Rainfall-Runoff Model Based on ANN with LM, BR and PSO as Learning Algorithms

Muhammad Rabani Mohd Romlay, Muhammad Mahbubur Rashid, Siti Fauziah Toha, Azhar Mohd Ibrahim

Abstract**:** ***Rainfall-runoff model requires comprehensive computation as its relation is a complex natural phenomenon. Various inter-related processes are involved with factors such as rainfall intensity, geomorphology, climatic and landscape are all affecting runoff response. In general there is no single rainfall-runoff model that can cater to all flood prediction system with varying topological area. Hence, there is a vital need to have custom-tailored prediction model with specific range of data, type of perimeter and antecedent hour of prediction to meet the necessity of the locality. In an attempt to model a reliable rainfall-runoff system for a flood-prone area in Malaysia, 3 different approach of Artificial Neural Networks (ANN) are modelled based on the data acquired from Sungai Pahang, Pekan. In this paper, the ANN rainfall-runoff models are trained by the Levenberg Marquardt (LM), Bayesian Regularization (BR) and Particle Swarm Optimization (PSO). The performances of the learning algorithms are compared and evaluated based on a 12-hour prediction model. The results demonstrate that LM produces the best model. It outperforms BR and PSO in terms of convergence rate, lowest mean square error (MSE) and optimum coefficeint of correlation. Furthermore, the LM approach are free from overfitting, which is a crucial concern in conventional ANN learning algorithm. Our case study takes the data of rainfall and runoff from the year 2012 to 2014. This is a case study in Pahang river basin, Pekan, Malaysia***.

Index Terms**:** ***Artificial neural network; rainfall-runoff; Levenberg Marquardt; Bayesian Regularization; Particle Swarm Optimization.***

I. INTRODUCTION

The research on earth-related systems have gain more attention as there are developing concern about natural and environmental changes. Important system such as rivers and their discharge give enormous impact on human activities[1], as it directly affects the community placement and sustenance. One of the common solutions of river related problem is to make a prediction model of rainfall-runoff system.

Rainfall-runoff is a perplexing interaction between precipitation and landscape aspects. There are many factors which affect runoff level such as storm characteristics, intensity of rainfall, duration of rainfall, geomorphology and climatic elements [2].

Revised Manuscript Received on September 15, 2019

Muhammad Rabani Mohd Romlay, Pursuing PhD, Mechatronics Engineering, International Islamic University Malaysia (IIUM).

Muhammad Mahbubur Rashid, Associate Professor in the Department of Mechatronics Engineering, IIUM, Kuala Lumpur.

Siti Fauziah Toha, Associate Professor, Department of Mechatronics Engineering, International Islamic University Malaysia (IIUM).

Azhar Mohd Ibrahim, Assistant Professor, International Islamic University Malaysia (IIUM)

Moreover, mitigating water related problem requires high level of accuracy [3]. Therefore, rainfall-runoff systems are strenuous to model [4]–[7]. Following this matter, researchers have opted for machine learning (ML) in an attempt to solve the problem. ML have been making a breakthrough for flood forecasting, providing vigorous and competent models that can efficaciously learn convoluted flood systems in an adaptive or flexible manner [8]. Amongst ML methods, support vector machine (SVM) is hugely popular, with auspicious results and exceptional generalization capability [8]. This is due to its suitability for linear and non-linear classification [9], enabling it to identify global optimum solution in flood models [10]. As promising as it was, SVM do possess some drawbacks. SVM require high computational cost and may provide unrealistic outputs [11]. There is also a flaw in the implementation of SVM in the case of seasonal flow prediction employing least-square support vector machine (LS-SVM) [11]. Another implementation of ML in flood forecasting which is steadily progressing is the decision tree method (DT). Classified as a fast algorithm [12], it has been tested to model flood modelling. Nevertheless, it still has some limitations. There are claims that its fitness to flood prediction is still yet to be seen and fully scrutinized [8].

Another convenient option to model rainfall-runoff is by using artificial neural network (ANN). In recent years, ANN has been tested in diversified forecast models for both hydrological system and non-hydrological system. Inspired by the biological neural network, it has provided solutions for tasks in various fields such as biology [13], machine vision [14]–[16], water flow estimation [17], ozone level prediction[18] and speech recognition [19]. Furthermore, ANN also plays an important part in nowcasting field of research; which is the ability to predict the evolution of geophysical field based from images of remote sensors in short-term scale [21]. Some even considered ANN to be the most suitable modelling technique, with generalization ability being its strength [20]. With faster speed compared to most conventional models, it offers an alternative to SVM which requires a higher computational cost. As it is independent, ANN does not place a relationship between dependent and independent variable. Amongst numerous ML techniques, ANN is deemed to be most suitable to represent modelling system for rainfall- runoff [8].

Nonetheless, the ANN model still leaves a wide range of possibilities as there are numerous learning algorithms which can be used to optimize ANN. Although hugely popular in flood forecasting, there is no clear-cut conclusion proclaimed with regards to which model functions better in a specific application [8].

The capacity of each learning algorithm may differ across distinctive types of task. Difference of model input might be in terms of period of prediction (daily or monthly) or amount of available data (scarce or plentiful). Therefore, there is a vital need to have a custom-tailored prediction model with a specific range of data, type of parameter and hour of prediction according to model's objective and data availability. In this research, we define the prediction model to be short-term (12 hours); specifically, to distinguish possibilities of flash-flood. There is a huge amount of available data for training purposes (5 rainfall stations with 1-hour interval recorded data from 2012 to 2014). Once prediction model criteria have been set up to cater to the specific needs in Sungai Pahang, it is imminent to determine the best learning method to train the ANN for the rainfall-runoff model.

Three distinctive, highly effective learning algorithms are proposed to train ANN which is the Levenberg Marquardt (LM), Bayesian Regularization (BR) and Particle Swarm Optimization (PSO) algorithms. LM algorithm, which was initially introduced by Kenneth Levenberg and Donald Marquardt [22] offers a fast and stable convergence model [23]. It is claimed to be more robust than most learning algorithm, due to its ability to find the solution even if it starts very far off the final minimum [23]. With the advantages and qualities mentioned, ANN trained by LM is considered to be very efficient for rainfall-runoff model [24]. The second learning algorithm to be implied for ANN training is the BR. Bayesian methods are widely used in astronomy, cosmology, and are gaining recognition in other fields [25]. It is a competent learning algorithm for large or deep feed-forward neural network [25]. Furthermore, the BR method offers a supervised learning approach as well as a statistical technique for classification [26]. It has shown promising results in rainfall predictions and other weather forecasting related field [25]–[27], therefore cementing its selection as the training algorithm for our rainfall-runoff prediction in Pekan, Pahang. The third learning algorithm to be evaluated is PSO. Initially developed for modelling social behaviour [28], PSO was fast recognized as an evolutionary technique in computational intelligence [29]. Its development has benefited a broad scope of complicated engineering and science optimization problems [30]. Due to its potential, PSO has been engaged in working out classification problems in many health domains, specifically in heart diseases and breast cancer [31], just to name a few. Additionally, the ANN trained by PSO has been applied in classifying Iris, Cancer, Diabetes, Hepatitis, Henan, and Cubic datasets [31]. Satisfactory results have also been achieved in the hydrological related problem such as river stage forecasting [32]. Hence, PSO is deemed suitable to be implied as the learning algorithm for ANN model of our system.

This study attempts to improve hydrological forecasting by acquiring data assimilation and accommodate the parameter of the neural network with a varying learning algorithm to determine which models' best suit flood-prone Pahang river in Pekan. The paper has been organized in the following manner. Sections 1 explains the introduction of the research, whereas Section 2 describes the Pahang river basin and details of the model and data sources. Section 3 introduces the methodology of the study. In Section 4 results of the neural network training is explained and further discussions are put forth in section5. Finally, the summary of the result and conclusions are discussed in Section 6.

II. STUDY REGION

Sungai Pahang in Pekan, is a flood-prone area with heavy rainfall usually occurs during October until the middle of February. From the year of 2012 until 2014, flood incidents have been recorded in January and December 2012 [33][34], December 2013 [35], and December 2014 [36], with the latest being one of the worst floods in the recent decade within Malaysia. As the relation between rainfall and runoff are too much subjective, it is crucial to have a predictive model of rainfall-runoff to anticipate and mitigate flood occurrence. The Pahang River, which is 459km in length is the longest river in Peninsula Malaysia. It starts at the confluence of Jelai and Tembeling rivers on the Titiwangsa Mountains, flows through Jerantut, Kerdau, Chenor, Lubuk Paku, Pekan and Kuala Pahang before finally channels into the South China Sea. The river crosses almost every district in the state of Pahang.

In Fig. 1, rainfall stations in Pekan district is shown with Sungai Pahang flowing towards the South China Sea. There are two monsoons (the northeast and southwest monsoons) and two inter-monsoon seasons occurring in Peninsular Malaysia. High total rainfall is recorded in Pahang Basin during the northeast monsoon period, amounting to almost 40% of Pahang's accumulated annual rainfall [37].

Fig. 1. Rainfall stations in Pekan, Pahang

Rainfall stations in Pekan which are involved in the study are shown in Fig. 1. For ease of operation, we numbered the rainfall station as Station 1 to 5 accordingly. The rainfall stations in which data were taken are situated in Paya Membang (Station 1), Kg. Serambi (Station 2), Kg. Temai Hilir (Station 3), Rumah Pam Pahang Tua (Station 4) and Kastam Kuala Pahang (Station 5). Our case study takes the data of rainfall runoff from the year 2012 to 2014. Rainfall data are obtained from the Department of Irrigation and Drainage (DID) Malaysia with a time interval of 1 hour between each reading.

III. MODEL STRUCTURE AND PARAMETER OPTIMIZATION TECHNIQUE

This section explains the procedure of the experiment as well as the parameters involved in constructing the rainfall-runoff model. The model structure is represented in a nonlinear autoregressive model with exogenous input (NARX). Then, multi-layered perceptron neural network (MLPNN) are employed with activation functions of sigmoid and linear functions for hidden and output layers respectively. The ANN is then trained with three different types of learning algorithms which is LM, BR and PSO. The parameter of the training algorithms in the simulation is discussed. Finally the results obtained are evaluated and compared by measuring the mean square error (MSE) and regression (R) value.

The basic relationship between input and output of a nonlinear system can be represented in a nonlinear autoregressive model with exogenous inputs (NARX) form, given by

$$\hat{y}(t) = f\begin{bmatrix} u(t), u(t-1), u(t-2), \ldots, u(t-n_u), \\ y(t-1), y(t-2), \ldots, y(t-n_y) \end{bmatrix} \quad (1)$$

Where $y(t)$ denotes the predicted output and $f(.)$, is the vector-valued nonlinear function of $y(t)$ and $u(t)$. Input $u(t)$ represents the height of collected rainfall, output $y(t)$ indicates the actual runoff level and $\hat{y}(t)$ indicates the predicted runoff level. The signal vectors applied to the input layer of the model are as follows

- *Present and past value of the rainfall, for instance,* $u(t)$,$u(t-1)$,…,$u(t-n_u)$*, which represents the model order of the system*
- *Delayed values of the runoff level, such as* $y(t-1)$,$y(t-2)$,…,$y(t-n_y)$

Next, we will see the configuration of ANN with the learning algorithms to model the rainfall-runoff system.

A. Artificial Neural Network (MLP-NN)

ANN is suitable for rainfall-runoff prediction [38], [39] as it offers relatively swift and flexible means of modelling [40]. Multilayer perceptron MLP is a form of ANN. It includes input variables, hidden layers and output layer consisting of output variable [41]. There are connection weights which act as the interconnecting link between the neuron layers. The weight which directly applied to one neuron without being connected with the previous neuron is applicable in certain circumstance which is known as bias. The activation function chosen is the sigmoid function

$$f(s) = \frac{1}{1+e^{-s}} \quad (2)$$

Where $s_i = \sum_{i=1}^{n} w_i x_i$, in which w_i are weights and x_i are input values. Further explanation of ANN implementation will be discussed as we introduce the learning algorithms in the following section.

Levenberg Marquardt Algorithm

The first learning algorithm to be implemented is Levenberg Marquardt (LM). LM is claimed to be a fast converging training algorithm [42] with vast usage in neural network fields. It is a modification of the classic Newton algorithm to find the best solution for minimization problem [43].

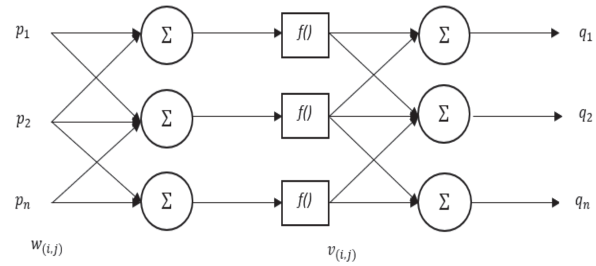

Fig. 2. Neural network with two-layered hidden neuron

Fig. 2 shows two-layered neural networks with certain variables and parameter. p_n denotes input, $w_{(i,j)}$ represents connecting weight of the first layer, $v_{(i,j)}$ as the connecting weight of the second layer and q_n as output. The net input to unit *i* of the first layer is

$$a(i) = \sum_{i=1}^{n} w_{(i,j)} p_i \quad (3)$$

The net output of unit *i* for the second layer

$$q(i) = \sum_{i=1}^{n} f\big(a(h)\big) v_{(i,j)} \quad (4)$$

The objective of ANN is to determine the relationship between input and output pairs. *h* is the hidden unit and *f( )* acts as the activation function. The chosen activation function is Sigmoid function

$$f(s) = \frac{1}{1+e^{-s}} \quad (5)$$

For update given by the Gauss-Newton method,

$$\Delta x = [J^T(x)J(x)]^{-1} J^T(x)e(x) \quad (6)$$

Whereas, the adjustment to the Gauss-Newton method by Levenberg-Marquardt is such

$$\Delta x = [J^T(x)J(x) + \mu I]^{-1} J^T(x)e(x) \quad (7)$$

In circumstances of scalar μ is zero, Eq. (6) is just the newton’s method, when μ is Large, Eq. (6) becomes gradient descent with small step size. The original description of Levenberg-Marquardt is given in [44].

Bayesian Regularization

The second learning algorithm to train ANN is Bayesian Regularization (BR). The regularization term favors small values for network weights and biases, and decrease tendency of a model to overfit noise in the training data [45]. The objective function for BR is shown below,

$$F = \beta * E_d + \alpha * E_w \tag{8}$$

where E_d is the sum of squared error, and E_w is the sum of square weight. Alpha α and beta β are Bayesian hyperparameters, variables which guide the optimization (minimal error or minimal weights) that the learning process must seek. The aim is to achieve a minimal error of cost functions using minimal weights. After the data is acquired, the post-distribution of weight can be revised through the Bayes' rule

$$P(w|D, \alpha, \beta, M) = \frac{P(D|w,\beta,M),P(w|\alpha,M)}{P(D|\alpha,\beta,M)} \tag{9}$$

$P(w|D, \alpha, \beta, M)$ represents posterior probability. $P(D|w, \beta, M)$ denotes the probability of data occurring given the weights *w*. $P(w|\alpha, M)$ is the weight estimation before data attained. $P(D|\alpha, \beta, M)$ is known as the normalization factor. A thorough study on BR can be seen here [46].

Particle Swarm Optimization

The foundation of Particle Swarm Optimization (PSO) can be represented as a bunch of random particles (solutions) which have their own random position and velocities. With the particles flowing over the hyperspace searching for possible solutions, a search for optima over a series of iteration was completed [47]. The particles learning process is based on its own experience and other particle's prior experience. In the hyperspace, each individual particle has its own best fitness position noted by the term personal best. Each particle maneuvers in accordance with a personal best and global best with a new velocity term for each particle throughout each iteration. The personal best and global best velocities are randomly weighted to yield new velocity rate for the particle.

$$v(t+1) = wv(t) + c_1 r_1[p(t) - x(t)] + c_2 r_2[g(t) - x_{11}(t)] \tag{10}$$

The training algorithm using PSO is shown in equation (9). $v(t)$ is the velocity of the particle, $x(t)$ is the position of the particle, c_1 and c_2 are the cognitive coefficient and social coefficient respectively, r_1 and r_2 are the cognitive components and social component, *w* is the inertial component, *p(t)* is the personal best and *g(t)* is the global best candidate.

$$x(t+1) = x(t) + v(t+1) \tag{11}$$

The basis function formulation is shown above in equation (11). $x(t)$ is the position of the particle, while $x(t+1)$ represents the updated position of the particle and $v(t+1)$ shows the particle's new updated velocity.

B. Rainfall-Runoff Modelling Technique

The training algorithm parameter setting is shown in this section with a different setting for LM, BR and PSO. Selected parameters set for ANN are shown in Table I.

Parameter \ Learning Algorithm	LM	BR	PSO
Maximum number of epoch	1000	1000	1000
Minimum performance gradient	1e-7	1e-7	-
Initial Marquardt parameter	0.001	0.001	-
Decrease factor of Marquardt parameter	0.1	0.1	-
Increase factor of Marquardt parameter	10	10	-
Maximum value of Marquardt parameter	1e10	1e10	-
PSO acceleration Constant (c_1 & c_2)	-	-	1.5

Table I. Learning algorithm and its parameter

If the simulation stopped because the maximum Marquardt parameter has been reached, it is a satisfactory indicator that the algorithm has converged. Further training after Marquardt parameter reached will degrade network learning. BR updates weight and bias value according to LM optimization. When training ANN with BR algorithm, it is crucial to let the simulation trained until the parameters have converged. For PSO, the minimum performance gradient was abandoned as PSO uses position and velocity update as performance function to determine its weights and biases. The acceleration constant c_1 and c_2 are tantamount to Marquardt parameter in LM and BR; it represents particle's movement towards personal best and global best position. The objective function is determined by MSE, with regression value R, at the end of the simulation.

C. Evaluation

Results evaluations were indicated by measuring the mean square error (MSE) and regression (R) value. Mean square error is the average squared difference between targets and outputs. Lower values are better. Zero means no error. The objective function is as follow

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_1 - Y_i)^2 \tag{12}$$

$\hat{Y}$ indicate vector of n^{th} output, and Y is the vector of observed values corresponding to the inputs to the function which produce the predictions. Regression R value measures the correlation between outputs and targets. R value of 4 means a close relationship, 0 value dictates a random relationship

$$R = \sum_{t-1}^{n} \frac{(Q_t - \bar{Q})(\hat{Q} - \bar{\hat{Q}})}{\sqrt{\sum_{t=1}^{n}(Q_t - \bar{Q})^2 \sum_{t-1}^{n}(Q_t - \bar{\hat{Q}})^2}} \tag{13}$$

$Q, \hat{Q}_1, \bar{Q}, \bar{\hat{Q}}$, denotes t^{t} input rainfall, output runoff level, input rainfall mean, and output runoff level mean, respectively. As the method of ANN evaluation is clearly discussed, next we look into the way of how the simulation is being set up.

D. Simulation Setup

The ANN is trained with observed runoff level per hour obtained from the year 2012 until the end of 2014. Mean areal hourly rainfall data of the 5 stations are calculated to represent the precipitation of the area. The data is usually divided into two classes; 70% for training and 30% for testing as shown in [48]. The testing part is further broken down into two, which is the testing and validation phase. In numbers, the 18377 data sets for training phase makes up a percentage of 70% of the whole data, 3938 data sets for validation phase with 15% and the rest of 3938 data sets for testing phase with the remaining 15%.

Pre-Test Simulation

In order to obtain the best configuration of each distinctive learning algorithm, we conducted a pre-test before the simulation of rainfall-runoff forecast model. In the pre-test, there are two manipulating variables to be evaluated. These variables are

- ***Number of Neuron in Hidden Layer***

Various input combinations are used in order to determine the best model to estimate runoff in using ANN. The number of hidden neurons of 3 and 10 are chosen for the varying variables. Based on the MSE and correlation value the best numbers of neurons in the hidden layer are obtained.

- ***Type of model***

Model I : $\boldsymbol{Q(t) = f(P(t-1), Q(t-1))}$
Model II : $\boldsymbol{Q(t) = f(P(t-1), P(t-2), Q(t-1), Q(t-2))}$
Model III : $\boldsymbol{Q(t) = f(P(t-1), P(t-2), P(t-3),}$
$\boldsymbol{Q(t-1), Q(t-2), Q(t-3))}$

Three models were developed to examine the effect of adding delays to neural network configuration. Each model represents number of delays to 1, 2 and 3 number of delays respectively. The input data of the model consist of antecedent rainfall, $P(t-1)\ldots P(t-n)$ and antecedent runoff level data, $Q(t-1)\ldots Q(t-2)$**.** The sign $\boldsymbol{t}$ denotes time, where P denotes input rainfall and Q is the sign for runoff level.

Rainfall-runoff forecast

After best configurations of PSO, LM and BR are obtained, we finally proceed with the comparison of rainfall-runoff forecast model. The comparison is done separately on each rain station data. To simulate runoff forecast 12 hours in advance, the input data of runoff level used for ANN is the data of 12 hours ahead of the forecast time. The best learning algorithm which produces optimum rainfall-runoff forecast is determine.

IV. RESULTS

E. Pre-Test Data Tabulation

Table II. MSE, R (correlation) and best configuration of ANN model trained by LM, BR and PSO. 10n/3rd implies 10 number of neurons in hidden layer with 3rd model order.

Rainfall stations	MSE			R(Correlation)			Best Configuration		
	LM	BR	PSO	LM	BR	PSO	LM	BR	PSO
Station1	1.2282×10^{-1}	1.0661×10^{-1}	4.9081×10^{-3}	3.66913	3.6704	3.65481	10n/3rd	10n/3rd	10n/1st
Station2	1.2577×10^{-1}	1.2416×10^{-1}	4.6445×10^{-3}	3.66846	3.66941	3.65231	3n/3rd	3n/ 3rd	10n/1st
Station3	1.2577×10^{-1}	1.206×10^{-1}	4.731×10^{-3}	3.66846	3.66914	3.65841	3n/ 3rd	10n/3rd	10n/1st
Station4	1.2328×10^{-1}	1.0661×10^{-1}	5.2627×10^{-3}	3.6691	3.6704	3.65275	10n/3rd	10n/3rd	10n/1st
Station5	1.2391×10^{-1}	1.1322×10^{-1}	4.8728×10^{-3}	3.66901	3.66966	3.65503	10n/3rd	10n/3rd	10n/1st

Table III. R and MSE of rainfall-runoff forecast with bolded values imply best result

Rain Station	*MSE*			*R correlation*			Best training algorithm
	LM	*BR*	*PSO*	*LM*	*BR*	*PSO*	
Station 1	$\mathbf{1.3031\times 10^{-1}}$	1.6866×10^{-1}	2.4436×10^{-1}	**3.66687**	3.66377	3.65481	**LM**
Station 2	1.5346×10^{-1}	$\mathbf{1.3317\times 10^{-1}}$	2.3124×10^{-1}	3.66474	**3.66698**	3.65440	**BR**
Station 3	1.2928×10^{-1}	$\mathbf{1.0033\times 10^{-1}}$	2.3554×10^{-1}	3.66742	**3.66962**	3.65841	**BR**
Station 4	$\mathbf{1.4694\times 10^{-1}}$	2.0731×10^{-1}	2.620×10^{-1}	**3.66494**	3.66008	3.65481	**LM**
Station 5	$\mathbf{1.4716\times 10^{-1}}$	1.8052×10^{-1}	2.4260×10^{-1}	**3.66585**	3.66336	3.65503	**LM**

Table II shows the result of each rainfall station for ANN trained by LM, BR and PSO. The best network architecture in terms of the R(correlation) value and MSE (mean square error) is presented. The best configuration column presents the best number of neurons in the hidden layer and optimum type of model for each individual learning algorithm on every rainfall station. The configuration obtained here will be used as a reference for the rainfall-runoff forecast model.

B. Rainfall-Runoff Forecast Model Performance

As the best configuration of neural network models of LM, BR and PSO have been acquired, next a 12-hour forecast of the rainfall-runoff model is simulated. To model rainfall-runoff forecast, the rainfall input data are kept unchanged, whereas the runoff level input data are observed as 12 hours lead of the current time. Regression plot and MSE performance graph are shown and discussed in the following part. The best learning model to train the neural network for each rainfall station can be deduced from the result.

Table III shows the result of the rainfall-runoff forecast. From the table, MSE performance graphs are constructed and test data results are compared. The network trained by LM algorithm shows the best results for rainfall Station 1, 4 and 5. Meanwhile, neural network trained by BR showed the best result for rainfall Station 2 and 3.

Exceptional accuracy of correlation coefficient was attained as the result of two important factors. The first is due to abundance of input data available. Large training data helps in achieving highly accurate deep learning process as shown in [49], [50]. The second factor is the positive effect of filtering out missing data in the rainfall and runoff level input. In some part in particular, the missing gaps within either rainfall or runoff data received from DID Malaysia are consecutive. This may occur due to many factors such as equipment dysfunction, errors in measurements or faults in data acquisition. There are several conventional methods to overcome this situation. Ignoring, deleting and interpolating missing data are the options when encountering such circumstances. In this research, we have opted for deletion of incomplete or missing data which in return produce high accuracy model. Moreover, ignoring missing value may disrupt the result of the analysis if its percentage is large, whereas interpolating may produce bias reading [51].

V. DISCUSSIONS

In this section, we will make a comparative evaluation of the learning algorithms of LM, BR and PSO based on the MSE graph comparison, time of training iteration and stability analysis.

F. Comparative Assessment

From Table 3 previously shown, MSE final results are observed but the error values during each iteration are unapparent. To make a specific analysis on the simulation, MSE performance plot of each rainfall stations are presented. From here we can infer the learning algorithms' performance in terms of error and number of iterations.

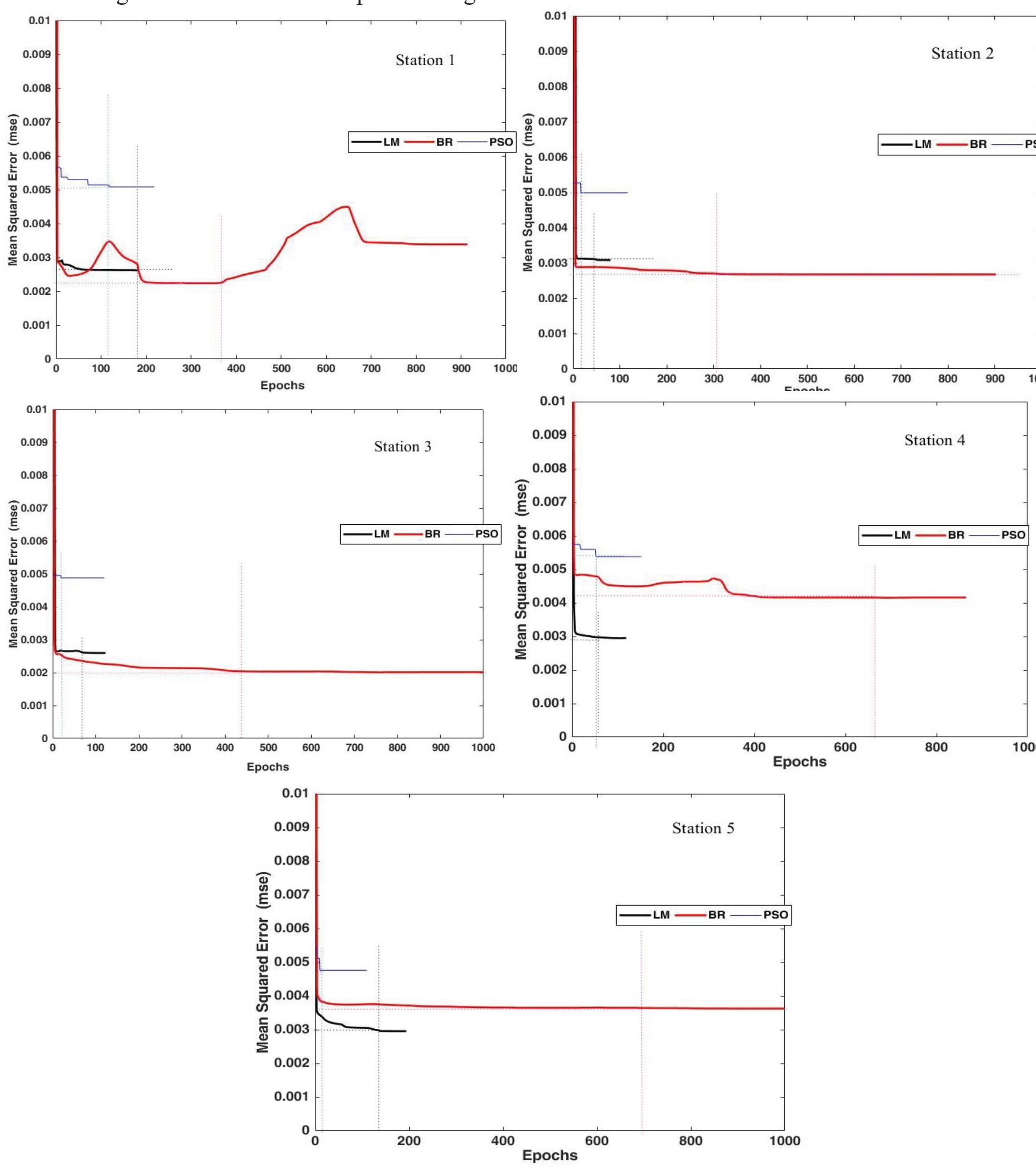

Fig. 2. MSE performance of the rainfall-runoff model for 5 rainfall stations. Line/training algorithm: Black-LM; red-BR, blue-PSO

The performance of ANN trained by LM, BR and PSO are placed on the same axis in Fig 3. First the physical of the MSE graphs are to be observed. For rainfall Station

1, 2 and 3, ANN trained by BR reached the lowest error when compared to other learning algorithms. The lowest error achieved is 2.2×10^{-3} , 2.68×10^{-3} and 2.02×10^{-3} respectively. While for rainfall Station 4 and 5 ANN trained by LM acquired the lowest error of 2.975×10^{-3} and 2.95×10^{-3} each. ANN trained by PSO algorithm accomplished the highest error value in general for all stations, when compared to LM and BR algorithm.

As ANN trained by LM and BR showed better accuracy, comparisons are made between the two. An important factor to bear in mind is that the lowest error attained in Fig 3 does not represent lowest MSE value. For instance, in Station 1 ANN trained by BR achieved lowest error but throughout the training LM attained least MSE value. Thought it seems ANN trained by BR achieved lower minimum error, it reaches the value in the later stage of the simulation with best error value recorded at epoch number 209, 305 and 438. However for ANN trained by LM, the best error value gained during earlier iteration recorded at epoch number 59 and 194 respectively. From here, we can draw the inference that ANN trained by LM converge faster than BR with least number of MSE. Further discussions on the rate of convergence and number of iterations will be discussed in the following section.

B. Numbers of training iteration

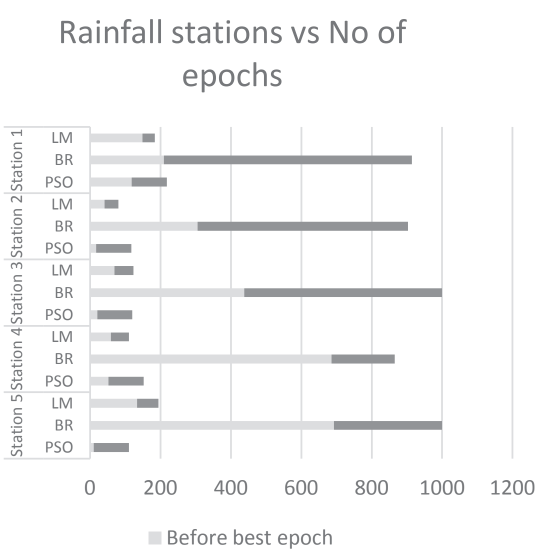

Fig. 3. Bar chart showing the number of iterations of each training

Fig. 4 shows the amount of iteration before and after best MSE value is achieved. From the bar chart, it can be said that neural network train by BR continues redundantly. Simulation for all 5 rainfall stations surpasses the 900 iterations benchmark and 2 of the simulation terminate as it completed the allowable maximum number of iterations.

For network trained by LM and PSO, early stopping through validation check was implied. This is done to prevent overtraining and overfitting of the model. LM is a fast convergence method [23], [42] as further supported by the analysis of the result. The bar chart in Figure 6 implies that the network training through LM algorithm acquires minimum error value at a faster rate than BR, but in a letter stage to PSO.

For network trained by PSO, the best MSE values are obtained at the early stage of training with recorded epoch number of 118, 17, 20, 52 and 10. However, the iteration prolonged even after the best error was attained. Thus we can say that PSO earned lowest error at least number of iteration, but terminate training later than LM. From here, we can conclude that LM terminate training fastest after acquiring best MSE value, followed by PSO and BR.

C. Stability

Another crucial aspect to be observed upon the ANN model is stability. In this area, training neural network by LM and PSO are adequate as the error recorded decreases over time. Meanwhile, for BR, MSE graph plot recorded several troughs along the simulation. This is noticeable for simulation of rainfall Station 1 and scarcely seen for Station 4. For precise analysis, we present the MSE graph of Station 1 and Station 4 for ANN trained by BR showing training and testing data plot. During the training phase, the MSE shows steady decrement over time as the parameters, weights and biases are being predetermined. As it comes towards the testing phase, the algorithms declined in performance. The MSE during training and testing is 2.43427e-3 and 3.38755e-3 for rainfall Station 1 respectively. Meanwhile, MSE during training and testing is 2.56150e-3 and 4.16389e-3 for rainfall Station 4.

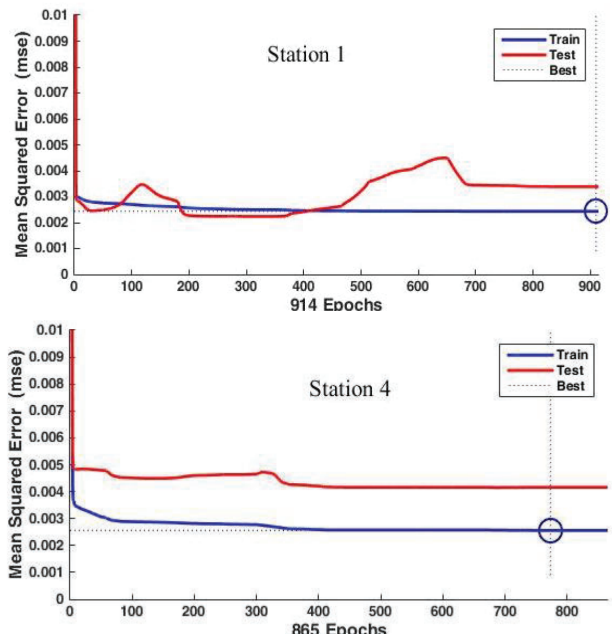

Fig. 4. MSE plot graph for ANN trained by BR for Station 1 and Station 4

Fig 5 shows smooth decrement of MSE during the training phase, while a few peaks and troughs can be seen for the testing part of the simulation for both rainfall Station 1 and Station 4. For rainfall Station 4, it is demonstrated that the testing error plot is closely following the trend set by the training error. Thus, even though the test data set shows more than one peaks and troughs, it is still generally accepted as it is superseding training data. In contradictory, the test set graph plot for rainfall Station 1 fluctuates until it reaches approximately just before the 700th epoch benchmark. As the training set error is decreasing, the test set error rises. This may well be the case of overfitting.

Overfitting is a circumstance where the neural network performs adequately on training data but deficient on generalization [52]. In other words, the network tends to learn the noise and irrelevant properties of training data which results in lower performance when tested with unseen data [53]. Cross-validation, early stopping and regularization are amongst the methods used in preventing overfitting [54]. BR usually consumes more time to converge, as illustrated in Figure 3 where training for Station 1, 2 and 4 achieve convergence after the 800th iteration benchmark. From here we deduced that training ANN with BR has a tendency of overfitting.

VI. CONCLUSION

With natural disaster and catastrophe involving flooding inflicts damage and loss of resources every year, the design of efficient flooding defense system is inevitable. This requires extensive data collection and complicated hydrological calculations. Therefore, the rainfall-runoff model provides a good solution to provide runoff and river flow forecast. The study presented a neural network model with varying learning algorithms. Distinctive neural network parameters such as the number of neurons in the hidden layer and the number of delays is investigated to acquire best-fit configurations for our case study in Pahang river.

The study shows the comparison of ANN models trained with LM, BR and PSO. Performance of the best model for all learning algorithm is then identified. Based on the research, we can deduce that ANN trained by LM achieved lowest MSE value with least number of iterations. Meanwhile, ANN trained by BR attained low MSE value but require highest number of iteration and has a tendency of overfitting. Whereas ANN trained by PSO acquire highest number of MSE value, with low number of iterations.

In conclusion, ANN model trained by LM shows the best evaluation for 12-hour forecast when compared to BR and PSO. Encouraging results were achieved to estimate runoff level in the flood-prone area by using ANN model. The rainfall-runoff models developed can be utilized to efficiently formulate hydrological studies or flood defense system in this particular area.

ACKNOWLEDGMENT

The author would like to thank Research Initiative Grant Scheme by IIUM (RIGS 16-337-0501) for funding the study and Department of Irrigation and Drainage Malaysia (DID) for the contribution of experimental data of rainfall and runoff information.

REFERENCES

1. A. C. Braga *et al.*, "Characterization of river flow fluctuations via horizontal visibility graphs," *Phys. A Stat. Mech. its Appl.*, vol. 444, pp. 1003–1011, 2016.
2. X. Song, F. Kong, C. Zhan, and J. Han, "Hybrid Optimization Rainfall-Runoff Simulation Based on Xinanjiang Model and Artificial Neural Network," *J. Hydrol. Eng.*, vol. 17, no. 9, pp. 1033–1041, 2012.
3. W. Taesombat and N. Sriwongsitanon, "A real rainfall estimation using spatial interpolation techniques," *ScienceAsia*, vol. 35, no. 3, pp. 268–275, 2009.
4. X. Z. Liang *et al.*, "Regional climate-weather research and forecasting model," *Bull. Am. Meteorol. Soc.*, vol. 93, no. 9, pp. 1363–1387, 2012.
5. P. C. Nayak, B. Venkatesh, B. Krishna, and S. K. Jain, "Rainfall-runoff modeling using conceptual, data driven, and wavelet based computing approach," *J. Hydrol.*, 2013.
6. Y. Wang, Y. Jiang, X. Lei, and W. Hao, "Rainfall-Runoff Simulation Using Simulated Annealing Wavelet BP Neural Networks," *2010 Int. Conf. Intell. Comput. Technol. Autom.*, pp. 963–967, 2010.
7. Andrea Petroselli & Abdiel Fernandez Alvarez (2012) The Flat-Area Issue in Digital Elevation Models and Its Consequences for Rainfall-Runoff Modeling, GIScience & Remote Sensing, 49:5, 711-734, DOI: 10.2747/1548-1603.49.5.711
8. Mosavi, A.; Ozturk, P.; Chau, K.-W. Flood Prediction Using Machine Learning Models: Literature Review. *Water* **2018**, *10*, 1536.
9. N. With, P. By, Y. B. Dibike, S. Velickov, D. Solomatine, and M. B. Abbott, "Model Induction with Support Vector Machines: Introduction and Application," *J. Comput. Civ. Eng.*, pp. 208–216, 2001.
10. M. S. Tehrany, B. Pradhan, S. Mansor, and N. Ahmad, "Flood susceptibility assessment using GIS-based support vector machine model with different kernel types," *Catena*, vol. 125, pp. 91–101, 2015.
11. D. A. Sachindra, F. Huang, A. Barton, and B. J. C. Perera, "Least square support vector and multi-linear regression for statistically downscaling general circulation model outputs to catchment streamflows," *Int. J. Climatol.*, vol. 33, no. 5, pp. 1087–1106, 2013.
12. M. S. Tehrany, B. Pradhan, and M. N. Jebur, "Spatial prediction of flood susceptible areas using rule based decision tree (DT) and a novel ensemble bivariate and multivariate statistical models in GIS," *J. Hydrol.*, vol. 504, pp. 69–79, 2013.
13. P. Sugunnasil, S. Somhom, W. Jumpamule, and N. Tongsiri, "Modelling a neural network using an algebraic method," *ScienceAsia*, vol. 40, no. 1, pp. 94–100, 2014.
14. A. R. F. Quiros, A. Abad, R. A. Bedruz, A. C. Uy and E. P. Dadios, "A genetic algorithm and artificial neural network-based approach for the machine vision of plate segmentation and character recognition," *2015 International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, Cebu City, 2015, pp. 1-6.
15. Y. Ding, T. Wang and X. Fu, "The connecting of images based on machine vision and neural network," *The Fourth International Workshop on Advanced Computational Intelligence*, Wuhan, 2011, pp. 391-394. doi: 10.1109/IWACI.2011.6160038
16. B. Wenfeng, S. Yaping, and X. Bingbing, "Development of machine vision positioning system based on neural network," *2011 Int. Conf. Mechatron. Sci. Electr. Eng. Comput.*, pp. 1683–1686, 2011.
17. T. Egawa, K. Suzuki, Y. Ichikawa, T. Iizaka, T. Matsui and Y. Shikagawa, "A water flow forecasting for dam using neural networks and regression models," *2011 IEEE Power and Energy Society General Meeting*, San Diego, CA, 2011, pp. 1-6. doi: 10.1109/PES.2011.6038925
18. L. H. Nghiem and N. Thi Kim Oanh, "Comparative analysis of maximum daily ozone levels in urban areas predicted by different statistical models," *ScienceAsia*, vol. 35, no. 3, pp. 276–283, 2009.
19. [19] J. Chien and T. Lu, "Tikhonov regularization for deep neural network acoustic modeling," *2014 IEEE Spoken Language Technology Workshop (SLT)*, South Lake Tahoe, NV, 2014, pp. 147-152. doi: 10.1109/SLT.2014.7078565
20. A.; P. I. Jain S. K. V., "Comparative Analysis of Event-based Rainfall-runoff Modeling Techniques—Deterministic, Statistical, and Artificial Neural Networks," *J. Hydrol. Eng.*, vol. 8, pp. 93–98, 2003.
21. F. S. Marzano, G. Rivolta, E. Coppola, B. Tomassetti, and M. Verdecchia, "Rainfall nowcasting from multisatellite passive-sensor images using a recurrent neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 11, pp. 3800–3811, 2007.
22. D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2. pp. 431–441, 1963.
23. G. Narang, S. Singh and A. Narang, "Robust face recognition method based on SIFT features using Levenberg-Marquardt Backpropagation neural networks," *2013 6th International Congress on Image and Signal Processing (CISP)*, Hangzhou, 2013, pp. 1000-1005. doi: 10.1109/CISP.2013.6745202
24. W. Suparta, W. S. Putro, M. S. J. Singh and M. F. Asillam, "The estimation of rainfall and precipitation variation during 2011 convective system using an artificial neural network over Tawau, Sabah," *2015 International Conference on Space Science and Communication (IconSpace)*, Langkawi, 2015, pp. 479-484. doi: 10.1109/IconSpace.2015.7283806
25. P. Graff, F. Feroz, M. P. Hobson, and A. Lasenby, "Neural Networks for Astronomical Data Analysis and Bayesian Inference," *2013 IEEE 13th Int. Conf. Data Min. Work.*, pp. 16–23, 2013.
26. V. B. Nikam and B. B. Meshram, "Modeling Rainfall Prediction Using Data Mining Method: A Bayesian Approach," *2013 Fifth Int. Conf. Comput. Intell. Model. Simul.*, pp. 132–136, 2013.

27. S. Hellman, A. McGovern, and M. Xue, "Learning ensembles of continuous Bayesian networks: An application to rainfall prediction," *Proc. - 2012 Conf. Intell. Data Understanding, CIDU 2012*, pp. 112–117, 2012.
28. J. Kennedy and R. Eberhart, "Particle Swarm Optimization," pp. 1942–1948, 1995.
29. K. W. Chau, "Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River," *J. Hydrol.*, 2006.
30. K. Gopalakrishnan, "Particle Swarm Optimization in Civil Infrastructure Systems: State-of-the-Art Review," in *Metaheuristic Applications in Structures and Infrastructures*, 2013.
31. K. K. Kuok, S. Harun, and S. M. Shamsuddin, "Particle Swarm Optimization Feedforward Neural Network for Hourly Rainfall-runoff Modeling in," no. 10, 2009.
32. K. W. Chau, "A split-step particle swarm optimization algorithm in river stage forecasting," *J. Hydrol.*, pp. 131–135, 2007.
33. TheStarOnline, "Sg Lembing cut off by flood," *Saturday, 14 Jan*, 2012.
34. MYsinchew.com, "Rise in flood victims in Negeri Sembilan , Johor hit again by flooding," *3 Dec*, vol. 3, p. 2012, 2012.
35. MalayMailOnline, "Heavy rain continues , over 38 , 000 flood victims throng relief centres in Pahang," *December 07*, 2013.
36. TheStarOnline, "Flood situation in Pahang still critical," *Monday, 29 Dec*, 2014.
37. J. Suhaila, S. M. Deni, W. a N. Zawiah Zin, and A. A. Jemain, "Trends in Peninsular Malaysia rainfall data during the southwest monsoon and northeast monsoon seasons: 1975-2004," *Sains Malaysiana*, vol. 39, no. 4, pp. 533–542, 2010.
38. S. Asadi, J. Shahrabi, P. Abbaszadeh, and S. Tabanmehr, "Neurocomputing A new hybrid arti fi cial neural networks for rainfall – runoff process modeling," *Neurocomputing*, 2013, DOI:10.1016/j.neucom.2013.05.023.
39. Y. Seo, S. Kim, O. Kisi, and V. P. Singh, "Daily water level forecasting using wavelet decomposition and artificial intelligence techniques," *Journal of Hydrology,* Volume 520, January 2015, Pages 224-243
40. [40] a. M. Jafri, Z. Hashim, M.L.Kavvas, Z.Q.Chen, and N.Ohara, "Development of Atmospheric Based Flood Forecasting and Warning System for Selected River Basins in Malaysia.," *J. Hydrol.*, vol. 356, no. 3–4, pp. 1–18, 2010.
41. I. Aichouri, A. Hani, N. Bougherira, L. Djabri, H. Chaffai, and S. Lallahem, "River Flow Model Using Artificial Neural Networks," *Energy Procedia*, vol. 74, pp. 1007–1014, 2015.
42. S. F. Toha and M. O. Tokhi, "ANFIS and Neural Network Techniques for Non Parametric Modeling," *Syst. Sci.*, vol. 36, 2010.
43. S. Kumar, A. Indian, and Z. Khan, "Neural Network Model for Prediction of Ground Water Level in Metropolitan Considering Rainfall- Rainfall Runoff as a Parameter," vol. I, no. 3, pp. 195–198, 2013.
44. M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.
45. M. S. Khan and P. Coulibaly, "Bayesian neural network for rainfall-runoff modeling," *Water Resour. Res.*, vol. 42, no. 7, pp. 1–18, 2006.
46. F. D. Foresee and M. T. Hagan, "Gauss-Newton Approximation to Bayesian Learning," *Neural Comput.*, vol. 4, no. 3, pp. 1930–1935, 1997.
47. M. R. M. Romlay, M. M. Rashid, and S. F. Toha, "Development of Particle Swarm Optimization Based Rainfall-Runoff Prediction Model for Pahang River , Pekan," *2016 Int. Conf. Comput. Commun. Eng.*, pp. 306–310, 2016.
48. V. Gholami, K. W. Chau, F. Fadaee, J. Torkaman, and A. Ghaffari, "Modeling of groundwater level fluctuations using dendrochronology in alluvial aquifers," *J. Hydrol.*, vol. 529, no. September, pp. 1060–1069, 2015.
49. M. A. R. L. W. Yaniv Taigman, Ming Yang, "DeepFace: closing the gap to human-level performance in face verification," *Acta Ophthalmol. Scand.*, vol. 73, no. 6, pp. 506–511, 1995.
50. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.
51. D. Priya, "A Review of Missing Data Handling Methods on Time Series Data," *Int. J. Eng. Technol. Sci.*, vol. 2, no. 2, p. 58, 2015.
52. A. B. van Wyk and A. P. Engelbrecht, "Overfitting by PSO trained feedforward neural networks," *IEEE Congress on Evolutionary Computation*, Barcelona, 2010, pp. 1-8. doi: 10.1109/CEC.2010.5586333
53. Z. K. Pourtaheri and S. H. Zahiri, "Ensemble classifiers with improved overfitting," *2016 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, Bam, 2016, pp. 93-97. doi: 10.1109/CSIEC.2016.7482130
54. R. Gencay and Min Qi, "Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging," in *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 726-734, July 2001. doi: 10.1109/72.935086

AUTHORS PROFILE

Muhammad Rabani Mohd Romlay received B.Eng (Hons) and M.Sc. degree in Mechatronics Engineering from International Islamic University Malaysia (IIUM). Currently pursuing his study and a candidate for PhD in the same institute. At the receiving end of several honours in international conferences and exhibitions such as in IRIE 2014 and ORCID WEC 2018, the focus of his work includes neural network, machine learning, fuzzy logic, rainfall-runoff, autonomous vehicle and navigation aids for the visually impaired people.

Muhammad Mahbubur Rashid. (M'07) was born in Naogaon, Bangladesh. He received the B.Sc. (Eng.) degree in electrical and electronic engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 1992, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Malaya, Kuala Lumpur, Malaysia, in 2003 and 2007, respectively. From 1994 to 2000, he was a Sub divisional Engineer (instrumentation and control) with the Bangladesh Power Development Board. Since 2007, he has been working as an Assistant Professor and Associate Professor in the Department of Mechatronics Engineering, IIUM, Kuala Lumpur.

Siti Fauziah Toha (B.Eng'03-M'06-PhD'10), is currently an Associate Professor at the Department of Mechatronics Engineering, International Islamic University Malaysia (IIUM). She completed her Ph.D in Automatic Control and Systems Engineering from The University of Sheffield. Her current research interest is battery management system, energy reduction for electric motorcycle development as well as hybrid renewable energy applications using computational intelligence methods such as Genetic Algorithms, Fuzzy Logic and Particle Swarm optimization.

Azhar Mohd Ibrahim an Assistant Professor at International Islamic University Malaysia (IIUM) has completed his Ph.D. at the School of Computer Sciences, Universiti Sains Malaysia under research theme of crowd dynamics. He obtained his B.Eng and MSc in Mechatronics Engineering from IIUM. His research focuses on evacuation models, intelligent system, surveillance systems and artificial intelligence. He has been receiving several honors including the recipient of the best student award for Master of Science by research, 29th IIUM convocation ceremony and the winner of the 3MT thesis competition held at Universiti Sains Malaysia in 2016.