Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Comparison of RNN, LSTM, and GRU Methods on Forecasting Website Visitors

I Nyoman Gede Arya Astawa[1] ✉ **I Putu Bagus Arya Pradnyana**[2] and **I Ketut Suwintana**[3]
[12] *Departement of Electrical Engineering, Politeknik Negeri Bali, Bali, Indonesia*
[3] *Departement of Accounting, Politeknik Negeri Bali, Bali, Indonesia*
Corresponding Author: I Nyoman Gede Arya Astawa, **E-mail**: arya_kmg@pnb.ac.id

| ABSTRACT
Forecasting is the best way to find out the number of website visitors. However, many researchers cannot determine which method is best used to solve the problem of forecasting website visitors. Several methods have been used in forecasting research. One of the best today is using deep learning methods. This study discusses forecasting website visitors using deep learning in one family, namely the RNN, LSTM, and GRU methods. The comparison made by these three methods can be used to get the best results in the field of forecasting. This study used two types of data: First Time Visits and Unique Visits. The test was carried out with epoch parameters starting from 1 to 500 at layers 1, 3, and 5. The test used first-time visit data and unique visit data. Although tested with different data, the test results obtained that the smallest MSE value is the LSTM method. The value of each MSE is 0.0125 for first-time visit data and 0.0265 for unique visit data. The contribution of this research has succeeded in showing the best performance of the three recurrent network methods with different MSE values.

| KEYWORDS

Forecasting, website visitors, deep learning

| ARTICLE INFORMATION

ACCEPTED: 25 July 2022 **PUBLISHED:** 01 August 2022 **DOI:** 10.32996/jcsts.2022.4.2.3

1. Introduction
Currently, the use of the internet for search queries for various information almost reaches 70,000 searches per second (Soleymani et al., 2016). Searching information on websites can generally be used for various purposes (Tarafdar & Zhang, 2016), such as predicting user behavior, recommending a product, and optimizing the website function (Gunter & Önder, 2016). Forecasting is the best way to know website visitors (Dwivedi & Sachdeva, 2016). However, many researchers cannot determine which method is best used to solve the problem of forecasting website visitors. To forecast website visitors is to collect data on how many people visit the website (Li et al., 2021). The forecasting data usually uses time-series data (Douglas C. Montgomery, 2015). The Recurrent Neural Network (RNN) method is one of the best methods used in forecasting to complete tasks related to time series data (Hojjat Salehinejad, 2017). The development of the RNN method to solve the vanishing gradient and exploding gradient problems, namely the Gated Recurrent Unit (GRU) and Long Short-Term Memory methods (LSTM). These two methods can be said to be new. Researchers (Mateus et al., 2021; Yang et al., 2020) compare the LSTM and GRU methods to see the performance and speed of training. Researchers (Yamak et al., 2019) get the best forecasting results consecutively ARIMA, GRU, and LSTM methods for Bitcoin prices. Researchers (Wibawa et al., 2020) forecast the number of unique visitors on electronic journal websites by selecting the learning rate and determining the number of neurons in the LSTM process, affecting the performance tests. One component of electronic journal accreditation is the number of visitors to the journal's website (Nurdiani, 2018). Therefore, forecasting is needed to see how many users or website visitors are needed and whether or not a promotion is needed to increase visitors to the journal website (S. Syahrial, 2010). The data needed for forecasting website visitors is data on first-time visits and unique visits.

This study focused on forecasting visitors to electronic journal websites. This research compared three recurrent networks: standard RNN, LSTM, and GRU. This research contributes to proving the best performance of the forecasting method seen from the MSE results.

2. Literature Review

There are many methods used in forecasting research. One of the best today is using deep learning methods. The deep learning method we use in this research is Recurrent Neural Network (RNN). Berradi [3] conducted research comparing RNN with PCA and without PCA. In this study, the RNN method was used to predict the stock price of Total Maroc from the Casablanca stock exchange. The MSE value obtained by RNN with PCA is smaller, namely 0.00596, while the MSE value obtained by RNN without PCA is 0.011835 [3].

Another study using deep learning was conducted by (Gunter & Önder, 2016) that discussed the RNN, LSTM, and GRU methods to carry out the task of forecasting the COVID-19 pandemic. The results obtained are very helpful in preparation for controlling the pandemic. Ji et al. (Ji et al. 2022) considered the temporal and nonlinear characteristics of canyon wind speed data, a hybrid transfer learning model based on CNN and GRU, to predict short-term canyon wind speed with less observational data. Then, Li (Li et al., 2021) discussed the GRU for performing useful remaining-of-life prediction tasks. The results of the effectiveness of the method used for the aero ropilsion system simulation data from NASA Researchers (Soleymani et al., 2016) discussed the LSTM method to perform forecasting tasks in the field of demand production. The results show that the LSTM method is superior to other methods. Another deep learning research is the study done by Huang (Yamak et al., 2019), which discusses the LSTM, GPR, and EMDD methods to perform forecasting tasks in wind speed in a short time. The results show that the LSTM deep learning method is superior to other methods.

Based on several previous studies, deep learning obtained very good results in overcoming forecasting problems. Thus, this research used the RNN family methods for forecasting website visitors. RNNs can study dependencies between sequential or time-series input data. The ability of sequential dependency learning makes the RNN method very popular and widely used.

This study used deep learning in one family, the RNN, LSTM, and GRU methods to forecast website visitors. Comparison of these three methods to get the best results in the field of forecasting. The data used in this study were data on visitors to the Journal of Education website: Theory, Research, and Development of the State University of Malang. There were two types of data used, namely First Time Visits data and Unique Visits data. First-time visit data are data on website visitors who have just visited the website for the first time, while unique visit data are people who visit the website during a specified period (Ledford & Tyler, 2007; Wibawa et al., 2020).

3. Methodology

This research used deep learning of three recurrent networks: standard RNN, LSTM, and GRU. The explanation of the three recurrent networks is as follows.

3.1 Recurrent Neural Network (RNN)

RNN is a very good Neural Network for forecasting using time series data (Zhang, 2012). RNN is an artificial neural network (ANN) structure that uses feedback, using its output as one of the inputs for the next process (Abbasimehr et al., 2020; Elman, 1990). The RNN architecture can be seen in Fig. 1.

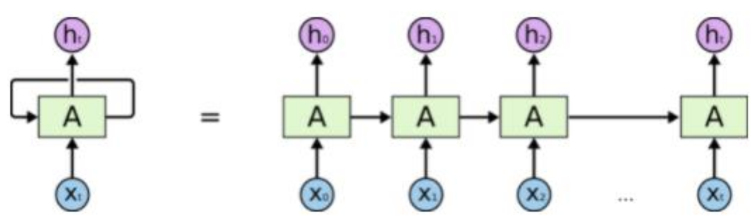

Fig 1. Recurrent Neural Network Architecture (Olah, 2015)

Figure 2 shows that the RNN has three layers, namely input (X), hidden (A), and output (h) (Apaydin et al., 2020). N input units is a vector sequence through time t i.e. xt = (x1, x2, ..., xN). Meanwhile, the recurrent hidden layer is connected directly to the input layer. Where M the hidden layer units is ht = (h1, h2, ..., hM). In the RNN method in the image on the right, the output process will refer to the previous calculation for each element in the sequence. RNN has a memory containing previously generated recorded information (Salehinejad et al., 2017).

RNN training can be done using a genetic algorithm (Reil & Husbands, 2002). Backpropagation is done to update based on the loss function, which is a function of weight (W) and bias (b). Parameters are shared equally on each time step in the network. The gradient for each output depends on the calculation of the current time step and the previous time step (Salehinejad et al., 2017). RNN training is divided into three parts: Forward propagation, Backward propagation, and Weight Update.

3.2 Long Short Term Memory (LSTM)

This study compared two methods: RNN, LSTM, and GRU. The LSTM and GRU methods are methods developed from the RNN method. LSTM can handle the vanishing gradient problem in training that is bound to happen in basic RNNs. In LSTM, there is a cell state. This cell state serves as memory or memory for a layer. Cell state values are manipulated using a gating system or a gate system. This gateway system consists of several simple neural network architectures to manage when data should be stored, used, or forgotten (Li et al., 2021).

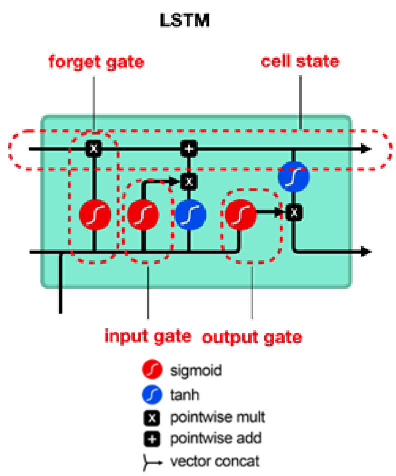

Fig 2. LSTM Architecture

Figure 2 shows the workflow of memory cells in each working LSTM neuron. There are three unit gates: the input gate, forget gate, and output gate (Huang et al., 2018). It contains three sigmoid activation functions and two tanh activation functions at each input to the neuron. The input gate determines whether an input will be added to the current cell state memory. Forget gate is useful for determining a memory at a previous time should be forgotten or not. While the output gate is useful for determining how much influence the cell state memory has on the prediction results that will be generated.

3.2.1 LSTM Training

Input gates consist of three sigmoid activation functions and one tanh activation function, each of which is useful for updating values and creating new value vectors to be stored in memory cells. As shown by equations 1 and 2:

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \qquad (1)$$
$$\tilde{C}_t = tanh(W_c.[h_{t-1}, x_t] + b_i) \qquad (2)$$

where: i_t is the Input gate, $\tilde{C}_t$ is a Memory cell, σ is the Learning rate, W_f a is the weight matrix of the forgetting gate, h_{t-1} is Output values at time points ti 1 and ti, x_t is the Input value, and b_i is Bias of input gate.

The cell gate will replace the value in the previous memory cell with the new memory cell value. This value is obtained by combining the values contained in the forget and input gates. With the Equation 3:

$$C_t = f_t * c_{t-1} + i_t * \tilde{C}_t \qquad (3)$$

where variable: C_t is Cell Gate, f_t is Forget gate, $\tilde{C}_t$ is a Memory cell, i_t is Input gate, W_f is Weight matrix of forgetting gate, and c_{t-1} is Cell status at time points ti 1 and ti

Forget gates information on each input data will be processed and selected, which data will be stored or discarded in memory cells. For output using the sigmoid activation function, if the value is 1, the data are stored, and if the value is 0, then the data are discarded (Li et al., 2021). With the Equation 4 :

$$f_t = \sigma\left(W_f . [h_{t-1}, x_t] + b_f\right) \qquad (4)$$

Where variable: f_t is Forget gate, σ is Learning rate, W_f is Weight matrix of forgetting gate, h_{t-1} is Output values at time points ti 1 and ti, x_t is the Input value, and b_f is Bias of forgetting gate.

The output gate is a sigmoid activation function and a tanh activation function, each of which is useful for determining the value of which part of the memory cell to be issued and placing that value in the memory cell. Finally, the value was issued by multiplying the two values. With the Equations 5 and 6:

$$o_t = \sigma(W_o . [h_{t-1}, x_t] + b_o) \qquad (5)$$
$$h_t = o_t * \tanh(C_t) \qquad (6)$$

Where variable: o_t is Output gate, σ is Learning rate, W_o is Weight matrix of output gate, h_{t-1} is Output values at time points ti 1 and ti, x_t is the Input value, and b_o is Bias of output gate.

3.3 Gated Recurrent Unit (GRU)

GRU is an architecture created by Kyunghun Cho in 2014 (Cho et al., 2014). The main purpose of creating a GRU is to make each recurrent unit capable of adaptively capturing dependencies on different time scales.

In the GRU, the component that regulates the flow of information is called a gate, and the GRU has two gates, namely the reset gate and the update gate. The gates are depicted as rectangles with their respective activation functions. The reset gate in the GRU determines whether new information should be forgotten or not, while the update gate is for remembering. GRU architecture is simpler than LSTM.

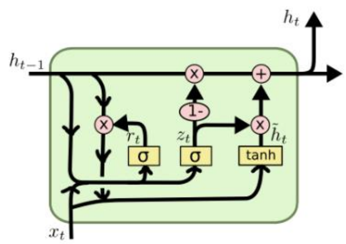

Fig 3. GRU Architecture

In Figure 3 above, r represents reset gates, and z represents update gates. Meanwhile, h and ĥ are activations and candidate activation. Activation and candidate activation are activation functions. The GRU uses two sigmoid activation functions and one tanh activation function

.

Equation 7 is for the reset gate:

$$r_t = \sigma(W_r . [h_{t-1}, x_r] + b_r) \qquad (7)$$

Where variable: r_t is reset gate, σ is Learning rate, W_r is Weight matrix of reset gate, h_{t-1} is output values at time points ti 1 and ti, x_r is the Input value, and b_r is Bias of resetting gate. The update gate is shown by equations 8, 9, 10:

$$z_t = \sigma(W_z . [h_{t-1}, x_z] + b_z) \qquad (8)$$
$$\tilde{h}_t = \tanh(W . [r_t * h_{t-1}, x_t]) \qquad (9)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t) \qquad (10)$$

Where variable: z_t is update gate, σ is Learning rate, W_z is Weight matrix of update gate, h_{t-1} is output values at time points ti 1 and ti, x_z is the Input value, and b_r is Bias of update gate.

The RNN method is very sensitive to fluctuations in time-series data, so the data must be normalized before entering into the neural network. MinMaxscaler is used to normalize the data (ArunKumar et al., 2021). The following equation represents the mathematically of MinMaxscaler:

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (11)$$

Where x is the original time-series data, x_n is the normalized time-series data, x_{min} and xmax are the minimum and maximum values of the time-series data.

The MinMaxscaler normalization method maintains the original distribution shape of the data and does not change the information embedded in the original data. The normalized data were divided into test and training data.

Mean Square Error (MSE) is the average squared error between the actual and forecast values. This method was used to test the performance of the proposed model. The MSE equation is as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - y)^2 \qquad (12)$$

Where yi is the actual value of the request, y is the value of the forecast result, and n is the amount of data.

The MSE method is generally used to check the estimation of the error value in forecasting. A low MSE value or a mean squared error value close to zero indicates that the forecasting results follow the actual data and can be used for forecasting calculations in the future.

4. Results and Discussion

This study used data on first-time visits and unique visits data. The journal website used in this study was the international journal website of the State University of Malang with the link journal.um.ac.id/index.php/jptpp/. The data used were data per month for two years. Data on first-time and unique visits are shown in Table 1.

Table 1. Data on first-time visits and data on unique visits in two years

Month	2018		2019	
	First Time Visits	Unique Visits	First Time Visits	Unique Visits
Jan	8852	9915	3947	5390
Feb	9415	10331	4727	6571
Mar	14863	16286	5776	6271
Apr	15.231	16.546	7.828	8.708
May	14439	15996	6682	7365
Jun	7.469	8.200	2.667	2.884
Jul	12247	13572	4425	4951
Aug	11.189	12.462	4.340	4.875
Sep	19.709	21.720	8.657	9.419
Oct	22.285	24.279	10.983	11.916
Nov	18.071	19.725	10.807	11.840
Dec	15.046	16.697	10.521	11.543

The number of data on first-time visits is 250176 visitors, and data on unique visits is 277462 visitors. The average number of first-time visits per month is 10424 visitors and the average number of unique visits per month are 11561 visitors. The training was carried out using python tools and the Google COLAB cloud computing platform, where the data used for training were 80%, and for testing were 20%. The methods used in this research are RNN, LSTM, and GRU. Based on the data, the three methods were trained at layers 1, 3, and 5. The training parameters for each method are input = 15, output = 1, back = 1, epochs 1 to 500, learning rate = 0.001.

Tests were conducted to determine the performance of RNN, LSTM, and GRU methods on journal website experiences. The epoch parameter started from 1 to 500 at layers 1, 3, and 5. Based on the random value in the initialization of the weight values, the training was carried out once in each layer.

4.1 The first test used first-time visit data.

The training results on the first-time visit data with testing obtained MSE values, as shown in Table 2. The data were trained from epochs 1 to 500, but the epoch data shown in the table were only epochs 100, 200, 300, 400, and 500.

Table 2. MSE value on first time visit data

epoch	Layer 1			Layer 3			Layer 5		
	RNN	LSTM	GRU	RNN	LSTM	GRU	RNN	LSTM	GRU
100	0,0167	0,0141	0,0156	0,0146	0,0141	0,0127	0,0141	0,0165	0,0176
200	0,0129	0,0126	0,0128	0,0129	0,0127	0,0126	0,0133	0,0128	0,013
300	0,0129	0,0125	0,0126	0,0127	**0,0125**	0,0127	0,0128	0,0126	0,0127
400	0,013	0,0125	0,0126	0,0128	0,0126	0,0127	0,0128	0,0127	0,0127
500	0,013	**0,0125**	0,0127	0,0127	0,0127	0,0127	0,0128	**0,0125**	0,0126

Table 2 shows the MSE value from the training results in the first-time visit data. The training carried out on layer 1 in the initial epoch, which is 1 to 100, shows that the MSE value is still large. Starting from epoch 300, the MSE results already look constant. The LSTM method shows the smallest MSE value in layer 1 from epoch 300 to 500 is 0.0125. The second test on layer 3 showed the same results for the LSTM method at epoch 300, and the MSE result was 0.0125. While for layer 5, the smallest MSE results show the smallest results for the LSTM method at epoch 500. From the three tests on layer 1, layer 3, and layer 5, the greater the epoch value, the smaller the MSE value, and it tends to be constant at epoch 500. The smallest MSE value in each layer from epoch 1 to 500 is 0.0125 for the LSTM method. The performance of the LSTM method is superior to others (Mateus et al., 2021; Yang et al., 2020). This is caused by the effect of the cell state on the resulting prediction.

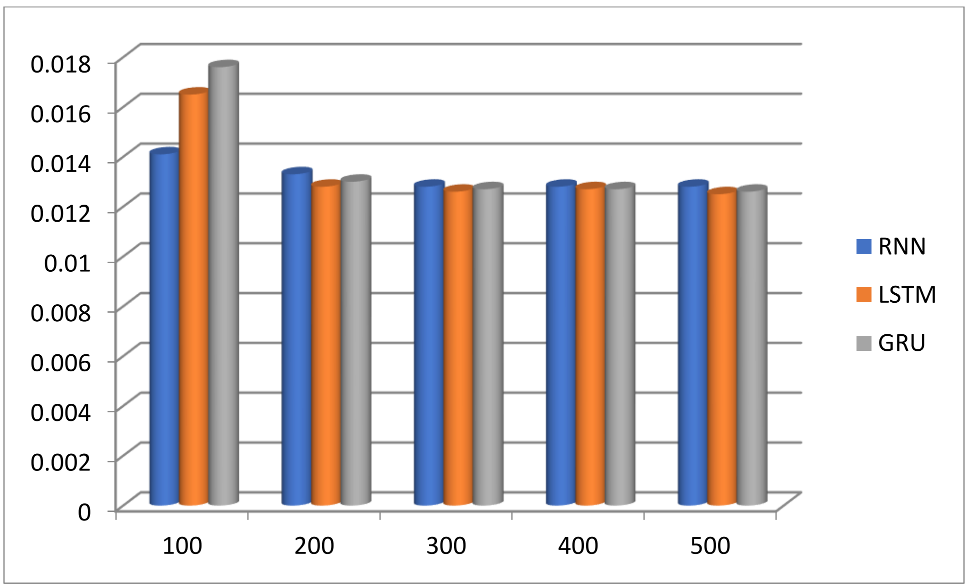

Fig 4. Graph of MSE value at layer 5 using first-time visit data

Figure 4 is a graphic image showing the MSE value for the LSTM method, which is smaller than the RNN and GRU methods from the smallest to the largest epoch.

4.2 The first test uses unique visit data

The results of the training on unique visit data by testing the MSE value are shown in table 3. The test was conducted to determine the RNN, LSTM, and GRU performance, with the same treatment as the first test. The training was carried out on layers 1, 3, and 5, while the epochs were from 1 to 500.

Table 3. MSE value of training results on unique visit data

epoch	Layer 1			Layer 3			Layer 5		
	RNN	LSTM	GRU	RNN	LSTM	GRU	RNN	LSTM	GRU
100	0,0276	0,0282	0,0276	0,0276	0,0294	0,0281	0,0272	0,0268	0,0285
200	0,0276	0,0274	0,0274	0,0276	0,0284	0,0276	0,0275	0,0266	0,0278
300	0,0283	0,0273	0,0277	0,0274	0,0274	0,0271	0,0273	0,0267	0,0276
400	0,0274	0,027	0,0271	0,0275	0,0268	0,0271	0,0273	0,0267	0,0271
500	0,0279	**0,0268**	0,0274	0,0274	**0,0267**	0,027	0,0273	**0,0265**	0,0269

Table 3 shows the MSE value of the training results on the unique visit data. The test results obtained the smallest MSE value at layer 1 epoch 500. The smallest MSE value for the LSTM method was 0.0268. While at layer 3 epoch 500, the smallest MSE value for the LSTM method is 0.0267. In layer 5 epoch 500, the smallest value for the LSTM method is 0.0265.

The smallest MSE value from epoch 1 to 500 is 0.0265 for layer 3 and layer 5 for the LSTM method. The smallest MSE value in each layer from epoch 1 to 500 is 0.0265 for the LSTM method.

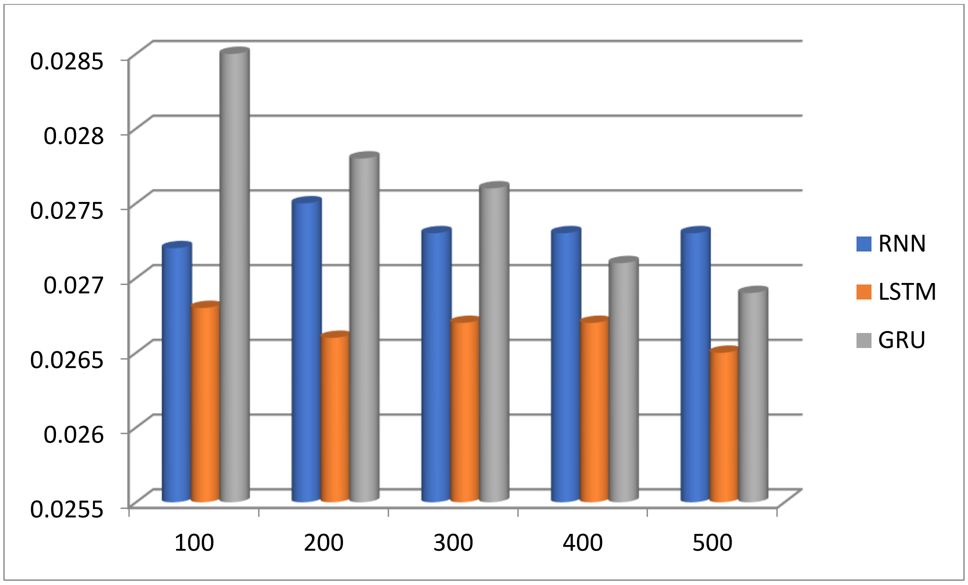

Fig 5. Graph of MSE values at layer 5 using unique visit data

Figure 5 shows a graph of the MSE value at layer 5. This image shows the MSE value for the LSTM method is smaller than the RNN and GRU methods from the smallest to the largest epoch.

The results of testing the MSE values in Table 2 and Table 3 prove the LSTM method is the best compared to RNN and GRU (Shewalkar, 2018).

5. Conclusion

This study compared the performance of the RNN, LSTM, and GRU methods for forecasting electronic journal visitors. The training was carried out at layers 1, 3, and 5 for each epoch from 1 to 500. The test used first-time visit data and unique visit data. Although tested with different data, the test results obtained the smallest MSE value for the LSTM method. The value of each MSE was 0.0125 for first-time visit data and 0.0265 for unique visit data. This study succeeded in showing the performance of the RNN, LSTM, and GRU methods with insignificant differences in MSE values. This is due to the LSTM architecture, which supports higher resampling rates, and can work on smaller and larger data sets. Provided that GPU technology can handle high computing load processes.

Funding: This research received no external funding.
Conflicts of Interest: "The authors declare no conflict of interest.
ORCID iD: https://orcid.org/0000-0003-1472-896X

References

[1] Abbasimehr, H., Shabani, M., & Yousefi, M. (2020). An optimized model using the LSTM network for demand forecasting. *Computers & Industrial Engineering*, *143*, 106435. https://doi.org/10.1016/j.cie.2020.106435

[2] Apaydin, H., Feizi, H., Sattari, M. T., Colak, M. S., Shamshirband, S., & Chau, K.-W. (2020). Comparative Analysis of Recurrent Neural Network Architectures for Reservoir Inflow Forecasting. *Water*, *12*(5). https://doi.org/10.3390/w12051500

[3] ArunKumar, K. E., Kalaga, D. V., Kumar, C. M. S., Kawaji, M., & Brenza, T. M. (2021). Forecasting of COVID-19 using deep layer Recurrent Neural Networks (RNNs) with Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) cells. *Chaos, Solitons & Fractals*, *146*, 110861. https://doi.org/10.1016/j.chaos.2021.110861

[4] Cho, K., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078 [cs.CL]*, 1724-1734.

[5] Douglas C. Montgomery, C. L. J., Murat K. (2015). *Introduction to Time Series Analysis and Forecasting*. Jhon Wiley.
[6] Dwivedi, N., & Sachdeva, S. (2016). Forecasting Visitor Accession Trend of Two Prominent Indian Health Journal Websites for the Period 2015–2020 Using Time Series Analysis. *Digital Medecine SSRN Elsevier*, *2*(2), 57-63.
[7] Elman, J. L. (1990). Finding Structure in Time. *Cogn. Sci*, *14*(2), 179-211.
[8] Gunter, U., & Önder, I. (2016). Forecasting city arrivals with Google Analytics. *Annals of Tourism Research*, *61*, 199-212.
[9] Hojjat Salehinejad, S. S., Joseph Barfett, Errol Colak, Shahrokh Valaee. (2017). Recent Advances in Recurrent Neural Networks. *arXiv:1801.01078*.
[10] Huang, Y., Liu, S., & Yang, L. (2018). Wind speed forecasting method using EEMD and the combination forecasting method based on GPR and LSTM. *Sustainability*, *10*, 3693. https://doi.org/10.3390/su10103693
[11] Ji, L., Fu, C., Ju, Z., Shi, Y., Wu, S., & Tao, L. (2022). Short-Term Canyon Wind Speed Prediction Based on CNN—GRU Transfer Learning. *Atmosphere*, *13*, 813. https://doi.org/10.3390/atmos13050813
[12] Ledford, J., & Tyler, M. E. (2007). *Google Analytics 2.0*. Wiley Publishing, Inc.
[13] Li, X., Law, R., Xie, G., & Wang, S. (2021). Review of tourism forecasting research with internet data. *Tourism Management*, *83*, 104245. https://doi.org/10.1016/j.tourman.2020.104245
[14] Mateus, B. C., Mendes, M., Farinha, J. T., Assis, R., & Cardoso, A. M. (2021). Comparing LSTM and GRU Models to Predict the Condition of a Pulp Paper Press. *Energies*, *14*(21). https://doi.org/10.3390/en14216958
[15] Nurdiani, A. (2018). *Pedoman Akreditasi Jurnal Ilmiah*. Direktorat Jenderal Penguatan Riset dan Pengembangan Kementerian Riset, Teknologi, dan Pendidikan Tinggi.
[16] Olah, C. (2015). Understanding LSTM Networks. *colah.github.io*.
[17] Reil, T., & Husbands, P. (2002). Evolution of central pattern generators for bipedal walking in a real-time physics environment. *IEEE Transactions on Evolutionary Computation*, *6*(2), 159-168. https://doi.org/10.1109/4235.996015
[18] Syahrial, K. M., and Nunung M. (2010). Analisa Statistik Pengunjung Situs Resmi Universitas Syiah Kuala (www.unsyiah.ac.id). *Jurnal Rekayasa Elektrika*, *9*(2), 49-54. https://doi.org/10.17529/jre.v9i2.165
[19] Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent Advances in Recurrent Neural Networks.
[20] Shewalkar, A. N. (2018). *Comparison of RNN, LSTM, and GRU on Speech Recognition Data* North Dakota State University]. North Dakota.
[21] Soleymani, M. R., Garivani, A., & Zare-Farashbandi, F. (2016). THE EFFECT OF THE INTERNET ADDICTION ON THE INFORMATION-SEEKING BEHAVIOR OF THE POSTGRADUATE STUDENTS. *Materia socio-medica*, *28*(3), 191-195. https://doi.org/10.5455/msm.2016.28.191-195
[22] Tarafdar, M., & Zhang, J. (2016). Analysis of Critical Website Characteristics: A Cross-Category Study of Successful Websites. *Journal of Computer Information Systems*, *46*(2), 14-24. https://doi.org/10.1080/08874417.2006.11645879
[23] Wibawa, A. P., Saputra, I. T., Utama, A. B. P., Lestari, W., & Izdihar, Z. N. (2020, 21-22 Oct. 2020). Long Short-Term Memory to Predict Unique Visitors of an Electronic Journal. 2020 6th International Conference on Science in Information Technology (ICSITech),
[24] Yamak, P. T., Yujian, L., & Gadosey, P. K. (2019). *A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting* Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence, Sanya, China. https://doi.org/10.1145/3377713.3377722
[25] Yang, S., Yu, X., & Zhou, Y. (2020, 12-14 June 2020). LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI),
[26] Zhang, G. P. (2012). Neural Networks for Time-Series Forecasting. In G. Rozenberg, T. Bäck, & J. N. Kok (Eds.), *Handbook of Natural Computing* (461-477). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-92910-9_14