Journal of Optimization in Industrial Engineering
Vol.15, Issue 1, Winter & Spring 2022, 187-193
DOI: 10.22094/JOIE.2021.1941252.1899

Research Article

Daily Rainfall Forecasting Using Meteorology Data with Long Short-Term Memory (LSTM) Network

Soo See Chai [a],*, Kok Luong Goh [b]

[a] *Faculty of Computer Science and Information Technology, University of Malaysia Sarawak (UNIMAS), 94300, Kota Samarahan, Sarawak, Malaysia*

[b] *International College of Advanced Technology Sarawak (i-CATS), Kuching, Sarawak, Malaysia*

Received 25 September 2021; Revised 09 November 2021; Accepted 18 November 2021

Abstract

Rainfall is a natural climatic phenomenon and prediction of its value is crucial for weather forecasting. For time series data forecasting, the Long Short-Term Memory (LSTM) network is shown to be superior as compared to other machine learning algorithms. Therefore, in this research work, a LSTM network is developed to predict daily average rainfall values using meteorological data obtained from the Malaysian Meteorological Department for Kuching, Sarawak, Malaysia. Six daily meteorology data, namely, minimum temperature (°C), maximum temperature (°C), mean temperature (°C), mean wind speed (m/s), mean sea level pressure (hPa) and mean relative humidity (%) from the year 2009 to 2013 were used as the input of the LSTM prediction model. The accuracy of the predicted daily average rainfall was assessed using coefficient determinant (R2) and Root Mean Square Error (RMSE). Contrary to the common practice of dividing the whole available data set into training, validation and testing sub-sets, the developed LSTM model in this study was applied to forecast the daily average rainfall for the month December 2013 while training was done using the data prior of this month. An analysis on the testing data showed that, the data is more spread out in the testing set as compared to the training data. As LSTM requires the right setting of hyper-parameters, an analysis on the effects of the number of maximum epochs and the mini-batch size on the rainfall prediction accuracy were carried out in this study. From the experiments, a five layers LSTM model with number of maximum epoch of 10 and mini-batch size of 100 managed to achieve the best prediction at an average RMSE of 20.67 mm and R2 = 0.82.

Keywords:*Rainfall; LSTM; Prediction; Meteorology; Time-Series Data*

1. Introduction

Accurate forecasting of rainfall is still remaining as a demanding issue in the field of meteorological services. Rainfall contributes significantly in hydrological cycle and its value is critical for water resource planning and management, flood risk prevention and reservoir operation which affects our community (Kumar and et all, 2019; Hernández and et all., 2016).

For some parts of the world, rainfall is the only source of freshwater. Therefore, predicting future rainfall events is very important to help human in planning and adapting strategies (Tran Anh, Duc Dang, & Pham Van, 2019). However, as the rainfall is resulting from various meteorological circumstances which are complex and the mathematical modelling for rainfall prediction is nonlinear (Kashiwao, 2017), the design of an effective rainfall prediction system is still remaining as a difficult task for researchers. Traditionally, rainfall prediction methods were mainly focusing on the Numerical Weather Prediction (NWP) and statistical model (Liu and et all., 2019). Yet, it was reported that, serious constraints had been noticed in these two models when there is a dynamic change and linear shift of rainfall (Darji, Dabhi, & Prajapati, 2015).

In recent years, data-driven models have gained popularity in the field of hydrological variables prediction problems (Mandal & Jothiprakash, 2012). Data-driven models are basically methods that use computational intelligence and machine learning algorithms with the existence of enormous volume of data accounting the modelled phenomena (Solomatine, 2006). There has been a lot of machine learning algorithms that are proposed for time series analysis. Among these various machine learning algorithms, deep-learning, which is rooted from conventional neural network, has shown to outperform its predecessors (Pouyanfar , 2018). This is due to the expansion and accessibility of data as well as the significant improvement in hardware technologies which had driven the advancement of deep learning studies. Deep learning uses graph theories with transformation between neurons to develop learning models with multiple layers. One of the major advantages of deep learning as compared to conventional neural network is that, the feature extraction in deep learning algorithms are performed automatically while the efficiency of the conventional neural network models relies on the goodness of the representation of the input data. In addition, features extraction is domain particular and involves tremendous amount of human work.

Recurrent neural network (RNN) is a unique kind of deep learning model which has the ability to explicitly

*Corresponding author Email address: rai.ammish@mountsafa.com

handle the order of input observations. In other words, RNN has the ability to learn the temporal context of the input sequences for better prediction. Long Short-Term Memory (LSTM) network, which is famous for its ability to adjust some hyper-parameters (Jia and et al., 2017) and capture the long temporal features of the input data, is a unique type of RNN model (Nakisa and et al., 2018). In machine learning, the values of the hyper-parameters are used to control the learning processes and fine-tuning these values is essential to exploit the network's functionality. Temporal characteristics of the historical rainfall data could be used to analyse and identify the phases of seasons. LSTM has shown great success in various areas, including rainfall prediction. However, the use of LSTM in short-term precipitation prediction, i.e. less than 24 hours, which has a mandatory impact on natural disaster management systems that happen in a very short period of time, like flash flood forecasting and thunder storm alert, is limited (Akbari Asanjan and et al., 2018). Long-term precipitation, on the other hand, typically range from a month to a seasonal forecast. Short-term precipitation prediction is an extremely tough problem due to the non-uniform and flawed characteristics of the meteorological structure over time (Kumar and et al., 2020).

LSTM models of different architecture and optimization algorithms were found to be used for precipitation prediction from the review. In addition, the selection of the hyper-parameters was mostly based on the processing power of the system used to develop the model. Although the effects of the hyper-parameters on the LSTM performance are acknowledged, the discussion on the selection on the hyper-parameters especially on the rainfall prediction using meteorology data is very little. Therefore, in this study, we propose the use of LSTM model for 1-day short-term rainfall prediction, focusing particularly on analysing the effects of two hyper-parameters, i.e. the number of maximum epochs and mini-batch size of the network model, in the prediction accuracy, using meteorology data.

2. Data

Daily mean precipitation (mm), together with the other six daily meteorology variables, namely daily minimum temperature (°C), daily maximum temperature (°C), daily mean temperature (°C), daily mean relative humidity (%), daily mean wind speed (m/s) and daily mean sea level pressure (hPa) were obtained for Kuching city from Department of Meteorology Malaysia (Sarawak Branch) for the year 2009 to 2013. The mean and standard deviation for each of these variables could be seen in Table 1.

Kuching, the capital city of Sarawak, exhibits a tropical rainforest climate with hot, humid and wet weather conditions throughout the year and is the wettest city in Malaysia. During the wet monsoon season, normally happens in the month of November to February, the city would be showered with high density of rainfall which sometimes can last for days. The average annual precipitation in Kuching is around 4200mm with approximately 250 rainy days per year. Due to the high precipitation intensity, flash flood is common in Kuching notably during the monsoon season. Therefore, there is an urgent need to be able to predict the future average daily rainfall. From Table 1, it could be seen that, the variation in terms of rainfall within a year is quite high. This is mainly due to the not so clear-cut two seasons in Kuching: drier season during the months April to September and wetter season from the month October to March.

Table 1
The mean and standard deviation (Std) distribution for all the seven variables obtained from the Department of Meteorology Malaysia (Sarawak Branch) for Kuching city from year 2009 to 2013.

Variables		**Year**				
		2009	**2010**	**2011**	**2012**	**2013**
Daily Max Temperature (^{0}C)	Mean	31.40	31.45	31.52	31.87	31.20
	Std	1.74	1.47	1.72	1.59	1.84
Daily Min Temperature (^{0}C)	Mean	23.49	23.27	23.48	23.57	23.66
	Std	0.59	0.63	0.61	0.60	0.61
Daily Mean Temperature (^{0}C)	Mean	26.35	26.15	26.33	26.52	26.33
	Std	0.89	0.84	0.90	0.88	0.87
Daily Mean Relative Humidity(%)	Mean	85.27	85.71	86.49	85.02	86.33
	Std	4.34	4.15	4.10	3.96	4.30
Daily Mean Wind Speed (m/s)	Mean	1.73	1.76	1.79	1.78	2.02
	Std	0.28	0.32	0.31	0.31	0.46
Daily Mean Sea Level Pressure (hPa)	Mean	1010.21	1008.80	1009.24	1009.21	1009.29
	Std	1.30	1.06	1.27	1.30	1.22
Rainfall (mm)	Mean	14.00	11.86	21.22	11.61	17.65
	Std	25.35	18.92	35.81	19.53	26.95

3. Methodology

There were a total of 1825 data for each variable (365 days x 5 years) obtained from the Department of Meteorology Malaysia (Sarawak Branch) for the year 2009 to 2013. The data will go through data pre-processing and normalization stages before the data division stage which will separate the data into training and testing data sets for the LSTM model. Figure 1 summarizes the overall methodology of this study.

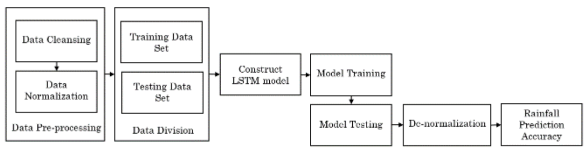

Figure 1. Overall methodology

3.1. Data cleansing

The data obtained contained defective values, which could be the result of missing values during data collection. As a pre-processing step, these values were cleaned from the data set. A total of 11 defective values were found from the total of 1825 data, resulting a remaining of 1814 data for further processing. These defective values were denoted as -1.1 in the original dataset.

3.2. Data Normalization/ De-normalization

The cleaned data is next normalized using z-score normalization in order to convert the variables to a common scale with an average of zero and standard deviation of one using the formula below:

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

where :

z : Normalized data,
x : Observed data
μ : Mean of the sample
σ : Standard deviation of the sample

In this research study, the normalization was done according to each variable. In other words, the mean and standard deviation of each variable were used to normalize the variable itself. Moreover, the training and testing data set used the standard deviation and mean from their own set of data for each variable.
After the LSTM model has predicted the rainfall values, these values will be de-normalized using the mean and standard deviation of the rainfall data of the training and testing set to compute the accuracy of the prediction for the training and testing data set respectively.

3.3. Data division

Contrary to the common practice of dividing the whole dataset into training, validation and testing data set, the prediction will be done for the month December 2013 in this study. The rationale of this is that, the forecasting is for a future unseen month. The training data set will consist of 59 months of meteorology data (from January 2009 to November 2013) while the testing data will be the 1-month of December 2013. The distribution of the mean and standard deviation of all the seven parameters in the training and testing data are listed in Table 2. From this table, it is clear that the testing data set has an average rainfall of much higher as compared to the training data. In addition, by looking at the higher standard deviation of the rainfall data of the testing data, it could be concluded that, the rainfall data in the testing data set is more spread out. Another observation from Table 2 is that, despite the large difference in the mean and standard deviation of the rainfall data in the testing data set, the mean and standard deviation for the other variables were quite consistent.

Table 2
The Mean and Standard deviation (Std) of the training and testing data set.

Variables		**Training Data Set**	**Testing Data Set**
Number of Data		1784	31
Daily Max Temperature (^{0}C)	Mean	31.99	30.40
	Std	1.89	1.79
Daily Min Temperature (^{0}C)	Mean	23.52	21.81
	Std	0.71	0.50
Daily Mean Temperature (^{0}C)	Mean	26.63	26.04
	Std	1.05	0.68
Daily Mean Relative Humidity(%)	Mean	84.49	88.17
	Std	4.91	3.70
Daily Mean Wind Speed (m/s)	Mean	1.80	2.24
	Std	0.32	0.60
Daily Mean Sea Level Pressure (hPa)	Mean	1009.55	1009.02
	Std	1.34	1.09
Rainfall (mm)	Mean	11.76	23.54
	Std	22.98	30.91

3.4. Long Short-Term memory (LSTM) network model

LSTM model is a unique type of Recurrent Neural Network (RNN) network model that was developed specially to model temporal series (Salman and et al. 2018). This special architecture was first presented by Hochreiter and Schmidhuber (Hochreiter & Schmidhuber 1997) to solve the vanishing and exploding gradients issues, i.e. the error backflow problem in the backpropagation algorithm, in RNN. This is mainly because the traditional RNN model could not learn the long-distance correlation in a long sequence as the gradient vector component deteriorate rapidly over the long distance (Bengio, Simard, & Frascon, 1994; Hochreiter, 1998). LSTM model offers a remedy to this by integrating memory units. With this memory units, the LSTM model will have the ability to learn when to disregard the past hidden states and when to update the hidden states with the given new information (Chen, 2016).

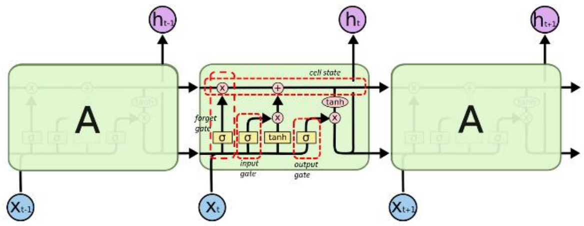

Fig. 2. A typical LSTM block (Sakinah and et al., 2019)

An LSTM block has hidden states as in the traditional RNN model. On top of this, the LSTM block is added with a cell state and three different gates, namely input gate, output gate, and forget gate as shown in Figure 2. The cell state, which is also the “memory” of the network, acts as a transport highway that carries relative information all the way down the sequence chain. Using Sigmoid activation function in the forget gate, the data enter into the LSTM model will be processed in order to decide which data to be stored or discard on the memory cell. In the next step, the decision is to be made on what new information will be stored in the cell state. This particular process is divided into two phases: 1. Using the sigmoid layer, called as the input gate layer, a decision is to be made on which values will be updated, and 2. Using tanh activation function to create a new value vector to store at the memory cell. The combination of the values from the forget and input gates will substitute the former value in the memory cell of the cell gate. Two operations will be implemented in the output gates. In the first operation, using sigmoid activation function, the decision on which part the memory cell to be freed will be made. In the second operation, a new value will be positioned in the memory cell using the tanh activation function. This new value will be calculated by multiplication of these two functions. The main parts of an LSTM network are the input and LSTM block/layer. The input layer is the layer where sequence or time series data are inputted into the model. The LSTM block/layer will learn the long-term dependencies between the time steps of the sequence data. In Figure 2, the repeating modules (labelled as A) in an LSTM architecture are shown. In the central block shown in Figure 2, input x_t together with an input in the time point t-1 are received. Next, it generates output h_t which is also an input in the time point t+1.

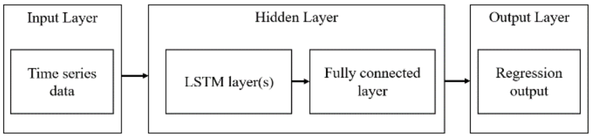

Fig. 3. LSTM architecture used.

In our proposed methodology, a five layers network model as shown in Figure 3 is developed: one input layer, two LSTM blocks/layers, one fully connected layer and one regression output layer. In the input layer, the six meteorology data (daily maximum temperature (^{0}C), daily minimum temperature (^{0}C), daily mean temperature (^{0}C), daily mean relative humidity (%), daily mean wind speed (m/s) and daily mean sea level pressure (hPa) were used. The hidden units in LSTM layer 1 and LSTM layer 2 are set to 10 in each layer. The output layer consisted of the one response which is the daily mean rainfall. The size of the fully connected layer was set to the number of response in the output layer, which is one in this case. The Adaptive Moment Estimation (ADAM) optimization algorithm (Kingma & Ba, 2014) is used to train the LSTM model. ADAM, an adaptive learning rate algorithm, will calculate the learning rates for each individual parameter. The advantages of ADAM included efficient stochastic optimization which only needs first-order gradients with little memory requirement (Jiang & Chen, 2017).

3.5. LSTM Hyper-parameters

Regardless of the application areas of LSTM, there are several hyper-parameters that affect the performance of the network. Hyper-parameters are configurations that are external to the model and whose values cannot be estimated from the data(Paper, 2020). These values require tuning in order to harness the ability of the network. The tuning process is tedious as it is normally non-intuitive, time-consuming and systematic trial-and-error procedures. In this research work, the effects of the two hyper-parameters, i.e. maximum epoch and mini batch size, on the effects on the LSTM performances are analysed. These two hyper-parameters are explained below:

Maximum number of Epochs

An epoch is defined as the thorough pass of the training algorithm over the entire training set. The maximum epoch is the maximum number of full pass of the training algorithm through the entire training set. In this study, the maximum number of epochs was set with values ranging from 10 to 1000, in order to analyse the effects of this hyper-parameter.

Mini batch-size

The mini-batch size is defined as a sub group of the training data set that is used to assess the gradient of the loss function and to update the weights. In neural network training, stochastic gradient descent optimization algorithm is used. A general gradient descent algorithm is defined as:

$$\theta = \theta - \eta * \nabla_\theta J(\theta) \quad (1)$$

where $J(\theta)$: objective function,

θ : model's parameters, and

η : learning rate which determines the size of the steps taken to reach a local minimum

A stochastic gradient descent algorithm is a variation of the general gradient descent algorithm and can be calculated using the formula below:

$$\theta = \theta - \eta * \nabla_\theta J(\theta;\ x^{(i)}; y^{(i)}) \quad (2)$$

where $x^{(i)}$ is the training example, and

$y^{(i)}$ is the label

Using the stochastic gradient descent optimization algorithm, the present state of the model will be used for prediction. The prediction will next be compared with the expected values and the difference of these two values is used to estimate the error or loss gradient (Brownlee, 2018). This loss gradient, which is a statistical estimate, is then used to update the model weights and the process is repeated. From here, it could be seen that, the more training examples used in the estimate, the more accurate this estimate will be and more likely that the weights of the network will be adjusted in a way that will improve the performance of the model. The improved estimate of the loss gradient comes at the cost of having to use the model to make many more predictions before the estimate can be calculated, and in turn, the weights updated. The number of training examples used in the estimate of the error gradient is a hyperparameter for the learning algorithm called the "batch size". Batch size controls the accuracy of the estimate of the error gradient when training neural networks(Masters & Luschi, 2018). A mini batch-size is a configuration of the batch size anywhere in between, e.g. more than 1 example and less than the number of examples in the training dataset. This sub-unit of the training data has two purpose: 1. to assess the gradient of the loss function, as well as, 2. to update the weights. The neural network weights are updated using the gradients values. However, as the gradients travel through the network, the gradients will start to vanish. There will be not much learning if the gradients values become extremely small. In this study, this number was set to 100 at the very beginning when running the analysis on the number of maximum epochs. Next, using the number of maximum epochs which achieves the lowest RMSE (Root Mean Square Error) value, the experiments on the mini-batch size, which ranges from 10 to 1000 were carried out.

4. Results & Analysis

4.1. Performance measurements parameters

To assess the efficiency of the developed model, two performance measurements are calculated, which are the Root Mean Square Error (RMSE) and correlation of determinant (R^2). The RMSE will assess how closely the predicted values match the observed values, i.e. measures that overall agreement between the actual and predicted values. There is no upper bound and the value 0 means the model is perfect. The calculation of RMSE was done using equation (3):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(Forecasted_i - Actual_i\right)^2}{N}} \tag{3}$$

The coefficient of determinant (R^2), also called goodness of fit, measures the strength of the linear relationship between the forecasted and actual rainfall value. R^2 ranges in between 0.0 to 1.0 with value of 1.0 suggests a perfect fit between the two variables and 0.0 would mean the model did not manage to model the data accurately.

In addition to the RMSE and R^2 values, the training elapsed time (sec), which is the amount of time MATLAB takes to accomplish the training process on the Intel® Xeon ® CPU at 2.4 GHz system with 32GB was also recorded.

4.2. Analysing the effects of maximum epochs

An epoch is defined as the thorough pass of the training algorithm over the entire training set. To analyse the effects of the maximum epoch, experiments with the maximum number of epochs being set to 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 500 and 1000 were carried out and the results in terms of RMSE (mm), R^2 and the average training elapsed time (sec) were shown in Table 3. For each experiment, the LSTM model is run ten times and the average RMSE, R^2 and training elapsed time were calculated. The size of the mini-batch was set to 100 for this part of experiment.

Table 3
Effects of the maximum epochs on the rainfall prediction using the proposed LSTM architecture.

#Max Epoch	**RMSE (mm)**	**R^2**	**Training Elapsed Time (sec)**
10	20.67	0.82	5.10
20	21.50	0.75	8.60
30	22.62	0.68	11.50
40	22.68	0.70	14.30
50	22.47	0.71	17.70
60	22.07	0.70	20.00
70	24.20	0.64	23.80
80	23.79	0.64	26.20
90	23.22	0.70	30.30
100	23.83	0.62	36.00
200	24.48	0.59	72.00
300	24.32	0.61	95.60
500	23.35	0.67	154.60
1000	23.67	0.65	338.60

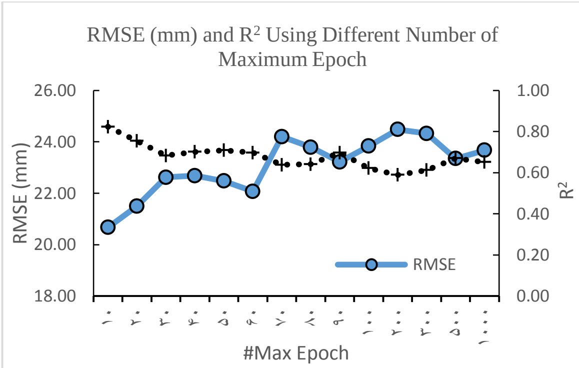

Fig. 3. Line graph showings the RMSE and R^2 obtained with different number of maximum epoch used.

From Table 3, it could be seen that, the best RMSE achieved was 20.67 mm with R^2 = 0.82 with the number of maximum epochs equals to 10. The worst RMSE value achieved was 24.48 mm with lowest R^2 = 0.59 when the maximum epochs were 200. There was a difference of around 18% between the lowest and highest RMSE obtained. It could be concluded that, when the number of maximum epochs increases, the accuracy of the LSTM model deteriorates, with RMSE values getting larger and R^2 values getting smaller. When the number of maximum epochs rises, more time was needed to train the LSTM model. The average training elapsed time for each maximum epochs are shown in Table 3. Therefore, it can be concluded that, the RMSE and R^2 values show that, the accuracy of the LSTM network model is affected by the number of maximum epochs used. In this case, the increase of the number of maximum epochs does not improve the accuracy of the prediction made.

4.3. Analysing the effects of mini-batch size

The mini-batch size is defined as a sub group of the training data set that is used to assess the gradient of the loss function and to update the weights. Using small batch size is known to achieve faster convergence as the LSTM model with small batch size will start to learn before seeing all the data. However, there is no guarantee that the network would converge to the global optima. In this part of the experiments, the mini-batch size was set to 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500 and 1000. For each mini-batch size, the LSTM model was trained and tested 10 times and the average RMSE, R^2 and training time elapsed were calculated. The maximum number of epochs was set to 10, which is the lowest RMSE with highest R^2 values obtained from the previous experiment.

Table 4
Effects of the different mini-batch size on the rainfall prediction using the proposed LSTM architecture.

Mini-batch Size	RMSE (mm)	R^2	Training Elapsed Time (sec)
10	21.48	0.78	5.30
20	22.26	0.77	5.20
30	21.40	0.79	5.30
40	21.47	0.77	5.50
50	22.80	0.71	5.30
60	21.21	0.79	5.40
70	21.86	0.77	5.40
80	21.49	0.76	5.40
90	21.50	0.78	5.40
100	20.67	0.82	5.10
200	20.83	0.79	5.50
300	21.21	0.79	5.20
400	21.48	0.78	5.30
500	21.81	0.76	5.40
1000	21.45	0.79	5.40

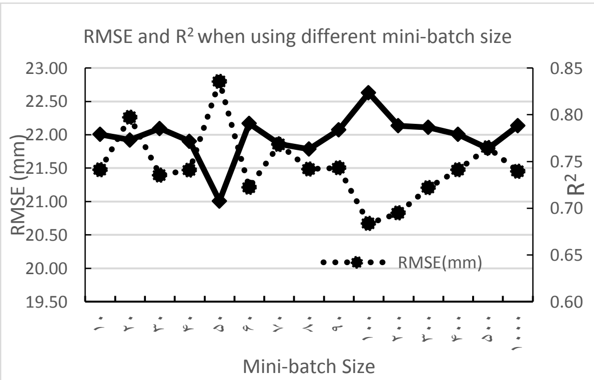

Fig. 4. Line graph showings the RMSE and R^2 obtained using different number of mini-batch size.

From Table 4, the lowest RMSE obtained was 20.67 mm with R^2 = 0.82 using mini-batch size of 100. Figure 4 shows the RMSE and R^2 values obtained in this part of the experiment. From Figure 4, it could be seen that, the performance of the LSTM model increases, although not consistently, when the mini-batch size increases from 10 to 100, and started to decrease after the 100 mini-batch size. The training elapsed time was quite consistent for all the different number of mini-batch size used.

4.4. Rainfall prediction

Using the proposed LSTM architecture as explained above, and with maximum epochs and mini-batch size set to 10 and 100 respectively, the network model is trained and tested for 10 times using 59 months of data for training and the last one month of the data for testing. The RMSE and R^2 for the rainfall prediction obtained are shown in Table 5. In addition, the elapsed time in seconds for training of the model is also shown. The average RMSE obtained was 20.67 mm with R^2 = 0.82. The average training elapsed time was 5.10 sec.

Table 5
RMSE, R^2 and Training Elapsed time (sec) for the 10 runs using the developed model to predict the daily mean rainfall (mm)

Run	RMSE (mm)	R^2	Training Elapsed Time (sec)
1	22.15	0.80	5.00
2	19.35	0.82	5.00
3	20.91	0.79	5.00
4	19.07	0.87	5.00
5	22.76	0.87	5.00
6	20.56	0.78	6.00
7	19.06	0.83	5.00
8	22.55	0.78	5.00
9	20.16	0.85	5.00
10	20.13	0.86	5.00
Average	20.67	0.82	5.10

5. Conclusion

In this research work, a five layers LSTM network model was designed and developed to predict rainfall values using six meteorology data obtained from the Department of Meteorology Malaysia (Sarawak Branch) for the year 2009 to 2013. An analysis on the number of maximum epochs and the mini-batch size of the LSTM network model on the accuracy of the daily rainfall prediction was done. It was revealed that, the prediction accuracy is influenced by these two parameters. For the maximum epochs, the increase of this number shows that, the accuracy of the rainfall prediction deteriorates. In terms of mini-batch size, the rainfall prediction accuracy increases when the mini-batch size increases from 10 to 100, and started to decrease after the 100 mini-batch size. The best prediction accuracy obtained was RMSE = 20.67 mm with R^2 = 0.82. A further improvement of the current research would be a study on the relationship between the size of training set on the prediction accuracy.

References

Kumar, D. , Singh, A., Samui, P. & Jha, R K..(2019), "Forecasting monthly precipitation using sequential modelling," *Hydrological sciences journal*, 64(6), 690-700 .

Hernández, E. Sanchez-Anguix,V. Julian, V. Palanca, J. & Duque, N. (2016) "Rainfall Prediction: A Deep Learning Approach," Cham,Springer International Publishing, in Hybrid Artificial Intelligent Systems, 151-162.

Tran Anh, D., Duc Dang, T. &. Pham Van, S. (2019) "Improved Rainfall Prediction Using Combined Pre-Processing Methods and Feed-Forward Neural Networks," *J — Multidisciplinary Scientific Journal*, 2,(1), 65-83, [Online]. Available: https://www.mdpi.com/2571-8800/2/1/6.

Kashiwao, Nakayama, Ando, T. K., S. K. M. Lee, & Bahadori, A. (2017) "A neural network-based local rainfall prediction system using meteorological data on the Internet: A case study using data from the Japan Meteorological Agency," *Applied Soft Computing*, 56, 317-330.

Liu,Q., Zou,Y. Liu, X. & Linge, N. (2019) "A survey on rainfall forecasting using artificial neural network," *International Journal of Embedded Systems*,11,(2), 240-249.

Darji, M. P., Dabhi,V. K. & Prajapati, H. B. (2015)"Rainfall forecasting using neural network: A survey," in *2015 International Conference on Advances in Computer Engineering and Applications*, IEEE, 706-713.

Mandal, T. & Jothiprakash, V. (2012) "Short-term rainfall prediction using ANN and MT techniques," *ISH Journal of Hydraulic Engineering*, 18(1), 20-26.

Solomatine, D. P. (2006)"Data‐ driven modeling and computational intelligence methods in hydrology," *Encyclopedia of hydrological sciences*.

Pouyanfar S. & *et al.*,(2018) "A survey on deep learning: Algorithms, techniques, and applications," *ACM Computing Surveys (CSUR),* 51(5), 1-36, 2018.

Jia, Y., Wu, J., Ben-Akiva, Seshadri, M. R. & Du, Y. (2017) "Rainfall-integrated traffic speed prediction using deep learning method," *IET Intelligent Transport Systems*, 11(9), 531-536, 2017.

Nakisa, B., Rastgoo, M. N., A. Rakotonirainy, F. Maire, & Chandran,V. (2018) "Long short term memory hyperparameter optimization for a neural network based emotion recognition framework," *IEEE Access*,6, 49325-49338.

Akbari Asanjan, A., Yang,T. Hsu,K. Sorooshian,S., Lin, J. & Peng, Q. (2018) "Short‐ term precipitation forecast based on the PERSIANN system and LSTM recurrent neural networks," *Journal of Geophysical Research: Atmospheres*,123(22), 12,543-12,563.

Kumar, A. Islam, T. Sekimoto, Y. Mattmann, C. & Wilson, B. (2020) "Convcast: An embedded convolutional LSTM based architecture for precipitation nowcasting using satellite data," *Plos one*,15(3), e0230114.

Salman, A. Heryadi, G., Abdurahman, Y. E. & Suparta,W. (2018) "Single layer & multi-layer long short-term memory (LSTM) model with intermediate variables for weather forecasting," *Procedia Computer Science,* 135, 89-98.

Hochreiter, S. & Schmidhuber, J.,(1997) "LSTM can solve hard long time lag problems," in *Advances in neural information processing systems*, 473-479.

Bengio,Y., Simard,P. & Frasconi,P., (1994) "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks,* 5(2),157-166.

Hochreiter, S.(1998) "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6,(2), 107-116.

Chen,Y., Liu, He,.S., Liu, S. K. & Zhao, J. (2016)"Event extraction via bidirectional long short-term memory tensor neural networks," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*: Springer, 190-203.

Sakinah, N., Tahir, Badriyah, M. T. & Syarif, I. (2019) "LSTM With Adam Optimization-Powered High Accuracy Preeclampsia Classification," in *2019 International Electronics Symposium (IES)*, IEEE, 314-319.

Kingma, D. P. & Ba J.(2014) "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*.

Jiang, S. & Chen, Y. (2017)"Hand gesture recognition by using 3DCNN and LSTM with adam optimizer," in *Pacific Rim Conference on Multimedia*, Springer, 743-753.

Paper, D.,(2020)"Scikit-Learn Classifier Tuning from Complex Training Sets," *Hands-on Scikit-Learn for Machine Learning Applications: Data Science Fundamentals with Python*, 165-188.

Brownlee, J. (2018) *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions*. Machine Learning Mastery.

Masters, D. & Luschi, C.(2018) "Revisiting small batch training for deep neural networks," *arXiv preprint arXiv:1804.07612*.

This article can be cited:
Chai, S., Goh, K. (2022). Daily Rainfall Forecasting Using Meteorology Data with Long Short-Term Memory (LSTM) Network. *Journal of Optimization in Industrial Engineering*, 15(1), 187-193.

http://www.qjie.ir/article_686436.html
DOI: 10.22094/joie.2021.1941252.1899