Employing Long Short-Term Memory and Facebook Prophet Model in Air Temperature Forecasting

Article *in* Communication in Statistics- Simulation and Computation · January 2021

DOI: 10.1080/03610918.2020.1854302

6 authors, including:

Toni Toharudin
Universitas Padjadjaran
85 PUBLICATIONS **781** CITATIONS

Resa septiani Pontoh
Universitas Padjadjaran
38 PUBLICATIONS **274** CITATIONS

Rezzy Eko Caraka
Badan Riset dan Inovasi Nasional
204 PUBLICATIONS **2,383** CITATIONS

Solichatus Zahroh
Utrecht University
15 PUBLICATIONS **220** CITATIONS

Communications in Statistics - Simulation and Computation

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/lssp20

Employing long short-term memory and Facebook prophet model in air temperature forecasting

Toni Toharudin , Resa Septiani Pontoh , Rezzy Eko Caraka , Solichatus Zahroh , Youngjo Lee & Rung Ching Chen

To cite this article: Toni Toharudin , Resa Septiani Pontoh , Rezzy Eko Caraka , Solichatus Zahroh , Youngjo Lee & Rung Ching Chen (2021): Employing long short-term memory and Facebook prophet model in air temperature forecasting, Communications in Statistics - Simulation and Computation, DOI: 10.1080/03610918.2020.1854302

To link to this article: https://doi.org/10.1080/03610918.2020.1854302

Published online: 19 Jan 2021.

COMMUNICATIONS IN STATISTICS - SIMULATION AND COMPUTATION®
https://doi.org/10.1080/03610918.2020.1854302

Employing long short-term memory and Facebook prophet model in air temperature forecasting

Toni Toharudin[a,b], Resa Septiani Pontoh[a], Rezzy Eko Caraka[b,c,d], Solichatus Zahroh[a], Youngjo Lee[b,c], and Rung Ching Chen[d]

[a]Department of Statistics, Faculty of Mathematics and Natural Science, Padjadjaran University, Indonesia; [b]Department of Statistics, College of Natural Science, Seoul National University, Seoul metropolis, Republic of Korea; [c]Research for Basic Sciences, Lab Hierarchical Likelihood, College of Natural Science, Seoul National University, Seoul Metropolis, Republic of Korea; [d]Department of Information Management, College of Informatics, Chaoyang University of Technology, Taiwan (R.O.C.)

ABSTRACT

One of information needed in weather forecast is air temperature. This value might change any time. Prediction of air temperature is very valuable for some communities and occasions. Therefore, high accuracy prediction is needed. Since the information about air temperature might vary over time, it is necessary to implement methods that can adapt to this situation. The use of neural network methods such as long short term memory (LSTM), nowadays, becomes popular in facing big data including unexpected fluctuation on the data. Thus, the model is used in this paper which provides long series data on air temperature. In addition, recently, Facebook announced an accurate method of forecasting, called Prophet model's, for data which have trend, seasonality, holidays, missing data, not to mention outliers. Hence, the forecast of five-year daily air temperatures in Bandung on this paper is modeled by LSTM and Facebook Prophet. The result shows that, for minimum temperature, Prophet performs better on maximum air temperature while LSTM performs better on minimum air temperature. However, the difference on the value of RMSE is not too large significant.

ARTICLE HISTORY
Received 8 April 2020
Accepted 17 November 2020

KEYWORDS
Air temperature; Forecasting; LSTM; Machine learning; Prophet model

1. Introduction

One of information needed in weather forecast is air temperature because human life is influenced by physical of the environment and climatic conditions. Nowadays, the uncommon natural phenomenon, called extreme weather, is significantly increasing. High air temperatures occur accidentally. The city of Bandung, the capital city of West Java Indonesia, located at 06° 57 ′LS 107° 37′ BT, has a tropical climate causing the sun shines throughout the year which somehow impacting the high air temperature. Figure 1 shows the five-year daily maximum and minimum air temperature plots. The plot shows that the data seem remain stationary. However, Figure 1 mentions if the period of air temperature is shortened, the plot shows some fluctuations indicating that the data is not. This condition is supported by the fact that the weathers' elements tend to fluctuate rapidly (Hafni, Pujiastuti, and Harjupa 2015). Therefore, it allows air temperature data patterns to be non-stationary. Prediction of air temperature is very valuable for some communities and occasions. Therefore, high accuracy prediction is needed. There are many ways that

CONTACT Rezzy Eko Caraka rezzyekocaraka@gmail.com; rezzy94@snu.ac.kr Research for Basic Sciences, Lab Hierarchical Likelihood, College of Natural Science, Seoul National University, Seoul Metropolis, Republic of Korea.

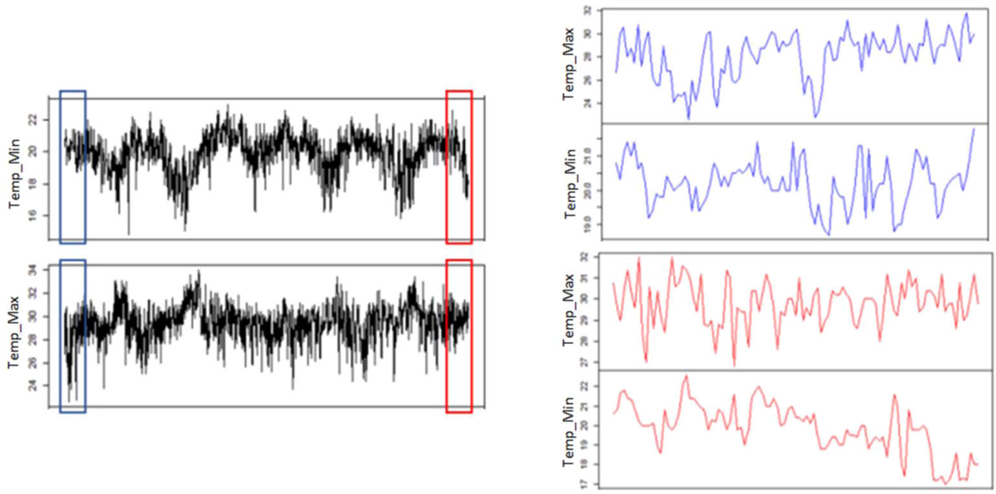


Figure 1. Daily maximum and minimum air temperature plots in Bandung (January 1st 2014 to June 30th 2019).

can be done, including by combining feature selection (Chen et al. 2020), combine parametric model and machine learning (Suhartono et al. 2012; Caraka et al. 2020). Since the information about air temperature might vary over time, it is necessary to implement methods that can adapt to this situation (Supari et al. 2017). The Meteorology, Climatology and Geophysics Agency (BMKG) of Bandung issues daily weather predictions that contain predictive values of various weather elements, including air temperature. Prediction value issued by BMKG Bandung is in the form of upper and lower limits of air temperature on that day (BMKG 2015; Caraka and Tahmid 2019).

Currently, BMKG weather prediction uses a numerical weather forecast, called Numerical Weather Prediction (NWP), based on the latest data and information from both observations and satellite monitoring (Putra, Saputro, and Kharisma 2018; Putra et al. 2019). However, the output of NWP is only exploratory (Supari et al. 2017). Therefore, alternative forecasting methods are needed as the alternative method to forecast the air temperature that can adjust to the changing on patterns of the air temperature (Rodríguez-Rajo, Frenguelli, and Jato 2003).

Various methods of predicting air temperature by several regions were previously being carried out by many expertise, such as Autoregressive Fractionally Integrated Moving Average (Ibrahim et al. 2011), fuzzy (Telesca et al. 2017), Autoregressive Integrated Moving Average (Patowary, 2017; Suhartono 2011), long memory (Hochreiter and Schmidhuber 1997; Caraka, Mahmud, and Sugiyarto 2016), SPLINE (Caraka and Devi 2016; Suparti, Prahutama, and Santoso 2018). Recently, due to its practicability, Neural Networks (NN) are becoming popular including on the air temperature forecast.

In addition, temperature values can fluctuate unexpectedly, so there are cases where old data can help the model identify trends and general movements that do not match the latest data (Akram and El 2016). Its ability to face the unexpected fluctuation on air temperature, has made the NN, currently, to be phenomenal. Since the daily air temperature contains long period sequence data, Long Short-Term Memory (LSTM) might be suitable to encounter this situation. Moreover, LSTM has been successfully applied on various sectors, such as stock market prediction (Lai, Chen, and Caraka 2019), flood forecasting (Le et al. 2019), sales forecasting (Helmini et al. 2019), travel time prediction (Yuan and Wang 2016), and weather prediction (Kumar and Singh 2019). Another popular forecasting method is Facebook Prophet composed by Facebook's Data Science Team intended to have balanced instinctive parameters without knowing the details of the fundamental model. It uses a decomposable time series model with three main model components: trend (Crawley 2012), seasonality (Hong et al. 2013), and holidays (Taylor and Letham 2018).

The outcome is the analyst-in-the-loop approach that utilizes human and machine automated undertaking. Prophet begins by demonstrating a time series using indicated parameters analyzing, producing forecasts, and evaluating them (Suhartono 2011). When the poor performance is recognized or an issue happens, Prophet Surfaces these issues to the analyst to enable them to comprehend what turned out badly and how to adjust the model depends on the criticism.

Prophet is optimized for the business forecast, but they claimed that Prophet makes it much straightforward to create a reasonable and accurate forecast. Prophet has been applied outside the business forecast, such as air pollution (Samal et al. 2019), bitcoin (Wu et al. 2018), and website traffic forecast (Subashini et al. 2019). Hence, the forecast for the air temperature will be used in this model too. The purpose of this research is to obtain forecasting models for daily maximum and minimum temperatures in the city of Bandung using the Long Short-Term Memory and Facebook Prophet Model.

This research is developing of previous research (Zahroh et al. 2019). Therefore, RNN cannot be used for data with a longer period of time and LSTM and Prophet's model can overcome this by adding a cell state so that the previous information can be used. It is expected that these methods can provide high accuracy so that it can be used to predict future air temperatures. The remainder of the paper is organized as follows. Section 2 provides the methodology. Section 3 offers results and discussion. Finally, conclusions and future research directions are indicated in Section 4.

2. Research method

2.1. Data sources and research variables

The data used in this research are historical data of daily maximum and minimum air temperatures in Bandung from January 1st, 2014 to June 30th, 2019 for 2007 periods. Forecasting methods that will be used in this research are Long Short Term Memory.

2.2. Artificial neural network

ANN was first introduced by McCulloch and Pits in 1943 (McCulloch and Pitts 1943). ANN is an information processing system that works in the same way with biological neural networks that are believed to be highly accurate (Fausett 1994). ANN has three types of layers: the input layer, output layer, and hidden layer. There are activation functions in the output and hidden layer. Some commonly used activation functions are *Sigmoid* and *Tanh* (tangent hyperbolic) (Toharudin et al. 2019). ANN is divided into two types, Feed Forward Neural Network (FFNN) and Recurrent Neural Network (RNN). FFNN is a network where connections between neurons in a layer do not form cycles, which means that the input only propagates forward from the input level to the output level (Hasbi, Budi, and Rukun 2018; Caraka, Baker, et al. 2019). When a feedback connection is added to the FFNN, the network is referred to as Recurrent Neural Network (RNN).

Because the neuron layer has its own connections, RNN is regarded as a network with memory (Lai, Chen, and Caraka 2019). RNN has unique characteristics that FFNN does not have, its architecture has at least one feedback loop so it can store data that brings information and will be accommodated for the next input. the optimization (Yang 2010) used in NN will also give different accuracy (Caraka, Chen, Yasin, et al. 2019).

However, RNN has a weakness, the occurrence of vanishing or exploding gradient problems during the training process as the periods for training data increase (Bengio, Simard, and

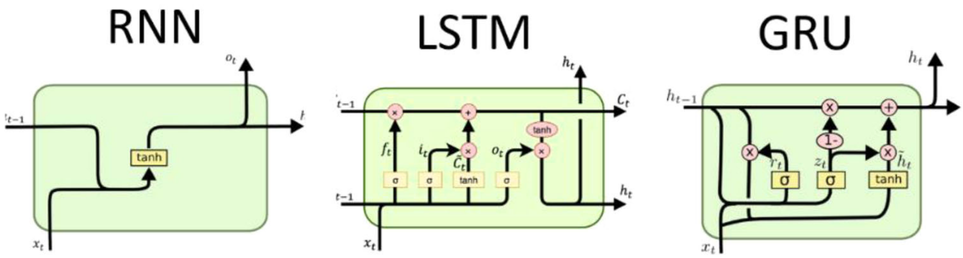


Figure 2. Network structure of RNN, LSTM, and GRU.

Frasconi 1994). Thus, if there is a gap between current data and earlier data, the RNN fails to connect the input, it is called long-term dependencies. The development of RNN that can overcome the problem of vanishing and exploding gradient is the Gated Recurrent Unit (GRU) which were first introduced by Cho et al. (2014) and Long Short Term Memory (LSTM), which were first introduced by Hochreiter and Schmidhuber (1997).

2.3. LSTM

GRU and LSTM have the same function, which is to find out whether there is a long-term dependency and to overcome the problem of vanishing and exploding gradient. LSTM does it through three gates, namely a forget gate that controls how much information needs to be removed, an input gate that controls how many cell states need to be stored, and an output gate that controls how many cell states are sent to the next cell have to. Refers to (Caraka, Chen, Supatmanto et al. 2019) see the visualization each gate. Whereas GRU works using two gates, a reset gate that lies between the previous activation function and the next candidate to eliminate previous information and an update gate that decides how much the candidate activation function is used to update the cell status. Each gate has weights and biases that are mutually independent of each other.

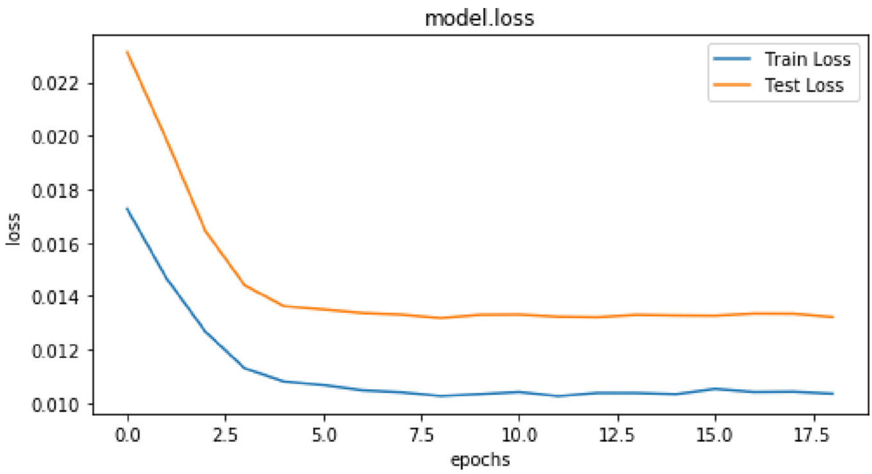
Figure 2 shows the structural differences from RNN, LSTM, and GRU. GRU uses fewer training parameters, so less memory is used and the training process is faster than LSTM. However, LSTM is able to achieve higher accuracy than GRU when long sequence data is used. The following steps must be carried out in LSTM: data pre-processing, data split, LSTM modeling (Karim et al. 2018).

The training process is carried out on training data with several network models and the best model is the model that has the smallest error value of various training models which is done by calculating the value of each gate function as much as the maximum epoch or until the target error is reached. The training process is carried out in a cell with three gates.

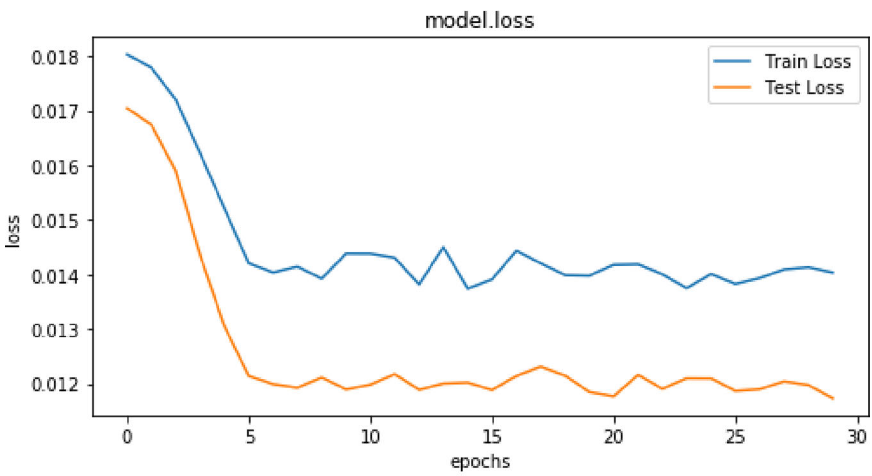
In the gate, there are several nomenclatures that are used: σ is the sigmoid activation function, *tanh* is the *tanh* activation function, $W_f, W_i,\ W_o,\ W_c$ are the weights of the forget gate, input gate, output gate, and cell state, $b_f,\ b_i,\ b_o,\ b_c$ are the biases of the forget gate, input gate, output gate, and cell state, h_{t-1} is the hidden state output at time *t-1*, x_t is the input at time *t*, and $\sim C_t$ is the intermediate cell state. LSTM section in Figure 3 is a flow of information ranging from forget gate to output gate. Information that passes through the forget gate (f_t) which has the following equation:

$$f_t = \sigma(W_f.[h_{t-1} + x_t] + b_f) \tag{1}$$

The input for forget gate, which is a hidden state at time $t-1$ and the current input at time *t*, entered through a sigmoid activation function. The output of this gate is 0 to 1 because the activation function used is sigmoid. If the value of f_t close to 0, the information from the previous cell is removed. If the value of f_t close to 1, this information is retained. Next, the output of this



(a)



(b)

Figure 3. Loss function chart for minimum air temperatures (a) and Maximum air temperatures (b) of the Best LSTM model.

gate is multiplied by element-wise multiplication with the cell state at time $t-1$. Next is the *input gate* (i_t). Input on this gate is the same as forget gate and the output produced is the same as the forget gate because it uses the same activation function, which is sigmoid, which has the following equation:

$$i_t = \sigma(W_i.[h_{t-1} + x_t] + b_i) \tag{2}$$

The output of the input gate is multiplied by element-wise multiplication with the output intermediate cell state ($\tilde{C}_t$), which has the following equation:

$$\tilde{C}_t = \tanh(W_C.[h_{t-1} + x_t] + b_C) \tag{3}$$

Cell state at time $t-1$ which has been multiplied with forget gate is updated with this output gate by element-wise addition to getting a new cell state (c_t) with the following equation:

$$c_t = (i_t * \tilde{C}_t + f_t * c_{t-1}) \tag{4}$$

The last gate is the output gate. Input on this gate is the same as the previous gate and the output produced is the same too because it uses the same activation function. Output gate (o_t) has the following equation:

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{5}$$

The new cell state that was previously obtained exits as input at time $t+1$ and through the tanh activation function to do an element-wise multiplication operation with the output gate, so that the output for the hidden state is obtained at time t (h_t) as in the following equation:

$$h_t = o_t * \tanh(c_t) \tag{6}$$

The training process in the cell is carried out until the learning process stops or reaches a predetermined stop condition. If not, then do the process again by updating the weights on the network. The criteria for stopping this process is reaching the maximum epoch or reaching the specified error target.

2.4. Facebook prophet model

Prophet is developed by Facebook's Data Science Team in 2017 (Vishwas and Patel 2020). It uses a decomposable time series model (Chung et al. 2014) with three main model components: trend, seasonality, and holidays.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \tag{7}$$

Where $g(t)$ is the trend function which models non-periodic changes, $s(t)$ is the seasonality that represents periodic changes (weekly and yearly), $h(t)$ are the effects of the holiday which occur on potentially irregular schedules, and ϵ_t is an error term that is not accommodated by the model respectively. Using time as a regressor, Prophet is trying to fit several linear and nonlinear functions of time as components. Two trend models that cover Facebook applications: a nonlinear saturating growth model and a piecewise linear model. A nonlinear model is typically modeled using the logistic growth model, which in its most basic form is

$$g(t) = \frac{C}{1 + \exp(-k(t-m))}, \tag{8}$$

Where C is the carrying capacity, k is the growth rate, m is an offset parameter. When the rate k is adjusted, the offset parameter must also be adjusted to connect the endpoints of segments. The piecewise logistic growth model is then:

$$g(t) = \frac{C(t)}{1 + \exp\left(-\left(k + \boldsymbol{a}(t)^T\boldsymbol{\delta}\right)\left(t - \left(m + \boldsymbol{a}(t)^T\boldsymbol{\gamma}\right)\right)\right)} \tag{9}$$

Where $\boldsymbol{\delta}$ and $\boldsymbol{\gamma}$ is a vector rate adjustment defines the change in the rate that occurs at the time s_j. The change points due to a phenomenon, which results in growth rate will change and the trend model is

$$g(t) = \left(k + \boldsymbol{a}(t)^T\boldsymbol{\delta}\right)t\left(m + \boldsymbol{a}(t)^T\boldsymbol{\gamma}\right) \tag{10}$$

where k is the growth rate, m is an offset parameter, $\boldsymbol{\delta}$ is the rate adjustment, and γ_j is set to $-s_j\delta_j$ to make the function continuous. In automatic change point's selection, $\delta_j \tilde{} \text{Laplace}(0, \tau)$ to fit the proposed model with seasonality effects and forecast based on it, it uses a Fourier series which provides a flexible model. Seasonal effects can be represented as in the following equation:

Table 1. The LSTM's parameter.

Characteristic	Specification
Network architecture	Learning rate : 0,001 (optimizer : Adam, RMSProp) Batch size : 1, 10, 20, 30 Number of hidden layers: 1 Number of neurons in the hidden layer: 1, 5, 10 Epoch : 10, 20, 30, 40, 50

$$s(t) = \sum_{n=1}^{N}\left(a_n\cos\left(\frac{2\pi nt}{P}\right) + b_n\sin\left(\frac{2\pi nt}{P}\right)\right) \tag{11}$$

Where P is a regular period.

Holidays and events often do not follow a periodic pattern, so their effects are not well modeled by a smooth cycle (Caraka, Bakar, et al. 2018). Prophet allows the analyst to provide a custom list of past and future events. A window around such days is considered separately and additional parameters are fitted to model the effect of holidays and events. Before analysis, splitting data into training and testing as the LSTM does.

2.5. Evaluation

After obtaining a model for prediction from the training process, the model is evaluated with testing data to get the predicted accuracy value of the model (De Gooijer and Hyndman 2006). When comparing forecasting methods with the same unit, RMSE is widely used (Armstrong and Collopy 1992). RMSE will be used with the following equation:

$$\text{RMSE} = \sqrt{\frac{1}{\boldsymbol{n}}\sum_{t=1}^{\boldsymbol{n}}(\boldsymbol{X_t} - \hat{\boldsymbol{X}}_{\boldsymbol{t}})^2} \tag{12}$$

Where n is the number of observations, $\boldsymbol{X_t}$ is the observed value, and $\widehat{\boldsymbol{X_t}}$ is the predicted value. A model can be said to be good if the smaller RMSE value is obtained. The range of RMSE values is $[0, \infty]$. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the RMSE is most useful when large errors are particularly undesirable (Chai and Draxler 2014; Caraka, Bakar, et al. 2019).

3. Results and discussion

This paper presents two different models of air temperature datasets. Thus, the Long Short-Term Memory (LSTM) and Facebook Prophet Model are used. All scripts have been written in Python programming language. As mentioned before, the dataset is splitting into training and testing. In this research, the data used amounted to 2007, with the composition of the data was 80% for training data (1605 data) and 20% for testing data (402 data). In LSTM, parameter initialization was carried out by trial and error. The parameters used to determine the optimal model are listed in Table 1.

In each gate, there are mutually independent weight and bias parameters. The weight initialization used is Glorot uniform, while the bias initialization is zero. Furthermore, weights and biases are updated with the backpropagation algorithm. Modeling in LSTM uses supervised learning, so the target will be determined first. The predictor data that will be used first is converted into 3-dimensional form, while the target data to be used is converted into a 2-dimensional form, namely [samples, time steps]. Furthermore, the training data that has been converted into a 3-dimensional form is used for LSTM modeling. From the model, an evaluation of the model is carried out on the test data so that the results of the model evaluation in the form of RMSE are

Table 2. RMSE value of LSTM model with different epochs.

	RMSE	
Maximum epoch	Max_Temp	Min_Temp
10	1,26	1,01
20	1,25	1,00
30	1,23	0,97
40	1,24	0,97
50	1,25	1,00

Table 3. RMSE value of LSTM model with different batch size.

	RMSE	
Batch size	Max_Temp	Min_Temp
1	1,23	0,97
10	1,38	1,37
20	1,35	1,36
30	1,42	1,36

Table 4. RMSE value of LSTM model with different number of neurons.

	RMSE	
Number of neurons	Max_Temp	Min_Temp
1	1,23	0,97
5	1,26	0,95
10	1,26	0,94

Table 5. RMSE value of LSTM model with different optimizer.

	RMSE	
Optimizer	Max_Temp	Min_Temp
adam	1,23	0,94
RMSProp	1,45	0,97

Table 6. RMSE value for LSTM and prophet model.

	RMSE	
Models	Max_Temp	Min_Temp
LSTM	1,23	0,94
Prophet	1,03	1,04

obtained. The parameter initialization was first carried out using the Adam optimizer, the batch size of 1, neurons in the hidden layer of 1 with various epochs. The lowest RMSE value is obtained with 30 epochs, so 30 epochs will be used with different batches, different neurons, and different optimizers to gained the lowest RMSE for forecasting. The RMSE value can be seen in Tables 2–6.

The lowest RMSE value is obtained with 30 epochs, 1 batch size (stochastic gradient descent), 1 hidden layer, 1 neuron in the hidden layer for maximum air temperature and 10 neurons for minimum air temperature, and using Adam Optimizer. Based on these characteristics, it can be seen a decrease in loss function until it reaches the local minimum in Figure 3. Based on the best model of LSTM, then it can be seen whether the LSTM or Prophet model has the smallest RMSE value. Hereafter, air temperature forecasting will use the model with the lowest RMSE.

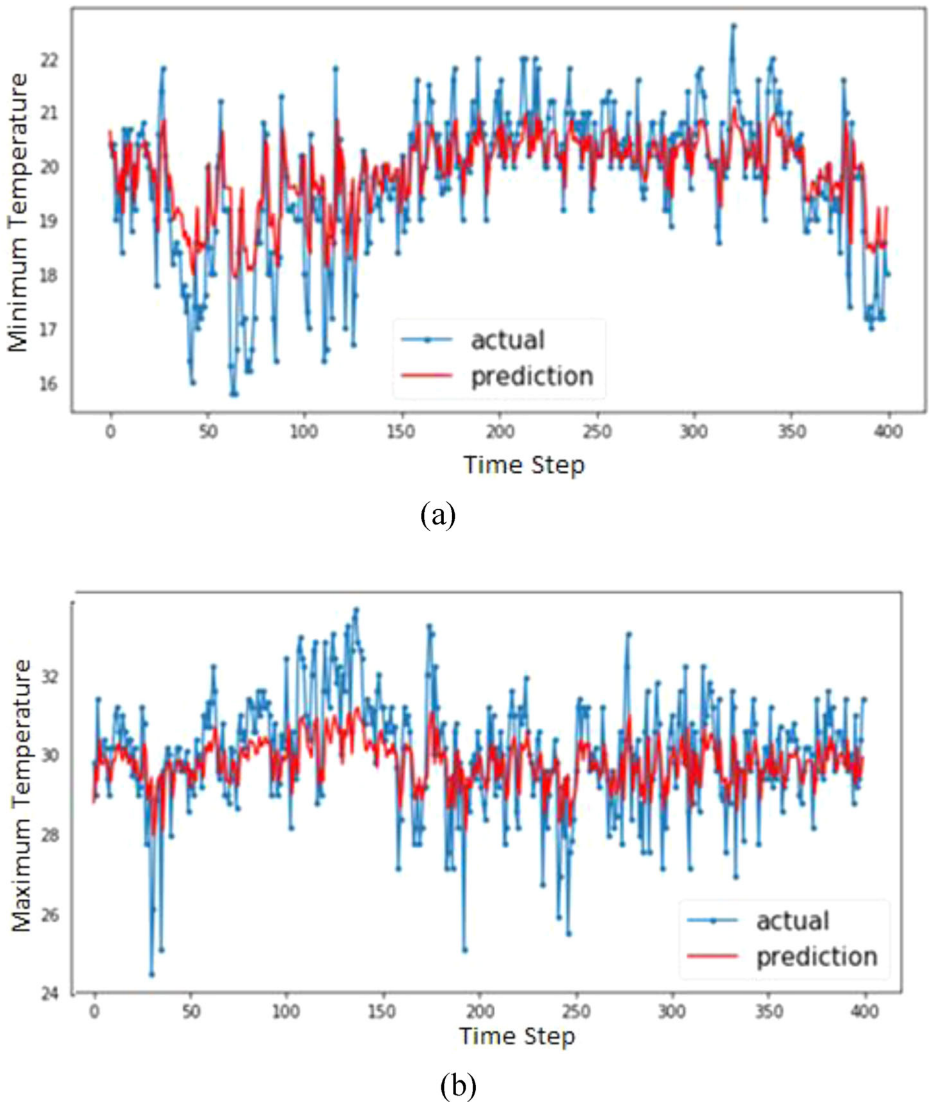


Figure 4. A plot between actual and prediction data on testing data for minimum air temperature (a) and Maximum (b) of the Best Model.

The best model for maximum air temperature is Prophet, while the LSTM is better for minimum air temperature. After determining the parameters that will be initialized to get the best model, the plot will be seen between the actual data and the predicted data from the best model for the testing data that can be seen in Figure 4.

4. Conclusion

Based on LSTM modeling conducted by trial and error it was found that the parameters for the best model were using the Adam optimizer with 1 batch size, 1 hidden layer, 10 epochs, 1 neuron in the hidden layer for maximum air temperature, and 10 neurons in the hidden layer. In this model, weights are initialized with Glorot uniform and biases are initialized by 0. So, that an RMSE of 1.23 is obtained for the maximum air temperature and 0.94 for the minimum air temperature. LSTM method is a non-parametric method that can be used for data with fast fluctuation. However, this method is considered unable to capture sharp fluctuations so the prediction values for the long term tend to be constant. The result of prediction generated from this method is only a value so that to obtain intervals value can be used Bayesian approach, Monte Carlo Dropout, or do resampling with Jackknife and Bootstrap methods. On the other hand, Prophet performs better than the LSTM model in forecasting maximum air temperature. RMSE obtained from Prophet is 1.03 for maximum air temperature and 1.04 for minimum temperature, where the error value can be said to be small enough to predict the air temperature. The advantage of this model is the duration of modeling is shorter than the LSTM model because there is no parameter to tune before, as the LSTM model. Both of LSTM and Prophet do well for air temperature forecasting. In future research, these methods can be used for other sectors to find out how well it can be used for other than weather forecasting, especially air temperature. Future work can perform comparisons of Learning rate, Batch size, Number of hidden layers, Number of neurons in the hidden layer, and consider the significant lag that comes from autoregressive and performing H-Likelihood (Caraka et al. 2020; Jin and Lee 2020).

Author contributions

T.T., R.S.P., R.E.C., and S.Z conceived and designed the experiments; R.S.P., R.E.C., and S.Z. performed the experiments; R.S.P., R.E.C., and S.Z. analyzed the data; R.S.P., R.E.C., and S.Z. preprocessed the base datasets; R.S.P., R.E.C. wrote the final paper; T.T., Y.L., and R.C.C provide the project funding. All authors read the paper and project administration. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported in part by the National Research Foundation of Korea under Grant [NRF-2019R1A2C1002408], in part by the Ministry of Science and Technology, Taiwan, under Grant [MOST-107-2221-E-324-018-MY2], and in part by the Universitas Padjadjaran under Grant [1733/UN6.3.1/LT/2020].

ORCID

Rezzy Eko Caraka http://orcid.org/0000-0002-1812-7478
Youngjo Lee http://orcid.org/0000-0001-9820-6434
Rung Ching Chen http://orcid.org/0000-0001-7621-1988

Data availability statement

The analysis code datasets used in this article available from the corresponding author upon reasonable request.

References

Akram, M., and C. El. 2016. Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. *International Journal of Computer Applications* 143 (11):7–11. doi: 10.5120/ijca2016910497.

Armstrong, J. S., and F. Collopy. 1992. Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting* 8 (1):69–80. doi: 10.1016/0169-2070(92)90008-W.

Bengio, Y., P. Simard, and P. Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* 5 (2):157–66. doi: 10.1109/72.279181.

BMKG. 2015. *Essentials of meteorology*. Meteorology, Climatology, and Geophysical Agency. Goodreads. doi: 10.1017/CBO9781107415324.004.

Caraka, R. E., S. Bakar, B. Pardamean, and A. Budiarto. 2018. Hybrid support vector regression in electric load during national holiday season. Proceedings - 2017 International Conference on Innovative and Creative Information Technology: Computational Intelligence and IoT, ICITech 2017. doi: 10.1109/INNOCIT.2017.8319127.

Caraka, R. E., S. A. Bakar, M. Tahmid, H. Yasin, and I. D. Kurniawan. 2019. Neurocomputing fundamental climate analysis. *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 17 (4):1818–27. doi: 10.12928/telkomnika.v17i4.11788.

Caraka, R. E., R. C. Chen, B. D. Supatmanto, M. Tahmid, and T. Toharudin. 2019. Employing moving average long short term memory for predicting rainfall. International Conference on Technologies and Applications of Artificial Intelligence (TAAI), 1–5. IEEE.

Caraka, R. E., R. C. Chen, H. Yasin, B. Pardamean, T. Toharudin, and S. H. Wu. 2019. Prediction of status particulate matter 2.5 using State Markov Chain stochastic process and HYBRID VAR-NN-PSO. *IEEE Access.* 7: 161654–65. doi: 10.1109/ACCESS.2019.2950439.

Caraka, R. E., and A. R. Devi. 2016. Application of non parametric basis spline (B-Spline) in temperature. *Statistika* 4 (2):68–74. Retrieved from http://jurnal.unimus.ac.id/index.php/statistik/article/viewFile/2236/2234.

Caraka, R. E., Y. Lee, R. C. Chen, and T. Toharudin. 2020. Using hierarchical likelihood towards support vector machine: Theory and its application. *IEEE Access* 8:194795–807. doi: 10.1109/ACCESS.2020.3033796.

Caraka, R. E., Z. Mahmud, and W. Sugiyarto. 2016. Model Long Memori Dalam Prediksi Suhu. *Seminar Hari Meteorologi Dunia 2016*, 222–36. Jakarta: Sekolah Tinggi Meteorologi Klimatologi dan Geofisika (STMKG).

Caraka, R. E., and M. Tahmid. 2019. *Statistika Klimatologi.* 1st ed. Yogyakarta: Mobius Graha Ilmu.

Chai, T., and R. R. Draxler. 2014. Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development* 7 (3):1247–50. doi: 10.5194/gmd-7-1247-2014.

Chen, R. C., C. Dewi, S. W. Huang, and R. E. Caraka. 2020. Selecting critical features for data classification based on machine learning methods. *Journal of Big Data* 7 (1):52. doi: 10.1186/s40537-020-00327-4.

Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. doi: 10.3115/v1/d14-1179.

Chung, J., G. Caglar, Cho, K. H., and Yoshua, B. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. NIPS 2014 Workshop on Deep Learning, 1–9.

Crawley, M. J. 2012. Time series analysis. In *The R Book*, 785–808. John Wiley & Sons. doi: 10.1002/9781118448908.ch24.

De Gooijer, J. G., and R. J. Hyndman. 2006. 25 Years of time series forecasting. *International Journal of Forecasting* 22 (3):443–73. doi: 10.1016/j.ijforecast.2006.01.001.

Fausett, L. V. 1994. *Fundamental of neural network: Architecture, algorithm and application.* 1st edition (December 19, 1993). Pearson.

Hafni, W., D. Pujiastuti, and W. Harjupa. 2015. Analisis variabilitas temperatur udara di daerah Kototabang periode 2003–2012. *Jurnal Fisika Unand* 4 (2):185–92.

Hasbi, Y., W. Budi, and S. Rukun. 2018. Feed forward neural network modeling for rainfall prediction. *E3S Web of Conferences* 73:05017. doi: 10.1051/e3sconf/20187305017.

Helmini, S., N. Jihan, M. Jayasinghe, and S. Perera. 2019. Sales forecasting using multivariate long short term memory networks. *PeerJ Preprints.* 7:e27712v1. doi: 10.7287/peerj.preprints.27712.

Hochreiter, S., and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9 (8):1735–80. doi: 10.1162/neco.1997.9.8.1735.

Hong, W. C., Y. Dong, W. Y. Zhang, L. Y. Chen, and B. K. Panigrahi. 2013. Cyclic electric load forecasting by seasonal SVR with chaotic genetic algorithm. *International Journal of Electrical Power & Energy Systems* 44 (1): 604–14. doi: 10.1016/j.ijepes.2012.08.010.

Ibrahim, L., B. Asare, M. Yakubu, and U. Dauda. 2011. Modeling of Sokoto daily average temperature: A fractional integration approach. *Nigerian Journal of Basic and Applied Sciences* 19 (1):21–30. doi: 10.4314/njbas.v19i1.69340.

Jin, S. and Y. Lee. 2020. A review of h-likelihood and hierarchical generalized linear model. *WIREs Computational Statistics.* Advance online publication. doi: 10.1002/wics.1527.

Karim, F., S. Majumdar, H. Darabi, and S. Chen. 2018. LSTM fully convolutional networks for time series classification. *IEEE Access.* 6:1662–9. LSTM Fully Convolutional Networks for Time Series. doi: 10.1109/ACCESS.2017.2779939.

Kumar, J., and A. K. Singh. 2019. Cloud resource demand prediction using differential evolution based learning. 7th International Conference on Smart Computing & Communications (ICSCC), Sarawak, Malaysia, 1–5.

Lai, C. Y., R. C. Chen, and R. E. Caraka. 2019. Long short term memory (LSTM) to predict the average stock market. The International Conference on Machine Learning and Cybernetics (ICMLC). Japan: IEEE.

Le, X. H., H. V. Ho, G. Lee, and S. Jung. 2019. Application of long short-term memory (LSTM) neural network for flood forecasting. *Water (Switzerland)* 11:2–19. doi: 10.3390/w11071387.
McCulloch, W. S., and W. Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics* 5 (4):115–33. doi: 10.1007/BF02478259.
Patowary, A. N. 2017. Monthly temperature prediction based on Arima model: A case study in Dibrugarh Station of Assam, India. *International Journal of Advanced Research in Computer Science* 8 (8):292–98. doi: 10.26483/ijarcs.v8i8.4590.
Putra, R. M., A. Kurniawan, A. Rangga, I. A. Ryan, M. Endarwin, and A. Luthfi. 2019. An evaluation graph of hourly rainfall estimation in Malang. IOP Conference Series: Earth and Environmental Science. doi: 10.1088/1755-1315/303/1/012031.
Putra, R. M., A. H. Saputro, and S. Kharisma. 2018. Visualization of volcanic ash distribution based on multispectral satellite imagery: A comparing method. ISSIMM 2018 - 3rd International Seminar on Sensors, Instrumentation, Measurement and Metrology, Proceeding. doi: 10.1109/ISSIMM.2018.8727641.
Rodríguez-Rajo, F. J., G. Frenguelli, and M. V. Jato. 2003. Effect of air temperature on forecasting the start of the Betula pollen season at two contrasting sites in the south of Europe (1995-2001). *International Journal of Biometeorology* 47 (3):117–25. doi: 10.1007/s00484-002-0153-z.
Samal, K. K. R., K. S. Babu, S. K. Das, and A. Acharaya. 2019. Time series based air pollution forecasting using SARIMA and Prophet Model. Proceedings of the 2019 International Conference on Information Technology and Computer Communications, Singapore, 80–85. doi: 10.1145/3355402.3355417.
Subashini, A., K. Sandhiya, S. Saranya, and U. Harsha. 2019. Forecasting website traffic using prophet time series model. *International Research Journal of Multidisciplinary Technovation* 1 (1):56–63.
Suhartono, 2011. Time series forecasting by using seasonal autoregressive integrated moving average: Subset, multiplicative or additive model. *Journal of Mathematics and Statistics* 7 (1):20–27. doi: 10.3844/jmssp.2011.20.27.
Suhartono, I. Puspitasari, M. S. Akbar, and M. H. Lee. 2012. Two-level seasonal model based on hybrid ARIMA-ANFIS for forecasting short-term electricity load in Indonesia. ICSSBE 2012 - Proceedings, 2012 International Conference on Statistics in Science, Business and Engineering: "Empowering Decision Making with Statistical Sciences. doi: 10.1109/ICSSBE.2012.6396642.
Supari, Tangang, F., L. Juneng, and E. Aldrian. 2017. Observed changes in extreme temperature and precipitation over Indonesia. *International Journal of Climatology* 37 (4):1979–97. doi: 10.1002/joc.4829.
Suparti, Prahutama, A., and R. Santoso. 2018. Mix local polynomial and spline truncated: The development of nonparametric regression model. *Journal of Physics: Conference Series* 1025 (2018):012102. doi: 10.1088/1742-6596/1025/1/012102.
Taylor, S. J., and B. Letham. 2018. Forecasting at scale. *American Statistician* 72 (1):37–45. doi: 10.1080/00031305.2017.1380080.
Telesca, V., D. Caniani, S. Calace, L. Marotta, and I. M. Mancini. 2017. Daily temperature and precipitation prediction using neuro-fuzzy networks and weather generators. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Part VI, LNCS 10409, 441–55. doi: 10.1007/978-3-319-62407-5_31.
Toharudin, T.,. R. E. Caraka, A. Bachrudin, S. A. Bakar, and D. Ambarsari. 2019. Copula-based feedforward neural network genetic algorithm cargo forecasting. *International Journal Advances in Soft Computing and Its Application* 11 (1):22–23.
Vishwas, B., and A. Patel. 2020. *Hands-on Time Series Analysis with Python. From Basics to Bleeding Edge Techniques*. Berkeley, CA: Apress. doi: 10.1007/978-1-4842-5992-4.
Wu, C. H., C. C. Lu, Y. F. Ma, and R. S. Lu. 2018. A new forecasting framework for bitcoin price with LSTM. IEEE International Conference on Data Mining Workshops, ICDMW, 168–75.
Yang, X. 2010. Nature-inspired metaheuristic algorithms. In *Nature-inspired metaheuristic algorithms second edition*. Elsevier. doi: 10.1016/B978-0-12-416743-8.00005-1.
Yuan, Y., and F. Y. Wang. 2016. Towards blockchain-based intelligent transportation systems. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, Rio de Janeiro, Brazil. doi: 10.1109/ITSC.2016.7795984.
Zahroh, S., Y. Hidayat, R. S. Pontoh, A. Santoso, F. Sukono, and A. T. Bon. 2019. Modeling and Forecasting Daily Temperature in Bandung. Proceedings of the International Conference on Industrial Engineering and Operations Management Riyadh, Saudi Arabia, 406–12.