

## Application of the PSO–neural network in rainfall–runoff modeling

Keshav Kumar<sup>a,\*</sup>, Vivekanand Singh<sup>b</sup> and Thendiyath Roshni<sup>b</sup>

<sup>a</sup> Department of Civil Engineering, Nalanda College of Engineering Chandl, Bihar, India

<sup>b</sup> Department of Civil Engineering, National Institute of Technology Patna, Bihar, India

\*Corresponding author. E-mail: keshav151008@nitp.ac.in

### ABSTRACT

In this study, rainfall–runoff (R–R) models were developed by assembling Particle Swarm Optimization (PSO) with the Feed Forward Neural Network (FFNN) and the Wavelet Neural Network (WNN). Performances of the model were compared with the wavelet ensembled neural network (WNN) and the conventional FFNN. The data from 1981 to 2005 were used for calibration and from 2006 to 2014 for validation of the models. Different combinations of rainfall and runoff were considered as inputs to the PSO–FFNN model. The fitness value and computational time of all the combinations were computed. Input combination was selected based on the lowest fitness value and lowest computational time. Four R–R models (FFNN, WNN, PSO–FFNN and PSO–WNN) were developed with the best input combination. The performance of the models was evaluated using statistical parameters (Nash–Sutcliffe Efficiency (*NSE*), *D* and root mean square error (*RMSE*)) and parameters vary in the range of (0.86–0.90), (0.95–0.97) and (68.87–84.37), respectively. After comparing the performance parameters and computational time of all four models, it was found that the PSO–FFNN model gave better values of *NSE* (0.89), *D* (0.97), *RMSE* (68.87) and less computational time (125.42 s) than other models. Thus, the PSO–FFNN model was better than the other three models (FFNN, WNN and PSO–WNN).

**Key words:** FFNN, optimization, PSO, rainfall–runoff, WNN

### HIGHLIGHTS

- The goals of this research are selection of input combination based on fitness value and computational time.
- Four R–R models (FFNN, WNN, PSO–FFNN and PSO–WNN) were developed with the best input combination. The performance of the models was evaluated using statistical parameters (*NSE*, *D* and *RMSE*).

## 1. INTRODUCTION

The rainfall–runoff process is non-linear, spatially distributed and changing with time hence, it cannot be easily described by simple models. A considerable research effort around hydrology during the past few decades has been devoted toward the development of computer-based models of rainfall–runoff processes. A rainfall–runoff model is used to simulate the runoff of a catchment for a given rainfall as the input (Kumar *et al.* 2018; Kumar *et al.* 2020). The estimation of catchment runoff is required to assess flood peaks, assessment of water for agricultural or industrial purposes, municipal, irrigation, wildlife protection and many more (Satheeshkumar *et al.* 2017). Several R–R models (black box models, conceptual models and distributed models) have been developed over the years. The black box models are mainly focused on the transfer function which creates a connection between inputs and outputs without considering the relationship between them. Distributed models require large computer resources and a large amount of data for the successful implementation of the model as compared to lumped models (Brirhet & Benaabidate 2016). Conceptual models require a huge computation for calibrating the parameters involved.

Black box models have some advantages such as simple mathematics, the least computational requirements and satisfactory results. Neural network models can find a relationship between input samples and can group samples similarly to cluster analysis. Neural networks have been applied in many areas of water resources such as the development of the rainfall–runoff model, stream flow forecasting, ground water modeling, etc. Neural network models provide better results when compared with other conceptual SAC–SMA (Sacramento Soil Moisture Accounting) models (Hsu *et al.* 1995), autoregressive models (Raman & Sunilkumar 1995),

This is an Open Access article distributed under the terms of the Creative Commons Attribution Licence (CC BY–NC–ND 4.0), which permits copying and redistribution for non-commercial purposes with no derivatives, provided the original work is properly cited (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

ARMAX models (Fernando & Jayawardena 1998), multiple regression models (Thirumalaiah & Deo 2000), linear and non-linear regressive models (Elshorbagy *et al.* 2000) and conceptual models (Salas 1993). Asadnia *et al.* (2014); Kalteh (2008); Kumar *et al.* (2008); Nourani *et al.* (2014); Nourani *et al.* (2011); Solaimani (2009); Sudheer *et al.* (2002) used the neural network model for rainfall–runoff studies.

However, the conventional neural network model has many drawbacks such as the performance of the network depends on initial weights and the solution reaching global optimum is not assured. To overcome these limitations, it is essential to develop an efficient method to optimize the neural network. Optimization techniques have been successfully employed for overcoming the limitations of the neural network in recent investigations. Nourani *et al.* (2011) have used a hybrid wavelet genetic programming (WGP) approach to optimize neural networks and found that the results of hybrid models are more satisfactory. Daneshmand *et al.* (2014); De Paola *et al.* (2016); Heydari *et al.* (2016) have used optimization techniques to find satisfactory results. Swarm intelligence-based Particle Swarm Optimization (PSO) technique is used to optimize the neural network. The main reason is to develop a hybrid neural network to optimize the weight of the network in minimum computational time. There is no overlapping and mutation calculation in PSO. The PSO search can be done by the speed of the particle. During optimization, the most optimistic particle transfers information to the other particle. The speed of re-searching the optimum value is very fast in PSO. Thus, this study shows the implementation of PSO with FFNN and WNN for rainfall–runoff modeling.

A few research works have been carried out in water resources using PSO (Jha *et al.* 2009; Khajeh *et al.* 2013). However, the authors could not find the application of PSO in rainfall–runoff modeling. Through this paper, the authors attempted to develop hybrid models with PSO to find the applicability of PSO in rainfall–runoff modeling. Hence, the main objective of this paper is to develop and compare the performance of hybrid models (PSO–FFNN, PSO–WNN and WNN) with the conventional neural network (FFNN). Furthermore, the fitness value and computational time of the four models have been computed and compared, which shows the efficiency of the developed models.

## 2. STUDY AREA AND DATA USED

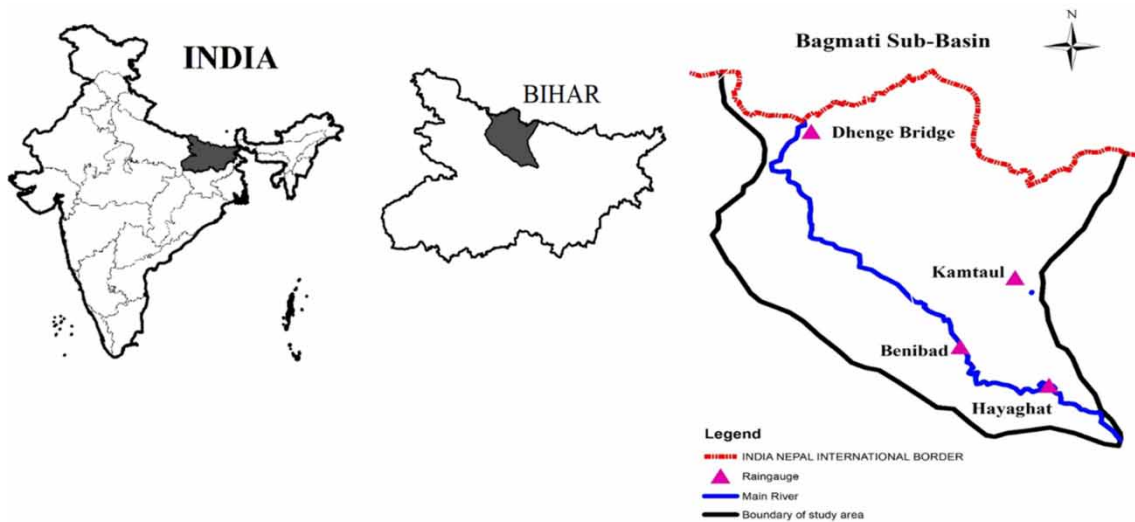
The Bagmati river is a perennial river in Nepal and India. It originates from the Shivpuri ranges of hills in Nepal at latitude 27°47'N and longitude 85°17'E and 16 km North East of Kathmandu at an altitude of 1,500 m above the mean sea level. The length of the river is 589 km and the catchment area of the Bagmati river basin is 14,384 km<sup>2</sup>. In the Bagmati river basin, 45% of the total area falls in the Bihar region of India and the rest lies in Nepal. It passes nearly 195 km in Nepal and the remaining 394 km in Bihar. The average annual rainfall of the Bagmati river basin including Adhwara is 1,255 mm. The land is mainly utilized for horticultural and agricultural purposes in the study area. 20% area is under non-agricultural uses such as roads, railways, waterbodies, buildings, etc. No forest cover is available in the study area. The climatic condition of the basin is changing due to the intrinsic topography. Temperature generally decreases with altitude and becomes high in the summer and low in the winter (Shrestha & Sthapit 2016). Selected rain gauge stations (Benibad, Dheng Bridge and Kamtaul) of the Bagmati river basin in Bihar (India) are shown in Figure 1.

For this study, monthly rainfall data of 34 years, i.e., from years 1981 to 2014 at three rain gauge stations in the Bagmati river basin have been used. These data have been collected from Indian Meteorological Department (IMD), Pune. Monthly runoff data at Hayaghat gauging site from 1981 to 2014 have been collected from Central Water Commission (CWC), Patna. Average rainfall over the basin has been computed using the Thiessen polygon method from data from three rain gauge stations (Deheng, Kamtaul and Benibad).

## 3. METHODOLOGY

### 3.1. Selection of input combination

Four combinations are shown in Table 1. The selection of input combination was done based on computational time and fitness value using the PSO–FFNN model. The combination, which shows less computational time and fitness value, was selected for the development of the models.



**Figure 1** | Map showing the main river and four rain gauge stations in the study area of the Bagmati river basin in India.

**Table 1** | Different input combinations used to develop models

Combination 1	Rainfall ( $t - 1$ ) runoff ( $t - 1$ ) rainfall ( $t$ )
Combination 2	Rainfall ( $t - 1$ ) rainfall ( $t$ )
Combination 3	Rainfall ( $t - 2$ ) runoff ( $t - 2$ ) rainfall ( $t - 1$ ) runoff ( $t - 1$ ) rainfall ( $t$ )
Combination 4	Rainfall ( $t - 2$ ) runoff ( $t - 2$ ) rainfall ( $t - 1$ ) rainfall ( $t$ )

### 3.2. Development of models

Selected input combination is used to develop the four models (FFNN, WNN, PSO-FFNN and PSO-WNN). The monthly rainfall and runoff data have been divided into two parts. In the first part, 70% of the data (i.e., from 1981 to 2005) have been used for calibration and in the second part, i.e., the remaining 30% (i.e., from 2006 to 2014) have been used for validation models.

#### 3.2.1. Development of a Feed Forward Neural Network

A Feed Forward Neural Network (FFNN) was used for the development of the FFNN model. The learning algorithm was adopted with a back propagation algorithm based on the generalized delta rule proposed by Rumelhart *et al.* (1986). In this algorithm, the connection weight between the nodes was adjusted by the strength of the signal in the connection and the total measure of the error.

The model error was computed by subtracting the model output from the observed output. This error was reduced by redistributing backward. The weights in the connection were adjusted in this step. The output was computed from the FFNN model using the adjusted weights. Back propagation was continued for a number of given cycles or until a prescribed error tolerance was reached. As mentioned by Dawson & Wilby (1998), different transfer functions and internal parameters were to be considered to make the network learning more generalized. The best fit structure of the ANN model was determined in the training process.

#### 3.2.2. Development of a Wavelet Neural Network

Wavelet is used to break the signals into various parts of frequency using the wavelet decomposition tool. Every component of frequency is compared with the original frequency signal. Wavelet transformation is mainly classified as a continuous and discrete wavelet. Wavelet is represented by a small wave function. Waves that decay in a finite period are known as small waves and long waves decay repeatedly over an infinite period. A wavelet function is described by the following equation.

$$\int_{-\infty}^{\infty} \Psi(u) du = 0 \quad (1)$$

$$\int_{-\infty}^{\infty} \Psi^2(u) du = 0 \quad (2)$$

$$\text{Admissibility condition, } C_{\Psi} = \frac{\int_{-\infty}^{\infty} |\Psi(f)|^2 df}{f} < \infty \quad (3)$$

where  $0 < C_{\Psi} < \infty$ ,  $u$  is the input wavelet signal

$$\Psi_{\lambda,t}(u) = \frac{\Psi}{\lambda} \left( \frac{u-t}{\lambda} \right) \quad (4)$$

where  $\Psi(\cdot)$  represents the mother wavelet,  $t$  indicates the translation parameter,  $t$  is the finite and  $\lambda$  is the dilation parameter and dilation  $\lambda > 0$ . The right side of Equation (4) is normalized so that  $\|\Psi_{\lambda,t}\| = \|\Psi\|$  for all  $\lambda$  and  $t$ .

The selection of the mother wavelet depends on the signal to be analyzed. Mainly, Morlet and Daubechies wavelet transform will be used as the 'Mother' wavelet (Shoaib *et al.* 2014). Daubechies wavelet shows a good interconnection between parsimony and data abundance; it gives approximately similar events over the observed time sequence and shows up in various patterns that most forecast models cannot distinguish them well.

Input signals are decomposed by wavelet using Daubechies Discrete Wavelets one-dimensional, up to the second level. The decomposition level of the wavelet is selected by  $\lceil \log_{10}(N) \rceil$  where,  $N$  is the total number of observation data. Input variables have been decomposed into detailed signals and approximate signals up to the second level. The Minmax threshold is used to denoise the decomposed signals. Then, these denoised signals are used for the WNN.

### 3.2.3. Particle Swarm Optimization

PSO is one of the new optimization techniques based on population. Each particle of PSO has its own velocity vector and position vector (Liu *et al.* 2008). A possible solution to a problem is represented by a velocity vector and a position vector. Position means rank is assigned to training data during calibration. The velocity vector is the execution time taken by the training data. Each particle stores its best global position obtained through interaction with its neighbor particles. The PSO algorithm manages its search by adjusting the position of particles and velocity vector. In PSO, the movement of particles is determined by the objective function. Particles nearer to the optimal solution have a lower velocity. Particles far from the optimal solution have a higher velocity.

Many researchers worked on optimization techniques such as Gravitational Search Algorithm, Cuckoo Search Algorithm, Krill Herd and PSO with Genetic Algorithm to solve problems such as aircraft landing, image processing, etc. (Cui *et al.* 2008; Liu *et al.* 2008; Agrawal & Kaur 2014; Tayarani *et al.* 2015; Chang *et al.* 2016; Girish 2016; Kakkar & Jain 2016; Wang *et al.* 2016).

PSO algorithm is mainly focused to optimize the network complexity and model performance. The connection weights of the models are adjusted by fitness evaluation Equation (5) during calibration.

$$E = \frac{1}{2} \sum_p \sum_{i=1}^q (d_i - y_i)^2 \quad (5)$$

where  $y_i$  is the output obtained from the models,  $d_i$  is the target value,  $q$  is the number of patterns used for calibration and  $p$  is the number of nodes in the output layer.

Both models are calibrated by lowering the fitness evaluation parameter in a search space. To obtain the fitness value, PSO finds the possible solution to a problem and measures its quality by using forward propagation through the model network.

PSO finds possible solutions. The quality of the possible solutions is measured by using forward propagation through the model network to obtain fitness evaluation values. The following steps are involved in PSO-based training algorithm:

- First decide the structure of the network and parameters of PSO.
- Initialize positions and velocities of a population. The position of everyone consists of network connection weights.

- Based on a group of calibration data, calculate the fitness value of each particle using Equation (5). Initialize the individual best position and the global best position.
- Update the position and the velocity for the whole particle swarm.
- If the stopping condition of the algorithm is not satisfied, move to step 3. Otherwise, terminate the iteration and get the best-optimized weights from the global best solution.

### 3.2.4. PSO–FFNN and PSO–WNN networks

Two models (PSO–FFNN and PSO–WNN) are developed for optimizing the connection weights of the network to produce the desired output. In Figure 2, the steps involved in the development of models are shown. PSO optimization techniques are assembled with FFNN and WNN. In PSO–FFNN and PSO–WNN, optimized weights are redistributed to the network links for getting the optimum result.

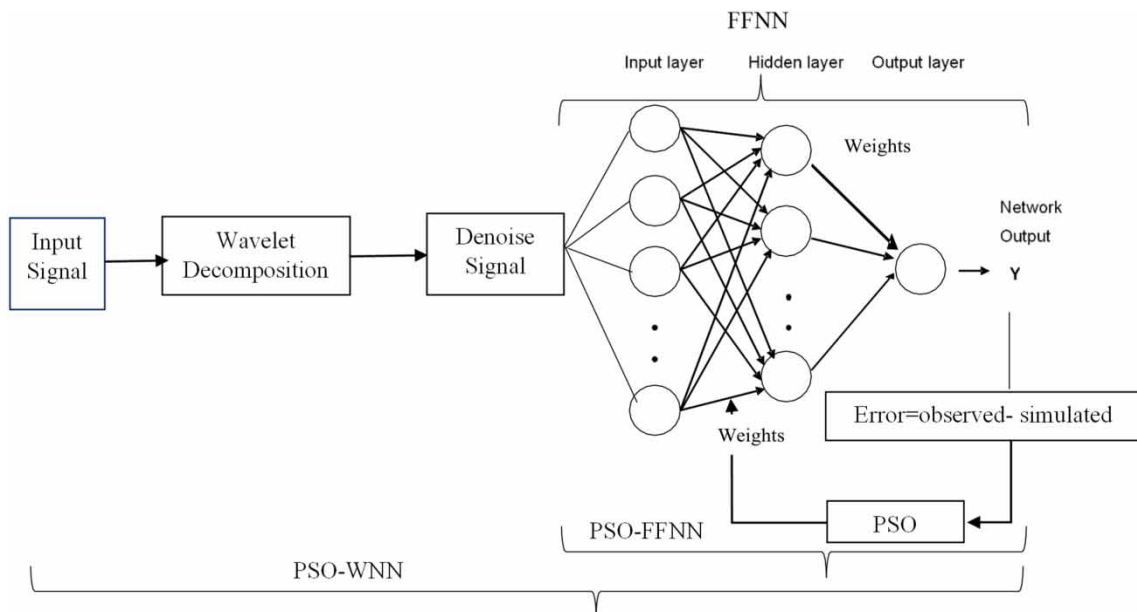


Figure 2 | Flow chart of four models (FFNN, WNN, PSO–FFNN and PSO–WNN).

### 3.3. Performance evaluation of models

The performance of models has been analyzed using the following performance indicators:

$$\text{Coefficient of correlation } (R) = \frac{\sum ty}{\sqrt{\sum T^2 \sum Y^2}} \tag{6}$$

$$\text{Nash – Sutcliffe Efficiency (NSE)} = 1 - \frac{\sum (t - y)^2}{\sum (t - \bar{t})^2} \tag{7}$$

$$\text{Index of Agreement } (D) = 1 - \frac{\sum (t - y)^2}{\sum (|t - \bar{y}| + |y - \bar{y}|)^2} \tag{8}$$

$$\text{Root mean square error (RMSE)} = \sqrt{\frac{\sum_{N=1}^N (t - y)^2}{N}} \tag{9}$$

$$\text{Coefficient of determination } (R^2) = \frac{N(\sum ty) - (\sum t)(\sum y)}{\sqrt{[N \sum t^2 - (\sum t)^2][N \sum y^2 - (\sum y)^2]}} \tag{10}$$

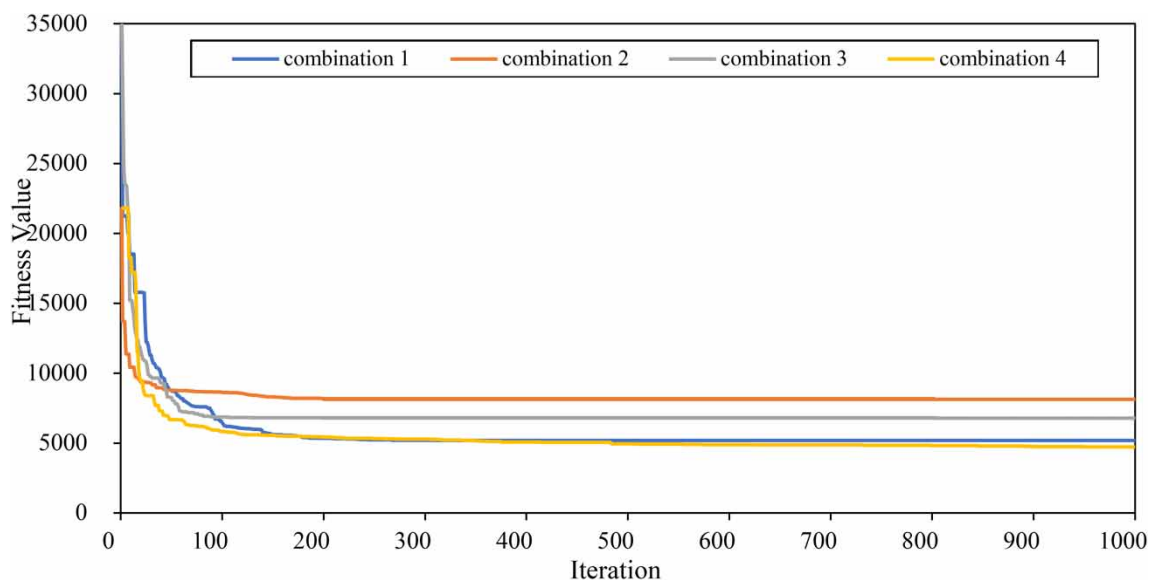
where  $N$  is the number of observations,  $y$  is the computed data,  $t$  is the observed data,  $T = t - \bar{t}$  in which  $\bar{t}$  represents the mean of the observed data and  $Y = y - \bar{y}$  in which  $\bar{y}$  represents the mean of the output data.

The root mean square error (*RMSE*) is used to measure the differences between predicted values and observed values. A correlation coefficient value lies between  $-1$  and  $+1$ , which quantifies the type of correlation and dependence, by developing meaningful statistical relationships between two or more variables in fundamental statistics. Nash–Sutcliffe Efficiency (*NSE*) is used to quantify how well a model simulation can predict the outcome variable and it shows the predictive power of any model. *NSE* is varying from  $-\infty$  to  $1$ .  $NSE = 0$  indicates that the model predictions are as accurate as the mean of the observed data,  $-\infty < NSE < 0$  indicates that the observed mean is a better predictor than the model.  $NSE = 1$  corresponds to a perfect match of the model to the observed data.

## 4. RESULTS AND DISCUSSION

### 4.1. Selection of input combination

The selection of input combination has been done by calculating the fitness value and computational time of the model. Figure 3 presents the comparison of fitness value of different combinations using PSO–FFNN. The fitness values of combinations 1 and 4 show almost similar results, but it was lower for combination 4. Combination 4 shows a lesser fitness value in the PSO–FFNN model compared to other combinations. Figure 4 presents the computational times of the model for different combinations.



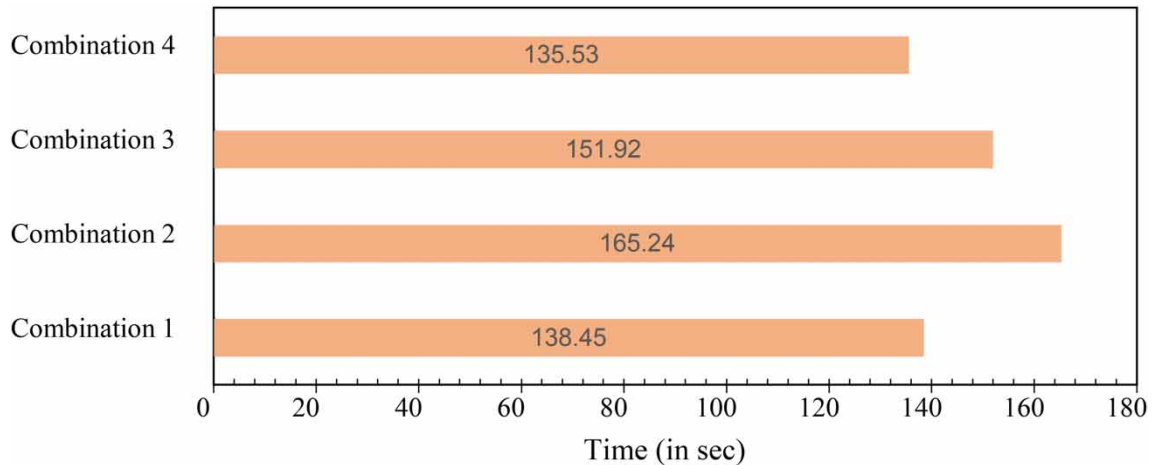
**Figure 3** | Comparison of different combinations using the PSO–FFNN with fitness value.

Computational time was one of the most important parameters in developing soft computing models. The computational time of combination 4 was lower, 135.5 s, as compared to the other three combinations. As a result, combination 4 was most suited for the development of models.

### 4.2. Calibration and validation of models

The selected combination has been used for the development of four models (FFNN, WNN, PSO–FFNN and PSO–WNN) in the Bagmati river basin. The monthly rainfall and runoff of the Bagmati river basin from 1981 to 2014 have been used for calibration and validation of the models. All four models have been calibrated using data from 1981 to 2005 and validated from 2006 to 2014. The selected combination was trained with error tolerance, the number of cycles for learning, the learning parameter and the neurons in the hidden layer as 0.01, 1,000, 0.1 and 1, respectively. The output values from the network were denormalized and compared with the observed targeted values. The performance criteria such as *NSE*, *RMSE* and *D* were used to examine the performance of the model. In the training of the above combination, the number of cycles was increased



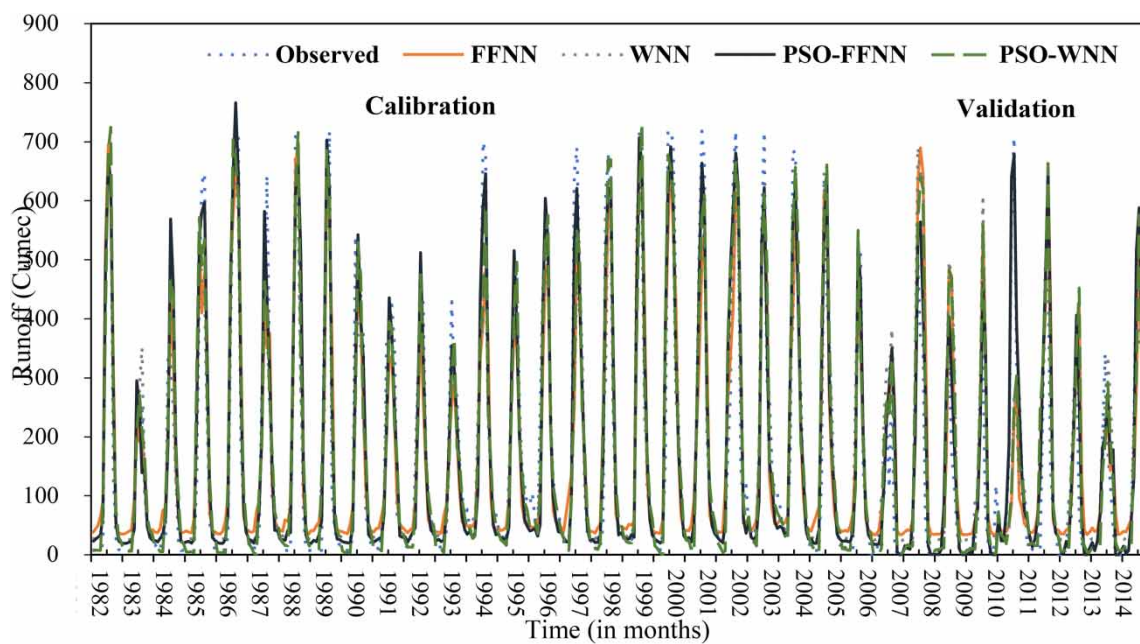


**Figure 4** | Time taken by different combinations in the PSO-FFNN.

in steps up to 2,000 and it was found that the convergence was not static for this number of cycles. For this initial combination, the *RMSE* of targeted and expected values was very high and the coefficient of correlation was low. Then, the network was trained with the increase or decrease values of error tolerance and varied values of the learning parameter. The learning parameter and the error tolerance were fixed with low *RMSE* and a high coefficient of correlation. The neurons in the hidden layer were increased from the minimum to the number high value of the coefficient of correlation. The number of neurons, which gave the highest *NSE* value, was selected for this combination. It was observed that convergence for this combination was achieved with the error tolerance, the learning parameter and the number of cycles and neurons in the hidden layer as 0.1, 0.9, 1,000 and 5, respectively. The *RMSE* was 68.87 and *NSE* was 0.90 in the PSO-FFNN model. To get the optimized structure of the hybrid model network, input combination 4 was used for calibration and validation of the models. It was noticed that the best convergence was achieved for the above combination by optimizing the weights of the network and internal parameters (error tolerance, learning parameter, number of hidden layer etc.) of the network.

**4.3. Performance analysis of the models developed**

Four models (FFNN, WNN, PSO-FFNN and PSO-WNN) have been calibrated using combination 4 from 1981 to 2005 and validated from 2006 to 2014. Figure 5 presents the comparison of the computed runoff of all four



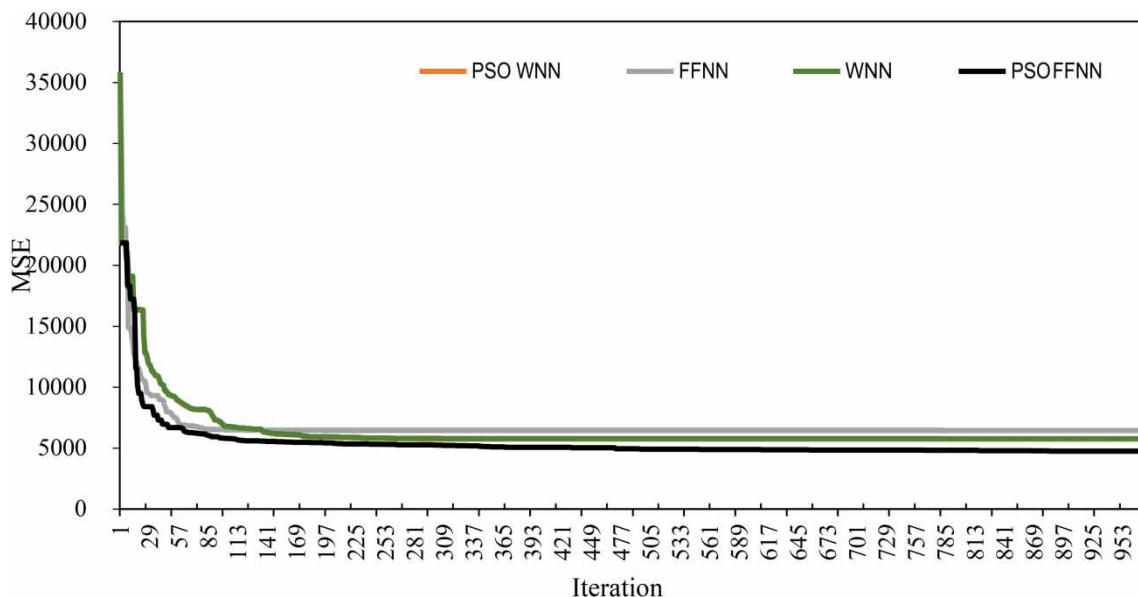
**Figure 5** | Calibration and validation of four models (FFNN, WNN, PSO-FFNN and PSO-WNN).

models with the observed runoff at the Hayaghat gauging site. All the peaks of the computed runoff hydrograph are not matching well with the observed peaks in FFNN and WNN models during the calibration and validation of datasets from 1981 to 2014. The coefficients of correlation are 0.93 and 0.94 during calibration and 0.92 and 0.93 during validation for the FFNN and WNN model, respectively. Computed runoff peaks are the same as observed in PSO–FFNN during calibration and validation. The coefficient of correlation and *RMSE* were 0.95 and 68.87 during calibration and 0.95 and 69.15 during validation, respectively. The computed statistical parameters of all the models have been presented in Table 2.

**Table 2** | Values of statistical parameters during calibration and validation of all the four models

Models/Statistical parameter	Calibration				Validation			
	FFNN	WNN	PSO–FFNN	PSO–WNN	FFNN	WNN	PSO–FFNN	PSO–WNN
<i>RMSE</i>	80.22	75.92	68.87	73.71	84.35	77.21	69.15	74.55
<i>NSE</i>	0.87	0.88	0.90	0.89	0.86	0.87	0.89	0.88
<i>D</i>	0.96	0.96	0.97	0.97	0.95	0.95	0.97	0.96
<i>R</i>	0.93	0.94	0.95	0.94	0.92	0.93	0.94	0.93
<i>R</i> <sup>2</sup>	0.87	0.89	0.90	0.89	0.86	0.88	0.89	0.88

During the calibration the performance parameters, i.e., *RMSE*, *NSE*, *R*, *D* and *R*<sup>2</sup> of PSO–FFNN are 68.8, 0.9, 0.97, 0.95 and 0.9, respectively, and 69.15, 0.89, 0.97, 0.94 and 0.89, respectively, during the validation of models. Performance parameters indicate that the PSO–FFNN model shows better results as compared to the other three models (FFNN, WNN and PSO–WNN). Optimization of error of all four models has been presented in Figure 6. The PSO–FFNN model gives lower error as compared to the other three models. The computation time of four models has also been calculated and presented in Figure 7. The computation time of PSO–FFNN, PSO–WNN, WNN and FFNN was 125.42, 135.92, 165.24 and 181.28 s, respectively. As a result, the PSO–FFNN model has less computation time as compared to the other three models. Based on these results, it is evident that the PSO–FFNN model is better than other models developed.

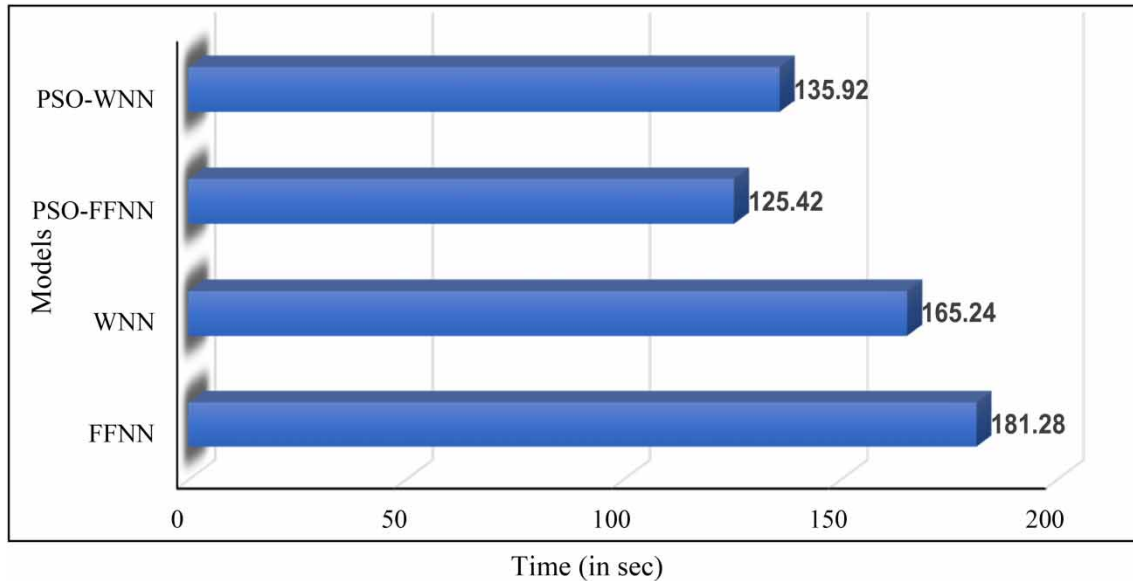


**Figure 6** | Optimization of error in four models.

## 5. DISCUSSION

Four models have been developed, i.e., FFNN, WNN, PSO–FFNN and PSO–WNN. It was found that the performance parameters, i.e., *RMSE*, *NSE*, *R*, *D* and *R*<sup>2</sup> of PSO–FFNN were 68.87, 0.9, 0.97, 0.95 and 0.9,





**Figure 7** | Comparison of computational time of all the models.

respectively, during the calibration of models and 69.15, 0.89, 0.97, 0.943 and 0.895, respectively, during the validation of models. Performance parameters of PSO-FFNN models show a 4–12% decrease in *RMSE*, a 1–2% increase in *NSE*, *D*, *R* and  $R^2$  which shows the better results among the other three developed models (FFNN, WNN and PSO-WNN). These results are in accordance with the results of Mazandaranizadeh & Motahari (2017).

## 6. CONCLUSIONS

In this paper, the application of PSO with a neural network to develop rainfall-runoff modeling of the Bagmati river basin was studied. Four models (FFNN, WNN, PSO-FFNN and PSO-WNN) are developed. The monthly rainfall and runoff from 1981 to 2014 of the Bagmati river basin were selected for the development of the models. Three rain gauge stations (Benibad, Dheng bridge and Kamtaul) were selected and the average rainfall was calculated using the Thiessen polygon method. The data from 1981 to 2005 were used for calibration of the models and from 2006 to 2014 for validation of the developed models. The selection of input combination was done using PSO-FFNN. Fitness value and computational time were computed for all combinations. Combination 4 (rainfall ( $t-2$ ) runoff ( $t-2$ ) rainfall ( $t-1$ ) rainfall ( $t$ )) shows less fitness value, i.e., 4,742 and less computational time, i.e., 135.53 s as compared with other three inputs combination. Input combination 4 was selected for the development of the models. Performance evaluation of the developed models was carried out using statistical performance parameters (*NSE*, *D*, *RMSE*). The performance parameters such as *NSE*, *D* and *RMSE* of all developed models vary in the range of (0.86–0.90), (0.95–0.97) and (68.87–84.37), respectively. The *NSE* values of FFNN, WNN, PSO-FFNN and PSO-WNN are 0.87, 0.88, 0.9 and 0.89, respectively, during calibration and 0.86, 0.87, 0.89 and 0.88, respectively, during validation of the models. PSO-FFNN models show better results as compared to the other models (FFNN, WNN and PSO-WNN). Computation times of the different developed models are also calculated and compared in this study. The computational time taken by PSO-FFNN, PSO-WNN, WNN and FFNN was 125.42, 135.92, 165.24 and 181.28 s, respectively. As a result, the PSO-FFNN model has lesser computation time (i.e., 125.42 s) as compared to the other three models. Finally, it was found that the PSO-FFNN model performs better than other models (FFNN, WNN and PSO-WNN).

## ACKNOWLEDGEMENTS

The authors would like to acknowledge the IMD (Indian Meteorological Department) and CWC (Central Water Commission) for providing the data for analysis.

## DATA AVAILABILITY STATEMENT

Data cannot be made publicly available; readers should contact the corresponding author for details.

## CONFLICT OF INTEREST

The authors declare there is no conflict.

## REFERENCES

- Agrawal, A. P. & Kaur, A. 2014 A Comparative Analysis of Memory Using and Memory Less Algorithms for Quadratic Assignment Problem. In: *Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit*. <https://doi.org/10.1109/CONFLUENCE.2014.6949357>.
- Asadnia, M., Chua, L. H. C., Qin, X. S. & Talei, A. 2014 Improved particle swarm optimization–Based artificial neural network for rainfall-Runoff modeling. *Journal of Hydrologic Engineering*. <https://doi.org/10.1515/aot-2015-0058>.
- Brirhet, H. & Benaabidate, L. 2016 Comparison of two hydrological models (Lumped and distributed) over a pilot area of the issen watershed in the Souss Basin, Morocco. *European Scientific Journal, ESJ*. <https://doi.org/10.19044/esj.2016.v12n18p347>.
- Chang, X., Yi, P. & Zhang, Q. 2016 Skey frames extraction from human motion capture data based on hybrid particle swarm optimization algorithm. *Studies in Computational Intelligence*. [https://doi.org/10.1007/978-3-319-31277-4\\_29](https://doi.org/10.1007/978-3-319-31277-4_29).
- Cui, G., Qin, L., Liu, S., Wang, Y., Zhang, X. & Cao, X. 2008 Modified PSO algorithm for solving planar graph coloring problem. *Progress in Natural Science*. <https://doi.org/10.1016/j.pnsc.2007.11.009>.
- Daneshmand, F., Karimi, A., Nikoo, M. R., Bazargan-Lari, M. R. & Adamowski, J. 2014 Mitigating socio-economic-environmental impacts during drought periods by optimizing the conjunctive management of water resources. *Water Resources Management*. <https://doi.org/10.1007/s11269-014-0549-7>.
- Dawson, C. W. & Wilby, R. 1998 An artificial neural network approach to rainfall-runoff modelling [Une Approche de La Modélisation Pluie-Deblt Par Ies Réseaux Neuronaux Artificiels. *Hydrological Sciences Journal*. <https://doi.org/10.1080/02626669809492102>.
- Elshorbagy, A., Simonovic, S. P. & Panu, U. S. 2000 Performance evaluation of artificial neural networks for runoff prediction. *Journal of Hydrologic Engineering*. [https://doi.org/10.1061/\(ASCE\)1084-0699\(2000\)5:4\(424\)](https://doi.org/10.1061/(ASCE)1084-0699(2000)5:4(424)).
- Fernando, D. A. K. & Jayawardena, A. W. 1998 Runoff forecasting using RBF networks with OLS algorithm. *Journal of Hydrologic Engineering*. [https://doi.org/10.1061/\(ASCE\)1084-0699\(1998\)3:3\(203\)](https://doi.org/10.1061/(ASCE)1084-0699(1998)3:3(203)).
- Girish, B. S. 2016 An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem. *Applied Soft Computing Journal*. <https://doi.org/10.1016/j.asoc.2016.04.011>.
- Heydari, F., Saghafian, B. & Delavar, M. 2016 Coupled quantity-quality simulation-optimization model for conjunctive surface-groundwater use. *Water Resources Management*. <https://doi.org/10.1007/s11269-016-1426-3>.
- Hsu, K.-I., Gupta, H. V. & Sorooshian, S. 1995 Artificial neural network modeling of the rainfall-runoff process. *Water Resources Research* **31**(10), 2517. <https://doi.org/10.1029/95WR01955>.
- Jha, G. K., Thulasiraman, P. & Thulasiram, R. K. 2009 Pso based neural network for time series forecasting. In: *Proceedings of the International Joint Conference on Neural Networks*. <https://doi.org/10.1109/IJCNN.2009.5178707>.
- Kakkar, M. & Jain, S. 2016 Feature Selection in Software Defect Prediction: A Comparative Study. In: *Proceedings of the 2016 6th International Conference – Cloud System and Big Data Engineering, Confluence 2016*. <https://doi.org/10.1109/CONFLUENCE.2016.7508200>.
- Kalteh, A. 2008 Rainfall-Runoff modelling using artificial neural networks (ANNs): modelling and understanding. *Caspian Journal of Environmental Sciences* **6**(1), 53–58.
- Khajeh, M., Kaykhaii, M. & Sharafi, A. 2013 Application of PSO-artificial neural network and response surface methodology for removal of methylene blue using silver nanoparticles from water samples. *Journal of Industrial and Engineering Chemistry*. <https://doi.org/10.1016/j.jiec.2013.01.033>.
- Kumar, P. R., Ramana Murthy, M. V., Eashwar, D. & Venkatdas, M. 2008 Time series modeling using artificial neural networks. *Journal of Theoretical and Applied Information Technology* **4**(12), 1259–1264.
- Kumar, K., Singh, V. & Roshni, T. 2018 Efficacy of neural network in rainfall-Runoff modelling of Bagmati River Basin. *International Journal of Civil Engineering and Technology (IJCIET)* **9**(11), 37–46.
- Kumar, K., Singh, V. & Roshni, T. 2020 Efficacy of hybrid neural networks in statistical downscaling of precipitation of the Bagmati River basin. *Journal of Water and Climate Change* **11**(4), 1302–1322.
- Liu, B., Wang, L. & Jin, Y. H. 2008 An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. *Computers and Operations Research*. <https://doi.org/10.1016/j.cor.2006.12.013>.
- Mazandaranzadeh, H. & Motahari, M. 2017 Development of a PSO-ANN model for rainfall-runoff response in basins, Case Study: Karaj Basin. *Civil Engineering Journal* **3**, 35–44.
- Nourani, V., Kisi, Ö. & Komasi, M. 2011 Two hybrid artificial intelligence approaches for modeling rainfall-runoff process. *Journal of Hydrology*. <https://doi.org/10.1016/j.jhydrol.2011.03.002>.
- Nourani, V., Baghanam, A. H., Adamowski, J. & Kisi, O. 2014 Applications of hybrid wavelet-artificial intelligence models in hydrology: a review. *Journal of Hydrology* **514**, 358–377. <https://doi.org/10.1016/j.jhydrol.2014.03.057>.

- Paola, F. D., Galdiero, E. & Giugni, M. 2016 A jazz-Based approach for optimal setting of pressure reducing valves in water distribution networks. *Engineering Optimization*. <https://doi.org/10.1080/0305215X.2015.1042476>.
- Raman, H. & Sunilkumar, N. 1995 Multivariate modelling of water resources time series using artificial neural networks. *Hydrological Sciences Journal*. <https://doi.org/10.1080/02626669509491401>.
- Rumelhart, D. E., Hinton, G. E. & McClelland, J. L. 1986 A general framework for parallel distributed processing. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* 1, 45–76.
- Salas, J. D. 1993 Analysis and modeling of hydrological time series. In: Maidment, D. R. (ed.). *Handbook of Hydrology*. McGraw-Hill, New York, pp. 19.1–19.72.
- Satheeshkumar, S., Venkateswaran, S. & Kannan, R. 2017 Rainfall–Runoff estimation using SCS–CN and GIS approach in the pappiredipatti watershed of the Vaniyar Sub Basin, South India. *Modeling Earth Systems and Environment*. <https://doi.org/10.1007/s40808-017-0301-4>.
- Shoaib, M., Shamseldin, A. Y. & Melville, B. W. 2014 Comparative study of different wavelet based neural network models for rainfall-runoff modeling. *Journal of Hydrology* 515, 47–58. <https://doi.org/10.1016/j.jhydrol.2014.04.055>.
- Shrestha, R. M. & Sthapit, A. B. 2016 Temporal variation of rainfall in the Bagmati River Basin, Nepal. *Nepal Journal of Science and Technology*. <https://doi.org/10.3126/njst.v16i1.14355>.
- Solaimani, K. 2009 Rainfall-runoff prediction based on artificial neural network (a case study: Jarahi Watershed). *American-Eurasian Journal of Agricultural & Environmental Sciences* 5(6), 856–865.
- Sudheer, K. P., Gosain, A. K. & Ramasastri, K. S. 2002 A data-driven algorithm for constructing artificial neural network rainfall-runoff models. *Hydrological Processes*. <https://doi.org/10.1002/hyp.554>.
- Tayarani, M. H. N., Yao, X. & Xu, H. 2015 Meta-heuristic algorithms in car engine design: a literature survey. *IEEE Transactions on Evolutionary Computation*. <https://doi.org/10.1109/TEVC.2014.2355174>.
- Thirumalaiah, K. & Deo, M. C. 2000 Hydrological forecasting using neural networks. *Journal of Hydrologic Engineering*. [https://doi.org/10.1061/\(ASCE\)1084-0699\(2000\)5:2\(180\)](https://doi.org/10.1061/(ASCE)1084-0699(2000)5:2(180)).
- Wang, G. G., Gandomi, A. H., Alavi, A. H. & Deb, S. 2016 A hybrid method based on krill herd and quantum-behaved particle swarm optimization. *Neural Computing and Applications*. <https://doi.org/10.1007/s00521-015-1914-z>.

First received 27 May 2022; accepted in revised form 28 November 2022. Available online 14 December 2022