Exploring geometrical patterns in streamflow timeseries: utility for forecasting?

Ramesh S. V. Teegavarapu

ABSTRACT

Streamflow time series often provide valuable insights into the underlying physical processes that govern responses of any watershed to storm events. Patterns derived from time series based on repeated structures within these series can be beneficial for developing new or improved data-driven forecasting models. Data-driven models, artificial neural networks (ANN), are developed in the current study for streamflow prediction using input structures that are classified by geometrically similar patterns. A new modular and integrated ANN architecture that combines multiple ANN models, referred to as pattern-classified neural network (PCNN), is proposed, developed and investigated in this study. The PCNN relies on the development of several independent local models instead of one global data-driven prediction model. The PCNN models are evaluated for one step-ahead prediction of daily streamflows for Reed Creek and Little River, Virginia, and Elkhorn Creek, Kentucky in the United States. Results obtained from this study suggest that the use of these patterns has improved the performance of the neural networks in prediction. The improved performance of the PCNN models can be attributed to prior classification of data benefiting generalization abilities. PCNN model outputs can also provide an ensemble of forecasts that help quantify forecast uncertainty.

Ramesh S. V. Teegavarapu
Department of Civil, Environmental and Geomatics Engineering,
Florida Atlantic University,
Boca Raton, Florida 33431,
USA
E-mail: rteegava@fau.edu

Key words | artificial neural networks, geometrical shapes, pattern classification, pattern-classified neural network (PCNN), singular neural network (SNN), streamflow forecasting

INTRODUCTION

Streamflow forecasting assumes the greatest importance from water resources management and hydrologic design perspectives. A number of tools ranging from time-tested statistical modeling techniques (Salas 1993) to the black-box approach (e.g. neural network models; Govindaraju & Rao 2000) are now available for streamflow prediction. Dudley (2004) developed the regression equations for unregulated, rural rivers using several years of recorded streamflow with the help of ordinary least square regression techniques. Lakshmanan et al. (2009) have focused on data-driven models for predicting streamflow using rainfall-runoff observations over the heavily instrumented Ft. Cobb basin in western Oklahoma, USA, using a dataset of ten hydrologic

events. Cherkassky et al. (2006) introduced a generic theoretical framework for predictive learning and relate it to data-driven and learning applications in earth and environmental science. Kilinc et al. (2006) discuss three different approaches for developing data-driven models using historical data of the nearby streams. Inductive models are capable of extracting the functional relationships between predictand and predictors without utilizing information about the physical processes driving the variations in predictors. One such class of inductive models are the artificial neural networks (ANNs).

ANNs, as one of the data-driven forecasting tools and universal function approximators, have received considerable

attention in the last two decades for numerous applications in areas of water resources and hydrology (Govindaraju & Rao 2000; ASCE 2001a, 2001b; Rivera et al. 2002; Teegavarapu & Chandramouli 2005; Mutlu et al. 2008; Ferreira & Teegavarapu 2012) and continue to be the most used data-driven approaches for hydrological forecasting. Neural networks are data-driven approaches that do not explicitly consider the underlying physical processes. Successful ANN applications in hydrology and water resources fields relate to: (1) reservoir operation (Raman & Chandramouli 1996), (2) streamflow forecasting (Muttiah et al. 1997; Thirumalaih & Deo 1998; Zhang & Govindaraju 2000), (3) water quality modeling (Maier & Dandy 2000), (4) rainfall-runoff modeling (Minns & Hall 1996; Fernando & Jayawardena 1998) and (5) groundwater modeling (Ranjithan et al. 1993; Rogers & Dowla 1994; Sun et al. 2016). The study reported in this short communication is not another application of neural networks for hydrologic time series forecasting but an exercise conducted to evaluate the utility of geometric patterns in streamflow time series forecasting.

NEURAL NETWORK MODELS FOR HYDROLOGIC APPLICATIONS

A wealth of literature is available on the application of neural networks in the general area of water resources and hydrology. An exhaustive review of literature of all works relevant to the application of neural networks reported by ASCE (2001a, 2001b) is a valuable reference for appreciation of ANN applications in the water resources field. Applications of ANNs in hydrology are mainly related to streamflow (Govindaraju & Rao 2000) and rainfall prediction (e.g. French et al. 1992; Teegavarapu & Mujumdar 1996). A comprehensive review of neural network applications in the area of surface water quality modeling was provided by Maier & Dandy (2000). Their review also highlights the use of problem-specific information to improve the performance of the neural networks for a specific application. French et al. (1992) developed a three-layered NN to forecast rainfall intensity fields in space and time. Tang & Fishwick (1993) studied the neural network approach as a model for time series forecasting and compared it with

the Box–Jenkins model. Smith & Eli (1995) used ANN to model rainfall runoff process. Hjelmfelt & Wang (1993) developed a neural network based on the unit hydrograph theory. Using linear superposition, a composite runoff hydrograph for a watershed was developed by appropriate summation of unit hydrograph ordinates and rainfall excess values. Carrierre (1996) developed a virtual runoff hydrograph system that employed a recurrent back-propagation ANN to generate runoff hydrographs. Liong et al. (2000) used ANN for flow forecasting and reported a high degree of accuracy even for a 7-day lead model. Kasiviswanathan et al. (2013) reported the use of ANN-based rainfall runoff models using ensemble simulations. Kasiviswanathan et al. (2018) evaluated the input uncertainty for improvement in ANN-based hydrological models for prediction/forecasting. Tiwari et al. (2013) used ANNs, wavelets and self-organizing maps for improving the reliability of river flow forecasting. All these works based on ANNs benefited from the use of inputs that are highly relevant to the hydrological process being modeled.

One of the difficult and essential aspects of neural network modeling for hydrological or any time series forecasting is the identification and selection of inputs for a given output. Several types of input structures have been used in a variety of situations taking help from statistical analysis or trial and error procedures (Maier & Dandy 2000). Summary statistics (i.e. mean and standard deviation of the time series data), day of the year and lagged values of streamflow are used as inputs for one step-ahead prediction of streamflows (Teegavarapu 1998) using ANNs. These input structures in ANNs have either improved the learning times or the performance of the neural networks compared to traditional ARMA models. Jain & Indurthy (2003) present a comprehensive comparative analysis for the event-based modeling techniques for rainfall-runoff models using deterministic, statistical and ANN approaches. Rivera et al. (2002) report a multivariate non-linear model for streamflow series generation at various geographical sites using ANN and classical multivariate autoregressive models. Han et al. (2006) indicated that ANNs still have several limitations that prevent them from practical applications. Pre-processing of data can simplify the learning task for neural network in many cases (Cherkassky & Mulier 1994). Partitioning of problems into independent sub-problems can

also be adopted to develop modularity and to improve learning and generalization of neural networks (Reed & Marks 1999). See & Openshaw (1999) classified flow data into 16 clusters using self-organizing maps (SOM). They noted that the patterns derived from time series based on repeated structures within hydrological time series can be beneficial for developing new or improved data-driven forecasting models. Information, if available, to the partitioning approach can be used to design robust neural network models. The present study focuses on the use of geometric patterns identified in the hydrological time-series to improve the prediction capabilities of neural networks.

The two main objectives of this study are: (1) explore the possibility of using simple geometrical patterns derived from streamflow time series in neural network models to improve their prediction performance; (2) to evaluate the performance of a neural network architecture that uses pattern-classified inputs in comparison to a singular neural network (SNN). An SNN is a neural network model that does not partition the inputs into groups and uses multiple ANNs. The contents in the paper are organized as follows. A brief introduction about the existence of geometric patterns in hydrologic time series is first provided. Details of the development of pattern classified neural networks and their application to one-step ahead prediction of streamflows in three rivers are provided next. Finally, results and analysis along with the conclusions are presented.

GEOMETRIC PATTERNS IN STREAMFLOW TIME SERIES

The process of exploring patterns in data time series is similar to a pattern recognition task in which repeatable structures are searched in observed data. Data carry information either about the process generating them or the phenomenon they represent (Zimmermann 1993). Structure is defined as the manner in which the information is organized so that relationships between the variables in the process can be identified (Bezdek 1981). Hydrological time series are no exception. A number of structures can be found in the time series data if the data are searched via mathematical or simple visual classification techniques. In many earlier studies (e.g. Panu et al. 1978; Unny et al. 1981;

Khalil et al. 1998) that were aimed at prediction or filling of missing streamflows, streamflow, data were arranged in groups (e.g. wet and dry clusters) based on statistical properties. These studies are referred to as group-based techniques for estimation or filling of streamflows. An excellent treatise on the existence of patterns and organization in precipitation can be found in an article by Foufoula-Georgiou & Vuruptur (2000). A set of time series observations can be characterized by a number of patterns as shown in Figure 1. These patterns in general can be geometrical shapes, mostly trapezoidal patterns. The number of patterns of a specific type (triangular, rectangular or trapezoidal) depends on the length of the data (data window) used and nature of data. Two elements of scale triplet (spacing, extent, and support) concept discussed by Bloschl & Sivapalan (1995) are relevant in this context. The triplet concept applies to both space and time. In the current study, the spacing is 1 day, and the extent is the number of observations (streamflow measurements) that is considered as one entity. Extent will help define the number of patterns. Often, one feature may dominate an entire time series.

In the current study, two types of patterns are identified and defined. The patterns that define geometric shapes are referred to as (1) leading type and (2) expected type. Leading type patterns are those which are known in advance and lead to a specific time interval in which a forecast is required. Expected patterns are those that are not yet realized and follow the already realized leading patterns in time. Observing and deriving simple geometrical shapes (patterns) in streamflow time series is a straightforward procedure (Figure 1). The number of patterns (θ) obtained depends on the number of streamflow values (nv) considered as one entity (i.e. extent). The relationship between
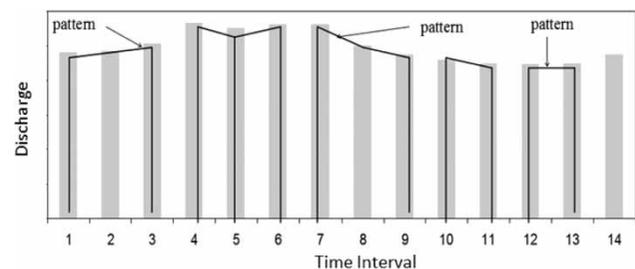

Figure 1 | Patterns in chronologically continues streamflow time series obtained by combining a set of streamflow values into groups of different lengths.

number of patterns and the values considered as one entity is given below:

$$\theta = 3^{nv-1} \tag{1}$$

Figure 2 shows the patterns (i.e. P1, P2 and P3) when two consecutive time series values are considered as one entity. Similarly, Figure 3 provides the patterns (i.e. P1, P2, P3, P4, P5, P6, P7, P8 and P9) when three consecutive values are considered.
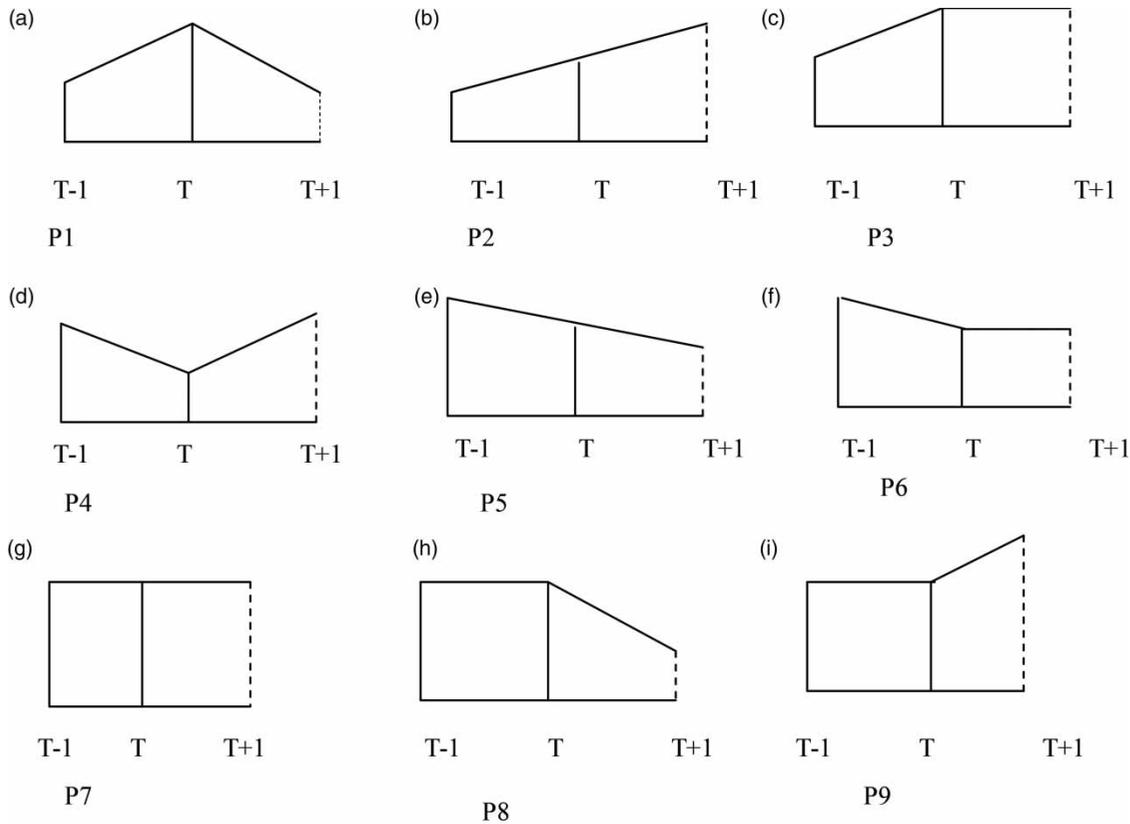
(a)

P1

(b)

P2

(c)

P3
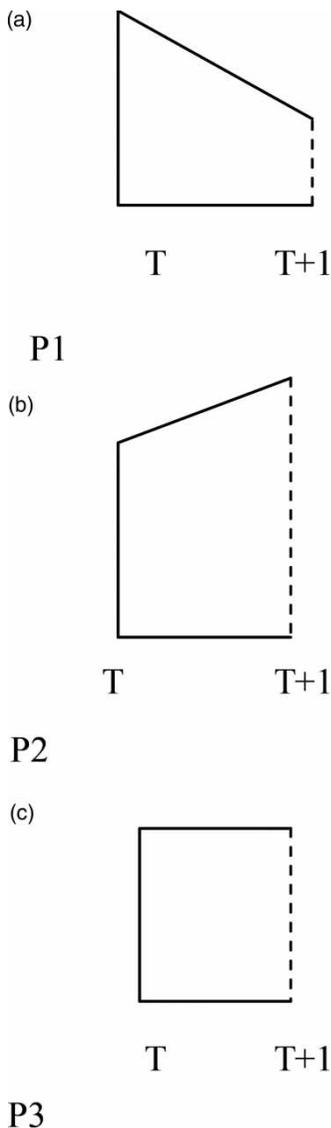
Figure 2 | Classification of streamflow time series into different geometric patterns for one-step-ahead prediction (one-lag). Prediction is required for time interval T + 1. Three patterns (P1, P2, P3) are possible.

PATTERN-CLASSIFIED NEURAL NETWORK

A new integrated neural network architecture combining a set of independent neural networks is developed in this study. The motivation to develop such a network is solely based on the modular neural networks developed by Zhang & Govindaraju (2000) for runoff prediction. The modular neural network architecture uses the algorithm developed by Jordan & Jacobs (1994). The input classification was done for different networks arbitrarily and the individual networks are referred to as expert networks. In the case of PCNN architecture, the inputs are initially classified into groups based on geometrical shapes identified in the time series. The number of independent ANN modules in the case of PCNN is equal to the number of patterns that are identified based on the lagged streamflow values. The input neurons are equal to the number of lags considered in the ANN model for prediction. The neural network models that do not use patterns are referred to singular neural networks (Zhang & Govindaraju 2000).

Network architecture

PCNN is an integrated architecture consisting of a set of neural network models that are trained independently using pre-identified and classified input patterns. The use of this structure allows for training of individual networks separately and also whenever new data becomes available. The number of independent networks will depend on the number of lags and patterns used. For example, if two lagged streamflow values are used for prediction purposes, then a total of nine (three in each category) networks need to be trained. Similarly, if only one lagged streamflow value is used as input, then three networks need to be trained. The architecture of PCNN is shown in Figure 4. The architecture of the singular network is shown in Figure 5. If only one hidden layer is used for the network, the network architecture of singular and PCCN can be explained by the following relationships (Equations (2)–(4)) that link outputs from hidden and output layer. The variables a_i and w_{ij} represent input (i.e. streamflow values) to the first layer node i and weight from input node i to hidden layer node j respectively in Equation (2). The

Figure 3 │ Classification of streamflow time series into different geometric patterns for one-step ahead prediction (with two lags). Prediction is required for time interval T + 1 based using information about possible patterns (P1, P2,… P9). Dotted lines refer to expected values to be forecasted.

number of input and hidden layer nodes are represented by p and j respectively:

$$\Phi_j = \sum_{i=1}^{p} a_i\, w_{ij} \quad \forall j \tag{2}$$

The variable Φ_j is the summation of products of inputs and weights from first layer to any hidden layer node j, φ is summation of products sigmoidal transformation of Φ_j and $w_{j,1}^o$ (weight attached to any hidden layer node j and output layer node) and f is the output (i.e. the predicted value of streamflow) of the neural network:

$$\varphi = \sum_{j=1}^{n} \left(\frac{1}{1+e^{-\Phi_j}} \right) w_{j,1}^o \quad \forall j \tag{3}$$

$$f = \frac{1}{1+e^{-\varphi}} \tag{4}$$

Once the networks are trained independently, the output (the forecasted streamflow value) can be obtained by integrating all the outputs from independent networks. The final output (f_{pcnn}) of the PCNN is given by Equation (5):

$$f_{pcnn} = \sum_{l=1}^{3^{nv-1}} f_l \beta_l \tag{5}$$

The variables f_1, f_2 … and so on are the outputs of the 3^{nv-1} neural networks for $nv = 2$, and β_1, β_2 …… $\beta_{3^{nv-1}}$ are the weights associated with each of the neural networks. The values of weights (relative frequencies) are calculated using the historical (training) data. The frequency value for any pattern is given by Equation (6):

$$\beta_n = \frac{\lambda_p}{\lambda_T} \tag{6}$$
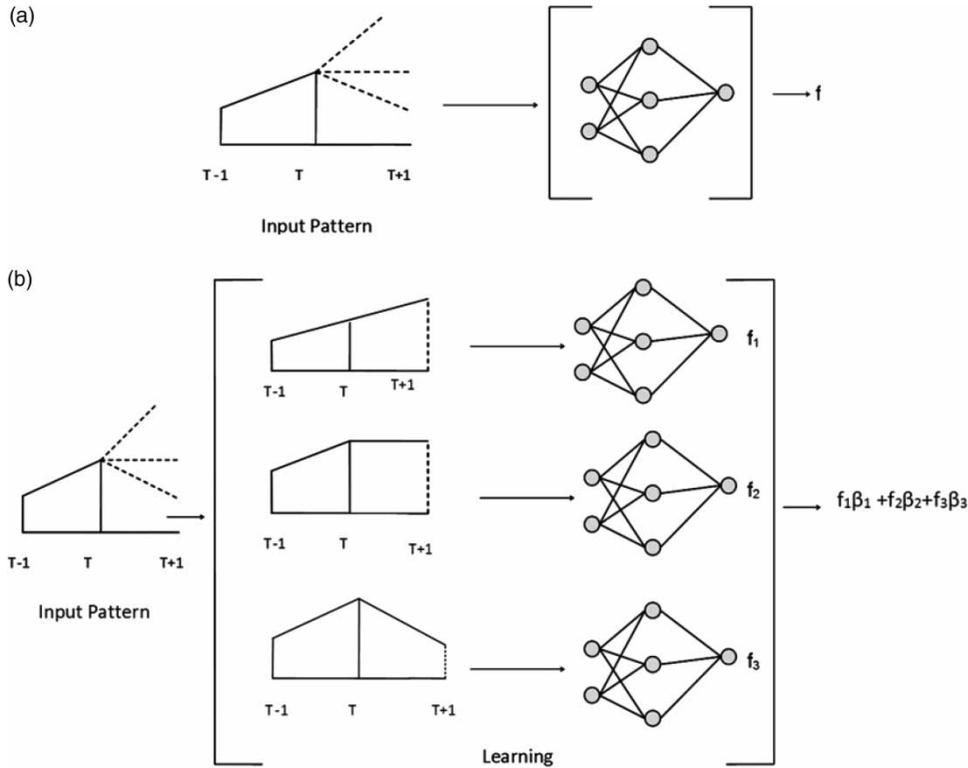
Figure 4 | Architectures for (a) singular network with one output (forecast) and (b) PCCN using pattern classified input structures with two lagged streamflow values with three weighted outputs (forecasts) resulting in one final value of forecast. Three networks are independently trained in PCNN.
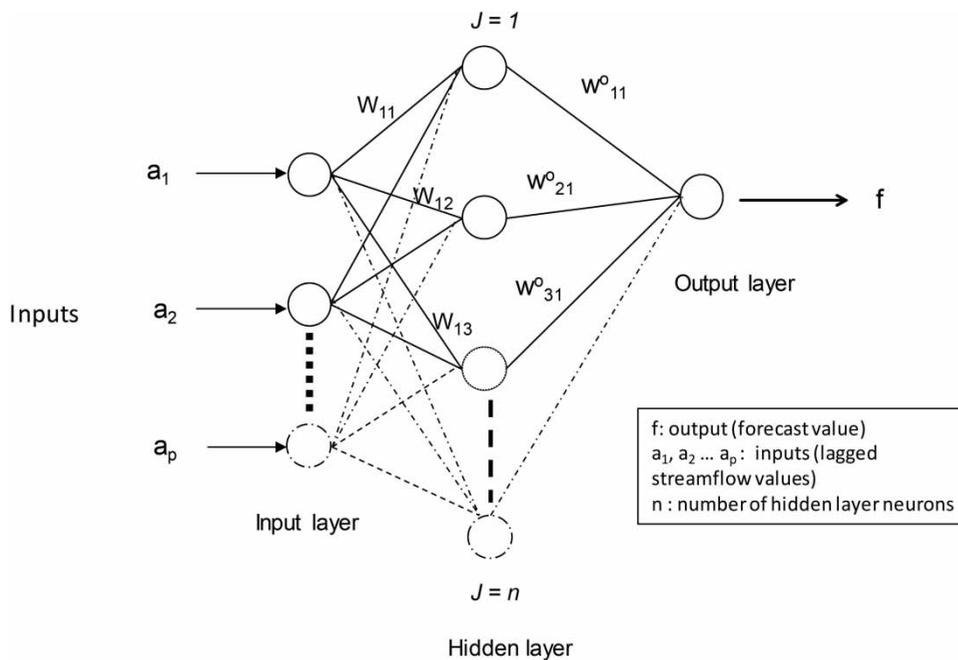
Figure 5 | Typical neural network architecture used for SNN with one input, hidden and output layers.

The variables λ_p and λ_T refer to number of patterns of specific type (e.g. P1 or P2 or P3 in case of one-lag shown in Figure 2) and the total number of leading patterns respectively. For nv greater than 2, the final output of PCNN is obtained by first identifying the sub-class to which a specific test pattern belongs and then using an appropriate set of trained ANNs for the patterns in that sub-class and finally obtaining an output (i.e. forecast) that is derived using weights obtained from Equation (6). For example, $nv = 3$ and there are three sub-classes with three patterns in each sub-class and they are: (1) P1, P2 and P3; (2) P4, P5 and P6 and (3) P7, P8 and P9. The output of PCNN is given by Equation (7):

$$f_{pcnn}^{sc} = \sum_{l=1}^{nv} f_l \beta_l \quad \forall sc \tag{7}$$

In the second step, output(forecast) is obtained using Equation (7) depending on the class to which the leading patterns belong. For any forecasting exercise, when $nv = 3$ a total of nine independent ANNs need to be trained. In this study, traditional feed-forward ANN (Freeman & Skapura 1991; Teegavarapu 1998) architecture has been adopted for singular and network used in PCNN. The architecture has three layers (input, hidden and output). The optimal number of nodes in the hidden layer can be determined by trial and error procedures. Details on the ANNs and the BP training algorithm can be found in Freeman & Skapura (1991). In the current study, each neural network model uses a variant of backpropagation algorithm, referred to as backpercolation (Jurik 1994) for training. Backpercolation is a learning algorithm which works in conjunction with the traditional backpropagation used for training of feedforward neural networks. In the backpercolation algorithm, the weights are not changed according to the error of the output layer as in backpropagation, but according to a unit error that is computed separately for each unit. This procedure effectively reduces the amount of training cycles needed.

Application of PCNN models

The PCNN models are applied to three rivers, Reed Creek and Little River in Virginia and South Elkhorn Creek in Kentucky. Availability of lengthy record of daily streamflows is the main reason for selecting two rivers for application and investigation of PCNN models. The locations (Reed Creek (latitude: 36°56′20″, longitude: 80°53′15″), Little River (latitude: 37°02′15″, longitude: 80°33′25″) and South Elkhorn Creek (latitude: 38°02′35″, longitude: 84°37′35″) of the streamflow gauging sites on these rivers in one of the major river basins in the US are shown in Figure 6. The major river basins are identified by 2-digit hydrologic unit code (HUC) numbers. The sites used for this study are located in the Ohio region identified by HUC number 05. Leading zeros are omitted in the representations of single digit HUC numbers in Figure 6. According to Köppen-Geiger climate classification (Kottek *et al.* 2006) based on temperature and precipitation observations for the period 1976–2000, the river basins for Reed Creek and Elkhorn are designated by three letter codes as *cfa*, and the Little river is *cfb*. The classifications *cfa* and *cfb* refer to warm temperate, fully humid, hot summer and warm summer respectively. The drainage areas for Reed Creek, South Elkhorn and Little Rivers are 668, 55 and 800 km^2 respectively.

Streamflow records are available for 1908–2000 for Reed Creek and 1928–2000 for Little River, Virginia. The Elkhorn Creek has a shorter observed record from 1950 to 2000. The average values of flows for Reed, Little, and Elkhorn streams are 7.50, 10.45 and 4.57 m^3/sec respectively. Similarly, for these sites, the standard deviations of flows at these sites are 10.13, 12.05 and 10.56 m^3/sec. The distributions of the streamflows at all three sites are positively skewed with large kurtosis. A Gamma distribution was found to be the best possible fit for all the three streamflow time series. The time series is systematically searched and the series is arranged in patterns comprising *leading* and *expected* types. Input structures, depending on the number of lags, are then prepared for training the individual neural network models. The relative frequencies required to calculate the weights in the PCNN network are calculated based on the training data. One-third of the available historical data is used for testing, and the rest is used for training. The leading patterns in the case of PCNN are obtained from the validation data (or test data) to obtain the forecasts using the appropriately trained networks. The performance of the PCNN network
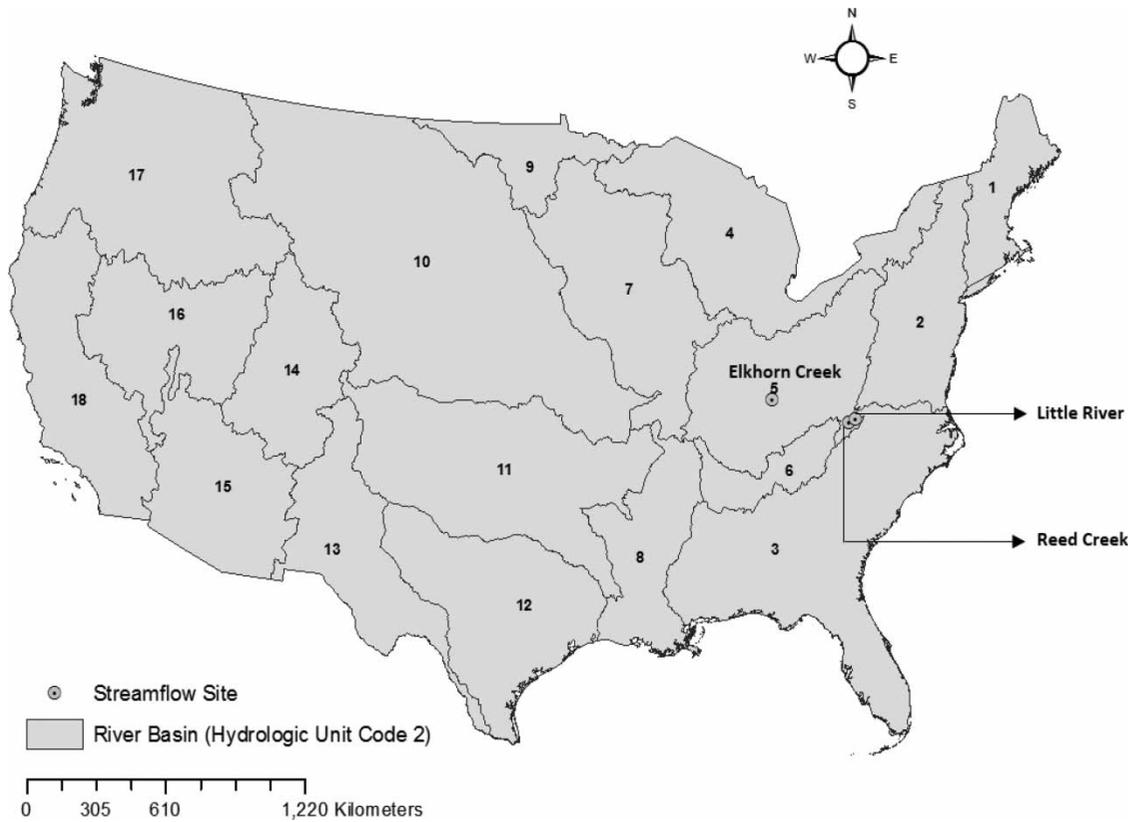
Figure 6 | Locations of three sites selected in one of the major river basins for evaluation of proposed methodology in this study. River basins numbered using HUC with leading zero omitted for single-digit codes.

is compared with that of an SNN which is also trained using the backpercolation training algorithm.

RESULTS AND ANALYSIS

PCNN models are applied for one step-ahead prediction of streamflows at Little River, Reed Creek, Virginia, and Elkhorn Creek, Kentucky. The predictions from PCNN are compared with those of SNN in each case. Two error measures namely, root mean squared error (RMSE) and mean relative error (MRE) and one performance measure, Nash–Sutcliffe efficiency criterion (NSEC) (Nash & Sutcliffe 1970) are used to assess the performance of the models. Details of the neural network architectures for different patterns are given in Tables 1 and 2. To facilitate the comparison of performances of singular NN and PCNN, the maximum number of hidden layer neurons are constrained to six in this study. The software used for

Table 1 | Neural network model architectures for different patterns for Little River and Reed Creek, Virginia and Elkhorn Creek, Kentucky

Type of geometrical pattern

Stream	P1	P2	P3
Reed Creek	1-3-1	1-2-1	1-6-1
Little River	1-5-1	1-1-1	1-6-1
Elkhorn Creek	1-5-1	1-1-1	1-2-1

1-3-1 architecture indicates one input-layer neuron, three hidden-layers and one output-layer neuron.

training ANN models iteratively choses the number of neurons until best performance on training data is achieved. This restriction may affect the performances of both SNN and PCNN.

Interesting observations can be made from the patterns obtained from the streamflow time series. The two streams, Reed Creek and Little River, which are evaluated in the current study are cross-correlated. The cross-correlation coefficient between these two streamflow time series was

Table 2 | Neural network model architectures for different patterns and streams

Type of geometrical pattern

| Stream	P1	P2	P3	P4	P5	P6	P7	P8	P9
Reed Creek	2-3-1	2-2-1	2-6-1	2-3-1	2-3-1	2-1-1	2-6-1	2-6-1	2-4-1
Little River	2-3-1	2-2- 1	2-6-1	2-6-1	2-6-1	2-1-1	2-6-1	2-1-1	2-2-1

2-3-1 architecture indicates two input-layer neurons, three hidden-layers and one output-layer neuron.

found to be equal to 0.76. This suggests that the flows in one stream can be estimated using the other. The percentages of patterns, P1, P2 and P3 for Reed Creek and Little River are 61.5, 29.5 and 9, and 61.3, 32.7 and 6 respectively. The pattern percentages are shown in Figures 7 and 8 respectively. These percentages are obtained based on training data. It is interesting to note that the percentages for patterns P1, P2 and P3 are similar. However, the existence of strong cross-correlation may indicate similarity in pattern percentages and not necessarily suggest the opposite. Similar conclusions can be made with the help of patterns, P1–P9.

It can be observed from Figures 7(a) and 8(a) that the percentage values for pattern P3 in case of Reed Creek and Little River are much lower than the percentages for the remaining patterns. Similarly, it is evident from Figures 7(b) and 8(b) that the percentages for patterns P6, P7, P8 and P9 in the case of time series in which two consecutive values are considered as one entity to develop patterns are much lower than those for the remaining patterns. One main reason is that consecutive patterns with equal values of streamflows are rare in nature unless the flow is controlled by a hydraulic structure. Lowest training times and mean squared error (MSE) values were realized for the ANNs that are employed for training the pattern P3. Details of the network architectures for different patterns are given in Tables 1 and 2. To make the comparison
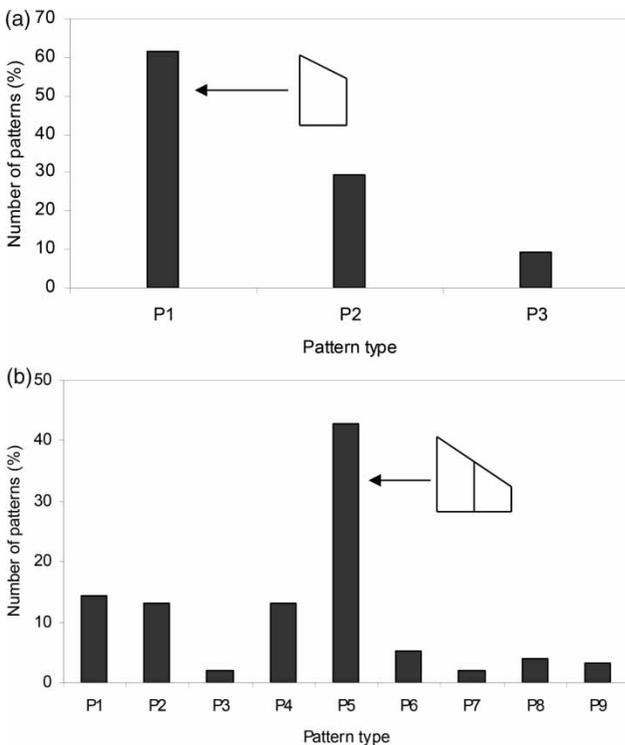
Figure 7 | Percentage of patterns identified in the time series for (a) input structure (number of lags = 1) and (b) input structure (number of lags = 2) for Reed Creek, Virginia, and dominant patterns.
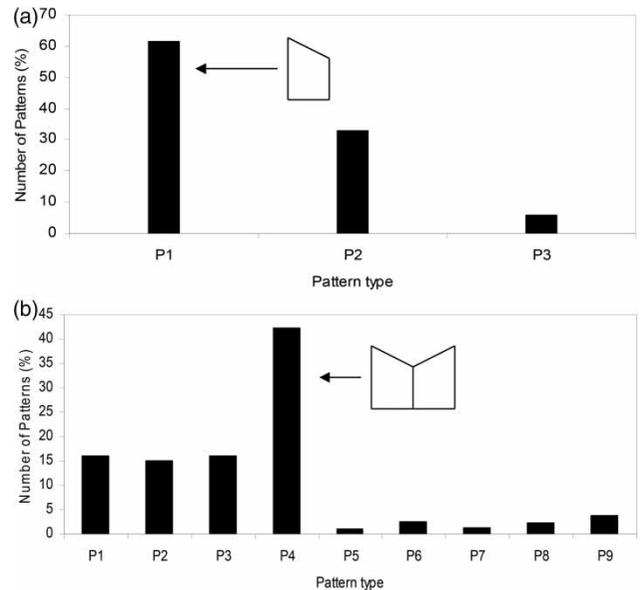
Figure 8 | Percentage of patterns identified in the time series for (a) input structure (number of lags = 1) and (b) input structure (number of lags = 2) for Little River, Virginia, and dominant patterns.

of SNN and PCNN possible, the maximum number of hidden layers in each network is constrained to a maximum of six in this study. The software used for training ANN models iteratively choses the number of neurons via a number of initial test simulations and a specific number is selected when best performance on the training data is achieved. The best networks in both the cases (lag = 1, lag = 2) used six neurons, which is the maximum number used in the study.

The performance measures, RMSE and MRE values, given in Table 3 indicate that the PCNN performed better than singular networks. The training is conducted in a controlled environment with all the parameters of training being the same for all the networks. Since the performance of ANN models can be improved by a number of trials using different parameters, by using a different configuration of networks, a standard for comparison of singular or PCNN models will not be possible. Therefore, all the training parameters, including the number of hidden layers and the number of neurons used, are fixed for all the networks. This strict constraint on the configuration may limit the network performance in the prediction process. Also, an unbiased comparison of prediction capabilities of singular and PCNN models requires this restriction. It is important to note that the error measures calculated are average values for the testing periods of 27, 23 and 16 years for Reed Creek, Little River, and

Elkhorn River, respectively. A 1% difference in RMSE value suggests on average one model is either overpredicting or underpredicting streamflow values by 1%. Any improvement, however minute it is, in the streamflow prediction can be considered significant as the predicted value is important from short term flood forecasting and protection perspectives. The error measures point primarily towards the conclusion that PCNN is a better network model compared to a SNN based on the limited experiments conducted in this study.

The existence of strong autocorrelation at different lags (up to two lags) is evident from Figure 9(a)–9(c) for three rivers and may suggest that the use of more than one lagged value of streamflow can be beneficial for improvement of PCNN's performance in prediction. Improvement in the performance of the models is also noticed when more than one lagged streamflow value is used in the ANN model. In some cases, the existence of the correlation between one- and two-lagged streamflow values, and thus the inter-correlation, may prevent the model based on two lags from being significantly better than a model based on a single lag. McCuen & Snyder (1986) provide a similar explanation for marginal performance improvement when lagged values used for modeling are increased in traditional ARMA models. The error measures point primarily towards the conclusion that PCNN is a better network model compared to a singular network. The PCNN models for the three rivers have consistently performed better than singular network models for both lag 1 and lag 2 cases. The predictions are consistent with the architecture of PCNN as shown in Figure 10. The results are shown to compare streamflow predictions from three networks within PCNN. The objective is mainly to show the predictions for different patterns. Depending on the pattern type and associated trained ANN model, PCNN predicts equal, higher or lower values of streamflow than the leading value. Calculation of weights (β_1, β_2 ...... $\beta_{3^{nv-1}}$) in the PCNN model is a crucial issue that can affect the value of final output. Mathematical programming (optimization) formulation can be used to obtain weights by selecting a portion of the testing data. The approach described by a nonlinear mathematical programming formulation is given by Equations (8)–(10). Different notation is used for weights (ω_1, ω_2 ... $\omega_{3^{nv-1}}$) calculated using an optimization formulation and those weights

Table 3 | Performance evaluations of SNN and pattern classified neural network (PCNN)

Stream	Network	Patterns/Lags	Performance error measure		
			RMSE (m³/sec)	MRE	NSEC
Little River	SNN	Lag = 1	9.55	0.134	0.35
		Lag = 2	9.45	0.126	0.37
	PCNN	Patterns = 3	9.35	0.131	0.38
		Patterns = 9	9.24	0.129	0.41
Reed Creek	SNN	Lag = 1	6.88	0.121	0.57
		Lag = 2	6.82	0.112	0.58
	PCNN	Patterns = 3	6.57	0.178	0.61
		Patterns = 9	6.37	0.123	0.63
Elkhorn Creek	SNN	Lag = 1	1.64	0.321	0.38
	PCNN	Patterns = 3	1.53	0.309	0.42

RMSE, root mean squared error; MRE, mean relative error; NSEC, Nash–Sutcliffe efficiency criterion.
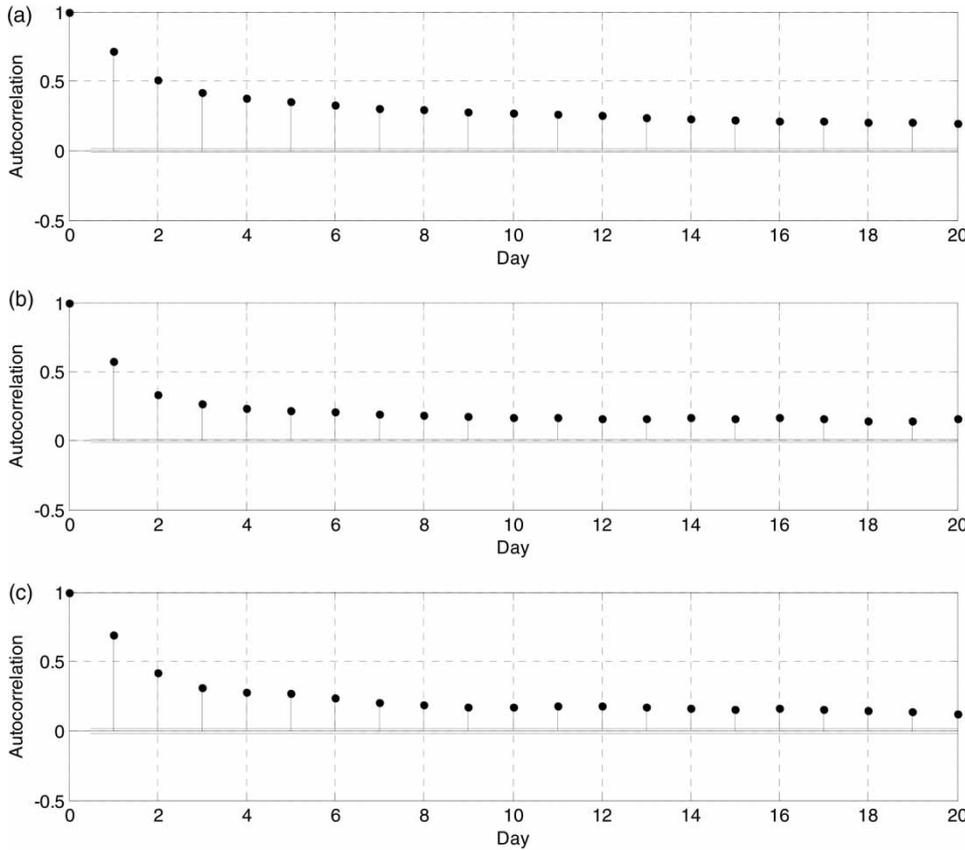
Figure 9 | Autocorrelograms for streamflow data of three rivers: (a) Reed Creek, (b) Little River, and (c) Elkhorn Creek. The horizontal lines close to zero refer bounds for Gaussian white noise process.

calculated by Equation (6) to avoid any confusion. Also, it is important to note that a part of the testing data is used to obtain the weights as observed values along with estimated ones and are used in the formulation as opposed to the usage of training data in the original approach presented.

Minimize:

$$\sum_{k=1}^{n} \left(\left(\sum_{l=1}^{N} \omega_l \theta_l^k \right) - \theta_m^k \right)^2 \tag{8}$$

Subject to:

$$\sum_{l=1}^{N} \omega_l = 1 \tag{9}$$

$$\omega_l \geq 0 \tag{10}$$

The variables θ_m^k is the observed streamflow value, θ_l^k is the estimated value from a singular network, l, within the PCNN, N is the number of patterns and n is the number of days. The objective function (Equation (8)) is used to minimize the difference between estimated and observed streamflow value over a period of n days. Constraint defined by Equation (10) will ensure that all the weights are positive and avoid negative forecast values. The formulation can be solved using any nonlinear optimization solver. In the current study, a genetic algorithm (Michalewicz 1996) as a solver is used to obtain optimal weights. One experiment is conducted to obtain weights using 10% of the testing data for the Little River. The weights obtained based on Equation (6) with a three-pattern PCNN model for $nv = 2$, and mathematical programming formulation described in this section using genetic algorithms (GA) are 0.06, 0.65, 0.31 and 0.2, 0.89 and 0.1, respectively. The performance values based on RMSE and MRE are 8.6 and 7.9 m^3/sec
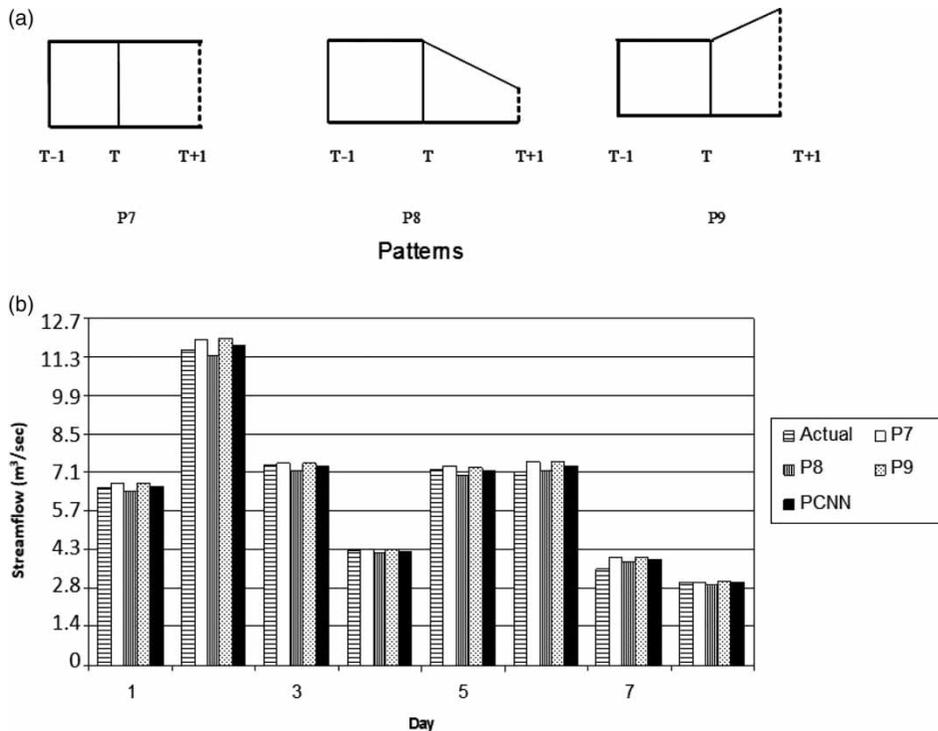
Figure 10 | Observed and predicted values of streamflows using three individual networks for patterns P7, P8, P9 and forecast with PCNN. Actual value refers to observed streamflow magnitude.

and 0.21 and 0.19 for a network with original and GA-based weights respectively. The network performance using these two measures improved approximately by 10% using weights from GA-based approach. NSEC values before and after optimization for this particular trial are 0.35 and 0.40 respectively. Low values of efficiency criterion are obtained for both the networks due to restrictions placed on the networks in terms of hidden layer neurons. Availability of conceptually acceptable weighting schemes other than those proposed in this study will help the practicing hydrologist in generating several forecast scenarios. The weights used in the individual neural network models are the values of relative frequencies or optimal weights and are dependent on the data used for developing the models.

GENERAL REMARKS

Lumped or distributed physically-based hydrologic simulation models (e.g. Chen *et al.* 2016) can be better alternatives to purely data-driven univariate time series forecasting models when exhaustive information about the physical parameters that characterize the hydrological processes is available. Distributed hydrological models still remain conceptually superior to data-driven modeling approaches as evident from recent applications for flood forecasting (Chen *et al.* 2011, 2017). Improvements in these modeling approaches are achieved by using parameter optimization (Chen *et al.* 2016) for improved performances in streamflow simulation and prediction. However, in many instances, the prohibitive cost of obtaining field data to derive knowledge about parameters, the time consumed for calibration and validation, model structure and parameter uncertainty, limit the usage of these models for many practical applications. In many past studies, traditional autoregressive models and ANNs have proven to be viable alternatives to physically-based models. Development of improved data-driven models including ANNs to achieve superior forecasting approaches can be seen as a beneficial exercise. This study is an attempt towards achieving improvements in ANN models for streamflow forecasting.

Hydrologic persistence, the tendency for high flows to follow high flows and for low flows to follow low flows, is assumed to be valid in many cases of hydrological time series (Matalas 1997). This persistence quantified using auto-correlation is often evident in hydrological time series and is the major contributing factor for the success of traditional univariate autoregressive moving average (ARMA) models used for forecasting. The local independent neural networks adopted in the case of PCNN are expected to capture the temporal variability of streamflows better compared to one global model developed for the entire time series. The results from the current study also confirm the widely acknowledged limitations of univariate data-driven models that use only those inputs of the lagged variable time series values and do not consider the underlying physical processes of hydrological phenomena.

The availability of previous discharge information as an input is crucial to the success of autoregressive models used for streamflow forecasting. However, the lack of causal inputs (e.g. rainfall, soil moisture state, etc.) relating to watershed processes and other hydrological parameters that affect runoff generation within any of these data-driven models also makes them inefficient short-term fore-casting tools. The division of time series data into patterns and development of separate ANNs in the current study contributed to the improvement of function approximation abilities of PCNN and also in modeling the rising and falling limbs of the individual storm hydrographs and other variations in between. The individual neural network models provide predictions based on the leading patterns, and the associated weights quantify the probabilities of occurrences relevant to predicted values.

The weights used in the individual neural network models that are part of PCNN are the values of relative frequencies of patterns derived based on the training data. It is obvious that in real-world situations testing data will not be available and therefore lengthy training data is essential to obtain these frequencies or weights and confidence in the forecasts. It should also be noted that the numeric values of the weights will depend on the training data length. In many situations, data collected over a number of years will implicitly contain the information about the land use and climate changes. In such instances, many traditional auto-regressive moving average approaches fail to provide predictions when the assumption of stationarity property is not valid. Data-driven approaches such as neural networks can capture the changes in the watershed over several years by utilizing different lengths of training data by using moving constant or expanding temporal windows. If recent data are added to the training set, neural networks can learn the most recent physical changes that have occurred in the watershed and this re-learning is reflected in the new weights attached to the neurons and also in the frequency values (weights) in the case of PCNN. Improved one step-ahead prediction can be obtained using these updated weights.

Although the computational time depends on the parameters and convergence criteria adopted in the training of ANNs, an approximate comparison of computation times can be made for singular and PCCN networks. For a given prediction problem, the computation time for PCNN is 3^{nv-1} times the time required to train the singular network for the value of $nv = 2$. For nv greater than 2, the time required for training will depend on the type of sub-class of patterns existing in the time series. In such cases, the computational time for PCNN is nv times that of SNN. Again, if all the 3^{nv-1} patterns exist in the time series, then the computational time for PCNN would be 3^{nv-1} times that of a singular network. The computational times will also depend on the number of neurons used in the hidden layer. The proposed PCNN may sometimes lead to a situation where there insufficient instances under a certain pattern. There are no strict guidelines to determine number of lags to be considered for developing PCNN models. Autocorrelograms can help in deciding the number of lagged streamflow values to be considered. The computational burden of training multiple networks will substantially increase once the number of lags considered is more than four.

The study in no way reflects an exhaustive analysis to conclude that PCNNs are better than singular neural networks for prediction problems when information relating to causal inputs is lacking, or its relevancy to a predicted variable is debatable in purely data-driven models. The study does not claim comprehensive evaluation of the use of geometric patterns in developing classifications among streamflow time series and improving prediction using ANNs. Extensions of the present study may include the

use of geometrical patterns formed by different lags and patterns along with total discharge values obtained by using the areas of the closed polygons formed by the geometrical patterns as inputs in PCNN models. The incorporation of geometrical properties into the hydrological modelling process is considered a viable concept that can offer substantial opportunities in adaptively modifying ANNs in a quest to develop superior operational forecasting solutions.

CONCLUSIONS

Utility of geometric patterns derived from streamflow time series for one-step ahead prediction is investigated in this study. ANN models are developed using inputs that are classified into patterns of similar geometrical shapes. A new integrated neural network architecture is proposed, investigated and experimented in this study. The network architecture is referred to as PCNN. The training and testing of models for streamflow prediction suggest that input structures can influence the performance of neural network models of streamflow time series. PCNN architecture implements the idea of developing multiple local models as opposed to one global model for streamflow prediction. Results from this study suggest that classification of inputs with the help of simple geometrical patterns can help to develop modular neural networks that can improve forecasting. An ensemble of forecasts that are possible using weights derived from information from the past data using PCNN can help quantify uncertainty in the forecast.

REFERENCES

ASCE 2001a Task Committee on Artificial Neural Networks in Hydrology, Artificial Neural Networks in Hydrology. I. Preliminary concepts. *J. Hydrol. Eng. ASCE* **5** (2), 115–123.

ASCE 2001b Task Committee on Artificial Neural Networks in Hydrology, Artificial Neural Networks in Hydrology. II Hydrologic applications. *J. Hydrol. Eng. ASCE* **5** (2), 124–137.

Bezdek, J. C. 1981 *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer, New York, London.

Bloschl, G. & Sivapalan, M. 1995 Scale issues in hydrologic modeling: a review. *Hydrological Processes* **9**, 251–290.

Carriere, P., Mohaghegh, S. & Gaskari, R. 1996 Performance of a virtual runoff hydrograph system. *J. Water Resour. Plan. Manage. ASCE* **122** (6), 421–427.

Chen, Y., Ren, Q. W., Huang, F. H., Xu, H. J. & Cluckie, I. 2011 Liuxihe model and its modeling to river basin flood. *J. Hydrol. Eng.* **16**, 33–50.

Chen, Y., Li, J. & Xu, H. 2016 Improving flood forecasting capability of physically based distributed hydrological model by parameter optimization. *Hydrol. Earth Syst. Sci.* **20**, 375–392.

Chen, Y., Li, J. & Wang, H. 2017 Large watershed flood forecasting with high resolution distributed hydrological model. *Hydrol. Earth Syst. Sci.* **21**, 735–749.

Cherkassky, V. & Mulier, F. 1994 Statistical and neural network techniques for nonparametric regression. In: *Selecting Models From Data, Lecture Notes in Statistics*, vol. 89. (P. Cheeseman & R. W. Oldford, eds). Springer Verlag, New York, pp. 383–392.

Cherkassky, V., Krasnopolsky, V., Solomatine, D. P. & Valdes, J. 2006 Computational intelligence in earth sciences and environmental applications: issues and challenges. *Neural Netw.* **19**, 113–121.

Dudley, R. W. 2004 *Estimating Monthly, Annual, and Low 7-day, 10-Year Streamflows for Ungaged Rivers in Maine, U.S.* Dept. of the Interior, U.S. Geological Survey, Information Services Denver, Colorado, pp. 1–30.

Fernando, D. A. K. & Jayawardena, A. W. 1998 Runoff forecasting using RBF networks with OLS algorithm. *J. Hydrol. Eng. ASCE* **3** (3), 203–209.

Ferreira, A. & Teegavarapu, R. S. V. 2012 Optimal and adaptive operation of a hydropower system with unit commitment and water quality constraints. *Water Resour. Manage.* **26** (3), 707–732.

Foufoula-Georgiou, E. & Vuruptur, V. 2000 Patterns and organisation in precipitation. In: *Spatial Patterns in Catchment Hydrology: Observations and Modeling* (R. Grayson & G. Bloschl, eds). Cambridge University Press, Cambridge, UK.

Freeman, J. M. & Skapura, D. M. 1991 *Neural Networks: Algorithms, Applications, and Programming Techniques*. Addison-Wesley, New Jersey.

French, M. N., Krajewski, W. F. & Cuykendal, R. R. 1992 Rainfall forecasting in space and time using a neural network. *J. Hydrol.* **137**, 1–37.

Govindaraju, R. S. & Rao, A. S. (eds). 2000 *Artificial Neural Networks in Hydrology*. Water Science and Technology Library, Springer, Dordrecht, The Netherlands.

Han, D., Kwong, T. & Li, S. 2006 Uncertainties in real-time flood forecasting with neural networks. *Hydrol. Process.* **21** (2), 223–228.

Hjelmfelt, A. T. & Wang, M. 1993 Predicting runoff using artificial neural networks. In: *Proceedings of the International Conference on Hydrology and Water Resources, New Delhi, India* (V. P. Singh & B. Kumar, eds). Springer, Netherlands.

Jain, A. & Indurthy, S. K. 2003 Comparative analysis of event-based rainfall-runoff modeling techniques – deterministic, statistical, and artificial neural networks. *J. Hydrol. Eng. ASCE* **8** (2), 93–98.

Jordan, M. I. & Jacobs, R. A. 1994 Hierarchical mixture of experts and the EM algorithm. *Neural Comput.* **6**, 181–214.

Jurik, M. 1994 *Backpercolation: Assigning Local Error in Feedforward Perception Networks, Jurik Research*. Available from: www. jurikres.com/down__/backperc.pdf (accessed December 2017).

Kasiviswanathan, K. S., Cibin, R., Sudheer, K. P. & Chaubey, I. 2013 Constructing prediction interval for artificial neural network rainfall runoff models based on ensemble simulations. *J. Hydrol.* **499** (2013), 275–288.

Kasiviswanathan, K. S., Sudheer, K. P. & He, J. 2018 Probabilistic and ensemble simulation approaches for input uncertainty quantification of artificial neural hydrologic models. *Hydrol. Sci. J.* **63** (1), 101–113.

Khalil, M., Panu, U. S. & Lennox, W. 1998 *Infilling of Missing Streamflow Values Based on Concept of Groups and Neural Networks*. Civil Engineering Technical Report No CE-98-2, Lakehead University, Canada.

Kilinc, I., Cigizoglu, H. K. & Zuran, A. 2006 A comparison of three methods for the prediction of future streamflow data. In *Conference on Water Observation and Information System for Decision Support*. Balkan Institute of Water and Environment, Macedonia, pp. 1–8.

Kottek, M., Grieser, J., Beck, C., Rudolf, B. & Rubel, F. 2006 World Map of the Köppen-Geiger climate classification updated. *Meteorol. Z.* **15**, 259–263.

Lakshmanan, V., Gourley, J. J., Flamig, Z. & Giangrande, S. 2009 A simple data-driven model for streamflow prediction. In: *7th International Conference in Artificial Intelligence and its Application to Environmental Science*. American Meteorological Society, Massachusetts, pp. 1–7.

Liong, S. Y., Lim, W. H. & Paudyal, G. N. 2000 River stage forecasting in Bangladesh: neural network approach. *J. Comput. Civil Eng. ASCE* **14** (1), 1–8.

Maier, H. & Dandy, G. C. 2000 Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling & Software* **15**, 101–124.

Matalas, N. 1997 Stochastic hydrology in the context of climate change. *Clim. Change* **37**, 89–101.

McCuen, R. H. & Snyder, W. M. 1986 *Hydrologic Modeling: Statistical Methods and Applications*. Prentice Hall, New Jersey, USA.

Michalewicz, Z. 1996 *Genetic Algorithms + Data Structure = Evolution Programs*. Springer, Berlin.

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

Mutlu, E., Chaubey, I., Hexmoor, H. & Bajwa, S. G. 2008 Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed. *Hydrol. Process.* **22** (26), 5097–5106.

Muttiah, R. S., Srinivasan, R. & Allen, P. M. 1997 Prediction of two-year peak stream discharge using neural networks. *J. Am. Water Resour. Assoc.* **33**, 625–630.

Nash, J. E. & Sutcliffe, J. V. 1970 River flow forecasting through conceptual models I – A discussion of principles. *J. Hydrol.* **10** (3), 282–290.

Panu, U., Panu, U. & Ragade, R. 1978 A feature prediction model in synthetic hydrology based on concepts of pattern recognition. *Water Resources Research* **14** (2), 335–344.

Raman, H. & Chandramouli, V. 1996 Deriving a general operating policy for reservoir using neural network. *Journal of Water Resources Planning and Management, ASCE* **122** (5), 342–347.

Ranjithan, S., Eheart, J. W. & Garrett Jr., J. H. 1993 Neural network based screening for ground water reclamation under uncertainty. *Water Resour. Res.* **29** (3), 563–574.

Reed, R. D. & Marks, R. J. 1999 *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press, Cambridge, MA.

Rivera, J. C. O., Bartua, R. G. & Andreu, J. 2002 *An Artificial Neural Networks Modeling Approach Applied to the Probabilistic Management of Water Resources Systems*. International Environmental Modeling and Software Society, iEMSs sessions, Switzerland, pp. 178–183.

Rogers, L. L. & Dowla, F. V. 1994 Optimization of ground water remediation using artificial neural networks with parallel solute transport modeling. *Water Resour. Res.* **30** (2), 457–481.

Salas, J. D.-J. 1993 Analysis and modeling of hydrological time series. In: *Handbook of Hydrology* (D. R. Maidment, ed.). McGraw-Hill, NY.

See, L. & Openshaw, S. 1999 Applying soft computing approaches to river level forecasting. *Hydrol. Sci. J.* **44** (5), 763–778.

Sun, Y. D., Wendi, K. D. E. & Liong, S.-Y. 2016 Application of artificial neural networks in groundwater table forecasting – a case study in a Singapore swamp forest. *Hydrol. Earth Syst. Sci.* **20**, 1405–1412.

Smith, J. & Eli, R. N. 1995 Neural network models for rainfall-runoff process. *J. Water Resour. Plan. Manage. ASCE* **121** (6), 499–508.

Tang, Z. & Fishwick, P. A. 1993 Feed forward neural nets as models for time series forecasting. *J. Comput.* **5** (4), 374–385.

Teegavarapu, R. S. V. 1998 Input structures for a neural network model used for streamflow forecasting. In: *Hydrology in a Changing Environment*, vol. 3 (H. Wheater & C. Kirby, eds). Wiley, Chichester, pp. 104–115.

Teegavarapu, R. S. V. & Chandramouli, V. 2005 Improved weighting methods, deterministic and stochastic data-driven methods for estimation of missing precipitation records. *J. Hydrol.* **312**, 191–206.

Teegavarapu, R. S. V. & Mujumdar, P. P. 1996 Rainfall forecasting using neural networks. In: *Proceedings of IAHR International Symposium on Stochastic Hydraulics I*. Balkema, Rotterdam, pp. 325–332.

Thirumalaiah, K. & Deo, M. C. 1998 River stage forecasting using artificial neural networks. *J. Hydrol. Eng. ASCE* **3**, 26–32.

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** (2), 486–502.

Unny, T. E., Panu, U. S., McInnes, C. D. & Wong, A. K. C. 1981
    Pattern analysis and synthesis of time dependent hydrologic
    data. *Adv. Hydrosci.* **12**, 222–224.
Zhang, B. & Govindaraju, R. S. 2000 Modular neural networks
    for watershed runoff. In: *Artificial Neural Networks in*

Hydrology (R. S. Govindaraju & A. Ramachandra Rao,
    eds). Kluwer Academic Publishers, Netherlands,
    pp. 73–91.
Zimmermann, H.-J. 1993 *Fuzzy Set Theory and its Applications*.
    Kluwer Academic Publishers, Boston.