Vol. 04, No. 04,(2020),181–188, **ISSN:2581-3242**

IJMLNCE JOURNAL

International Journal of Machine Learning and Networked Collaborative Engineering

Journal Homepage: http://www.mlnce.net/home/index.html

DOI : https://doi.org/10.30991/IJMLNCE.2020v04i04.005

Compare model multi-input RNN, LSTM and GRU for prediction of irrigation canal's water level in Red river delta, North Vietnam

[a]Dang Nguyen Ha Thai, [b]Dat Nguyen Quang*

High School for Gifted Students, Hanoi University of Science - VNU, Hanoi, Vietnam

[a] nguyenhathaidang_hsgs19@hus.edu.vn

[b] nguyenquangdat@hus.edu.vn

Abstract

Forecasting water level on Red River is an important problem in Vietnam. We need to replace water level predicting models that based on experiences of hydro-meteorologists by machine learning models which provide faster as well as more accurate results. Therefore, we have applied several best machine learning methods with artificial neural networks such as ANN, RNN, LSTM, and GRU, compared these models. The results indicated that LSTM is most appropriate to Red River data, with 153.5% better than the worst model ANN (in MSE), and 1.58% better than the second best model GRU (in MSE).

Keywords

Water level,
Compare,
Forecasting,
Neural network.

1. Introduction

Red River Delta in Northern Vietnam is a crucial agricultural area in Vietnam. Nowadays, irrigation is a problem that should be highly concerned because climate change has changed farming condition significantly. Therefore, there is an enormous impact on rice cultivation in Vietnam. In water level measuring station on several major rivers of Northern of Vietnam, scientists still utilize some obsolete machine learning approachs. For example, in a station in Hanoi, measuring and forecasting system of Artificial Neural Network are applied with the support from experiences of agricultural experts. Nowadays, there are various methods with higher precision, and a shift to updated approach to predicting is a necessary under the circumstance of climate change which fluctuate the water level unstably. Consequently, we want to propose a new idea that fits to the situation of Red River and, specifically, water level measuring in Hanoi.

2. Related Work

Mohamad Javad Alizadeh et al. [**1**] predicted the rainfall and runoff in the Tolt River basin by using the WANN model. Their results shown that including the 1 month ahead rainfall predictions in the rainfall forecasting 2 months ahead improves the WANN model's performance about 15%, 15%, and 17% in terms of NSE, R2, and RMSE, respectively.

Youngmin Seo et al. [**2**] developed and applied two hybrid models: Wavelet-based Artificial Neural Network (WANN), Wavelet-based Adaptive Neuro-Fuzzy Inference System (WANFIS) for forecasting daily

* *corresponding author*
nguyenquangdat@hus.edu.vn

water level in the Andong dam watershed, South Korea. Based on statistical performance indexes the WANN and WANFIS models are found to perform better efficiency than the ANN and ANFIS models.

Baowei Wang et al. [**3**] introduced a two-layer RNN framework to apply to 2 types of data from some cities in China, with LSTM and GRU for the first and second layer respectively. The research finding indicated that this architecture was more suitable than others to produce the result more accurately. For instance, RMSE of this model was lower than that of LSTM.

By deploying NB-IoT, 7 dissimilar machine learning techniques and Lufta application to dataset, Andreas Lepperod et al. [**4**] did research to trial air quality in Trondheim in Norway. After that, they pointed out that DART excelled at forecasting the general quality of the air for the whole contaminants studied while GRU produced the best outcomes in monitoring the variations of air pollution.

Li et al. in 2018 [**5**] used LSTM model for stock price forecasting the data taken of Chinese stock market. They compared with the LSTM, RNN and MI-LSTM-N, the model generated the best results of MSE is 0.996 x 10-3.

Bahrudin H. et al. in 2019 [**6**] gave a prediction of the water level in Vrana Lake (Croatia). They used LSTM, RNN and FFNN. The authors shown that the proposed model LSTM has better results, with RMSE are 8.47 & 22.84 (6 and 12 months), and R are 0.907 & 0.808 (in the same 6 and 12 months).

Juntao Zhang et al. at 2020 [**7**] used RNN for prediction water level in China (in 5 dams), get the result are MSE of 1.30, MAE of 1.00 and R2 of 0.56. These values were lower than MSE, MAE and higher than R2 compared with LSTM, and better than ANN.

In the same year 2020, Faruq et al. [**8**] applied a LSTM model to forecast the hourly water level on data taken from the Sulaiman Bridge on the Klang River, Malaysia for flood anticipation. The results of this model were concluded to be effective (RMSE is 0.20593, R2 is 0.844).

3. Methodology

3.1. Artificial neural network - ANN

ANN is one of the most widely used models for time series forecasting. Actually doing a non-linear functional mapping from past values (y_{t-1}, y_{t-2}, ... , y_{t-p}) to the future value y_t:

$$y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-p}, w) + \varepsilon_t$$

with the w is vector of parameters, the f is the activate function.

Single hidden layer feed–forward network modelis characterized by a network of three layers of simple processing units connected by links. Inputs (y_{t-1}, y_{t-2}, ... , y_{t-p}) and output (y_t) associated with each other by the mathematical expression:

$$y_t = \alpha_o + \sum_{j=1}^{q} \alpha_j g\left(\beta_{o,j} + \sum_{i=1}^{p} \beta_{i,j} y_{t-i}\right) + \varepsilon_t$$

where α_j (j : 0 to q) and $\beta_{i,j}$ (i : 0 to p ; j : 1 to q) are the parameters of the model (called the connection weights).

3.2. Recurrent neural network – RNN

Current step, in the feed-forward neural networks, is represented by N-1 previous steps. Whereas, the RNN is a neural network which has additional connections between adjacent time steps. Previous values are represented with recurrent connections even if the distance between current value and these values is infinite. While the feed-forward neural networks compress just only 1 past step, the RNN model can compress the whole history in low dimensional space. Self-connections from a node to itself over time are allowed and share their weights to the different time steps [**10**]. Therefore the RNN is an effective tool for modeling the time series data. At step t, the model receives the input value at the current input x_t, the hidden node values at previous history h_{t-1}, and calculates the value at the current hidden node h_t. The output value y_t is computed from the hidden node value h_t. Consequently, y_t depends not only on x_t but also x_{t-1}.

Let $(x_0, x_1, \ldots, x_T)$ denote the input vector x, $(h_0, h_1, \ldots, h_T)$ denote the hidden states of the recurrent layer h and $(y_0, y_1, \ldots, y_T)$ denote the output vector y. In the RNN model, the functions among states are represented in terms of a closely coupled system given by following equations:

$$\begin{aligned} \boldsymbol{x_t} &= (\boldsymbol{W_t} \cdot \boldsymbol{h_t}) \\ \boldsymbol{h_t} &= \boldsymbol{g}(\boldsymbol{V} \cdot \boldsymbol{h_t}) \\ \boldsymbol{y_t} &= \boldsymbol{f}(\boldsymbol{U} \cdot \boldsymbol{x_t}) \end{aligned}$$

where W_t denotes the current vector of parameter, and U and V are vector of weights that was learned (the connection weights from the input layer x to the hidden layer h). The functions f is *tanh* or *sigmoid*, the function g is *softmax* function:

$$\boldsymbol{f}(\boldsymbol{z}) = \frac{\boldsymbol{1}}{\boldsymbol{1} + \boldsymbol{e}^{-\boldsymbol{z}}}$$

$$\boldsymbol{g}(\boldsymbol{z}) = \frac{\boldsymbol{e}^{\boldsymbol{z}}}{\sum_{\boldsymbol{k}} \boldsymbol{e}^{\boldsymbol{z_k}}}$$

RNNs create a short-term memory, so it can solve with position invariance well, which cannot be done by feed-forward networks.

RNNs are able to handle short-term dependencies in data series. The weakness of the recurrent neural network is a capability of solving long-term dependence [**11**, **12**].

3.3. Long-short term memory – LSTM

As we have mentioned, RNNs are able to perform well against short-term dependencies in data series, but its weakness lies on longer ones. In 1997 [**11**], LSTM model was proposed by Hochreiter and Schmidhuber as a variant of RNN which has the ability to confront the existing problem of this model. In fact, in order to store information along the training period, LSTM has been designed with 3 gates and memory cells [**11**, **12**, **13**].

Input of LSTM unit at *t*-step is x_t, and output at *t-1*-step is (h_{t-1}). The input are filled by gates (with *sigmoid* function). The output value is in range [*0, 1*]. If the output is 0, all inputs are removed. If not, all information is passed. Then, the output at *t*-step (h_t) and the *cell state* at *t*-step are solved by taking those below steps:

Step 1: Forget gates dispose data from the *cell of state*:

$$\boldsymbol{f_t} = \boldsymbol{\sigma}\left(\boldsymbol{W_f} \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_i}\right)$$

Step 2: New data are kept in the *cell state*. In the first phase of this step, the input gate (with the function *sigmoid*) gets the new values, then updates this node. During the second phase, *tanh* class inputs a new vector $\tilde{\boldsymbol{c}}_{\boldsymbol{t}}$.

$$\begin{aligned} \boldsymbol{i_t} &= \boldsymbol{\sigma}(\boldsymbol{W_i} \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_i}) \\ \tilde{\boldsymbol{c}}_{\boldsymbol{t}} &= \boldsymbol{tanh}(\boldsymbol{W_{\tilde{c}}} \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_{\tilde{c}}}) \end{aligned}$$

Step 3: *state cells* are updated from c_{t-1} to c_t:

$$\boldsymbol{c_t} = \boldsymbol{f_t} \otimes \boldsymbol{c_{t-1}} \oplus \boldsymbol{i_t} \otimes \tilde{\boldsymbol{c}}_{\boldsymbol{t}}$$

Step 4: Compute the output. Initially, the data which will be the outside of the LSTM unit are evaluated by the output gate evaluates. Then, \$state\$ becomes a passed node which lies between -1 to 1. The final output will be calculated by this formula:

$$\begin{aligned} \boldsymbol{o_t} &= \boldsymbol{\sigma}(\boldsymbol{W_o} \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_o}) \\ \boldsymbol{h_t} &= \boldsymbol{o_t} \otimes \mathbf{tanh}\ (\boldsymbol{h_t}) \end{aligned}$$

with W_f, W_i, $\boldsymbol{W_{\tilde{c}}}$, W_o are parameters of LSTM and b_f, b_i, $\boldsymbol{b_{\tilde{c}}}$, b_o are biases of LSTM model.

3.4. Gated recurrent unit – GRU

The gated recurrent unit model (GRU), which is a simplification of LSTM, was proposed by KyungHyun Cho in Oct 2014 [**9**]. It performs on a variety of application, uses some functions of gates.

Unlike LSTM, the GRU model doesn’t have a memory cell. We can summarise GRU's operations in the following formulas:

$$\boldsymbol{h_t} = (\boldsymbol{1} - \boldsymbol{z})\boldsymbol{h_{t-1}} + \boldsymbol{z_t}\widetilde{\boldsymbol{h}}_t$$
$$\boldsymbol{z_t} = \boldsymbol{\sigma}(\boldsymbol{W_h x_t} + \boldsymbol{U_z h_{t-1}})$$
$$\widetilde{\boldsymbol{h}}_t = \left(\boldsymbol{W_h x_t} + \boldsymbol{U}(\boldsymbol{r_t} \cdot \boldsymbol{h_{t-1}})\right)$$
$$\boldsymbol{r_t} = \boldsymbol{\sigma}(\boldsymbol{W_r x_t} + \boldsymbol{U_r h_{t-1}})$$

where the values h_t , z_t denote GRU's output, r_t is the update gate and reset gate, $\widetilde{\boldsymbol{h}}_t$ is the candidate output, and W_z, W_h, W_r, U_z, and U_r are the matrices in GRU.

4. Experiment and results

4.1. Data and Criteria for comparison

4.1.1. Data and area

We utilized data of water level measuring station in Hanoi which is measured in months of rainy season (from June 14 to September 14 each year) from 2017 to 2020. These data are important because they affect Hanoi, capital of Vietnam and also largest city in Northern Vietnam.

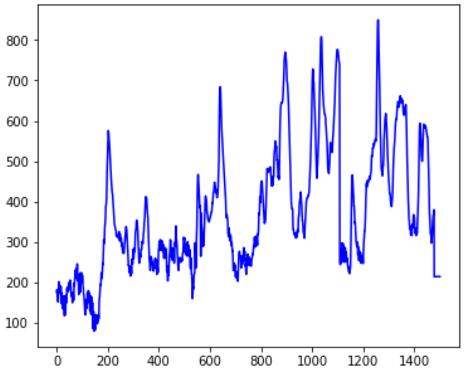

Figure 1: The dataset at Hanoi's station in Red river.

4.1.2. Criteria for comparison

2 criteria are used in order to compare the results: Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE):

$$MSE = \frac{1}{n} \cdot \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$
$$MAPE = \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{n} \right|$$

4.2. Model training

We trained each model 30 times, the results were taken as the average of the program runs.

4.3. Result

The dataset was trained with all of the neural network models, the ANN, RNN, LSTM, GRU (with the same parameters).

The ***Table 1*** gives information on the result of the model after applying actual our data.

Table 1: Results of data.

Model	Compare	
	MSE	MAPE
ANN	5.423559	0.616261
RNN	3.588753	0.396943
LSTM	**2.138652**	**0.228286**
GRU	2.172862	0.234749

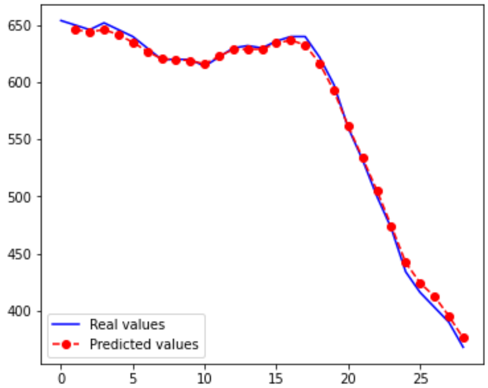

Figure 2: Results of ANN models for water level in data of Hanoi's station.

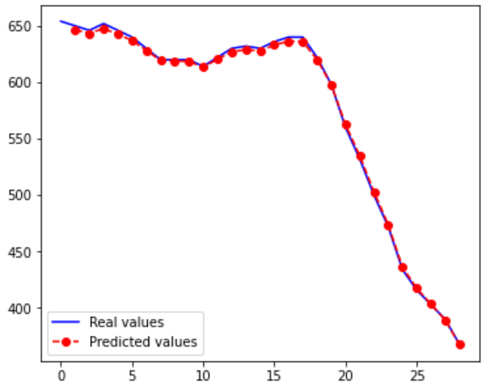

Figure 3: Results of RNN models for water level in data of Hanoi's station.

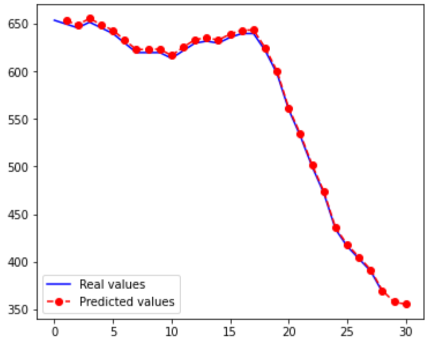

Figure 4: Results of LSTM models for water level in data of Hanoi's station.

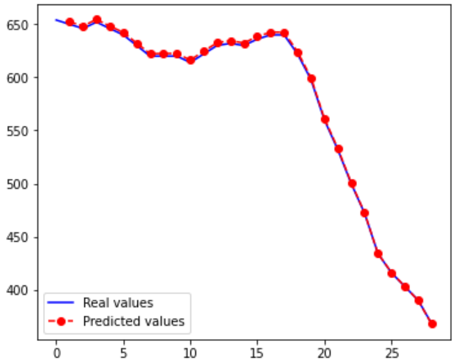

Figure 5: Results of GRU models for water level in data of Hanoi's station.

We can inferred from ***Table I*** that LSTM is most appropriate to Red River data. The test scores of LSTM are 2.138652 MSE and 0.228286 MAPE while test scores of GRU - which is the 2nd best - are 2.172862 MSE and 0.234749 MAPE, which are 1.58% and 2.89% larger than those of LSTM, respectively. Next, test scores of RNN are 3.588753 MSE and 0.396943 MAPE which are 67.5% and 73.8%, respectively, greater than those of LSTM. Finally, test score of ANN are worst, was 5.423559 MSE and 0.616261 MAPE, greater than LSTM 153.5% and 169%.

5. Conclusion

We will apply machine learning models to the irrigation management system of the metering station in Hanoi, replacing part of the manual calculation. At the same time, we will also use the opinions of experts in the irrigation industry to be able to apply the most appropriate in practice.

With the successful use of the above system of machine learning models, in the future, we will try to apply new and stronger methods in forecasting:

- Hybrid model.
- Multi-time series model.
- Online learning.

Acknowledgment

We thanks to the WARM group members for the help during the editing this article. This article is supported by group WARM, Thuy Loi University, Hanoi, Vietnam.

References

[1] M.J. Alizadeh, M.R. Kavianpour, O. Kisi, and V. Nourani (2017) A new approach for simulating and forecasting the rainfall-runoff process within the next two months, In: Journal of Hydrology, vol. 548, pp. 588-597, Elsevier, https://doi.org/10.1016/j.jhydrol.2017.03.032

[2] Y. Seo, S. Kim, O. Kisi, and V.P. Singh (2015) Daily water level forecasting using wavelet decomposition and artificial intelligence techniques, In: Journal of Hydrology, vol. 520, pp. 224-243, Elsevier, https://doi.org/10.1016/j.jhydrol.2014.11.050

[3] B. Wang, W. Kong, H. Guan and N.N. Xiong (2019) Air quality forecasting based on gated recurrent long short term memory model in Internet of Things, In: IEEE Access, vol. 7, pp. 69524-69534, IEEE, https://doi.org/10.1109/ACCESS.2019.2917277

[4] A. Lepperod, H.T. Nguyen, S. Akselsen, L. Wienhofen, and P. Ozturk and W. Zhang (2019) Air Quality Monitor and Forecast in Norway using NB-IoT and Machine Learning, In: International Summit Smart City 360°, pp. 56-67, Springer, https://doi.org/10.1007/978-3-030-51005-3_7

[5] Y. Li and H. Cao (2018) Prediction for tourism flow based on LSTM neural networks, In: Procedia Computer Science 129, pp. 277-283, Elsevier, https://doi.org/10.1016/j.procs.2018.03.076

[6] B. Hrnjica and O. Bonacci (2019) Lake Level Prediction using Feed Forward and Recurrent Neural Networks, In: Water Resources Management, vol. 33, pp. 2471-2484, Springer, https://doi.org/10.1007/s11269-019-02255-2

[7] J. Zhang, Z. Zhang, Y. Weng, S. Gosling, H. Yang, C. Yang, W. Li, Q. Ma (2020) Using Recurrent Neural Network for Intelligent Prediction of Water Level in Reservoirs, In: IEEE 44th Annual Computers, Software, and Applications Conference, IEEE, https://doi.org/10.1109/COMPSAC48688.2020.0-108

[8] A. Faruq, S.S. Abdullah, A. Marto, M.A.A. Bakar, Samin and A. Mubin (2020) River water level forecasting for flood warning system using deep learning long short-term memory network, In: 3rd International Conference on Engineering and Applied Technology, IOP Science, https://doi.org/10.1088/1757-899X/821/1/012026

[9] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio (2014) Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, In: EMNLP, pp. 1724-1734, arXiv, https://arxiv.org/abs/1406.1078v3.

[10] L. Peng, S. Liu, R. Liu and L. Wang (2018) Effective long short-term memory with differential evolution algorithm for electricity price prediction, In: Energy, vol. 162, pp. 1301-1314, Elsevier, https://doi.org/10.1016/j.energy.2018.05.052

[11] S. Hochreiter and J. Schmidhuber (1997) Long short-term memory, In: Neural Computation, vol. 9, issue 8, pp. 1735-1780, MIT Press, https://doi.org/10.1162/neco.1997.9.8.1735

[12] R. Pascanu, T. Mikolov and Y. Bengio (2013) On the difficulty of training recurrent neural networks, In: Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine

Learning Research, vol. 28, pp. 1310-1318, https://proceedings.mlr.press/v28/pascanu13.html

[13] F.A. Gers, J. Schmidhuber and F. Cummins (2000) Learning to Forget: Continual Prediction with LSTM, In: Neural Computation, vol. 12, issue 10, pp. 2451-2471, MIT Press, https://doi.org/10.1162/089976600300015015

Author's Biography

Dang Nguyen Ha Thai, a student in High School for Gifted Student, Hanoi University Science – Vietnam National University, Hanoi, Vietnam. He is majoring in Mathematics and Informatics. He has experience in programming and scientific research.

Dat Nguyen Quang, a PhD. Student, a teacher in Hanoi University Science – Vietnam National University, Hanoi, Vietnam. His favorite majors are Big Data, Machine Learning, Artificial Intelligence. He has published over 10 works in journals and scientific conferences.