A comparison of particle swarm optimization and genetic algorithm for daily rainfall-runoff modelling: A case study for Southeast Queensland, Australia

Mahsa Jahandideh Tehrani[*a], Graham Jenkins[a], and Fernanda Helfer[a]

[a] *School of Engineering and Built Environment, Griffith University, Australia*

Abstract: Real-time and short-term prediction of river flow is essential for efficient flood management. To obtain accurate flow predictions, a reliable rainfall-runoff model must be used. This study proposes the application of two evolutionary algorithms, Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), to train the Artificial Neural Network (ANN) parameters in order to overcome the ANN drawbacks, such as slow learning speed and frequent trapping at local optimum. These hybrid ANN-PSO and ANN-GA approaches were validated to equip natural hazard decision makers with a robust tool for forecasting real-time streamflow as a function of combinations of different lagged rainfall and streamflow in a small catchment in Southeast Queensland, Australia. Different input combinations of lagged rainfall and streamflow (delays of one, two and three days) were tested to investigate the sensitivity of the model to the number of delayed days, and to identify the effective model input combinations for the accurate prediction of real-time streamflow, which has not yet been recognized in other studies. The results indicated that the ANN-PSO model significantly outperformed the ANN-GA model in terms of convergence speed, accuracy, and fitness function evaluation. Additionally, it was found that the rainfall and streamflow with 3-day lag time had less impact on the predicted streamflow of the studied basin, confirming that the flow of the studied river is significantly correlated with only 2-day lagged rainfall and streamflow.

Keywords: Real-time streamflow; Evolutionary computation; ANN; PSO; GA; and Rainfall-runoff modelling.

[*] Corresponding author. Address: Griffith School of Engineering, Gold Coast Campus, Griffith University, QLD 4222, Australia.
E-mail address: mahsa.jahandideh-tehrani@griffithuni.edu.au

1. Introduction

Rainfall-runoff modelling is a complex process due to the high spatial and temporal variability of rainfall patterns and catchment characteristics (Tokar and Johnson 1999). Rainfall-runoff models can be classified as conceptual or empirical models (Machado et al. 2011). Conceptual models, such as lumped models (e.g., NAM[1], HBV[2], etc.) and distributed models (e.g., SHE[3], SWAT[4], etc.), provide mathematical description of the hydrological cycle, based on the physical process of the water flow over the entire catchment. Conceptual models are associated with a challenging calibration process, besides requiring a large amount of input data. The availability of these data, and their non-linear interaction, make the estimation of runoff difficult with these models (Kuok et al. 2010; Machado et al. 2011). Therefore, hydrological modellings of data scarce catchments become limited due to lack of land use and land cover changes, meteorological, and soil moisture data. In such catchments, where limited data (e.g., rainfall and streamflow) are available, empirical models are highly efficient. Empirical models are based on building a stable non-linear rainfall-runoff relationship, while not considering the physical processes of the hydrological cycle. These models are easy to apply and require less data as well as low computational costs (Machado et al. 2011). There are different examples of empirical rainfall-runoff modelling approaches, such as autoregressive integrated moving average (ARIMA), genetic programming (GP), multiple linear regressive model (MLR), autoregressive-moving average with exogenous term model (ARMAX), etc. The stated models are easy to use but indicate poor performance in modelling non-linear hydrological process (Zhang et al. 2018). In order to tackle this drawback, many attention has been focused on artificial intelligence-based (AI-based) data driven models (Nourani et al. 2014), such as the Artificial Neural Network (ANN). Basically, ANN models outperform ARIMA, MLR, and ARIMAX in optimizing model parameters, hydrological forecasting, and data processing (Chitsaz et al. 2016; Zhang et al. 2018). The ANN

[1] Nedbor Affstromnings Model
[2] Hydrologiska Byråns Vattenbalansavdelning
[3] Systeme Hydrologique Europeen
[4] Soil and Water Assessment Tool

model represents the model input and output by its data-driven relationship (Piotrowski et al. 2011). Many researchers have suggested that similar or better predictions have been obtained from the ANN approach rather than the conceptual methods (Carcano et al. 2008; Napolinato et al. 2010).

Over the past decade, the ANN technique has been widely employed in water resources management, such as in rainfall-runoff forecasting (Elshorbagy et al. 2010; Ba et al. 2017), flood frequency analysis (Aziz et al. 2013), ungauged catchment flood prediction (Aziz et al. 2017), river flow prediction (Imrie et al. 2000), and drought forecasting (Hosseini-Moghari et al. 2017). ANN presents strong learning ability without the provision of any pre-defined model structure. There are different types of algorithms for training the ANN, which compares the observed and modelled data, and updates the ANN model parameters at each iteration to improve the ANN learning process. Back propagation algorithm (BPA) and its variants (e.g., conjugate gradient (CG), gradient descent) have been widely applied for training the ANN. BPA searches the error through applying the steepest descent method. In this approach, the ANN parameters are adjusted by moving a small step towards the negative gradient of the energy function (Chiang et al. 2004; Asadnia et al. 2018). Using gradient search methods often leads to inconsistent and less efficient performance of the ANN as the stated techniques are unable to capture the nonlinear rainfall-runoff process, and suffers from slow convergence rate (Jain and Srinivasulu, 2004). In order to improve the poor performance of gradient descent based methods, Levenberg-Marquardt (LM) training method has been used in many studies (De Vos and Rientjes 2008; Taormina et al. 2012). This method is based on coupled gradient descent (GD) with Gauss-Newton (GN) methods. However, this method also indicates slow convergence rate (Asadnia et al. 2018). Therefore, in order to tackle challenges such as slow convergence speed, unpredictable behavior of error function, large nonlinear interdependence of ANN parameters, and risk of being trapped in local optimum, a global optimization strategy is required to combine deterministic and probabilistic approaches and to direct solutions towards the global optima through competitive and systematic evolution process (Jain and Srinivasulu, 2004). To overcome these drawbacks, and also to improve ANN performance, a number of researchers have applied evolutionary techniques, such as genetic algorithm (GA) and particle swarm optimization (PSO) (Kuok et al. 2010), to build a hybrid and improved ANN model. The evolutionary algorithms are critical optimization tools, which are able to solve complex, non-linear, and discrete optimization problems (Roy and Singh 2020). The main challenges associated with evolutionary algorithms is the adjustment of algorithmic parameters and complex evolutionary process (Jahandideh-Tehrani et al. 2019). However, stated shortcomings can be addressed through using more efficient and less required algorithmic parameter specifications. For instance, as recommended by Roy and Singh (2020), PSO is simple and easy to implement by indicating good exploitation ability with less complex evolutionary operators. Additionally, the process of combining evolutionary algorithms with other statistical methods can significantly improve their search process and obtained optimum solutions (Jahandideh-Tehrani et al. 2019; Roy and Singh, 2020). GA was first introduced by Holland in (1975), which is inspired by natural evolution and Darwinian principles of selection, mutation, and recombination (crossover) with various mathematical operators (Jahandideh-Tehrani et al. 2019). PSO was first proposed by Kennedy and Eberhart in 1995, which is inspired by the group behavior of animals, including bird flocks and fish schools. Both PSO and GA are population-based algorithms, which are first initialized with a population of random solutions, after which the solutions are updated over a series of generations to search for the optimum solution (Asadinia et al. 2014).

A hybrid method of PSO and ANN has been developed in this study to improve the performance and training process of the ANN. This method is called feedforward artificial neural network-particle swarm optimization (ANN-PSO), which improves the convergence rate of ANN and avoid solutions falling into local optimum in optimization problems. PSO consists of a population (swarm) of candidate solutions (particles), where each particle is characterized with a position and a velocity. The idea of the hybrid ANN-PSO is to obtain the best set of weights and biases (particle position) where particles (problem solution) are flying through problem space to reach the best solution, and this process will prevent the solutions falling into local optimum (Kuok et al. 2010). Kouk et al. (2010) applied the hybrid ANN-PSO for hourly rainfall-runoff modelling in the Sungai Bedup Basin in Malaysia. They also analysed the sensitivity of the ANN-

PSO to the length of the calibration period, number of hidden layers, delays of days, number of particles and maximum iteration. They concluded that the hybrid ANN-PSO can successfully model rainfall-runoff relationships, particularly for high peaks. However, considering that the rainfall of the current day was used as the model input, predicting the streamflow was impossible in the stated study. Another application of the hybrid ANN-PSO has been investigated by Asadnia et al. (2014). They improved the PSO technique for training the ANN in order to predict water levels of the Heshui Watershed in China. They compared results from the ANN trained by the Levenberg-Marquardt network algorithm (LM-NN) with results from PSO-based ANN. They found that ANN-PSO models outperform LM-NN by improving convergence rate.

As mentioned before, GA initializes the search process through generating a group of random solution represented as the population of chromosomes at a time and navigates the search space through three evolutionary operations: selection, crossover, and mutation to obtain the global optimum. According to the fitness calculations for each chromosomes, the selection operator filters the chromosomes with higher fitness value for survival or producing offspring in the gene pool of the next generation. Next, crossover operator exchanges information of candidate solutions (genetic characteristics) between two random positions, in the hope of producing better offsprings. During mutation process, the entire population is explored to bring forth any possibility of improvement in the search (Chang 2008; Chandwani et al. 2015). Similar to the hybrid ANN-PSO, the combination of GA and ANN has been developed to enhance the speed of convergence towards optimal values of weights and bias in rainfall-runoff modelling process (Parasuraman and Elshorbagy 2007). Jain and Srinivasulu (2004) applied coupled GA to train the ANN, and indicated that the coupled ANN-GA predicted the daily flow more accurately than the conventional back proportion (BP)-ANN. In another study by Kasiviswanathan et al. (2013), GA was used to train the conventional ANN for the purpose of streamflow prediction using 12-year hydrological recorded data. Based on the obtained results and comparison between ANN-GA and conventional ANN, it was reported that ANN-GA had faster training capability as well as more accurate modeling results. More recently, an application of the ANN-GA to streamflow forecasting has been investigated by Moeeni et al. (2017). They applied GA as an optimization tool to identify the optimum number of hidden layer in ANN. They also compared the performance of the ANN-GA with the Seasonal Autoregressive Integrated Moving Average (SARIMA) model. They concluded that ANN-GA is more efficient in the wet years and flood flows forecasting.

Many studies have investigated the applications of simple ANN in the areas of hydrology and water resources without focusing on combining an optimization algorithm with ANN (e.g., Zhang et al. 2000; Wang et al. 2017; Ali et al. 2017; Gholami et al. 2018; Lee et al. 2019). This literature review demonstrated that further attention is required on using optimization methods in ANN modelling. As the study of ANN-PSO and ANN-GA in hydrology, particularly rainfall-runoff modelling, is still at an early stage, and little research has been conducted on these models for solving hydrological problems and real-time flow prediction, the investigation of these models in hydrology is highly recommended (Zhang et al. 2000; Sedki et al. 2009). There is also a need for further study to build enough confidence for the application of ANN-PSO and ANN-GA. Additionally, reliable real-time prediction of river flow is essential for forecasting floods and reducing the damages of flood events. Evolutionary algorithms perform as optimization tools to improve the training process and convergence rate of ANN models, which has not been considered in the literature. This study aims to develop and evaluate both ANN-PSO and ANN-GA rainfall-runoff models to forecast real-time streamflow in the Nerang River Basin, Southeast Queensland, Australia. The PSO and GA algorithms were coupled with the ANN model to optimize the values of weights and biases in the ANN model. Historical rainfall and discharge time series were used for training and testing the hybrid models (ANN-PSO and ANN-GA). Additionally, the performance of ANN-PSO was compared with that of ANN-GA under six scenarios, which were defined based on input combinations of different delays of days of rainfall and streamflow.

2. Materials and Methods

This section presents the theory and operation of the PSO algorithm, the GA algorithm, and the ANN model. Next, the weight training process of the proposed hybrid ANN-PSO and ANN-GA are illustrated. Additionally, six applied scenario of model inputs, the selected case study (Nerang River Basin) and the model performance evaluation criteria are outlined at the end of current section.

2.1. Particle Swarm Optimization (PSO)

PSO is a population-based stochastic optimization technique inspired by the social behaviour of bird flocks and fish schools (Silva et al. 2010). This algorithm is initialized with random particles (solutions), which are specified with random positions and velocities. The algorithm searches the problem space for optima through a series of iterations. During the iterations, the particles search for potential solutions and learn over time based on their own experiences and other particles' experiences. The movement of each particle is guided by its best fitness position, which is called personal best (pbest). The overall best value of all particles is also guided by the best-known position of the whole population, which is called global best (gbest). During each iteration, the velocity and position of each particle are updated regarding both pbest and gbest values. This update process is performed by estimating a new velocity term for each particle in relation to each particle distance from pbest and gbest (Knight et al. 2015). The new position of the *i* th particle is calculated by

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{1}$$

where $X_i(t+1)$; $X_i(t)$ = the positions for the *i* th particle at time (*t*+1) and *t*, respectively; and $V_i(t+1)$ = the new velocity for the *i* th particle at time (*t*+1). The new velocity is determined by

$$V_i(t+1) = w \times V_i(t) + C_1 \times r \times \left(X_i^{pbest} - X_i(t)\right) + C_2 \times r \times (X_i^{gbest} - X_i(t)) \tag{2}$$

where $V_i(t)$ = the current velocity for the *i* th particle at time *t*; *w* = the inertia weight; C_1; C_2= the weighting coefficients for the personal best and global best positions, respectively; X_i^{pbest}= the *i* th particle's best known position; X_i^{gbest}= the best position known to the swarm; and *r* = a random number between 0 and 1. C_1 and C_2 should be specified properly to obtain optimal solution. The flowchart for basic PSO algorithm is shown in Figure 1.

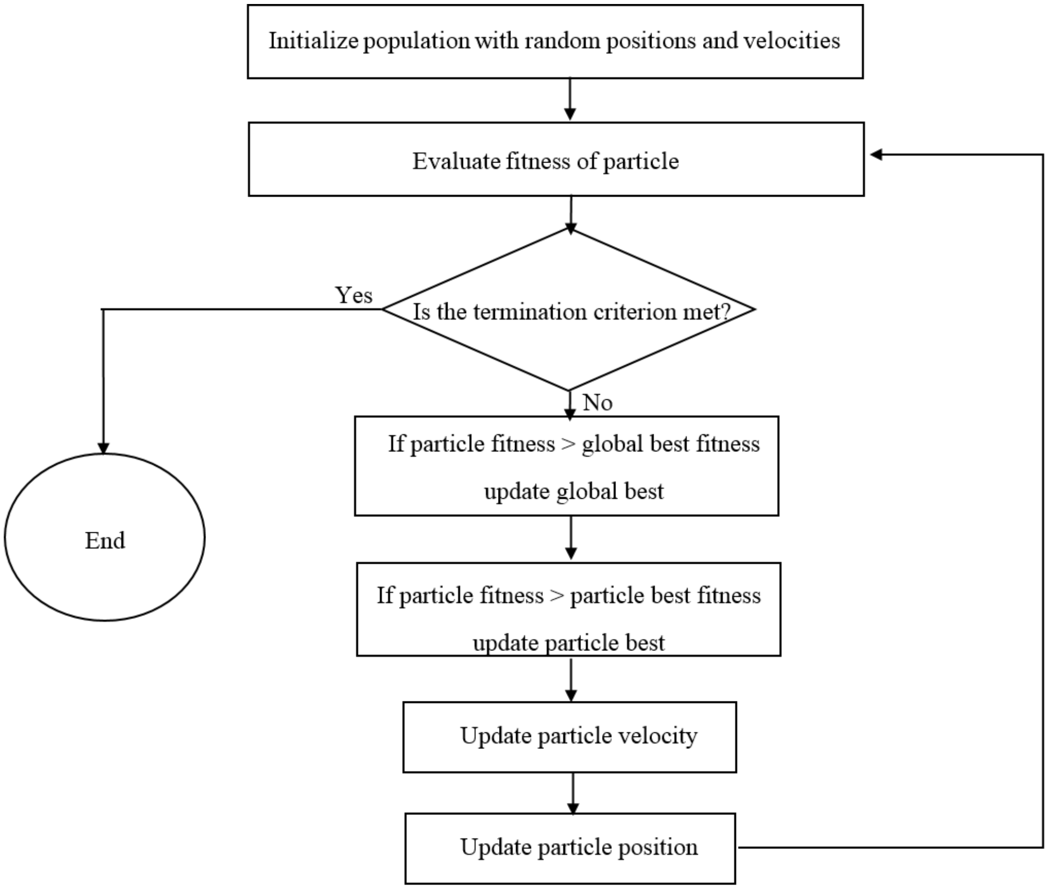
Initialize population with random positions and velocities
Evaluate fitness of particle
Is the termination criterion met?
Yes
No
End
If particle fitness > global best fitness update global best
If particle fitness > particle best fitness update particle best
Update particle velocity
Update particle position

Figure 1. Flowchart of basic PSO algorithm with all steps involved from beginning until termination conditions met

2.2. Genetic Algorithm (GA)

GA is a population-based heuristic, stochastic optimization algorithm developed by Holland (1975). This algorithm is inspired by the concept of Darwinian natural selection and natural genetics, which tends to mimic processes observed in natural evolution (Abe et al. 2004; Chang 2008). GA specifies the decision variables of an optimization problem to an artificial chromosome, where decision variables (genes) are joined together to form a string of values (chromosomes) (Parasuraman and Elshorbagy 2007). A collection of chromosomes is named a population, which is defined as one of the potential solutions. The fitness of each chromosome is based on the object function measurements (Chang 2008). As shown in Figure 2, GA initializes with random populations and progresses based on three basic operators (selection, crossover/recombination, and mutation) to improve the evaluation of solutions through iterations (generations). The selection process is defined as implementing chromosomes with higher fitness value for survival or generating offspring in the gene pool of the next generation. Crossover operates between pairs of chromosomes through exchanging corresponding parts of their gene sequences across a randomly selected point. Mutation is carried out through random alteration of the chromosome (Parasuraman and Elshorbagy 2007).

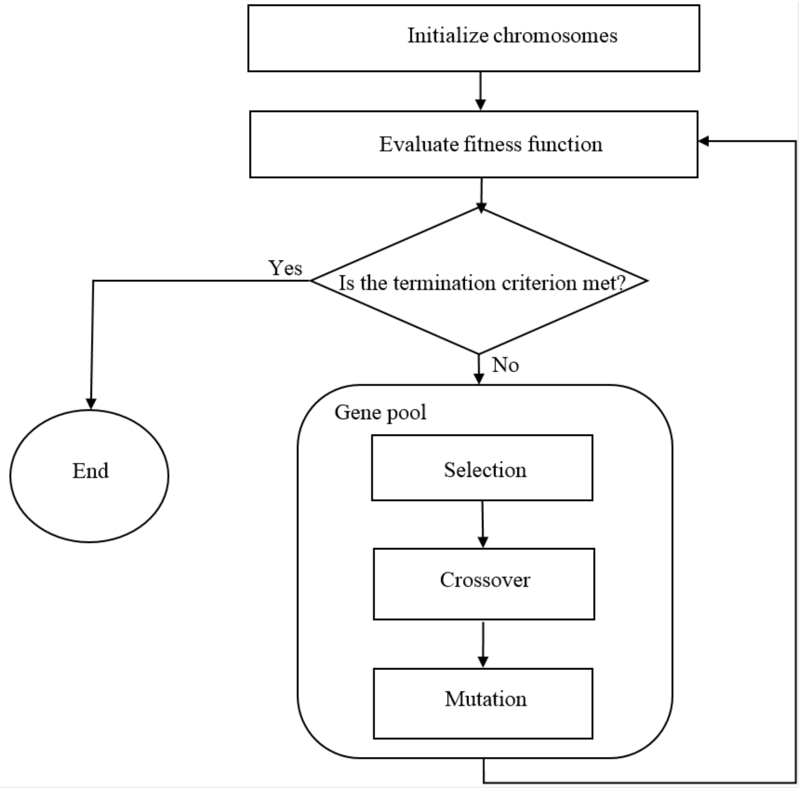

Figure 2. Flowchart of basic GA with all steps involved from beginning until termination conditions met

2.3. Artificial Neural Network (ANN)

2.3.1. Introduction to ANN

The adopted ANN in the current study is based on the structure of the multilayer perception (MLP). The MLP normally consists of three layers: input layer, hidden layer, and output layer. Each layer can include various numbers of neurons (nodes) which link to the neurons in the next layer with the connections called weights (*w*). The inputs of the first layer (input layer) are called external inputs, symbolized by *X* (Fig. 3). Each neuron receives an array of inputs and yields an output. As illustrated in Figure 3, the output of the input layer will become the input of the neurons in the next layer (hidden layer). Similarly, the output of the neurons in the hidden layer will become the input of the neurons for the next layer (output layer). Finally, the output layer indicates the simulated results.

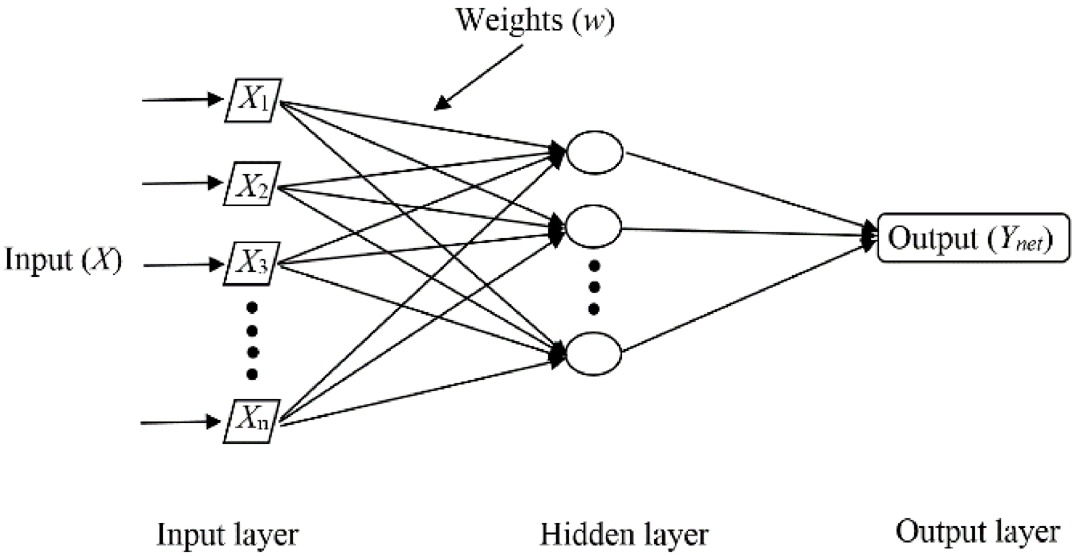

Figure 3. Layout of the MLP network with one hidden layer. The circles represent neurons

The output of each layer is determined by Equation 3:

$$Y_{net} = \sum_{i=1}^{N} Y_i \, w_i + b_i \tag{3}$$

where N = the total number of neurons of a layer; Y_{net}= the output of a layer; Y_i= the input of a layer; w_i= the weights between neurons; and b_i= the bias. In summary, the MLP neural network constitutes a nonlinear parameterized map from input vector to output vector. The parameters, weights, and biases should be adjusted such that the difference between network output and observed output is minimized.

2.3.2. Artificial Neural Network – Particle Swarm Optimization (ANN-PSO)

To improve the convergence rate and learning speed, PSO was coupled with the ANN to optimize the weights and biases of the MLP neural network. A set of weights and biases of the ANN build the position of each particle in PSO. Therefore, the dimension of each particle indicates the total number of weights and biases.

Figure 4 summarizes the steps of ANN-PSO. The training process of ANN-PSO starts with the creation of random particles, which are characterize by random positions (weight and bias). At this stage, the training error (fitness function) should be computed. In the current study, the training error is calculated based on the mean squared error (*MSE*) between the observed and simulated streamflow. As suggested by Christiansen et al. (2014), large differences between observed and modelled runoff values are given more importance using *MSE*, and it is more suitable for accurate peak flow predictions. Concerning the high complex hydrological process over extreme events (peak flow conditions), *MSE* has been selected for error estimation in the current study. The *MSE* is determined by equation 4 as follows:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(Q_{o(t)} - Q_{s(t)})^2 \tag{4}$$

where $Q_{o(t)}$= observed streamflow at the *t* th day; $Q_{s(t)}$ = simulated streamflow at the *t* th day; and *n* = total number of days.

After calculating the training error, the weights and biases (positions and velocity of particles) of the ANN will be updated for the next iteration. This update is performed to reduce the *MSE* at each iteration. This process should be continued until the termination criterion is met. After completion of the model, a set of optimum weights and biases will be generated and introduced to the ANN model. The ANN will apply the optimum weights and biases to simulate the outputs.

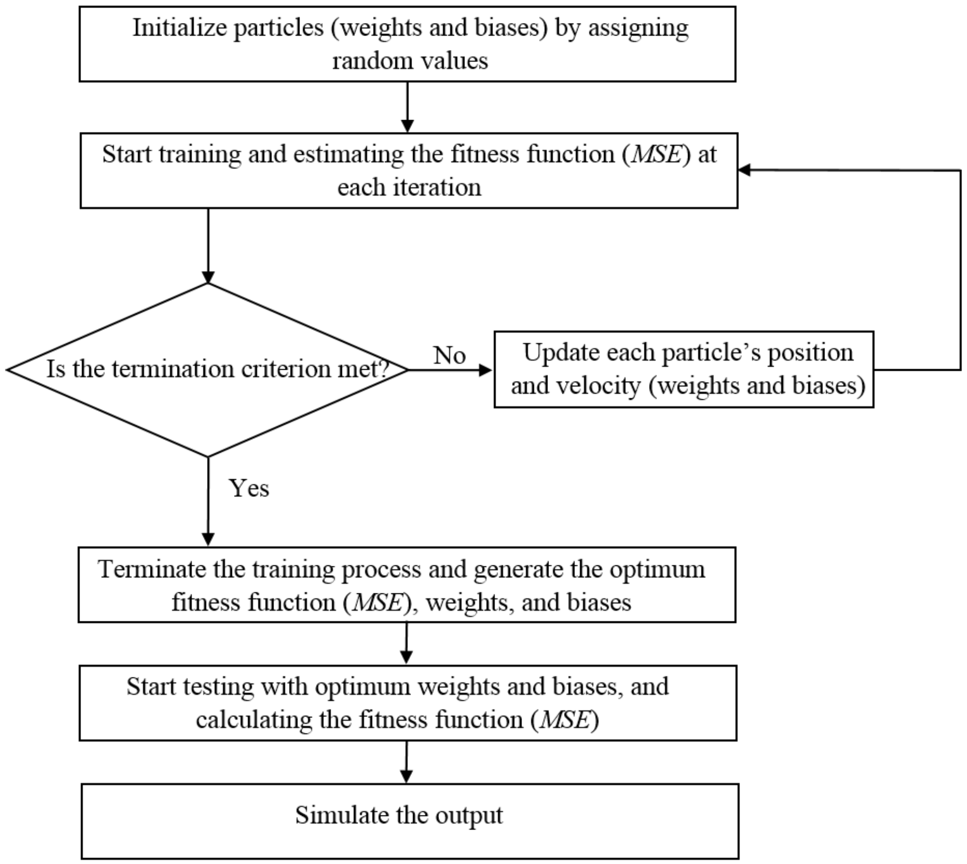
Initialize particles (weights and biases) by assigning random values
Start training and estimating the fitness function (MSE) at each iteration
Is the termination criterion met?
No
Update each particle's position and velocity (weights and biases)
Yes
Terminate the training process and generate the optimum fitness function (MSE), weights, and biases
Start testing with optimum weights and biases, and calculating the fitness function (MSE)
Simulate the output

Figure 4. Flowchart of the ANN-PSO training process with all steps involved from beginning until termination conditions met

2.3.3. Artificial Neural Network – Genetic Algorithm (ANN-GA)

In order to increase solution stability and improve the performance of ANN, such as convergence rate and computational time, the GA was combined with the ANN for weight training process. Similar to the coupled ANN-PSO, the purpose of coupling the GA with the ANN is to optimize the weights and biases of the MLP neural network. As can be seen in Figure 5, the hybrid ANN-GA is initialized through generation of random chromosomes (weights and biases). Next, the fitness function (*MSE*) is evaluated using equation 4. If the termination criterion is met, the training process is stopped, and optimum values of weights and biases are obtained. Otherwise, the chromosomes of the initial population are updated through selection, crossover, and mutation operators. As a result of such update, a new generation (iteration) of chromosomes is built, which feeds the ANN model to estimate the fitness function over the second iteration. This process is carried out until the termination criterion is satisfied in the proposed the ANN-GA model.

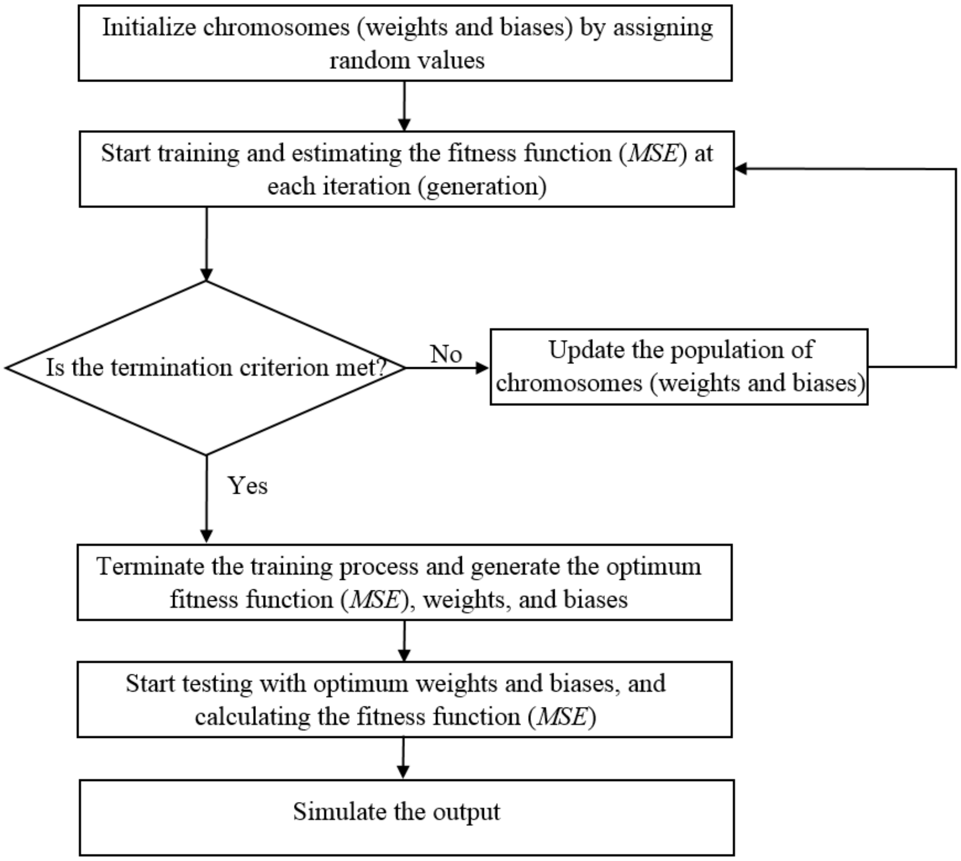
Initialize chromosomes (weights and biases) by assigning random values
Start training and estimating the fitness function (MSE) at each iteration (generation)
Is the termination criterion met?
No
Update the population of chromosomes (weights and biases)
Yes
Terminate the training process and generate the optimum fitness function (MSE), weights, and biases
Start testing with optimum weights and biases, and calculating the fitness function (MSE)
Simulate the output

Figure 5. Flowchart of the ANN-GA training process with all steps involved from beginning until termination conditions met

2.4. Scenario Definition for Model Input

In order to compare the performance of the hybrid ANN-PSO and ANN-GA models for streamflow forecasting under different input data, six model input scenarios were assumed. The stated scenarios were defined based on combinations of rainfall and streamflow with various lag times (up to three-day delay) as inputs for the models. Additionally, applying different combinations of lagged rainfall and streamflow is beneficial to identify the sensitivity of the studied river to hydrological conditions of the previous days. In others words, the assumed scenarios clarify the correlation between the current streamflow and previous days streamflow and rainfall conditions, which determines the best input combination, which yields the most accurate predictions. Table 1 presents the six applied scenarios in the current study. The input data of the six stated scenarios consists of lagged rainfall R_{t-1}, R_{t-2},R_{t-3}, and lagged streamflow, Q_{t-1}, Q_{t-2}, Q_{t-3}, where t indicates the current day (time instant). Therefore, R_{t-1}, R_{t-2},R_{t-3}, denote the rainfall of one previous day, two previous days, and three previous days, respectively. Similarly, Q_{t-1}, Q_{t-2}, Q_{t-3} denote the streamflow of one previous day, two previous days, and three previous days, respectively. The output of all the six scenarios (model inputs) is the streamflow for the current day, Q_t.

Table 1. Rainfall (*R*) and streamflow (*Q*) input combinations for the six scenarios considered in this study. The subscript *t* denotes the time interval (day) and the subscript numbers represent the time lags

Scenario	Input combination
1	R_{t-1}, Q_{t-1}
2	R_{t-1}, R_{t-2}, Q_{t-1}
3	R_{t-1}, Q_{t-1} Q_{t-2}
4	R_{t-1}, R_{t-2}, Q_{t-1}, Q_{t-2}
5	R_{t-1}, R_{t-2},R_{t-3}, Q_{t-1}, Q_{t-2}
6	R_{t-1}, R_{t-2},R_{t-3}, Q_{t-1}, Q_{t-2}, Q_{t-3}

As can be seen in Table 1, Q_t and R_t are not applied as input as the model is developed for the purpose of predicting streamflow from the rainfall and streamflow data of previous days.

Before conducting the ANN-PSO and ANN-GA computations, the rainfall and streamflow data needs to be normalized to enable the model to capture low, medium, and high flow regimes. Over the process of normalization, the rainfall and streamflow data are transformed into the range of -1 and +1. The equation of the normalization method is as follows:

$$x_n = \left(\frac{x - x_{min}}{x_{max} - x_{min}}\right) \times 2 - 1 \tag{8}$$

where x_n= normalized value; x = original value; x_{min}= minimum value; and x_{max}= maximum value.

2.5. Case Study and Data

As shown in Figure 6, the case study was developed for the Nerang River Basin, with the outlet at the Glenhurst gauging station. The Nerang River Basin, a sub-basin of the South Coast Basin, is located in Southeast Queensland, Australia, between latitude 27° 58′ 30″ S - 28° 16′ 30″ S and longitude 153° 9′ 0″ E - 153° 30′ 0″, and has a total area of approximately 493 *km*^{2}, and the delineated Nerang River Basin for the current research is 250 *km*^{2}, as shown in Figure 6.

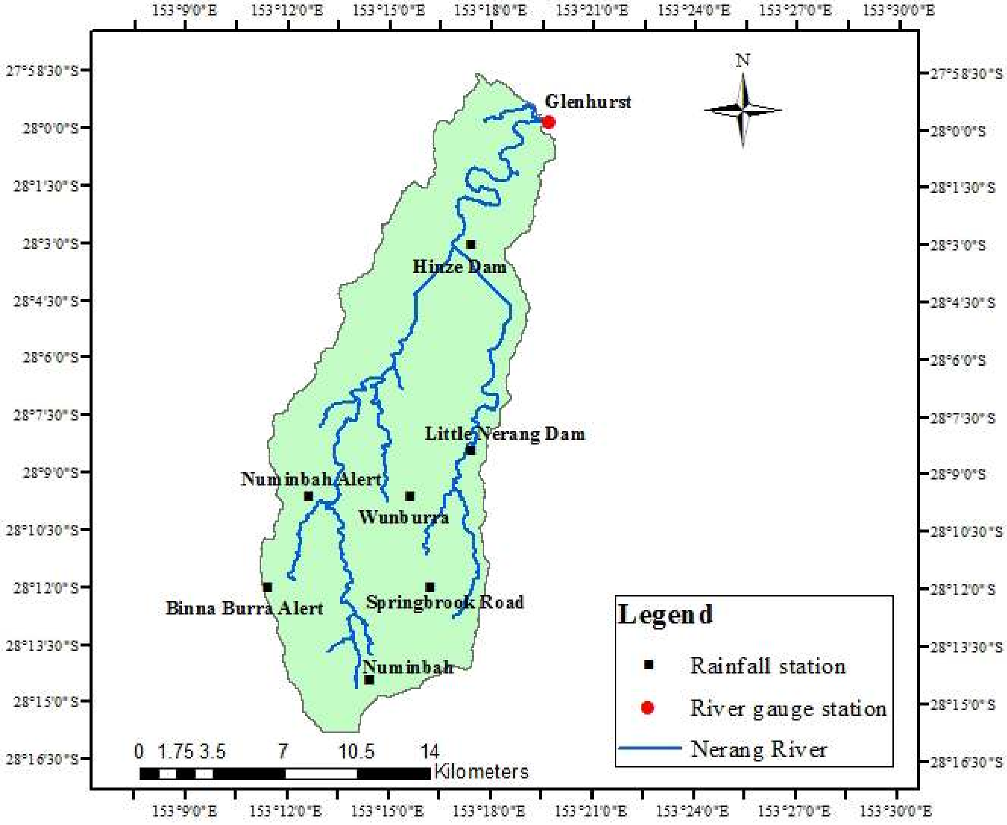

Figure 6. Map of the Nerang River Basin, showing the Nerang River, seven rainfall stations, and the basin outlet (Glenhurst river gauge station)

In order to calibrate and validate the model, daily time series of rainfall and discharge for the period 2005 to 2018 were obtained. 70% and 30% of the data was randomly selected and applied for both proposed models (ANN-PSO and ANN-GA) training (calibration) and testing (validation), respectively.

The observed daily discharges of the Glenhurst gauging station (153° 18′ 36.1″E, 27° 59′ 59.9″S) were provided by the Water Monitoring Information Portal (WMIP). This gauging station is located 22.7 km upstream the river mouth, and according to the WMIP records, the long-term (1968-2018) average streamflow (1968-2018) and rainfall (1991-2018) are 2.72 *m^3/s* and 3.39 *mm*, respectively. Additionally, the average evaporation is 4.06 *mm* over 2005-2018. The average monthly discharge and water level at the Glenhurst site over 1968-2018 is indicated in Figure 7. It can be seen that the Nerang River experiences high discharges and water levels from January to June (wet season), and low discharges and water levels from July to December (dry season). The highest average discharge (almost 6 *m^3/s*) and water level (almost 0.7 *m*) occurs in February, whereas September has the lowest average discharge (0.3 *m^3/s*) and water level (0.4 *m*) (Jahandideh-Tehrani et al. 2020).

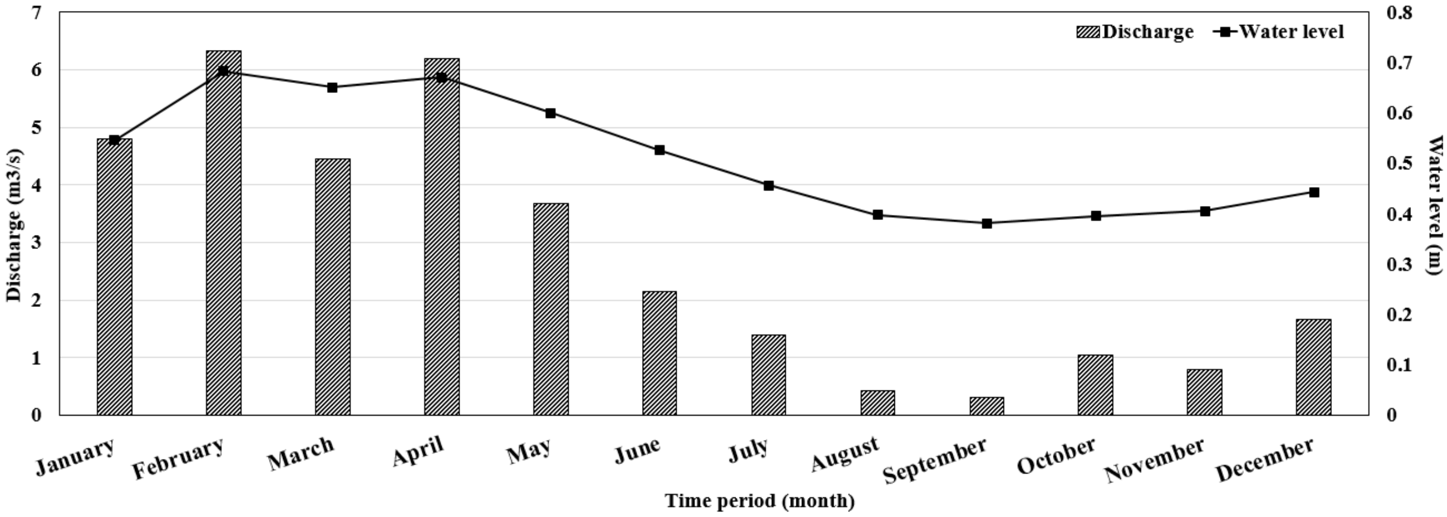

Figure 7. Average monthly discharges and water levels at the Glenhurst site for the period 1968-2018 (obtained from Jahandideh-Tehrani et al. 2020)

The daily rainfall time series of seven gauging stations were selected within the Nerang River Basin, and the data was obtained from the Bureau of Meteorology (BOM). As shown in Figure 6, the gauging stations were Hinze Dam (153.29° E, 28.05° S), Numbinbuh Alert (153.21° E, 28.16° S), Wunburra (153.26° E, 28.16° S), Little Nerang Dam (153.29° E, 28.14° S), Binna Burra Alert (153.19° E, 28.20° S), Sprinbrook Road (153.27° E, 28.20° S), and Numinbah (153.24° E, 28.24° S) with mean long-term rainfall of 3.76 , 3.16, 4.75, 4.45, 4.15, 5.83, and 6.58 *mm*, respectively. The weighted mean long-term rainfall of the entire delineated catchment is 4.40 *mm*. There was some missing rainfall data in the 14-year study period

(2005-2018). The data gaps were resolved using the Arithmetic Mean Method in SPSS software. Next, the weighted average of the daily total rainfall values from the seven rainfall stations was computed using the Thiessen Method. The covered area of each rainfall gauging station was estimated as 81.72 *km*^{2} for Hinze Dam, 35.89 *km*^{2} for Numbinbuh Alert, 27.26 *km*^{2} for Wunburra, 33.73 *km*^{2} for Little Nerang Dam, 16.27 *km*^{2} for Binna Burra Alert, and 29.86 *km*^{2} for Sprinbrook Road.

2.6. Performance Indices

To assess and compare the performance of the proposed models, two commonly used statistical measurements were applied: correlation coefficient (R^2) and Nash-Sutcliffe efficiency coefficient (*NSE*). R^2 values range between zero and one, where zero indicates no linear correlation, and one represents perfect linear relationship between the simulated and observed values. *NSE* indicates the ability of the model in simulating values away from the mean. The larger the value of R^2 and *NSE*, the better the model performance. The R^2 and *NSE* are determined as follows:

$$R^2 = \frac{\frac{1}{N-1}\sum_{i=1}^{n}(S_i - \mu_s)(O_i - \mu_o)}{\sigma_s \sigma_o} \tag{5}$$

$$NSE = 1 - \frac{\sum_{i=1}^{N}(O_{(i)} - S_{(i)})^2}{\sum_{i=1}^{N}(O_{(i)} - \mu o)^2} \tag{6}$$

where N = total number of observations; μ_s and μ_s = average simulated and observed streamflow, respectively; O_i = observed streamflow on i th day; S_i = simulated streamflow on i th day; and σ_s and σ_o = standard deviation of the simulated and observed streamflow, respectively.

3. Results and Discussions

To ensure an accurate prediction of streamflow by both ANN-PSO and ANN-GA models, trial and error was performed to determine the optimal configuration of ANN-PSO and ANN-GA. In order to efficiently compare the performance of PSO and GA in weight training process, similar ANN architecture (three neurons and one hidden layer), initial population size (100), and maximum iteration (500) were implemented. Additionally, over the trial and error process, the optimal PSO was found with the configuration of $C_1 = 1.5$, $C_2 = 2.5$. Additionally, the best configuration of GA was associated with selection probability of 0.15, crossover probability of 0.50, and mutation probability of 0.35.

Considering that the initial population (weight and bias) of both ANN-PSO and ANN-GA models were generated randomly, each of the six stated scenarios were performed three times and the average values of performance indices (R^2 and *NSE*) were computed for comparison. Table 2 presents the estimated performance indicators of the streamflow forecast obtained from the ANN-PSO and ANN-GA models for a different number of delays for day (under six defined scenarios).

Table 2. Performance indicators of the ANN-PSO and ANN-GA models for training and testing under six scenarios

Scenario	Run	Training				Testing			
		R^2		*NSE*		R^2		*NSE*	
		PSO	GA	PSO	GA	PSO	GA	PSO	GA
1	1	0.90	0.89	0.82	0.79	0.84	0.83	0.71	0.68
	2	0.93	0.88	0.86	0.78	0.93	0.84	0.87	0.69
	3	0.93	0.87	0.86	0.72	0.85	0.85	0.71	0.71
	Average	0.92	0.88	0.85	0.76	0.87	0.86	0.76	0.69
2	1	0.94	0.86	0.89	0.73	0.89	0.84	0.79	0.71
	2	0.93	0.88	0.86	0.75	0.86	0.86	0.73	0.72
	3	0.95	0.87	0.91	0.74	0.94	0.84	0.85	0.69
	Average	0.94	0.87	0.89	0.74	0.90	0.85	0.79	0.71
3	1	0.93	0.86	0.87	0.71	0.84	0.85	0.71	0.70
	2	0.94	0.83	0.88	0.67	0.85	0.84	0.71	0.70
	3	0.91	0.90	0.82	0.82	0.95	0.83	0.89	0.71
	Average	0.93	0.86	0.86	0.73	0.88	0.84	0.77	0.70
4	1	0.93	0.84	0.86	0.72	0.88	0.84	0.74	0.72
	2	0.90	0.86	0.81	0.72	0.94	0.84	0.85	0.72
	3	0.92	0.74	0.85	0.71	0.87	0.83	0.76	0.70
	Average	0.92	0.85	0.84	0.72	0.90	0.84	0.78	0.71
5	1	0.91	0.86	0.83	0.72	0.87	0.73	0.74	0.53
	2	0.89	0.81	0.79	0.64	0.93	0.79	0.79	0.62
	3	0.91	0.85	0.83	0.71	0.90	0.85	0.80	0.69
	Average	0.90	0.84	0.82	0.69	0.90	0.79	0.78	0.61
6	1	0.89	0.87	0.80	0.75	0.94	0.82	0.81	0.61
	2	0.92	0.80	0.85	0.64	0.88	0.83	0.77	0.69
	3	0.94	0.82	0.88	0.66	0.85	0.81	0.72	0.65
	Average	0.92	0.83	0.84	0.68	0.89	0.82	0.77	0.65

According to Table 2, ANN-PSO outperformed the ANN-GA model over both training and testing periods under all six stated scenarios by showing higher average performance indicators (R^2 and NSE). Therefore, it is determined that ANN-PSO performed more satisfactory in rainfall-runoff modelling process and streamflow prediction.

Considering the ANN-PSO model, scenario 2 outperformed the other scenarios by showing the best model performance indicators ($R^2 = 0.94$ and $NSE = 0.89$ for the training period, and $R^2 = 0.90$ and $NSE = 0.79$ for the testing period). Therefore, it is determined that in the studied basin, the river flow of the Nerang River was more sensitive to the rainfall of the two previous days and streamflow of the previous day concerning the ANN-PSO model. Additionally, the rainfall and streamflow with 3-day lag time had less impact on the Nerang River flow. Therefore, the input combination of R_{t-1}, R_{t-2}, and Q_{t-1} yielded the most accurate streamflow predictions in terms of using the ANN-PSO model. It can also be observed that scenario 3 indicated high performance indices ($R^2 = 0.93$ and $NSE = 0.86$ for the training period, and $R^2 = 0.88$ and $NSE = 0.77$ for the testing period), which placed this scenario in the second position of application. The input combinations of scenario 3 are R_{t-1}, Q_{t-1}, and Q_{t-2}, which confirmed the statement that the Nerang River flow was more highly correlated with the rainfall and streamflow conditions of the two previous days. Scenario 4 also performed with the combination of a 2-day lagged rainfall and streamflow combination. However, the computed performance indices were slightly smaller than the ones in scenarios 2 and 3.

As shown in Table 2, the ANN-GA model performed almost similarly in rainfall-runoff modelling under scenarios 1, 2, 3, and 4. The reason behind the slightly lower performance for scenarios 5 ($R^2 = 0.84$ and $NSE = 0.69$ for the training period, and $R^2 = 0.79$ and $NSE = 0.61$ for the testing period) and 6 ($R^2 = 0.83$ and $NSE = 0.68$ for the training period, and $R^2 = 0.82$ and $NSE = 0.65$ for the testing period) is that there was slightly less agreement between observed and simulated streamflow compared to scenarios 1, 2, 3, and 4. Hence, it can be concluded that the streamflow of the current day is less correlated with 3-day lagged time of rainfall and streamflow in the studied basin. Additionally, too many number of the network inputs generate a network system with more equations than free variables, which leads to overspecification of a network system. As a result, the network indicates poor simulation performance in capturing the existing trends in data (Asadnia et al. 2014).

In terms of convergence rate, Figure 8 shows a comparison of the fitness functions for the ANN-PSO and ANN-GA models over 500 iterations. This comparison was made for scenarios 1, 2, and 3 over the training period (graphs a, b and c, respectively). It can be seen that the convergence speed of ANN-PSO is greater than the speed of ANN-GA in all three scenarios. Over 500 iterations, the ANN-PSO model yielded lower *MSE* values (reaching a minimum of 0.0011 and 0.0009 m^6/s^2 for scenarios 1 and 3, respectively) compared to those of ANN-GA (which reached a minimum of 0.0013 and 0.0012 m^6/s^2 for scenarios 1 and 3, respectively). For scenario 2, both models yielded *MSE* values as low as 0.0009 m^6/s^2, but the ANN-PSO model was faster than the ANN-GA model in reaching this value. Additionally, as indicated in Figure 8, the optimal solution obtained by ANN-PSO in the initial stage was significantly better than that of ANN-GA, confirming that fewer iterations are required for ANN-PSO to reach an optimal solution. For instance, for scenario 2, ANN-PSO yielded an *MSE* value equal to 0.0009 m^6/s^2 after 25 iterations, whereas the ANN-GA needed 436 iterations to reach this value. Thus, in addition to the increased accuracy shown by the ANN-PSO model as compared to the ANN-GA model, the former was also significantly faster than the latter. Thus, it was concluded that the speed, efficiency, and accuracy in convergence are greater for ANN-PSO.

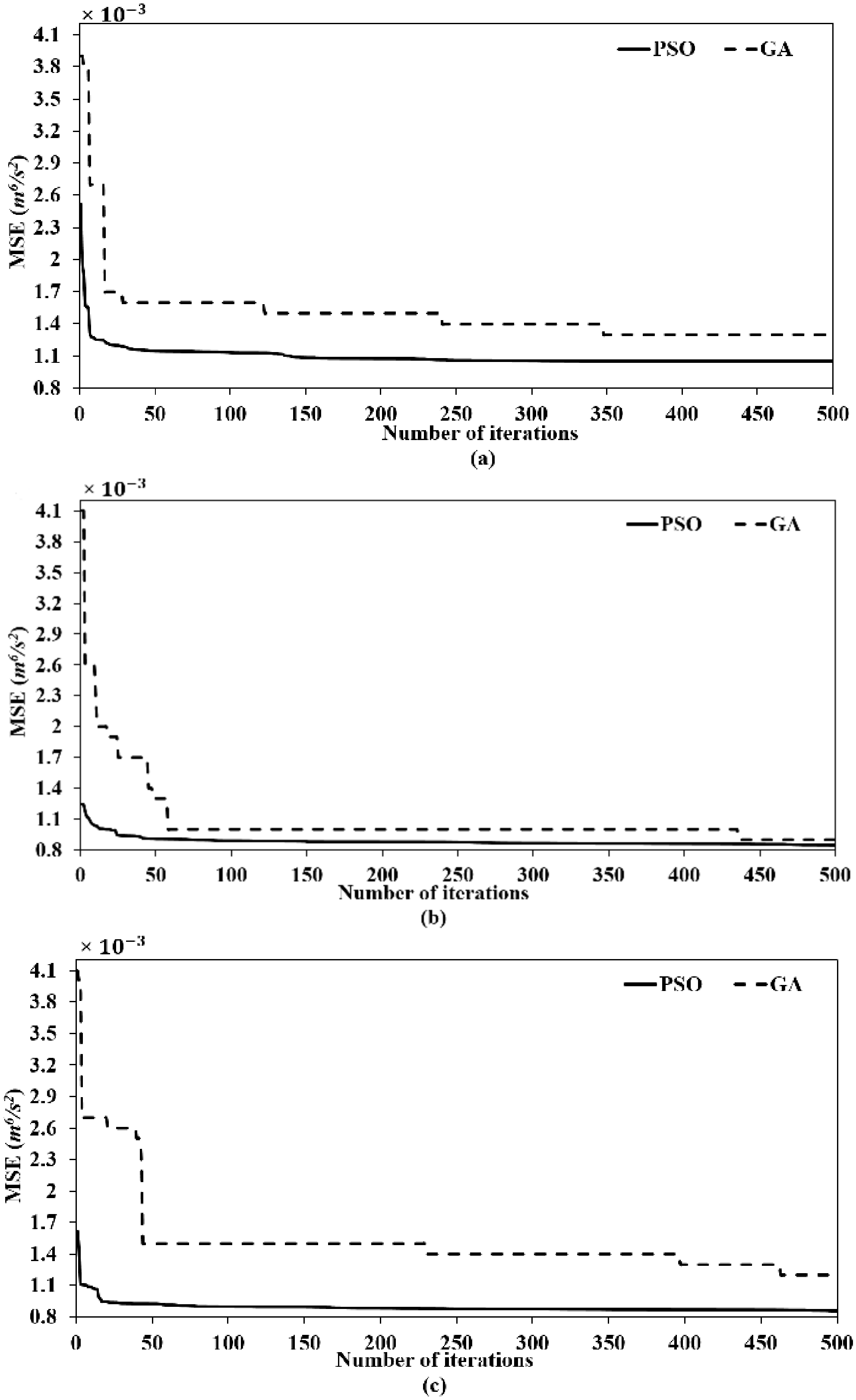

Figure 8. Comparison of the fitness function (*MSE*) plots of ANN-PSO and ANN-GA for (a) scenario 1; (b) scenario 2; and (c) scenario 3 over the training period

Figure 9 represents the correlation between observed and predicted daily flows achieved with the ANN-PSO (graphs a, b and c) and ANN-GA (graphs d, e and f) models for the best runs of scenarios 1, 2, and 3 over the testing period. Figure 9 clearly highlights the superiority of ANN-PSO over ANN-GA as there is stronger correlation between observed and predicted streamflows in all scenarios for this model. Figures 9 (a), 9 (b), and 9 (c) clearly shows that almost all points are located close to the 1:1 line, which confirms that the predicted streamflow generally agrees well with the observed streamflow in ANN-PSO modelling. However, it can be seen that a few points are placed below the 1:1 line, particularly in cases of high streamflow values, indicating that the ANN-PSO model sometimes underestimates the values of high streamflow (Zhang et al. 2018). This could be explained by the fact that the rainfall-runoff relationship becomes more complex in case of high (peak) flow conditions. Additionally, considering that peak flow events are scarce compared to normal flow condition, fewer data (extreme rainfall and streamflow) are available to feed the ANN-PSO and ANN-GA (data-driven) models. As a result, fewer extreme data are available to train the data-driven models, which leads to less accuracy in extreme streamflow prediction.

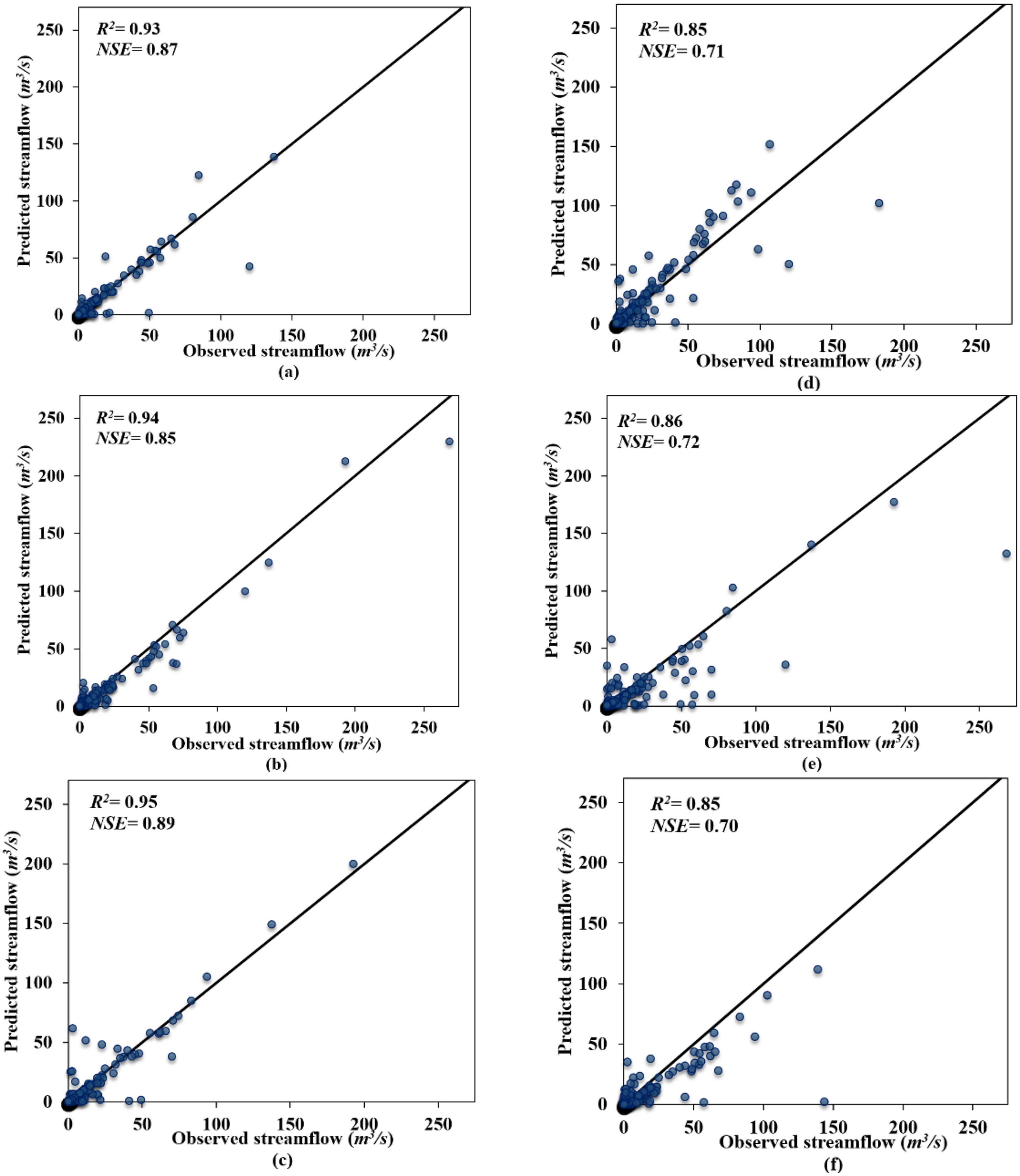

Figure 9. Performance comparison of (a) scenario 1; (b) scenario 2; and (c) scenario 3 for the best runs of ANN-PSO and (d) scenario 1; (e) scenario 2; and (f) scenario 3 for the best runs of ANN-GA

4. Conclusions

This study addressed the complex process of rainfall-runoff modelling as well as real-time streamflow prediction using two data-driven techniques. To obtain an efficient rainfall-runoff model and to improve the accuracy of real-time streamflow prediction, the PSO and GA algorithms were coupled with the ANN technique separately. The performance and efficiency of two proposed hybrid models, ANN-PSO and ANN-GA, were compared to determine the best model in terms of efficiency, convergence speed, and accuracy. Additionally, sensitivity of the proposed models to days of delay for rainfall and streamflow were evaluated through building six input combinations (scenarios), and comparing the performance indices (R^2 and *NSE*) of the modelling results. Considering that the initial weights and biases of both the ANN-PSO and ANN-GA models were created randomly, each model was run three times under six scenarios, and the average values of the performance indicators were computed to compare the six scenarios, and to select the input combination that yielded the most accurate streamflow predictions.

The proposed ANN-PSO and ANN-GA models were applied for real-time streamflow forecasting of the Nerang River Basin, Southeast Queensland, Australia. Input data consisted of 1) the weighted rainfall of seven gauging stations, which was computed using the Thiessen method, and 2) discharge of the Glenhurst station (at the outlet of the basin). The results of the current study revealed that both proposed models achieved satisfactory agreement between the observed and predicted streamflow for both low and high flow conditions. Additionally, the comparison between ANN-PSO and ANN-GA demonstrated that the use of PSO as an optimization tool for weight training process can result in higher convergence speed, accuracy, and performance indicators. Additionally, the results of input sensitivity assessment revealed that with regard to the ANN-PSO model, the input combination of R_{t-1}, R_{t-2}, and Q_{t-1} yielded the most accurate streamflow predictions (R^2 = 0.94 and *NSE* = 0.89, over the training period, and R^2 = 0.90 and *NSE* = 0.79 over the testing period) as compared to the other input combinations. Considering the ANN-GA model, scenarios 5 and 6 showed the least satisfactory performance, meaning that the rainfall and streamflow with

3-day lag time had less impact on the streamflow of the studied basin. Hence, the Nerang River flow is more dependent on the rainfall of the two previous days and streamflow of the previous day.

5. Limitations and Future Directions

The results of the current study demonstrated the benefits of applying the coupled ANN-PSO and ANN-GA techniques for rainfall-runoff modelling, and confirmed the ability of these tools for real-time streamflow forecasting, however, few limitations of the current study should be addressed in future research. It is recommended to investigate the capability of other evolutionary algorithms in optimizing the ANN parameters. Additionally, the current research limited to the Nerang River Basin area. Therefore, in terms of future research directions, it is recommended to validate the general ability of proposed tools (ANN-PSO and ANN-GA) in other catchments in Australia, and particularly in Southeast Queensland region. It is also recommended to apply proposed model to streamflow prediction using evaporation input data to provide more scenarios of model input combinations as the current research was limited to rainfall and streamflow input combinations, while evaporation characteristics of the basin can also affect streamflow prediction. Moreover, the sensitivity of the model to the number of neurons, initial swarm size, initial chromosomes, number of iterations, and number of hidden layers can also be studied in potential future research.

6. Acknowledgement

Funding for this project has been provided by Griffith University Postgraduate Research School through the GUPRS scholarship, and Griffith University International Postgraduate Research School through the GUIPRS scholarship. The authors would also like to acknowledge the support of the Water Monitoring Information Portal (WMIP), Queensland Government (Australia) in their provision of the streamflow data, and the Bureau of Meteorology (Australia) for providing the rainfall data.

7. References

Asadnia M, Chua LHC, Qin XS, Talei A (2014) Improved particle swarm optimization-based artificial neural network for rainfall-runoff modelling. J. Hydrol. Eng. 19:1320-1329

Ali Z, Hussain I, Faisal M, Nazir HM, Hussain T, Shad MY, Shoukry AM, Gani SH (2017). Forecasting drought using multilayer perceptron artificial neural network model. Adv. Meteorol. https://doi.org/10.1155/2017/5681308

Aziz K, Haque MM, Rahman A, Shamseldin AY, Shoaib M (2017) Flood estimation in ungauged catchments: Application of artificial intelligence based methods for Eastern Australia, Stoch. Environ. Res. Risk Assess. 31:1499-1514

Abe A, Kamegawa T, Nakajima Y (2004) Optimization of construction of tire reinforcement by genetic algorithm. Optim. Eng. 5:77-92

Aziz K, Rahman A, Fang G, Shrestha S (2013) Application of artificial neural networks in regional flood frequency analysis: A case study for Australia. Stoch. Environ. Res. Risk Assess. 28:541-554

Ba H, Guo S, Wang Y, Hong X, Zhong Y, Lio Z (2017) Improving ANN model in runoff forecasting by adding soil moisture input and using data preprocessing techniques. Hydrol. Res. 49:744-760

Chang LC (2008) Guiding rational reservoir flood operation using penalty-type genetic algorithm. J. Hydrol. 354:65-74

Chitsaz N, Azarnivand A, and Araghinejad S (2016) Pre-processing of data-driven river flow forecasting models by singular value decomposition (SVD) technique. Hydrolog. Sci. J. 61:2164-2178

Carcano EC, Bartolini P, Museli M, Piroddi L (2008) Jordan recurrent neural network versus IHACRES in modelling daily streamflow. J. Hydrol. 362:291-307

Chiang YM, Chang LC, and Chang FJ (2004) Comparison of static-feedforward and dynamic-feedback neural networks for rainfall–runoff modeling. J. Hydrol. 290:297-311

Chandwani V, Vyas SK, Agrawal V, Sharma G (2015) Soft computing approach for rainfall-runoff modelling: A review. Aquat. Procedia, 4:1054-1061

Christiansen NH, Voie PET, Winther O, and Hogsberg J (2014) Comparison of Neural Network Error Measures for Simulation of Slender Marine Structures. J. Appl. Math. https://doi.org/10.1155/2014/759834

De Vos, NJ, Rientjes, THM (2008) Multiobjective training of artificial neural networks for rainfall-runoff modeling. Water Resour. Res., 44:1-15

Elshorbagy A, Corzo G, Srinvasulu S, Solomatine DP (2010) Experimental investigation of the predictive capabilities of data driven modelling techniques in hydrology. Part 2. Application. Hydrol. Earth. Syst. Sci. 14:1943-1961

Gholami V, Booij, MJ, Nikzad Tehrani E, Hadian MA (2018) Spatial soil erosion estimation using an artificial neural network (ANN) and field plot data. Catena. 163: 210-218

Holland JH (1975) Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press.

Hosseini-Moghari SM, Araghinejad S, and Azarnivand, A (2017) Drought forecasting using data-driven methods and an evolutionary algorithm. Model Earth Syst. Environ. 3:1675-1689

Imrie CE, Durucan S, Korre A (2000) River flow predication using artificial neural networks: Generalization beyond the calibration range. J. Hydrol. 233:138-153

Jahandideh-Tehrani M, Bozorg-Haddad O, A. Loáiciga H (2019) Application of non-animal–inspired evolutionary algorithms to reservoir operation: an overview. Environ. Monit. Assess. https://doi.org/10.1007/s10661-019-7581-2

Jahandideh-Tehrani M, Helfer F, Zhang H, Jenkins G, and Yu Y (2020) Hydrodynamic modelling of a flood-prone tidal river using the 1D model MIKE HYDRO River: calibration and sensitivity analysis. Environ. Monit. Assess. https://doi.org/10.1007/s10661-019-8049-0

Jain A, Srinivasulu S (2004) Development of effective and efficient rainfall–runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network techniques. Water Resour. Res. https://doi.org/10.1029/2003WR002355

Kasiviswanathan KS, Cibin R, Sudheer KP, Chaubey I (2013) Constructing prediction interval for artificial neural network rainfall runoff models based on ensemble simulations. J. Hydrol. 499:275-288

Kennedy J, Eberhart RC (1995) Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, Perth, WA, Australia, Australia, November 27-December 1, pp 1942-1948

Kuok KK, Harun S, Shamsuddin SM (2010) Particle swarm optimization feedforward neural network for modelling runoff. Int. J. Environ. Sci. Tech. 7:67-78

Knight JT, Singer DJ, Collette MD (2015) Testing of a spreading mechanism to promote diversity in multi-objective particle swarm optimization. Optim. Eng. 16:279-302

Lee S, Lee KK, Yoon H (2019) Using artificial neural network models for groundwater level forecasting and assessment of the relative impacts of influencing factors. Hydrogeol. J. 27:567-579

Moeeni H, Bonakdari H, Fatemi SH, Zaji AH (2017) Assessment of stochastic models and a hybrid artificial neural network-genetic algorithm method in forecasting monthly reservoir inflow. INAE Lett. 2:13-23

Machado F, Mine M, Kaviski E, Fill H (2011) Monthly rainfall-runoff modelling using artificial neural networks. Hydrol. Sci. J. 56:349-361

Nourani V, Baghanam AH, Adamowski J, Kisi O (2014) Applications of hybrid wavelet–artificial Intelligence models in hydrology: a review. J Hydrol. 514:358–377

Napolinato G, See L, Calvo B, Savi F, Heppenstall AA (2010) A conceptual and neural network model for real-time flood forecasting of the Tiber River in Rome. Phys. Chem. Earth, 35:187-194

Parasuraman K, Elshorbagy A (2007) Cluster-based hydrologic prediction using genetic algorithm-trained neural networks. J. Hydrol. Eng. 12:52-62

Piotrowski AP, Napiorkowski JJ (2011) Optimizing neural Networks for river flow forecasting – evolutionary computation methods versus the Levenberg – Marquardt approach. J. Hydrol. 407:12-27

Roy B, Singh MP (2020) An empirical-based rainfall-runoff modelling using optimization technique. Int. J. River Basin Manag. 18:49-67

Sedki A, Ouazar D, El Mazoudi E (2009) Evolving neural network using real coded genetic algorithm for daily rainfall–runoff forecasting. Expert Syst. Appl. 36:4523-4527

Silva AP, Ravagnani MASS, Biscaia EC, Caballero JA (2010) Optimal heat exchanger network synthesis using particle swarm optimization. Optim. Eng. 11:459-470

Taormina R, Chau KW, and Sethi R (2012). Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon. Eng. Appl. Artif. Intell. 25:1670-1676

Tokar AS, Johnson PA (1999) Rainfall-runoff modelling using artificial neural networks. J. Hydro. Eng. 4:223-239

Wang J, Shi P, Jiang P, Hu J, Qu S, Chen X, Chen Y, Dai Y, Xiao Z (2017) Application of BP neural network algorithm in traditional hydrological model for flood forecasting. Water. https://doi.org/10.3390/w9010048

Zhang C, Shao H, and Li Y (2000) Particle swarm optimization for evolving artificial neural network, Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, Nashville, TN, USA, October 8-11, pp 2487-2490

Zhang Z, Zhang Q, Singh VP, and Shi P (2018) River flow modelling: comparison of performance and evaluation of uncertainty using data-driven models and conceptual hydrological model. Stoch. Env. Res. Risk A. 32:2667–2682