1.
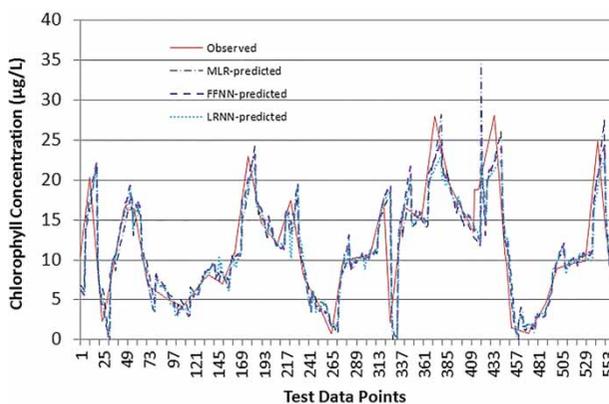

Figure 6 │ Time series plot of 1-week ahead prediction of chlorophyll concentrations using Original inputs for Data #2.
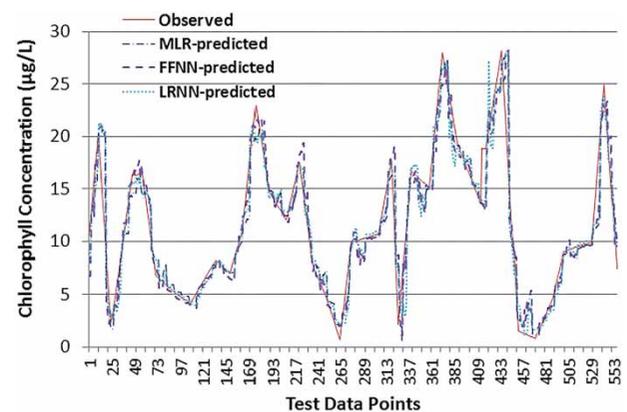

Figure 8 │ Time series plot of weekly prediction of chlorophyll concentration (µg/L) using PLC-Wavelet inputs for Data #2.

As far as the comparison of performances between FFNN and LRNN models is concerned, the LRNN model produced slightly better forecast performance than the FFNN model for Data #1 while both FFNN and LRNN produced similar forecasting performance for Data #2, which is demonstrated through three performance indicators presented in Table 2. On the other hand, both FFNN and LRNN outperformed the MLR for all three types of inputs. However, it is interesting to note that the performance gap between MLR and ANN models substantially reduced when PLC-Wavelet inputs were used.

Sensitivity of wavelet parameters

A sensitivity analysis was conducted to understand the effect of different levels of decomposition and the different wavelet functions used for producing the PLC-Wavelet inputs. In this study, three different decomposition levels are investigated and then four different wavelet functions are studied on Data #1 (i.e., monthly inflow values to Lake Eildon) using the FFNN model.

As mentioned earlier, for decomposition level 3 of an input variable, there are four probable component inputs (i.e., D_1, D_2, D_3, and A_3), which were used as model inputs in this study (as shown in Table 1). The probable inputs for decomposition levels 1 and 2 of an input variable are A_1 and D_1 (for level 1) and D_1, D_2, and A_2 (for level 2). For Data #1, the PLC technique selected four inputs and thus there will be a total of 8, 12, and 16 model inputs for the three different levels of decomposition, respectively. The sensitivity analyses for the three levels of decomposition for Data #1 are shown in Table 3. As can be seen from this table, level 2 of decomposition produced the best forecasting performance based on the MSE and E values, whereas level 3 is slightly better than level 2 decomposition based on the MAPE value.

Among the discrete wavelet functions available within the Matlab toolbox, the four commonly used wavelet functions, namely, the Haar wavelet, the Daubechies wavelet of order 3, Biorthogonal wavelet, and the Symlets wavelet of order 4 are used for the sensitivity analysis in this study. The FFNN model and level 2 of wavelet decomposition is used to compare the forecasting performance of the four different wavelet functions. The results presented in Table 4 show that the Daubechies wavelet of order 3 produced the best

Table 3 | Performance of the FFNN model on the testing dataset of Data #1 for the three levels of input decomposition

		Performance indicators for data #1		
PLC-Wavelet inputs	hn	MSE	MAPE	E
Level 1 (8 inputs)	12	0.021	81.4	0.743
Level 2 (12 inputs)	20	0.018	78.9	0.786
Level 3 (16 inputs)	16	0.022	77.8	0.732

hn, number of hidden neurons.

forecasting performance based on the MSE and E values, whereas the Symlets wavelet of order 4 produced the best forecasting performance based on the MAPE value.

DISCUSSION

As expected, the MLR model is not suitable for non-linear datasets, as demonstrated by its low performance on both the case study datasets. However, when the hybrid technique is used to preprocess the inputs, the MLR model has a significantly improved forecasting performance that makes it still useful for on-line forecasting. Therefore, sophisticated models like ANN models should be used to improve forecasting performance only when simple models like MLR produce poor performances. This observation further clarifies one of the common beliefs of the M-competitions presented in Crone *et al.* (2011) that 'sophisticated methods are not better than simpler methods'.

The LRNN model did not significantly outperform the FFNN model, while it has a high computational time compared to that required by the FFNN model. This is

Table 4 | Performance of the FFNN model on the testing dataset of Data #1 for different wavelet functions

		Performance indicators for Data #1		
Wavelet function	hn	MSE	MAPE	E
Haar wavelet	20	0.018	78.9	0.786
Daubechies wavelet of order 3	4	0.016	62.4	0.802
Biorthogonal wavelet	16	0.018	51.1	0.782
Symlets wavelet of order 4	12	0.020	38.4	0.755

hn, number of hidden neurons.

understandable since the LRNN model has feedback loops, which require additional computing resources.

It is observed that the use of PLC for input selection leads to a marginal improvement in the forecasting accuracy of both datasets and for all three models. On the other hand, the use of the DWD technique along with PLC inputs significantly improves the forecasting performance of all three models for both datasets. However, it still involves trial and error to identify the optimal DWD parameters. The sensitivity analysis of the different DWD parameters indicated that there are differences in forecasting performance for different levels of decomposition and with different wavelet functions. The level 2 decomposition and the Symlets wavelet might be the initial choice for these two DWD parameters.

The DWD technique has the ability to decompose the time series signal into low and high frequency. Low frequency is often viewed as representative for mean values and high frequency is for variation (Kisk 2010). Such decomposition is similar to de-trending (or separating of the non-stationary signal), which makes it easier for ANN models or even MLR models to simulate time series data with improved forecasting performances. On the other hand, conventional de-trending (e.g., differencing) is often carried out for traditional statistical models (e.g., ARMA models), which require non-stationary signals to accurately forecast time series data. However, a negative effect of conventional de-trending is the loss of variation and peak values, which are vital for hydroinformatics applications such as streamflow and rainfall forecasting. Therefore, conventional de-trending is often not used in forecasting studies using data-driven models (e.g., Wang *et al.* 2009; Kisk 2010).

Although the hybrid technique used in this study can improve the forecasting performance of both datasets, the forecasting accuracy is still not as high as expected. This indicates that the underlying mechanism of time series data is not yet fully captured and further development in the understanding of the underlying mechanism and mathematical modeling is necessary.

CONCLUSION

This study investigated a hybrid technique using PLC (for input selection) and DWD (for separating the non-stationary part) for improving the forecasting performance of regression-based models such as ANN models and MLR models. Based on the forecasting results of two real-world case study datasets, this study concluded that the hybrid technique can significantly enhance the forecasting performance of both the simple MLR model and ANN models. Furthermore, the LRNN model produced similar forecasting performance to that of the FFNN model. As far as input selection is concerned, the use of the PLC technique can significantly reduce the number of model inputs while producing a minor impact on the forecasting performance of ANN models. The input transformation using the DWD technique leads to significant improvement in the performance of the ANN models as well as the MLR model. This was demonstrated not only through improved performance indicators, but also through better capturing of peak values. Thus, this study provided useful understanding regarding the improvement in performance of ANN models when used in conjunction with significant input selection and wavelet-based data transformation, which could be useful for researchers and practitioners involved in autoregressive time series forecasting of real-world applications.

REFERENCES

Adamowski, J. & Sun, K. 2010 Development of a coupled wavelet transform and neural network method for flow forecasting of non-perennial rivers in semi-arid watersheds. *J. Hydrol.* **390**, 85–91.

Arunkumar, R. & Jothiprakash, V. 2013 Reservoir evaporation prediction using data-driven techniques. *J. Hydrol. Eng.* **18**, 40–49.

Babovic, V. & Keijzer, M. 2002 Rainfall runoff modeling based on genetic programming. *Nordic Hydrol.* **33**, 331–343.

Cannas, B., Fanni, A., See, L. & Sias, G. 2006 Data preprocessing for river flow forecasting using neural networks: wavelet transforms and data partitioning. *Phys. Chem. Earth Parts A/B/C* **31**, 1164–1171.

Cao, H., Recknagela, F. & Orr, P. T. 2013 Enhanced functionality of the redesigned hybrid evolutionary algorithm HEA demonstrated by predictive modelling of algal growth in the Wivenhoe Reservoir, Queensland (Australia). *Ecol. Modell.* **252**, 32–43.

Carcano, E. C., Bartolini, P., Muselli, M. & Piroddi, L. 2008 Jordan recurrent neural network versus IHACRES in modelling daily streamflows. *J. Hydrol.* **362**, 291–307.

Catalao, J. P. S., Pousinho, H. M. I. & Mendes, V. M. F. 2011 Short-term wind power forecasting in Portugal by neural networks and wavelet transform. *Renewable Energy* **36**, 1245–1251.

Cheng, C.-H. & Wei, L.-Y. 2009 Volatility model based on multi-stock index for TAIEX forecasting. *Expert Syst. Appl.* **36**, 6187–6191.

Choi, T.-M., Yu, Y. & Au, K.-F. 2011 A hybrid SARIMA wavelet transform method for sales forecasting. *Decision Support Syst.* **51**, 130–140.

Coulibaly, P., Anctil, F. & Bobee, B. 2001 Multivariate reservoir inflow forecasting using temporal neural networks. *J. Hydrol. Eng.* **6**, 367–376.

Crone, S. F., Hibon, M. & Nikolopoulos, K. 2011 Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *Int. J. Forecasting* **27**, 635–660.

Elshorbagy, A., Corzo, G., Srinivasulu, S. & Solomatine, D. P. 2010 Experimental investigation of the predictive capabilities of data driven modeling techniques in hydrology – Part 2: application. *Hydrol. Earth Syst. Sci.* **14** (10), 1943–1961.

Fernando, T. M. K. G., Maier, H. R. & Dandy, G. C. 2009 Selection of input variables for data driven models: an average shifted histogram partial mutual information estimator approach. *J. Hydrol.* **367**, 165–176.

Güldal, V. & Tongal, H. 2010 Comparison of recurrent neural network, adaptive neuro-fuzzy inference system and stochastic models in Egirdir lake level forecasting. *Water Resour. Manage.* **24**, 105–128.

Islam, A. S. 2010 Improving flood forecasting in Bangladesh using an artificial neural network. *J. Hydroinform.* **12**, 351–364.

Jung, N.-C., Popescu, I., Kelderman, P., Solomatine, D. P. & Price, R. K. 2010 Application of model trees and other machine learning techniques for algal growth prediction in Yongdam reservoir, Republic of Korea. *J. Hydroinform.* **12** (3), 262–274.

Kisk, O. 2010 Wavelet regression model for short-term stream flow forecasting. *J. Hydrol.* **389** (3–4), 344–353.

Kisl, O. & Cimen, M. 2011 A wavelet-support vector machine conjunction model for monthly streamflow forecasting. *J. Hydrol.* **399** (1–2), 132–140.

Liong, S. Y., Gautam, T. R., Khu, S. T., Babovic, V., Keijzer, M. & Muttil, N. 2002 Genetic programming: a new paradigm in rainfall runoff modeling. *J. Am. Water Resour. Assoc.* **38** (3), 705–718.

Liu, Z., Zhou, P., Chen, G. & Guo, L. 2014 Evaluating a coupled discrete wavelet transform and support vector regression for daily and monthly streamflow forecasting. *J. Hydrol.* **519**, 2822–2831.

Maier, H. R., Jain, A., Dandy, G. C. & Sudheer, K. P. 2010 Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions. *Environ. Modell. Software* **25**, 891–909.

Makridakis, S., Wheelwright, S. C. & Hyndman, R. J. 2003 *Forecasting Methods and Applications*. John Wiley and Sons (Asia), Chennai, India.

May, R. J., Maier, H. R., Dandy, G. C. & Fernando, T. M. K. G. 2008 Non-linear variable selection for artificial neural networks using partial mutual information. *Environ. Modell. Software* **23**, 1312–1326.

Minns, A. W. & Hall, M. J. 1996 Artificial neural networks as rainfall-runoff models. *Hydrol. Sci. J.* **41** (3), 399–417.

Muttil, N. & Chau, K. W. 2006 Neural network and genetic programming for modelling coastal algal blooms. *Int. J. Environ. Pollut.* **28**, 223–238.

Muttil, N. & Chau, K.-W. 2007 Machine-learning paradigms for selecting ecologically significant input factors. *Eng. Appl. Artif. Intell.* **20**, 735–744.

Parlos, A. G., Rais, O. T. & Atiya, A. F. 2000 Multi-step-ahead prediction using dynamic recurrent neural networks. *Neural Networks* **13**, 765–786.

Samarasinghe, S. 2006 *Neural Networks for Applied Sciences and Engineering*. Auerbach Publications, New York, USA.

Sun, Y., Doan, C. D., Dao, A. T., Liu, J. & Liomg, S.-Y. 2014 Improving numerical forecast accuracy with ensemble Kalman filter and chaos theory: case study on Ciliwung river model. *J. Hydrol.* **512**, 540–548.

Taylor, L. W. 2009 Using the Haar wavelet transform in the semiparametric specification of time series. *Econ. Modell.* **26**, 392–403.

Tiwari, M. K., Song, K. Y., Chatterjee, C. & Gupta, M. M. 2013 Improving reliability of river flow forecasting using neural networks, wavelets and self-organising maps. *J. Hydroinform.* **15**, 486–502.

Tran, H. D., Muttil, N. & Perara, B. J. C. 2015 Selection of significant input variables for time series forecasting. *Environ. Modell. Software* **64**, 156–163.

Tsay, R. S. 2005 *Analysis of Financial Time Series*. John Wiley and Sons, Hoboken, NJ, USA.

Valipour, M., Banihabib, M. E. & Behbahani, S. M. R. 2013 Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir. *J. Hydrol.* **476**, 433–441.

Vojinovic, Z., Kecman, V. & Seidel, R. 2001 A data mining approach to financial time series modelling and forecasting. *Int. J. Intell. Syst. Accounting Finance Manage.* **10**, 225–239.

Wang, W.-C., Chau, K.-W., Cheng, C.-T. & Qiu, L. 2009 A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *J. Hydrol.* **374** (3–4), 294–306.

Wei, S. K., Zuo, D. P. & Song, J. X. 2012 Improving prediction accuracy of river discharge time series using a Wavelet-NAR artificial neural network. *J. Hydroinform.* **14** (4), 974–991.

Wu, C. L., Chau, K. W. & Li, Y. S. 2009 Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques. *Water Resour. Res.* **45**, 432–455.

Yu, X., Liong, S. Y. & Babovic, V. 2004 EC-SVM approach for real-time hydrologic forecasting. *J. Hydroinform.* **6**, 209–223.