Flood Prediction Using Multilayer Perceptron Networks and Long Short-Term Memory Networks at Thu Bon-Vu Gia Catchment, Vietnam

Duy Vu Luu[1], Thi Ngoc Canh Doan[2], Khanh Le Nguyen[2], and Ngoc Duong Vo[3](✉)

[1] The University of Danang – University of Technology and Education, 48 Cao Thang, Danang, Vietnam
ldvu@ute.udn.vn

[2] The University of Danang – University of Economics, Danang, Vietnam
{canhdtn,khanh.le}@due.edu.vn

[3] The University of Danang – University of Science and Technology, Danang, Vietnam
vnduong@dut.udn.vn

Abstract. There is a significant change in the amplitude of rainfall between the rainy season and the dry season at Thu Bon-Vu Gia catchment in Vietnam. 65% to 80% of the annual rainfall is in the rainy season. Therefore, Thu Bon - Vu Gia catchment is a highly flood prone region. Floods frequently occur in this area and destroy critical infrastructure. This study compares Multilayer Perceptron (MLP) networks and Long Short-Term Memory (LSTM) networks in forecasting floods at Thu Bon-Vu Gia catchment. Discharges at the downstream point are predicted by utilizing periodic rainfall and flow data at upstream locations. Both models do not use other hydrologic, geological and meteorological data, which have low quality at the study site. Both models are reliable to forecast the flood in the catchment when the values of RMSE and NSE of the models are about 320 m^3/s and 0.5 respectively.

Keywords: Multilayer Perceptron (MLP) · Long Short-Term Memory (LSTM) · Thu Bon-Vu Gia catchment

1 Introduction

Floods create large social, economic, and environmental disruption, adversely affecting both individuals and communities. For example, in 2007, the flood in Quang Nam province, Vietnam caused approximately US$ 53 million worth of property and crop damage [1]. To help people better respond to floods appropriately, hydrological models, such as forecasting flood ones, are considered one of the workable solutions [2].

There are three types of hydrological models, comprising conceptual models, physical models and empirical models in which each model has its own advantages and

N.-S. Vo et al. (Eds.): INISCOM 2021, LNICST 379, pp. 393–402, 2021.
https://doi.org/10.1007/978-3-030-77424-0_32

disadvantages. First, conceptual models are suitable to predict floods in different physiographic regions, even ungauged catchments [3]. Second, physical models offer deep insight into real hydrological systems [3] and require large volumes of accurate data, such as meteorological and hydrological data. Third, without analyzing hydrological processes, empirical models are able to simulate relationships between inputs and outputs by utilizing mathematical equations. Therefore, the empirical models are suitable for this catchment because they do not require many kinds of data, such as geological and meteorological data, which have low quality at the study site.

In empirical models, various architectures and algorithms are adopted in forecasting floods. As mentioned by Thirumalaiah and Deo [4], all training algorithms should be tested in a specific condition to ensure their validity. Meanwhile, currently, there is a lack of applications of neural networks to Thu Bon-Vu Gia catchment. Therefore, this paper aims to compare the prediction ability of the LSTM model and the MLP model which are empirical models at the area.

2 Study Area and Data

The Thu Bon-Vu Gia river system is one of the largest river systems in Vietnam. The catchment area is over 10,000 km^2, partly meeting water requirements of Quang Nam province and Da Nang city. However, frequent floods and droughts in the Thu Bon-Vu Gia river system make water supplies come under increased stress. Furthermore, natural disasters cause 6.3% damage to Quang Nam province's GDP annually, even 20% in years with severe floods [5].

The paper aims to establish a model to generate discharge from rainfall at Nong Son gauging station in the catchment. Due to insufficiency of rainfall and discharge data for less than 24 h in this catchment, we use daily data provided by the Mid-Central Region

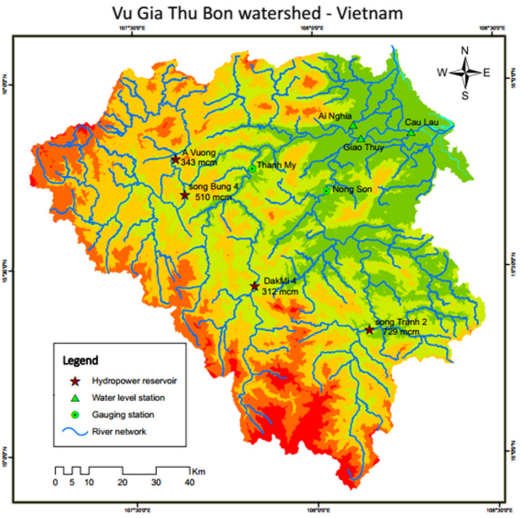

Fig. 1. Thu Bon-Vu Gia catchment.

Centre for Hydrometeorological Forecasting, which are normalized before being used. The input data include rainfall data in Tra My, Nong Son, Tien Phuoc and Hiep Duc station, and discharge data in Nong Son station from 1991 to 2010 (Fig. 1).

3 Methodology

3.1 Introduction to MLP

MLP has been popular in recent years. It has a wide range of applications, such as hand-written character recognition [6]. It includes an input layer, an output layer, and one or more hidden layers. This technique does not include any direct data-flow loop.

The nodes in the layers are fully or partially connected to each other. The relationships between nodes are presented by the weights. In the input layer, the nodes receive input data. In the hidden layers, input data of a node is multiplied by the weight and then added with a bias value. It is then transferred via an activation function. The outcome is sent to the output layer (Fig. 2).

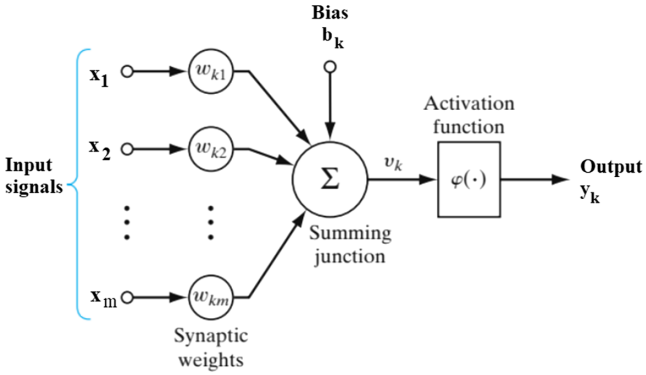

Fig. 2. An MLP with one hidden layer [7]

The operation of a node k at the hidden layers is calculated as:

$$I_k = \sum w_{ki} x_i + b_k \tag{1}$$

$$y_k = \varphi(I_k) \tag{2}$$

in which y_k is the output of the node k; φ is an activation function; x_i is an input i of the node k; w_{ki} is the weight; b_k is the weight.

There are some kinds of the activation function, such as Log-sigmoid transfer function, Hyperbolic tangent transfer function, and Purelin transfer function Training.

To modify the weight values, the backpropagation through time (BPTT) algorithm is recommended. There are four main steps in BPTT. First, the initial weight values are generated randomly. Second, the weight values are used to produce the output values.

Third, the error values between the observed outputs and modeled outputs are calculated as

$$E = \frac{1}{2}(Y_{obs} - Y_{mod})^2 \tag{3}$$

where E is the global error function; Y_{obs} is the observed output; Y_{mod} is the modeled output. Finally, the error values are used to adjust weights to minimize the error. The gradient of the error term is calculated as the formula (4). The model parameters are modified using the gradient descent rule as the formula (5).

$$\frac{\partial E}{\partial W} = \sum\nolimits_{t=1}^{T} \frac{\partial E_t}{\partial W} \tag{4}$$

$$W \leftarrow W - \alpha \frac{\partial E}{\partial W} \tag{5}$$

3.2 Introduction to Long Short-Term Memory

Recurrent Neural Networks (RNN) has developed since the 1980s. It's applications range from generating text [8], language modeling [9] to forecasting time series [10]. The RNN model builds a directed cycle based on connections among nodes. It helps the RNN model remember prior important information in the data.

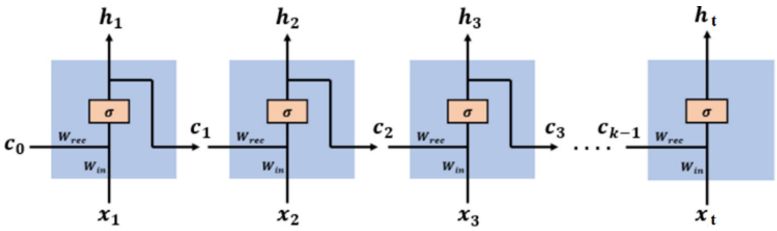

Fig. 3. Unfolding of recurrent neural networks [11]

A basic RNN model shown in Fig. 3 includes one input layer, one hidden layer and one output layer. At time step t, the model input includes an input x_t and a network state c_{t-1} storing previous information. Afterwards they are transferred to the hidden layer. Weight matrices, W_{rec} and W_{in}, go with c_{t-1} and x_t respectively. The activation function is the sigmoid function in the hidden layer. The output h_t is used to calculate the prediction error E. Next, a backpropagation through time (BPTT) training algorithm is also used to update the weight matrices.

BPTT has two common problems, which are the vanishing gradient and the exploding gradient. While the vanishing gradient makes the gradient quickly move to zero, the exploding gradient causes the parameters become too large.

LSTM, a kind of RNN, is recommended to help the model avoid the vanishing gradient problem. LSTM is applied in many topics, from prediction to face detection.

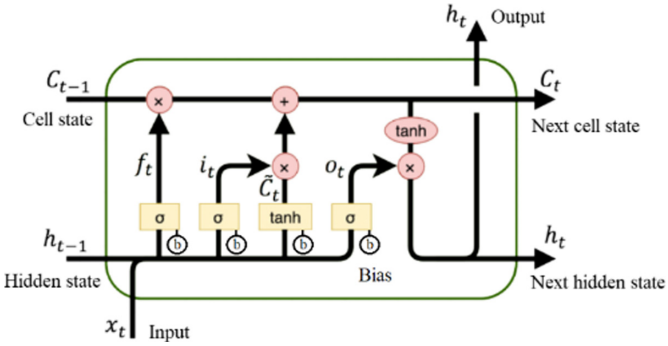

Fig. 4. LSTM cell [11]

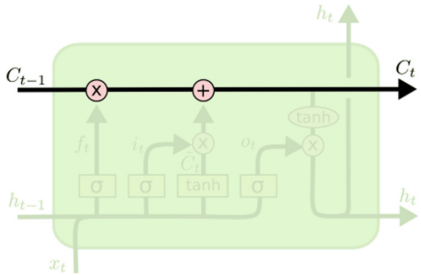

Fig. 5. Cell state in LSTM [11].

LSTM can remember prior important things and forget unnecessary information [12]. A basic LSTM model has three gates: an input gate, an output gate and a forget gate. The gates have different weights and biases (Figs. 4 and 5).

In the forget gate, f_t, input, h_{t-1} and x_t, are passed or forgot. Output of this gate being from 0 to 1 is computed as

$$f_t = \sigma(W_t.[h_{t-1}, x_t] + b_i] \tag{6}$$

where σ is the sigmoid function; w_t is the weight value; h_{t-1} is output of the module at the last time $(t-1)$; x_t is input at time t, and b_i is the bias value. When f_t is 1, the module completely stores the whole information. In contrast, when f_t is 0, the module completely forgets the received information (Fig. 6).

The input gate controls what information is transferred to the cell state. The output is determined by using the following equations:

$$i_t.\tilde{C}_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \cdot tanh(W_c.[h_{t-1}, x_t] + b_c \tag{7}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t \tag{8}$$

where C_t and C_{t-1} are the cell states at time t and $t-1$ respectively (Fig. 7).

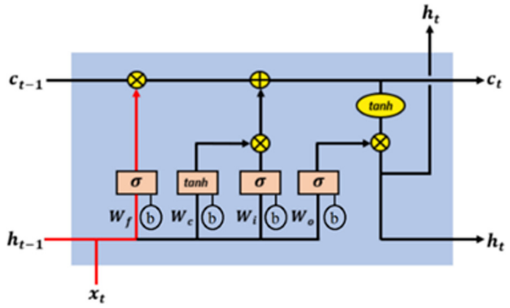

Fig. 6. The *LSTM forget gate* [11].

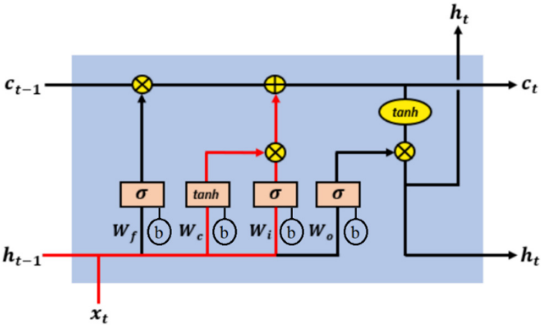

Fig. 7. The LSTM input gate [11].

The output gate controls the data flow in the cell state transferred to the next step. The output value, h_t, is computed as

$$h_t = O_i \tanh(C_t) \tag{9}$$

$$O_t = \sigma(W_0[h_{t-1}, x_t] + b_0] \tag{10}$$

where b_o is the bias value; and W_o is the weight value at the output gate (Fig. 8).

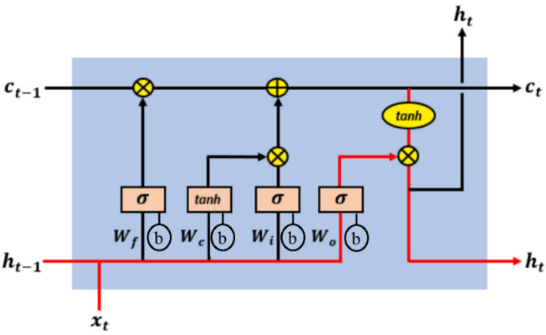

Fig. 8. The LSTM output gate [11].

4 Model Development

The research builds both the MLP model and the LSTM model. They generate the daily discharge data from the daily rainfall data in the catchment between 1991 and 2010. The rainfall data at Tra My, Tien Phuoc, Hiep Duc and Nong Son station, and the discharge data at Nong Son station are divided into three parts, namely training, validation and testing set. The training process uses the training data set (from 1991 to 2008) to generate a relationship between rainfall and discharge. The validation process aims to evaluate how well the current model performs while tuning the parameters of the models. The validation process uses the data from 2008 to 2009. It also helps control the learning rate, avoid overfitting, and select a model among different trained models. Finally, the last year (2010), which is the testing set, is used for performance evaluation.

This paper adopts Python being an open source programming language with libraries, namely Numpy, Pandas, Matpolib, Math and Keras.

4.1 Model Training and Evaluation

There are some aspects that this paper carefully considers while training the models. First, it is the hyperparameters selection including learning rate, timesteps, the number of input neurons, hidden layers, batch size, and epochs. Second, during the training process, overfitting is considered as a serious problem. Dropout regularization is a brilliant solution to prevent the model from overfitting [13]. Finally, both models are Adam which is the optimization algorithm. Adam is a recommended way to update the weights iteratively when it has different successful applications, such as computer vision [14] and natural language processing [15]. The algorithm is computationally efficient, also requires little memory. The trial and error method is adopted to choose the suitable hyperparameters, based on the model evaluation criteria. The training process is repeated and stopped when the evaluation criteria reach an acceptable value.

The Nash–Sutcliffe model efficiency coefficient (NSE) and Root Mean Square Error (RMSE) are used as the model evaluation criteria. NSE is used to indicate the efficiency of the model. The value of NSE ranges from $-\infty$ to 1. NSE value getting closer to 1 indicates a perfect model with an estimation error variance equal to zero. The RMSE measures the differences between the modeled values and the observed values. The RMSD value is always non-negative. If the value of RMSE is 0, it is a perfect fit to the data.

$$NSE = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_t - y_t)}{\sum_{i=1}^{n}(\hat{y}_t - \overline{y}_t)} \tag{11}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_t - y_t)} \tag{12}$$

Where $\hat{y}_t$ is observed discharges, y_t is modeled discharges, and $\overline{y}_t$ is mean of observed discharges at time t.

4.2 Validation Results

In the validation, the project uses the values of RMSE and NSE to find the optimal LSTM and MLP models from the proposed models with the different hyperparameters (Table 1).

Table 1. The selected LSTM models and MPL model

Items	The LSTM model	The MPL model
Timesteps	6 days	5 days
The number of hidden layers	1	1
Training parameter	- Learning rate: 0.001 - The number of epochs: 138 - batch_size: 73 - Dropout rate: 0.1 - Early stopping: yes	- Learning rate: 0.001 - The number of epochs: 93 - batch_size: 73 - Dropout rate: 0.1 - Early stopping: yes
The number of neurons at input, hidden, and output layer	5, 4, and 1	5, 7, and 1

4.3 Test Results

The above selected models use the unseen testing dataset, the year 2010, to evaluate their forecasting ability. The data in 2010 is selected for testing because a historic flood occurred in November 2010. The comparison between two models is shown in Table 2 and Fig. 9. The values of RMSE are not high, about 320 m^3/s compared to the maximum peak discharge of 6,520 m^3/s. The values of NSE are closer to 1. Moreover, there are small differences in the model evaluation criteria between both models.

Table 2. The results of the models in the testing phase.

Model	RMSE (m^3/s)	NSE	Modeled peak (m^3/s)	Observed peak (m^3/s)	Error (%)
LSTM	321.607	0.524	6,262	6,520	4
MLP	315.172	0.592	6,064	6,520	7

Figure 9 illustrates the relationship between the modeled discharges and the observed data from January 1991 to December 2010. There are similar trends among the observed data and modeled data. Both the LSTM and MLP model have the ability to produce the same pattern with the observation in the catchment. They also capture the flood peak. The modeled peak flow rate is closer to the observed data, 6,262.3 m^3/s and 6,520 m^3/s respectively on 16th November 2010.

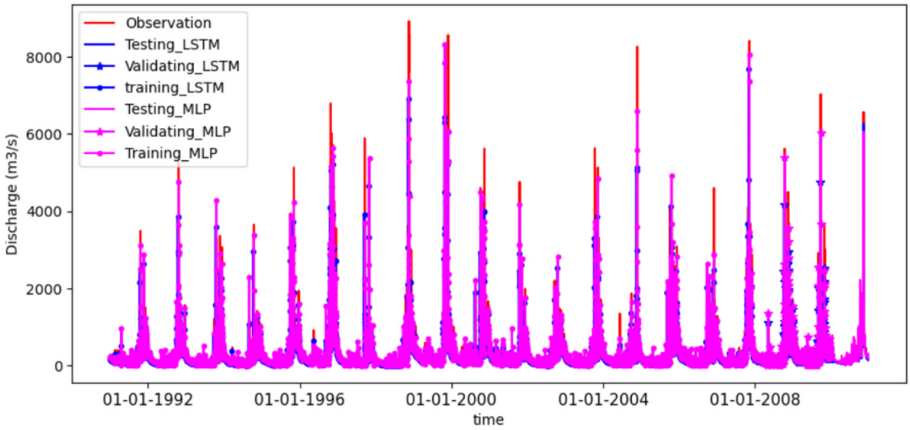

Fig. 9. The observed flows and modeled flows at Nong Son station in the training, validating and testing phase from January 1991 to December 2010.

However, the LSTM and MLP models often underestimate the historical flood peaks. The reason can be from operation of reservoirs, which makes discharge unexpectedly bigger in the points on flood events. Therefore, the next research should consider the reservoir operation in this catchment.

5 Conclusion

This study develops and compares the ability of the LSTM and MLP models to forecast discharge in Thu Bon-Vu Gia catchment. It proves that both LSTM and MLP have the same ability to forecast floods in the catchment.

They require only rainfall and discharge data, but they can produce accurate results. These models are good solutions because they do not require many kinds of data, which have low quality in the catchment. Moreover, although there are differences between the modeled data and the observed data, both models are useful for giving early warning in the catchment.

Operation of reservoirs effecting the results should be considered in future studies.

Acknowledgment. This research is funded by University of Technology and Education - The University of Danang under the project number T2020-06-155.

References

1. Navrud, S., Tran, T., Tinh, B.: Estimating the welfare loss to households from natural disasters in developing countries: a contingent valuation study of flooding in Vietnam. Glob. Health Action **5**, 17609 (2012)
2. Yoshimura, K., et al.: Development of flood forecasting system over Japan and application to 2018 Japan floods event. In: Geophysical Research Abstracts, vol. 21 (2019)

3. Jajarmizadeh, M., Harun, S., Salarpour, M.: A review on theoretical consideration and types of models in hydrology. J. Environ. Sci. Technol. **5**(5), 249–261 (2012)
4. Thirumalaiah, K., Deo, M.: River stage forecasting using artificial neural networks. J. Hydrolog. Eng. **3**(1), 26–32 (1998)
5. Chau, V., Cassells, S., Holland, J.: Economic impact upon agricultural production from extreme flood events in Quang Nam, central Vietnam. Nat. Hazards **75**(2), 1747–1765 (2014). https://doi.org/10.1007/s11069-014-1395-x
6. Pal, A., Singh, D.: Handwritten English character recognition using neural network. Int. J. Comput. Sci. Commun. **1**(2), 141–144 (2010)
7. Choi, M.: [ANN] Making model for binary classification (2018). https://www.kaggle.com/mirichoi0218/ann-making-model-for-binary-classification
8. Sutskever, I., Martens, J., Hinton, G.E.: Generating text with recurrent neural networks. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 1017–1024 (2011)
9. Mikolov, T., Zweig, G.: Context dependent recurrent neural network language model. In: 2012 IEEE Spoken Language Technology Workshop (SLT), pp. 234–239. IEEE (2012)
10. Rout, A.K., Dash, P., Dash, R., Bisoi, R.: Forecasting financial time series using a low complexity recurrent neural network and evolutionary learning approach. J. King Saud Univ.-Comput. Inf. Sci. **29**(4), 536–552 (2017)
11. Arbel, N.: How LSTM networks solve the problem of vanishing gradients (2018). https://medium.com/datadriveninvestor/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577
12. Yuan, Q., Wei, S.: Aligning network traffic for serial consistency and anomalies with a customized LSTM model. In: 2018 IEEE International Conference on Progress in Informatics and Computing (PIC), pp. 322–326. IEEE (2018)
13. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
14. Liu, Z., Cao, Y., Wang, Y., Wang, W.: Computer vision-based concrete crack detection using U-net fully convolutional networks. Autom. Constr. **104**, 129–139 (2019)
15. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018). https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper.pdf