Flood Prediction Using Seasonal Autoregressive Integrated Moving Average (SARIMA) Model

Abdulrazak Yahya Saleh, Roselind Tei

Abstract: ***This paper aims to evaluate the performance of the Seasonal Autoregressive Integrated Moving Average (SARIMA) model for the purpose of flood forecasting. Seven datasets are provided by the Drainage and Irrigation Department (DID) for Sungai Bedup, Serian, Sarawak, Malaysia; and these loads of valuable information are used to evaluate the performance of the SARIMA algorithm. A distinctive network was trained and tested using the daily data obtained from the DID from the years 2014 to 2017. The performance of the algorithm was evaluated based on the technique of Root Mean Square Error (RMSE) by comparing with the Long Short Term Memory Network (LSTM) and Backpropagation Network (BP). Among the seven datasets, the Sungai Bedup set shows a small testing error rate, which is (0.008), followed by Sungai Meringgu (0.011), Semuja Nonok (0.023), Bukit Matuh and Sungai Busit with the same value (0.025); and lastly the value of Sungai Merang is (0.029). The results prove that the SARIMA model can be employed reliably to forecast the water level of Sungai Bedup with the lowest RMSE value, which is 0.008. Meanwhile, LSTM has a RMSE value of 0.08 and Backpropagation has an RMSE value of 0.711. More discussions will be provided to demonstrate the effectiveness of the model in flood prediction.***

Index Terms: ***Artificial Neural Seasonal Autoregressive Integrated Moving Average (SARIMA), Long Short Term Memory (LSTM), Backpropagation (BP)***

I. INTRODUCTION

Flood is a natural calamity, and Malaysia experiences it almost every year in varying degrees of magnitude. Throughout Malaysia, an estimated 9% of the total land area of Malaysia, including Sabah and Sarawak, is vulnerable to flood, and approximately 4.82 million people are affected by this disaster annually [1]. Over the past decade, different kinds of modelling and data types had emerged in an attempt to forecast the flood events [2]. Excessive rainfall can cause flooding in rural regions and urban areas alike, which may undergo demographic changes from time to time. [3], 50 years of data reveal that 41% of all the natural disasters are related to severe weather conditions or water-related events such as flood. The historical records of the catchments are important information that can facilitate investigating the time series of flash floods occurring hourly [4]. An early accurate prediction of the occurrences will help to over-come logistic problems of evacuation and mitigate the impacts of flood events. Different principles have been used to forecast floods, such as computer simulations based on the watershed demographic model, principle of hydrological, hydraulic components and groundwater flow model [5]. However, these methods only can predict the occurrence of flood events for certain catchment or basin based on specific water-level values. The target in this research is employing a big pool of historical flood data to predict an accurate output; the results will then be utilised by the policy makers in implementing measures to reduce the impacts of floods, not only on the society but also on the environment. The SARIMA model is used in this research due to its ability to learn from the past data in solving complex problems, and this model has been widely used in the field of forecasting [6]. By applying the SARIMA technique, the computational models that contain numerous processing layers can learn the data given with multiple levels of abstraction [7]. The catchment of this study is part of the Sadong Basin, and it is located 80 km away from Kuching. [8], the area of the whole Sadong Basin is about 3550 km2 while the total length of the main river is 150 km, as shown in Figure 1. The datasets are obtained from the DID for the years of 2014 to 2017. Forecasting is conducted on seven different gauging stations in Serian Division; which are Sungai Bedup, Bukit Matuh, Semuja Nonok, Sungai Busit, Sungai Merang, Sungai Meringgu and Sungai Tep.

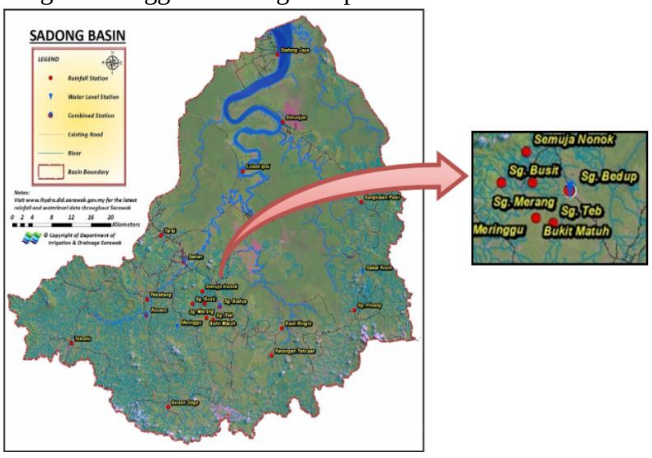

Fig. 1: Location of the Bedup River, Sarawak, Malaysia.

II. METHODOLOGY

To evaluate the performance of the SARIMA model, several experiments are conducted on seven real datasets given by DID. The characteristics of the seven datasets are shown in Table 1, These are the real-world datasets given by the DID, which are similar in terms of the number of available samples, datasets characteristics (Multi-variate), and Features (2). Each station name in the dataset is presented as an input pattern; meanwhile, the features show the water level

Revised Manuscript Received on June 05, 2019

Abdulrazak Yahya Saleh, FSKPM Faculty, University Malaysia Sarawak (UNIMAS), Kota Samarahan, 94300 Sarawak, Malaysia.

Roselind Tei, FSKPM Faculty, University Malaysia Sarawak (UNIMAS), Kota Samarahan, 94300 Sarawak, Malaysia.

per hour with the corresponding measurement (m), like the river level and danger level.

Table 1: Summary of datasets used

Data Set	Characteristics	Features
Sungai Bedup	Multivariate	2
Bukit Matuh	Multivariate	2
Sungai Teb	Multivariate	2
Sungai Merang	Multivariate	2
Sungai Meringgu	Multivariate	2
Semuja Nonok	Multivariate	2
Sungai Busit	Multivariate	2

This research divides the original dataset into two parts: 70% of the data in the whole dataset are used for training purpose to estimate the parameters' values of SARIMA, and the remaining 30% of the data for testing to evaluate the performance of the model. This re-search applies the data for a period of three and a half years for the training, and six months for the testing based on the details of Table 2.

Table 2: Training and Testing Data

Network	Training	Testing
2014-2017	Year 2014- January to June 2017	July – December 2017

The SARIMA model is a very powerful tool for handling the dependency between the input variables, and is able to hold and learn from a long sequence of training and testing. A schematic representation shows the flow of the SARIMA model in this study. Figure 2 briefly describes the schematic representation of the SARIMA model.

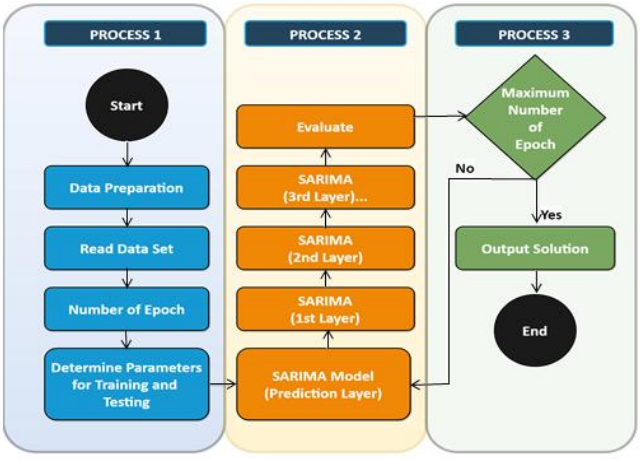

Fig. 2: Schematic representation of the proposed SARIMA model

$$\varphi(B) = 1-\varphi_1 B - \varphi_2 B^2 - \cdots - \varphi_P B^P \quad (2)$$

A. Seasonal Autoregressive Moving Average (SARIMA)

The idea of applying the Seasonal Autoregressive Integrated Moving Average (SARIMA) model is to perform the autocorrelation in the series by modelling in the collaborative way and capturing it directly. The SARIMA model contains a strong underlying mathematical and statistical theory, and it is easier to generate the predictive intervals. This model is very flexible and captures many different types of patterns [9].

The key concept in the SARIMA model is the order, which is the differencing. AR is an Autoregressive Model; it is one of the categories of SARIMA and has been classified as AR(*q*). The AR includes predictors that are lagged versions of the series. A more complex model is the Autoregressive Moving Average Model, and has been classified as ARMA: *(p, q)*. The ARMA model includes predictors, in addition to the p-lagged series and q-lagged versions of the forecast errors. This forecast error is called the moving average component of this model. The concept of ARMA model is to capture all forms of autocorrelation by including lags of the series and the forecast errors [9]. SARIMA *(p, d, q)* is the latest upgraded model of the previous algorithms of AR and ARMA. The differencing method is compatible and can be applied in the SARIMA model, which has two sets of parameters, which are *(p, d, q)* and *(P, D, Q)*. The lowercase *d* refers to the *lag-1* differencing and the uppercase *D* refers to the seasonal differencing. For a cycle with *M* seasons, the lag-M differencing is used to remove the seasonality. The parameter values of uppercase *D* denote decisions whether to perform seasonal differencing: *0* indicates that no seasonal differencing will be executed; and *1* indicates that seasonal differencing will be executed once. The SARIMA model requires the user to specify the parameter values of *(p, d, q)* and *(P, D, Q)* [9]. The SARIMA model contains many flexible techniques that capture the autocorrelations in all kinds of forms. It has a strong statistical foundation, and it is easy to obtain automated prediction intervals. SARIMA does require a stationary series which has no trend, no seasonality, and constant autocorrelation. The SARIMA method contains a strong underlying mathematical and statistical theory, and it is easy to generate the predictive intervals. The equations from (1) to (7) are based on the study

$$y_t = \emptyset_{t-1} + \emptyset_2 y_{t-2} + \emptyset_p y_{t-p} + u_t - \theta_1 u_{t-1} - \theta_2 u_{t-2} - \theta_q u_{t-q} \quad (1)$$

done for weather forecasting [9].

The SARIMA (p, d, q) model is defined with the equation as follows:

Where

P denotes *AR* parameters,

q moving average *(MA)* parameters,

$\emptyset_1, \emptyset_2$, the real parameters,

$\emptyset_p$, autoregressive coefficient,

$\theta\,(t = 1,2,q)$ are moving coefficients,

u_t independent white noise,

$\{y_t\}$ is zero.

Lag operator (B) equation is as shown below:

Where

$\varphi(B)$ is autoregressive operator

Stationerised sequence z_t by the mean of difference based on the equation:

$$z_t = (1-B)^d y_t = \nabla^d y_t \quad (3)$$

Where

d is the number of regular referencing,

y_t is corresponding to ARIMA (p, d, q).

The autocorrelation of the data series is measured by the autocorrelation coefficient; the equation is defined as below:

$$rk = \frac{\sum_{t=k+1}^{n} e_t\, e_t - k}{\sum_{t=1}^{n} e_1^2} (k = 1, 2, \dots, m), \quad (4)$$

Where

n is the number of cases in a particular sample of series for the white noise test,

m is the maximum amount of lag.

The test statistic equation is defined as below:

$$Q = n(n+2)\sum_{k=1}^{m} \frac{r_k^2}{n-k} \quad (5)$$

Given the degree of confidence of 1- α, if

$$Q < x_{\alpha}^2 \text{ (m-p-q)} \quad (6)$$

Where

Q fits the x^2 distribution at the significance of $1-\alpha$

Linear regression models determine the associated parameters for each monthly data series.

$$y_t = a_i y_j, \max + b_i y_j, \min + c_i y_j, \text{avg} + d_i \quad (7)$$

Where

a_i, b_i, c_i, d_i, are the coefficients in the model for *i* month parameters,

$y_i$max, $y_i$min, $y_i$avg, truncated mean to the *j* class where the time series of *i* month is identified in cluster analysis

The calculation of data validation accuracy for the dataset is given in equation (8)

$$RMSE \sqrt{\frac{\sum_{i=1}^{n} (P_i - O_i)^2}{n}} \quad (8)$$

Where:

P_i is a vector of *n* prediction, O_i is the vector of the observed value that corresponds to the output.

The root mean square error (RMSE) is a type of measure of error. It is a very frequently used measure for the differences between the value predicted by an estimator or a model and the actual observed values. Root mean square error is defined as the square root of differences between the predicted values and observed values. The individual differences in this calculation are known as "residuals". The RMSE method estimates the magnitudes of the errors. It is a good measure of accuracy which is used to compare the forecasting errors from different estimators for a certain variable, but not between the variables, since this measure is scale dependent.

B. Long Short-Term Memory (LSTM)

The LSTM architecture model is a block of memory cells which can maintain its state. Most studies incorporated many of the improvements that had been made to the LSTM architecture model since its original formulation. LSTM is now applied to many learning problems which differ in the significance of scale and the nature of the problems, the improvements of which were initially tested. A schematic memory block of the LSTM architecture model can be seen in Figure 3. LSTM consists of these components: three different gates, namely input gates, forget gates, output gates; block input; a single cell of the constant error carousel; an output activation function; and peephole connections. The output of the block is connected back to the block input and all of the gates [10].

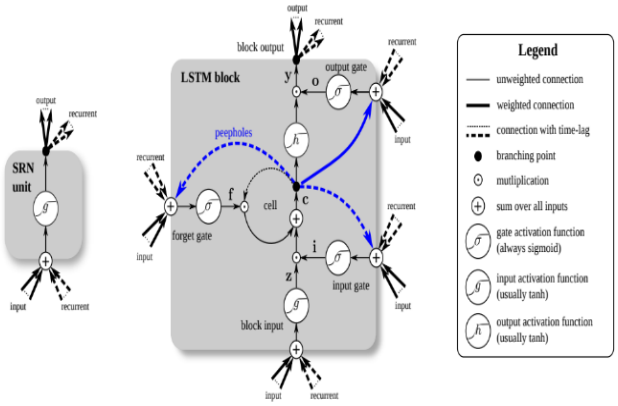

Fig. 3: Schematic of the Long Short-Term Memory Block [11]

The final weight derivatives are found by summing the derivatives at each time-step, where *O* is the objective function used for training; *f() (frequently noted as σ(.))* is the standard logistic sigmoid function defined in Equation (9); *g()* and *h()* are the transformations of *function ()*,the range of which is *[-2,2]* and *[- 1,1]*.

$$\delta_j^t = \frac{\partial 0}{\partial a_j^t} \quad (9)$$

The order in which Equation (10) to Equation (14) are calculated during the forward and backward passes is important and should proceed as specified below [11]. As with the standard LSTM, all states and activations are set to zero at $t = 0$, and all δ terms are zero at $t = T + 1$. LSTM will decide what information should be thrown away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer"; where $ht-1$ and xt, and outputs a number between *0* and *1* for each number in the cell state $Ct-1$.

Input Gates:

$$a_l^t = \sum_{i=1}^{I} w_{il}\, x_i^t + \sum_{h=1}^{H} w_{hl}\, b_h^{t-1} + \sum_{c=1}^{C} w_{cl}\, s_c^{t-1} + b_l^t = f(a_l^t) \quad (10)$$

Forget Gates:

$$a_\emptyset^t = \sum_{i=1}^{I} w_{i\emptyset}\, x_i^t + \sum_{h=1}^{H} w_{h\emptyset}\, b_h^{t-1} + \sum_{c=1}^{C} w_{c\emptyset}\, s_c^{t-1} + b_\emptyset^t = f(a_\emptyset^t) \quad (11)$$

Cells:

$$a_c^t = \sum_{i=1}^{I} w_{ic}\, x_i^t + \sum_{h=1}^{H} w_{hc}\, b_h^{t-1} + s_c^t = b_\emptyset^t s_c^{t-1} + b_l^t = g(a_c^t) \quad (12)$$

Output Gates:

$$a_w^t = \sum_{i=1}^{I} w_{iw}\, x_i^t + \sum_{h=1}^{H} w_{hw}\, b_h^{t-1} + \sum_{c=1}^{C} w_{cw}\, s_c^{t-1} + b_w^t = f(a_w^t) \quad (13)$$

Cell Outputs:

$$b_c^t = b_w^t h(s_c^t) \quad (14)$$

Where:

- w_{ij} the weight of the connection from unit *i* to unit *j*
- a_i^t the network input to some unit *j* at time *t*
- b_i^t the value of the same unit after the activation function has been applied
- ι input gate, $\emptyset$ forget gate, *w* output gate
- *C* set of memory cells of the block
- s_c^t state of cell c at time *t*
- *f* the activation function of the gates, *g* cell input activation functions, *h* cell output activation functions
- *I* the number of inputs, *K* the number of outputs, *H* number of cells in the hidden layer

C. Back Propagation (BP)

BP refers to one of the categories in the artificial neural Network (ANN), which consists of the different interconnected layers in its neuron. BP is a type of "Deepest-Descent" technique. If provided with an appropriate given number of the hidden layer units, BP will be able to minimise the error of the nonlinear function with high complexity. The number of the vertices connected to the input layer is determined by the number of input and output patterns. Training of the BP consists of three stages: advanced feed training, error calculation and weight adjustment [12]. A schematic of the BP architecture model can be seen in Figure 4.

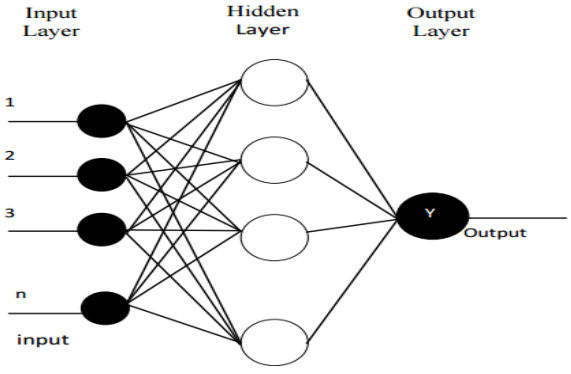

Fig. 4: Schematic of the BP Model

The activation function is a mathematical algorithm operated by the output of signal y. This function is used to enable or disable the neurons. BP is determined by the specific weight and activation function using the sigmoid binner. The values lay between the interval 0 and 1 [12]. The BP algorithm is described in the following stages from Equations (15) to (25):

Stage 1: Feed Forward

All the input in the hidden layers unit is calculated based on zj (j=1,2,…,p)

$$z\_net_j = v_{j0} + \sum_{i=1}^{n} x_i\, v_{ji} \quad (15)$$

Sigmoid activation output 1 function:

$$z_j = f(z\_net_j) = \frac{1}{1+e^{-z\_net_j}} \quad (16)$$

All the network input is calculated in unit yk (k=1,2,…,m)

$$y\_net_k = w_{k0} + \sum_{j=1}^{p} z_i\, w_{ki} \quad (17)$$

Sigmoid activation output 2 function is shown below:

$$y_k = f(y\_net_k) = \frac{1}{1+e^{-y\_net_k}} \quad (18)$$

Stage 2: Error Calculation

Calculate δ output unit based on error in each output unit yk (k=1,2,…,m):

Calculate the rate of wkj weight change with α acceleration rate momentum:

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (19)$$

Where,
K=1,2,…,m; j=0,1,…,p

Calculate δ based on error in each hidden unit Zj(j=1,2,…,p):

$$\delta\_net_j = \sum_{k=1}^{m} \delta_k\, w_{kj} \quad (20)$$

Calculate the vij weight change with α acceleration rate:

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (21)$$

Where,
j=1,2,…,p; j=0,1,…,n

Stage 3: Change of weight

The weight change of the line leading to the output units:

$$w_{kj} = (new) = w_{kj}(now) + \Delta w_{kj} \quad (22)$$

Where,
K=1,2,…,m; j=0,1,…,p

The weight change of the line leading to the hidden units:

$$v_{kj} = (new) = v_{kj}(now) + \Delta v_{ij} \quad (23)$$

Where,
j=1,2,…,p; j=0,1,…,n

Update the epoch values

$$epoch = epoch + 1 \quad (24)$$

The network output is compared with the target by calculating the E error using the formula below:

$$E = t + y_k \quad (25)$$

III. SARIMA RESUTLS

This section presents the forecasting results of SARIMA by using the seven datasets. These results reveal the generalisation capability of the SARIMA model for all the seven datasets. The results of all the datasets involved are analysed based on the RMSE technique. The results of the proposed method for each dataset are analysed and presented in the following subsections. In Table 3, the best result is highlighted in bold fonts. The Sungai Bedup data produce the best result compared with all the other datasets, respectively. In this context, the Sungai Bedup prediction contains a small RMSE value of 0.008. The results of SARIMA for all the datasets show low values of RMSE except that of the Sungai Merang dataset, as shown in Figure 5. This is the main reason: there are gaps of missing information in the dataset of Sungai Merang given by DID. In this context, the missing data might have resulted and affected the accuracy of forecasting. Moreover, the results indicate that the SARIMA model shows superior testing and low RMSE values for all the datasets compared with LSTM and BP, which show high RMSE values in all the datasets.

Table 3: Results of SARIMA

Station No	Dataset	RMSE
1006403	Sungai Bedup	**0.008**
1005079	Sungai Meringgu	0.011
100637	Semuja Nonok	0.023
1006033	Bukit Matuh	0.025
1005447	Sungai Busit	0.025
1105035	Sungai Teb	0.027
1005080	Sungai Merang	0.029

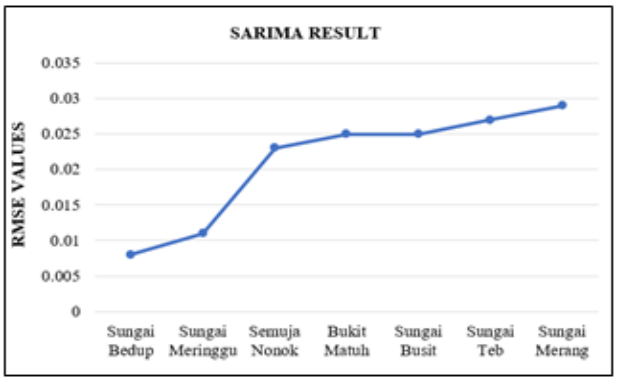

Fig 5: SARIMA Results for the seven datasets

IV. COMPARATIVE RESULT

This research focuses on forecasting flood and the accuracies are measured by RMSE values. The performance of the developed models is evaluated by comparing with those of LSTM and BP. More details about the methodology and the findings of applying both LSTM and BP can be found in [13]. Table 4 and Figure 6 show a comparative analysis of the LSTM model and BP model respectively. Results from this study evidently prove that the SARIMA model is reliable in forecasting the water level of Sungai Bedup with the lowest RMSE value (0.008), compared with those of the LSTM (0.711) and BP (0.08). In this research the SARIMA modelling framework captures the important drivers in the flood forecasting.

Table 4: Comparison between the three models

Dataset	Sungai Bedup		
	SARIMA	Backpropagation	LSTM
RMSE	**0.008**	0.711	0.08

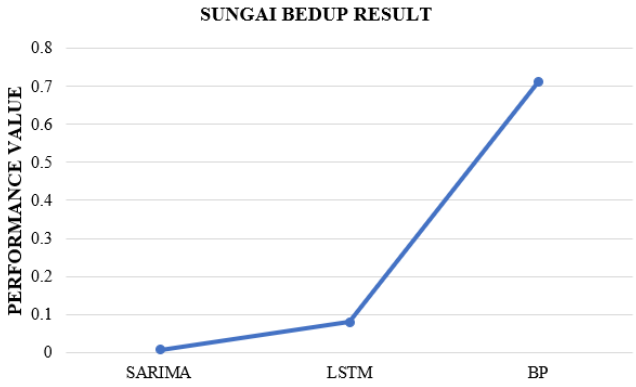

Fig 6: Graph of Weekly Forecasting

V. CONCLUSION

The SARIMA modelling framework captures the important drivers of the datasets. In the future, a combination of experience and good discerning skills will help an individual in determining the best method for a successful modelling effort. More datasets are required in future studies to improve the accuracy of forecasting. The latest generation of spiking neural network could be implemented in future investigations. Another study area could be selected

to carry out flood forecasting by predicting the water levels and tide levels by utilising rainfall datasets provided by DID. The flood prediction system can be adapted to develop other expert systems for purposes such as medical diagnosis, agricultural and malware detection. The model can be embedded in devices that use the mobile app platform. Early flood prediction helps government to save lives and minimise the damages to properties and the environment.

ACKNOWLEDGMENT

This research is supported and funded by Universiti Malaysia Sarawak (UNIMAS), under the Special Grant Scheme (F04/SpGS/1547/2017).

REFERENCES

1. Official Website of Department of Irrigation and Drainage Sarawak (DID).[s.d.].retrieved<http://www.did.sarawak.gov.my/modules/web/pages.php?mod=webpage&sub=page&id=319>. Accessed 03/Jul./18.
2. Hapuarachchi HAP, Wang QJ, Pagano TC (2011), A review of advances in flash flood forecasting. *Hydrological Processes* 25(18): 2771–2784, 2011. ISSN: 08856087, DOI 10.1002/hyp.8040.
3. Miller SG (2017), Wind, Rain, Heat: Health Risks Grow with Extreme Weather. *Live Science*. Retrieved <https://www.livescience.com/57936-climate-change-extreme-weather-health.html>. Accessed 26/Sep./17.
4. Archer DR, Parkin G, Fowler HJ (2017), Assessing long term flash flooding frequency using historical information. *Hydrology Research* 48(1): 1–16. ISSN: 0029-1277,10.2166/nh.2016.031.
5. Moges MA (2017), Suitability of Watershed Models to Predict Distributed Hydrologic Response in the Awramba Watershed in Lake Tana Basin. *Land Degradation & Development* 28(4): 1386–1397. ISSN: 10853278, DOI: 10.1002/ldr.2608.
6. Kruskal JB et al (2017), Big Data and Machine Learning—Strategies for Driving This Bus: A Summary of the 2016 Intersociety Summer Conference. *Journal of the American College of Radiology* 14(6); 811–817. ISSN: 15461440, DOI: 10.1016/j.jacr.2017.02.019.
7. Lecun Y, Bengio Y, Hinton G (2015), Deep learning. *Nature* [s.l.] 521(7553): 436–444. ISSN: 0028-0836, DOI: 10.1038/nature14539.
8. Bustami R et al. (2007), Artificial Neural Network for Precipitation and Water Level Predictions of Bedup River. *IAENG International Journal of Computer Science* 34(2): 228-233.
9. Wang HR, Wang C, Lin X, Kang J (2014), An Improved ARIMA Model for Precipitation Simulations. *Nonlinear Processes Geophysics* 21(6): 1159–1168. Retrieved: <www.nonlin-processes-geophys.net/21/1159/2014/>. Accessed 14/Apr/2018.
10. Kong YL et al. (2017), Long Short-Term Memory Neural Networks for Online Disturbance Detection in Satellite Image Time Series. *Remote Sensing* 10(3): 452. ISSN: 2072-4292, DOI: 10.3390/rs10030452.
11. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, 28(10), 2222-2232.
12. Suseno S, Suryono S, Endro J (2018), Backpropagation Neural Network Algorithm for Water Level Prediction. *International Journal of Computer Applications (0975-8887)* 179(19): 45–51. DOI: 10.5120/ijca2018916336
13. Saleh AY, Tei R (2018), Flood prediction of Sungai Bedup, Serian, Sarawak, Malaysia using Deep Learning. *International Journal of Engineering & Technology* 7(3.22): 55–58. DOI: 10.14419/ijet.v7i3.22.17125