ILKOM Jurnal Ilmiah Vol. 14, No. 1, April 2022, pp. 57-62
Accredited 2nd by RISTEKBRIN No. 200/M/KPT/2020; E-ISSN 2548-7779 | P-ISSN 2087-1716

LSTM-based Multivariate Time-Series Analysis: A Case of Journal Visitors Forecasting

Anggie Wahyu Saputra [a,1]; Aji Prasetya Wibawa [a,2*]; Utomo Pujianto [a,3]; Agung Bella Putra Utama [a,4]; Andrew Nafalski [b,5]

[a] Universitas Negeri Malang, Jl. Semarang No.5, Malang, 65145, Indonesia
[b] University of South Australia, 101 Currie St, Adelaide SA 5001, Australia
[1] anggiew39@gmail.com; [2] aji.prasetya.ft@um.ac.id; [3] utomo.pujianto.ft@um.ac.id; [4] agungbpu02@gmail.com; [5] andrew.nafalski@unisa.edu.au
** Corresponding author*

Article history: Received February 03, 2022; Revised March 10, 2022; Accepted March 12, 2022; Available online April 30, 2022

Abstract

Forecasting is the process of predicting something in the future based on previous patterns. Forecasting will never be 100% accurate because the future has a problem of uncertainty. However, using the right method can make forecasting have a low error rate value to provide a good forecast for the future. This study aims to determine the effect of increasing the number of hidden layers and neurons on the performance of the long short-term memory (LSTM) forecasting method. LSTM performance measurement is done by root mean square error (RMSE) in various architectural scenarios. The LSTM algorithm is considered capable of handling long-term dependencies on its input and can predict data for a relatively long time. Based on research conducted from all models, the best results were obtained with an RMSE value of 0.699 obtained in model 1 with the number of hidden layers 2 and 64 neurons. Adding the number of hidden layers can significantly affect the RMSE results using neurons 16 and 32 in Model 1.

Keywords: Forecasting; Multivariate; Long Short-term Memory; Sessions.

Introduction

Forecasting is estimating future information using past data [1]. It plays a significant role in various fields of life, one of which is planning to anticipate conditions that will occur in the future. Other functions of forecasting is to become benchmarks in decision-making [2]. One of the characteristics of a good decision is based on considering something that has happened. The future has uncertainty problems, so forecasts will never be 100% accurate [3]. However, appropriate methods and hyperparameters can make forecasts with a low error rate to provide good forecast values [4].

Many forecasting cases have been carried out, including the forecasting of sessions in an electronic journal. Several forecasts of these sessions have been carried out, including forecasting using the multilayer perceptron (MLP) method, which yields (RMSE) of 0.137826 [5]. Using single exponential smoothing MLP for session forecasting generates an RMSE of 0.7554 [6]. Then using the backpropagation neural network (BPNN), get MAPE 0.301 [7]. Meanwhile, research that uses long short-term memory (LSTM) produces an RMSE of 13.76 [8], and when using smoothing, LSTM has a MAPE of 0.08098 [9]. All of them have proven effective in solving prediction problems. However, the previous studies only used univariate data types, which only used one attribute. So, the time-series analysis was univariate [10]. While the dataset used has four attributes: sessions, page views, visitors, and new visitors. If all attributes are considered to affect the prediction results, then the analysis to be carried out is multivariate prediction [11].

This research focuses on applying multivariate analysis of the number of sessions to electronic journals using the LSTM algorithm. All attributes will be used to generate session predictions. LSTM is a derivative of a neural network that can be used for time-series data modeling [12]. LSTM evolves the recurrent neural network (RNN) architecture built and designed to overcome RNN problems when dealing with vanishing gradients and exploding gradients [13]. The advantage of the LSTM algorithm is that it can predict data over a relatively long period and can overcome long-term dependence on the input [14]. LSTM performance measurement is done with RMSE (root mean square error) in various architectural scenarios.

https://doi.org/10.33096/ilkom.v14i1.1106.57-62

Method

The method used in forecasting sessions for this electronic journal is LSTM. The forecasting process using LSTM has several stages described as follows.

The first process is dataset collection. The dataset is obtained through the website https://statcounter.com/ which is activity data on the number of visitors to the knowledge engineering and data science (KEDS) journal portal. The dataset was from January 1, 2018, until December 31, 2020. All the attributes in the dataset consisting of sessions, page views, visitors, and new visitors will be used in the forecasting process, with the forecasting target being the sessions attribute. Sessions were chosen because they are crucial indicators of a journal's success as a benchmark for wide distribution in accelerating the journal accreditation system. The dataset used can be seen in **Table 1.**

Table 1. Dataset

Day	Date	Sessions	Page views	Visitors	New Visitors	Target
Monday	01/01/18	3	3	3	3	2
Tuesday	01/02/18	2	2	1	1	1
Wednesday	01/03/18	1	1	1	1	1
Thursday	01/04/18	1	1	1	1	2
……….	……….	……….	……….	……….	……….	……….
……….	……….	……….	……….	……….	……….	……….
Tuesday	12/29/2020	9	24	7	4	19
Wednesday	12/30/2020	19	69	17	8	12
Thursday	12/31/2020	12	41	12	10	12

The dataset will be used in three types of models in this study. The distribution is based on training and testing data, as shown in **Table 2.**

Table 2. Data Usage Model Scenarios

Model	Training	Testing
1	2018	2019
2	2019	2020
3	2018/2019	2020

The second process is preprocessing by normalizing the data. Normalization minimizes errors by converting the actual value input data into values with a range of 0 to 1 [15]. The normalization technique used is min-max [16]. The min-max normalization equation can be seen in (1) [17].

$$X' = \frac{(x - \min_x)}{(\max_x - \min_x)} \tag{1}$$

X' is the result of normalization, x is the data to be normalized, $\min_x$ is the minimum value of all data, max_x is the maximum value of the entire data.

After the data is normalized, the next step is forecasting. Forecasting is done using LSTM. The LSTM architecture consists of a memory cell and three gates: input, forget, and output [18]. The input gate plays a role in regulating how much information must be stored in a sold state. The gate prevents the cell from storing useless data. Forget gate is responsible for setting a fixed value in the memory cell. Output Gate regulates how many values in the memory cell will be used for output. There are several computational stages in the LSTM method, which can be seen in (2)-(7) [19], [20].

$$f_t = \sigma\,(w_f.[h_{t-1}, x_t] + b_f \tag{2}$$

$$i_t = \sigma\,(w_i.[h_{t-1}, x_t] + b_i \tag{3}$$

$$\tilde{c}_t = \tanh\,(w_c.[h_{t-1}, x_t] + b_c \tag{4}$$

$$c_t = f_t\,.\,c_{t-1} + i_t.\,\tilde{c}_t \tag{5}$$

$$o_t = \sigma\,(w_o.[h_{t-1}, x_t] + b_0) \tag{6}$$

$$h_t = o_t\,.\tanh(c_t) \tag{7}$$

f_t, i_t, $\tilde{c}_t$, o_t, successively the forget gate, input gate, intermediate cell state, and output gate. σ dan tanh as the activation function. w_f, w_i, w_c, w_o is the weight value, b_f, b_i, b_0 is the bias value. h_{t-1} is the previous period input value and x_t input value on t. c_{t-1} is the old state, c_t is the current cell state, h_t is the hidden state.

The LSTM architecture used in this study consists of several models and scenarios, hidden layers, and the number of neurons. Architectural variations apply and adopt previous research [21]–[23]. The scenario for using the hidden layer (HL) starts from 2 to 10. For the scenario with the number of neurons using 16, 32, 64, 128, 256. Other LSTM parameters used are dropout 0.2, batch size 16, and epoch 500. The activation function used is linear, MSE type loss, and adam as the optimizer. The LSTM pseudocode from this study can be seen in **Table 3.**

Table 3. Pseudocode LSTM

```
Begin
 input: train_X, train_y
 initialize ()
 normalization (train_X, train_y)
 reshape (X, shape ← [train_X.shape[0],1,train_X.shape[1])
 LSTM_model = Sequential (LSTM (units, return_sequences, input_shape, dropout), Dense
 (unit, activation)
 LSTM_model.compile (loss_function, optimizer)
 LSTM_model.fit (train_X, train_y, epochs, batch_size)
End
```

After the forecasting test process is carried out, the next step is to evaluate. Evaluation of forecasting results using the root mean square error (RMSE). The purpose of the evaluation is to verify the effectiveness and display the magnitude of the error generated by a prediction model [24]. RMSE is used to detect irregularities or outliers in the prediction system built [25]. The RMSE formula is shown in (8) [26].

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_i - \check{y}_i)^2}{n}} \tag{8}$$

$RMSE$ is the root mean square error, $\sum$ is the total value, n is the amount of data, and y_i is the actual value, $\check{y}_i$ is the forecast value.

The last process compares the RMSE values and tests the significance of adding hidden layers and neurons from all existing models. The comparison aims to find out the best results that can be used for suggestions on the use of architecture. The following section can show the complete forecasting results of all models and their analysis.

Findings and Discussion

In this study, three models are processed with various scenarios. The scenario consists of different numbers of hidden layers and neurons. The RMSE results of the experiments that have been carried out are shown in **Table 4.**

Table 4. RMSE Experiment Results

HL	Neuron														
	Model 1					*Model 2*					*Model 3*				
	16	*32*	*64*	*128*	*256*	*16*	*32*	*64*	*128*	*256*	*16*	*32*	*64*	*128*	*256*
2	1.816	1.372	**0.699**	0.830	1.027	1.319	1.159	**0.851**	1.270	1.446	1.177	0.834	**0.701**	0.853	0.962
3	2.324	1.951	1.463	1.079	1.124	1.220	1.122	1.247	1.504	2.357	1.136	0.947	1.315	0.996	1.137
4	9.835	9.813	9.790	9.788	9.779	9.669	9.492	3.586	1.434	1.774	1.534	1.247	1.530	1.191	5.805
5	9.843	9.824	9.810	9.791	9.782	9.613	9.587	9.531	9.486	9.420	9.960	9.827	9.733	9.695	9.779
6	9.849	9.829	9.807	9.795	9.778	9.624	9.583	9.529	9.486	9.416	9.930	9.838	9.754	9.668	9.829
7	9.846	9.830	9.810	9.791	9.780	9.632	9.571	9.539	9.477	9.412	9.936	9.847	9.719	9.683	9.899
8	9.847	9.828	9.809	9.789	9.779	9.629	9.592	9.546	9.468	9.408	9.937	9.845	9.716	9.687	9.772
9	9.848	9.825	9.812	9.729	9.783	9.614	9.586	9.540	9.489	9.406	9.957	9.839	9.742	9.681	9.762
10	9.846	9.827	9.811	9.789	9.781	9.592	9.595	9.534	9.482	9.402	9.940	9.858	9.734	9.671	9.775

In **Table 4**, Model 1 produces the best RMSE 0.699 obtained from the architecture of hidden layer 2 and neuron 64. The use of hidden layers 2 and 3 produces a relatively good RMSE, while in the use of hidden layers 4 to 10, there are differences in RMSE, which is far from the initial 2 hidden layers. The number of hidden layers in model 1 resulted in a higher RMSE. Using the number of neurons in the RMSE hidden layer, the best in model 1 has decreased and increased in value. Adding the number of neurons in Model 1 resulted in better RMSE.

Based on **Table 4**, the best RMSE in Model 2 is 0.851 with 2 hidden layers and 64 neurons. The increasing hidden layers in Model 2 result in a fluctuating and unpatterned RMSE. Here, adding the number of hidden layers does not affect the improvement of the RMSE. The increase in the number of neurons decreases the number of RMSE starting at hidden layers 5 to 10.

From **Table 4,** Model 3 produces the best RMSE in the experiment with hidden layer 2, and the number of neurons is 64 with a value of 0.701. The number of hidden layers in model 3 resulted in the RMSE not showing an up or downtrend. There was a significant increase in the RMSE from adding hidden layers 3 to 4. In hidden layers 5 to 10, the addition of the neurons from 16 to 128 decreased, and the number of neurons increased to 256.

A significance test was conducted using the Paired Sample T-Test to determine the effect of adding a hidden layer and the number of neurons to the resulting RMSE. Paired Sample T-Test is used to find out between two paired samples. The significant test results by adding the number of hidden layers and the number of neurons can be seen in **Table 5** and **Table 6.**

Table 5. The Significance Test Results of Model 1 on the Addition of the Number of Hidden Layers

Neuron				
16	*32*	*64*	*128*	*256*
0.0277	**0.0405**	0.0664	0.0764	0.0695

Table 6. The Significance Test Results of Model 1 on the Addition of the Number of Neurons

Hidden layer								
2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0.0884	0.0899	0.1099	0.1100	0.1100	0.1100	0.1100	0.1100	0.1100

Based on the results from **Table 5** and **Table 6** in Model 1, adding a hidden layer can significantly affects the statistical results in neurons 64 and 32. While in models 2 and 3, the significance test results show that adding a hidden layer and neurons has no significant effect on RMSE performance (the resulting value is more than 0.05. The overall results show that adding the number of hidden layers and the number of neurons to the RMSE value has no significant effect. The best LSTM architecture in this study consists of 2 hidden layers and 64 neurons.

Conclusion

The results of this study indicate the utility of LSTM-based multivariate time-series analysis for journal visitor forecasting. Adding the number of hidden layers or the number of neurons in the LSTM method cannot be generalized because the resulting RMSE performance value changes up and down or vice versa, which does not show an increasing trend of hidden layers of neurons. In addition, based on the significance test results, there is no significant effect on the addition of the number of hidden layers and the number of neurons in the RMSE results. The future research will propose using hyperparameter tuning to find the best model for LSTM-based multivariate time-series forecasting.

References

[1] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[2] M. Bruno Soares, M. Daly, and S. Dessai, “Assessing the value of seasonal climate forecasts for decision-making,” *WIREs Clim. Chang.*, vol. 9, no. 4, Jul. 2018, doi: 10.1002/wcc.523.

[3] T. Hong and S. Fan, “Probabilistic electric load forecasting: A tutorial review,” *Int. J. Forecast.*, vol. 32, no. 3, pp. 914–938, Jul. 2016, doi: 10.1016/j.ijforecast.2015.11.011.

[4] N. Somu, G. R. M R, and K. Ramamritham, “A hybrid model for building energy consumption forecasting using long short term memory networks,” *Appl. Energy*, vol. 261, p. 114131, Mar. 2020, doi: 10.1016/j.apenergy.2019.114131.

[5] A. P. Wibawa, W. Lestari, A. B. P. Utama, I. T. Saputra, and Z. N. Izdihar, “Multilayer Perceptron untuk Prediksi Sessions pada Sebuah Website Journal Elektronik,” *Indones. J. Data Sci.*, vol. 1, no. 3, Dec. 2020, doi: 10.33096/ijodas.v1i3.15.

[6] M. A. B. Ferdinand, A. P. Wibawa, I. A. E. Zaeni, and H. A. Rosyid, “Single Exponential Smoothing-Multilayer Perceptron Untuk Peramalan Pengunjung Unik,” *J. Mob. Forensics*, vol. 2, no. 2, pp. 62–70, 2020, doi: https://doi.org/10.12928/mf.v2i2.2034.

[7] A. P. Wibawa, Z. N. Izdihar, A. B. P. Utama, L. Hernandez, and Haviluddin, “Min-Max Backpropagation Neural Network to Forecast e-Journal Visitors,” in *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Apr. 2021, pp. 052–058, doi: 10.1109/ICAIIC51459.2021.9415197.

[8] A. P. Wibawa, I. T. Saputra, A. B. P. Utama, W. Lestari, and Z. N. Izdihar, “Long Short-Term Memory to Predict Unique Visitors of an Electronic Journal,” in *2020 6th International Conference on Science in Information Technology (ICSITech)*, Oct. 2020, pp. 176–179, doi: 10.1109/ICSITech49800.2020.9392031.

[9] A. P. Wibawa, R. R. Ula, A. B. P. Utama, M. Y. Chuttur, A. Pranolo, and Haviluddin, “Forecasting e-Journal Unique Visitors using Smoothed Long Short-Term Memory,” in *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, Oct. 2021, pp. 609–613, doi: 10.1109/ICEEIE52663.2021.9616628.

[10] H. Li, “Time works well: Dynamic time warping based on time weighting for time series data mining,” *Inf. Sci. (Ny).*, vol. 547, pp. 592–608, Feb. 2021, doi: 10.1016/j.ins.2020.08.089.

[11] K. Backhaus, B. Erichson, S. Gensler, R. Weiber, and T. Weiber, *Multivariate Analysis*. Wiesbaden: Springer Fachmedien Wiesbaden, 2021.

[12] X. Yan, W. Weihan, and M. Chang, “Research on financial assets transaction prediction model based on LSTM neural network,” *Neural Comput. Appl.*, vol. 33, no. 1, pp. 257–270, Jan. 2021, doi: 10.1007/s00521-020-04992-7.

[13] M. Lechner and R. Hasani, “Learning Long-Term Dependencies in Irregularly-Sampled Time Series,” *arXiv*, 2020.

[14] G. Van Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model,” *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5929–5955, Dec. 2020, doi: 10.1007/s10462-020-09838-1.

[15] L. Munkhdalai, T. Munkhdalai, K. H. Park, H. G. Lee, M. Li, and K. H. Ryu, “Mixture of Activation Functions With Extended Min-Max Normalization for Forex Market Prediction,” *IEEE Access*, vol. 7, pp. 183680–183691, 2019, doi: 10.1109/ACCESS.2019.2959789.

[16] S. E. Buttrey, “ Data Mining Algorithms Explained Using R ,” *J. Stat. Softw.*, vol. 66, no. Book Review 2, 2015, doi: 10.18637/jss.v066.b02.

[17] J. Zhou, H. Chang, X. Cheng, and X. Zhao, “A Multiscale and High-Precision LSTM-GASVR Short-Term Traffic Flow Prediction Model,” *Complexity*, vol. 2020, pp. 1–17, Jun. 2020, doi: 10.1155/2020/1434080.

[18] I.-F. Kao, Y. Zhou, L.-C. Chang, and F.-J. Chang, “Exploring a Long Short-Term Memory based Encoder-

Decoder framework for multi-step-ahead flood forecasting," *J. Hydrol.*, vol. 583, p. 124631, Apr. 2020, doi: 10.1016/j.jhydrol.2020.124631.

[19] G. Hussain, M. Jabbar, J.-D. Cho, and S. Bae, "Indoor Positioning System: A New Approach Based on LSTM and Two Stage Activity Classification," *Electronics*, vol. 8, no. 4, p. 375, Mar. 2019, doi: 10.3390/electronics8040375.

[20] K. Wang, X. Qi, and H. Liu, "Photovoltaic power forecasting based LSTM-Convolutional Network," *Energy*, vol. 189, p. 116225, Dec. 2019, doi: 10.1016/j.energy.2019.116225.

[21] S. Kaushik *et al.*, "AI in Healthcare: Time-Series Forecasting Using Statistical, Neural, and Ensemble Architectures," *Front. Big Data*, vol. 3, no. March, Mar. 2020, doi: 10.3389/fdata.2020.00004.

[22] Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, "EA-LSTM: Evolutionary attention-based LSTM for time series prediction," *Knowledge-Based Syst.*, vol. 181, p. 104785, Oct. 2019, doi: 10.1016/j.knosys.2019.05.028.

[23] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Comput. Ind. Eng.*, vol. 143, no. July 2019, p. 106435, May 2020, doi: 10.1016/j.cie.2020.106435.

[24] I. I. Zulfa, D. Candra, R. Novitasari, F. Setiawan, A. Fanani, and M. Hafiyusholeh, "Prediction of Sea Surface Current Velocity and Direction Using LSTM," *Indones. J. Electron. Instrum. Syst.*, vol. 11, no. 1, pp. 93–102, 2021, doi: 10.22146/ijeis.63669.

[25] A. Jadidi, R. Menezes, N. de Souza, and A. de Castro Lima, "A Hybrid GA–MLPNN Model for One-Hour-Ahead Forecasting of the Global Horizontal Irradiance in Elizabeth City, North Carolina," *Energies*, vol. 11, no. 10, p. 2641, Oct. 2018, doi: 10.3390/en11102641.

[26] S. Bouktif, A. Fiaz, A. Ouni, and M. Serhani, "Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches," *Energies*, vol. 11, no. 7, p. 1636, Jun. 2018, doi: 10.3390/en11071636.