Daily Temperature Prediction Using Recurrent Neural Networks and Long-Short Term Memory

Ike Sri Rahayu, Esmeralda C Djamal, Ridwan Ilyas
Department of Informatics, Jenderal Achmad Yani University, West Java, Indonesia
ikesri.rahayu@student.unjani.ac.id, esmeralda.contessa@lecture.unjani.ac.id,
ilyas@lecture.unjani.ac.id

Abdul Talib Bon
Department of Production and Operations, University Tun Hussein Onn Malaysia, Malaysia
talibon@gmail.com

Abstract

Temperature is one of the parameters that need to be considered because it is related to daily activities. Besides, the temperature is influenced by other parameters such as humidity, rainfall, and wind speed in the surrounding area. Data was obtained from Meteorology, Climatology, and Geophysics (BMKG) Bandung, West Java, from 2000-2019. This paper builds a model that can predict daily temperatures over the next three days with five classes, namely "Cold", "Cool", "Normal", "Warm" and "Hot" using Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM). Before being predicted, pre-processing is needed to improve data quality consisting of interpolation, feature extraction, normalization, and segmentation. We use two optimization models, SGD and Adam. The results of this study prove that using Adam produces the best testing 90.92% for training data and 80.36% for test data. The amount of data used and the sharing of data can also affect the accuracy of the results obtained.

Keywords
temperature, prediction, RNN, LSTM

1. Introduction

Temperature is one of the essential weather parameters because it is useful for various industrial, agricultural, energy, environmental needs (Abdel-Aal 2004). In the community's environmental requirements, the temperature can affect clothing that is suitable for daily life (Kwon and Choi 2012). The increase in the surface temperature of the earth, known as global warming, causes changes in climate patterns. As a result of climate change, including drought, bad weather, in the field of the industry can cause food shortages, increase the spread of disease, damage to infrastructure, and damage to natural resources as people's livelihoods (Ben et al. 2015). The temperature rises or falls can make the body uncomfortable using clothes that are not appropriate to the ambient air temperature. This allows the public to have to adjust clothing materials to be used at certain temperatures. Therefore, several studies have conducted temperature predictions.

Previous studies predicted temperatures used some methods. Some of them are Support Vector Regression (SVR) to heating or air conditioning systems (Paniagua-Tineo et al. 2011); Fuzzy Time Series (FTS) for handle temperature prediction problems (Chen, Member, and Hwang 2000); Backpropagation and Genetic Algorithms to determine temperature dependence on specific data (Singh, Bhambri, and Gill 2011); Empirical Mode Decomposition (EMD) and Least Squares Support Vector Machine (LS-SVM) to predict changes that occur over time regularly (Ding-cheng and Chun-xiu 2010), and Artificial Neural Network (ANN) to predict snow avalanches (A. Ganju and Piyush Joshi 2012). Some studies predict average temperatures using maximum, minimum, and average air temperature data (Afzali, Afzali, and Zahedi 2012) monthly (Alvares et al. 2013), weekly (Kumar 2012), daily (He et al. 2009).

One way to be able to predict temperature is by utilizing machine learning such as Deep Learning. Deep Learning has several variations, one of which is Recurrent Neural Networks (RNN). Some studies used RNNs to predictions for stock returns (Rather, Agarwal, and Sastry 2015), estimated short-term housing costs (Kong et al. 2019), summer prediction (Saha and Mitra 2016), predictions of electrical voltage instability (Ibrahim and El-Amary 2018). RNN is also used to predict by generating two types of models to estimate weather data 24 and 72 hours (Zaytar and Amrani 2016). RNN is often used for sequential data such as time series, financial data, weather, video (Fente and Kumar Singh 2018), audio (Feng et al. 2017), and text (Yogatama et al. 2017).

The research builds a model that can predict daily temperatures using RNN. The training data used as input is by using four parameters, namely temperature, humidity, rainfall, and wind speed in the last 20 years taken from the Meteorology Climatology and Geophysics Agency (BMKG) in Bandung. The data will be pre-processed by interpolating data to improve data that is not measured and is not readable. The data will then be conducted training using RNN, which will produce weights. The weight will be used to predict the temperature of the air in the next three days.

2. Data

In this study, temperature prediction uses other weather parameters, such as temperature, humidity, rainfall, and wind speed from 2000 to 2019—data obtained from Meteorology, Climatology, and Geophysics (BMKG) Bandung for the past 20 years. Weather data produces 7305 data stored in the format (.csv). The weather parameter data can be seen in Table 1.

Table 1. Climate Dataset

Days to-	**Date**	**Temperature (°C)**	**Humidity (%)**	**Rainfall (mm)**	**Wind Speed (m/s)**
1.	01-01-2000	23.5	78	2.8	3
2.	02-01-2000	22.8	83	0	3
…	…	…	…	…	…
15	15-01-2010	22.9	77	7.2	2
16.	16-01-2010	22	83	8888	2
17.	17-01-2010	22.4	83	1.5	2
…	…	…	…	…	…
7304.	30-12-2019	24.4	80	0	1
7305.	31-12-2019	23.4	88	0.8	0

3. Methods

Weather parameter data will enter the pre-process stage to correct data that was not measured, calculate the average value every three days for each parameter, grouping data into a data set and normalize data for each feature in the zero to one range. Then in the training process using RNN and LSTM to get the weights that will be used as new input neurons in the prediction process.

3.1. Pre-processing

Pre-processing is the initial stage of the data input process which is where the data will pass through several processes. These pre-processing stages include interpolation, feature extraction, normalization, and segmentation.

3.1.1. Interpolation

Interpolation is used to find the middle value between two values to correct data that is not measured. Interpolation calculations are shown in equation (1).

$$X_a = \frac{(X_b + X_c)}{2} \tag{1}$$

The above calculation explains that X_a is the value to be searched for, X_b is the value that existed before X_a, and X_c is the value that comes after it X_a.

3.1.2. Feature Extraction

The feature extraction process is used to find the average value every three days from each weather parameter, which is temperature, humidity, rainfall, and wind speed. The calculation of feature extraction is shown in equation (2).

$$\sum_{i=1}^{14} = average\ X_i \tag{2}$$

Explained it, $\sum_{i=1}^{14}$ is the amount of data used and $average\ X_i$ is calculating the average of the data used.

3.1.3. Normalization

Normalization is needed when there are differences in the range of values held by air temperature, humidity, rainfall, and wind speed. Because of variations in the range of values, the data may become too large or too small. Therefore, the data normalization process is needed to make the data values uniform. The normalization process is done by using the equation (3).

$$Z = \frac{x - \min()}{\max() - \min()} \tag{3}$$

Where x is data that will be normalized, max is the highest data in the column, and min is the lowest data in the column.

3.1.4. Segmentation

Segmentation is the process of grouping data into a data set. In this study in one data, 3 days will be grouped into three months using the overlap process. The overlap is done so that the data have interrelations between one data with other data and prevent data discontinue. The results of segmentation can be seen in Figure 1.

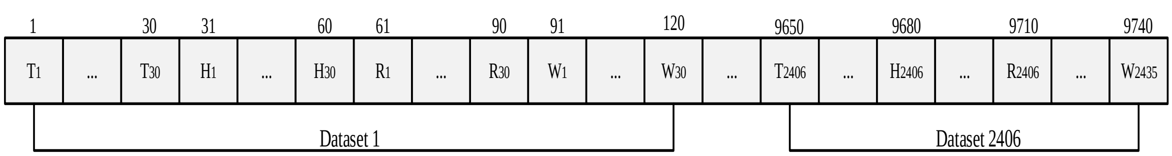

Figure 1. Data Segmentation

The first input parameter is the temperature (T), which starts from one to three days to 30 or three months, then followed by other parameters, namely humidity (H), rainfall (R), and wind speed (W) until the dataset to 2406. The results of this segmentation process are 2406 datasets sorted by time with four input parameters.

3.2. Recurrent Neural Networks

Recurrent Neural Networks is a variation of the methods of Deep Learning. RNN is an artificial neural network that uses recurrence by utilizing past data. RNN is used to estimate a situation in the future (Abdel-Nasser and Mahmoud 2017), RNN is a modification of Feedforward Neural Network with the characteristic of using feedback from output to input. Besides relying on input, the RNN output also depends on the previous state of the network that acts as memory (Alhagry, Aly, and A. 2017).

RNN consists of an input layer, a hidden layer, and output layer. RNN has a hidden layer that is connected to the hidden layer and the next input layer. Based on Figure 2, RNN has characteristics that each input is interconnected with one another. The input layer has a sequence length of a data set in a time-based sequence of features. The data set consists of four weather parameters, namely, temperature, humidity, rainfall, and wind speed. The number of neurons entered for the first training process is 120 neurons generated from the segmentation process is 2406 datasets. Neurons in the input layer are connected to neurons in the hidden layer. The hidden layer calculation is seen in equation (4), and the output calculation uses the softmax function, which can be seen in equation (5).

$$h_t = (Ux_t + Wh_{t-1}) \tag{4}$$
$$o_t = softmax(Vh_t) \tag{5}$$

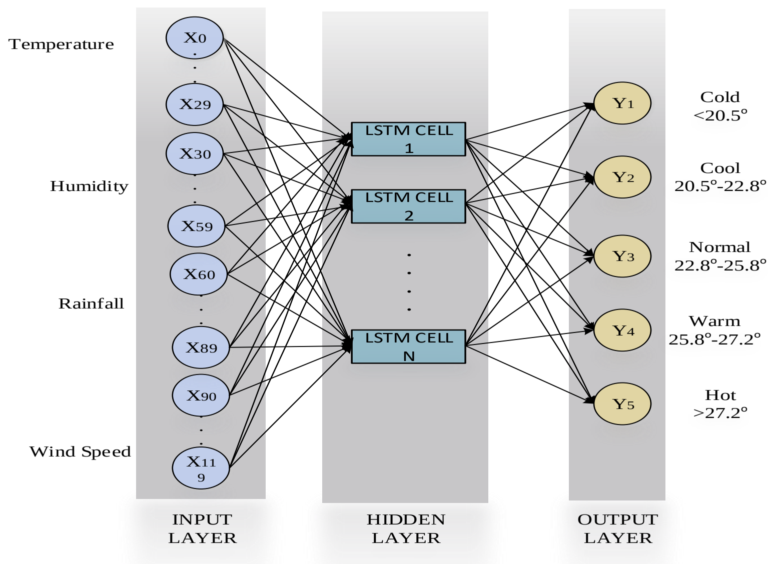

Figure 2. RNN LSTM Temperature Forecasting Structure

RNN has the disadvantage of short-term memory. There are several RNN processing models such as Gated Recurrent Unit (GRU), Backpropagation Through Time (BPTT), and Long Short Term Memory (LSTM) (Ningsih, Djamal, and Najmurrakhman 2019). Therefore, in making predictions using quite a lot of data and having difficulty in maintaining previous information, this is called vanishing gradient or short-term memory. RNN overcomes these weaknesses by using the Long Short-Term Memory (LSTM) processing model. LSTM has advantages that can handle vanishing gradient problems (Kong et al. 2019).

3.3. Long Short-Term Memory

Long Short Term Memory (LSTM) is a processing module of the Recurrent Neural Network that is capable of learning long-term dependencies (Kök, Şimşek, and Özdemir 2018). The difference between LSTM and traditional RNN neural networks is that each neuron in LSTM is a memory cell (Tsai, Zeng, and Chang 2018).

LSTM can manage memory at each input through cell memory or gate units in each of its neurons as shown in Figure 3. Each neuron contains three gates, namely the input gate, forget gate, and output gate. Forget gate in this LSTM serves to regulate the flow of information. So, LSTM can learn which information will be stored or delete information that is passed. This gate can also carry out relevant information in making predictions.

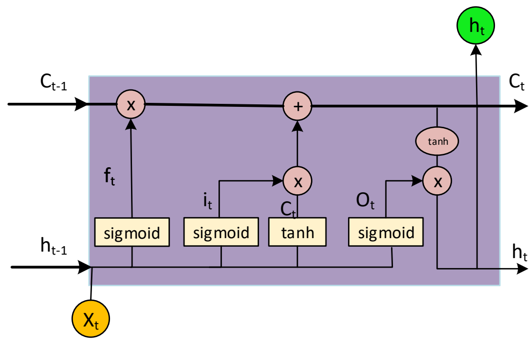

Figure 3. A LSTM Cell

Each LSTM network produces two-cell state values that will be reused with new input values and the output values will be stored in temporary memory. Memory in LSTM is called cells that take input from the previous state (h_{t-1}) and current input (x_t). The collection of cells decides what will be stored in memory and what will be deleted from memory (Yao, Huang, and Jia 2018).

RNN and LSTM are distinguished by the existence of a cell state that is used as a pathway to connect the data flow from each gate. Where the gate consists of the Binary Sigmoid function (σ) and multiplication operations. The ReLU activation function is changing data into a range (0-x) that can be seen in equation (6). The first step in LSTM is to decide what information will be removed from the cell state called forget gate using equation (7).

$$ReLU(x) = max\ (0, x) \tag{6}$$
$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{7}$$

Where the values h_{t-1} and x_t are ranged from zero to one for each cell. Information will be saved if the value obtained is close to one, otherwise, the information will be deleted if the value obtained is close to zero. The second step is to decide on new information to be stored. In this step there are two parts, the first is the input gate which determines the value to be stored and updated cell state using equation (8) and determines the new cell candidate ($\tilde{C}_t$) using the activation function tanh using equation (9).

$$i_t = \sigma(W_f.[h_{t-1}, x_t] + b_i) \tag{8}$$
$$\tilde{C}_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{9}$$

In updating the cell state by summing the old cell state (C_t) with the candidate cell state ($\tilde{C}_t$) using equation (10) where the old cell state is multiplied by forget state and then added to the result of the multiplication of candidate cells with input gate.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{10}$$

The final step in LSTM is the output gate to determine the output value that will be generated based on the information provided from the cell state where the calculation using the Sigmoid Binary activation function can be seen in equation (11) and the results of the calculation are multiplied by the Tanh activation function that has been updated using equation (12).

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{11}$$
$$h_t = o_t * \tanh(C_t) \tag{12}$$

Where σ is a sigmoid activation function. tanh is a tangent activation function. W_f, W_c, W_o are weight weights. h_{t-1} is the previous hidden state. b_i, b_c, b_o are bias vectors. Next, calculate the error in the output layer using the Mean Square Error (MSE). MSE represents the average absolute error between the predicted results and the target value. MSE calculations are performed using equation (13).

$$MSE = \nabla E = \sum_{k=1}^{m}(t_k - y_k)^2 \quad (13)$$

Where k is the output neuron m is the target value is the number of neurons in the output layer, y_k is the result of activation of the output layer and t_k is every target value k.

4. Result and Discussion

Climate data were obtained from the Meteorology, Climatology, and Geophysics Agency (BMKG) of Bandung City for 20 years with parameters used namely temperature, humidity, rainfall, and wind speed.

4.1. Comparing Two Optimization Models

Comparing the two optimization models is done to find out which is more appropriate to use in predicting temperatures. Optimization models used in testing are Stochastic Gradient Descent (SGD) and Adaptive Moment Estimation (Adam). Both of the optimization models are used to update the weights during training. Tests carried out using 100 epochs. Accuracy results from testing the two optimization models can be seen in Table 2.

Table 2. Comparing Two Optimization Models

No	Optimization Model	Training Data		Testing Data	
		Loss	Accuracy (%)	Loss	Accuracy (%)
1.	SGD	0.01271	87.24	0.01346	76.48
2.	Adam	0.01041	90.92	0.01079	80.36

The use of Adam's optimization model results in better accuracy because Adam repeats iteratively to update the weight so that it produces a smaller loss value, while SGD only uses a few sets of data at random to produce a changeable loss value. The graph of the test results using the SGD and Adam optimization models is shown in Figure 4. The accuracy results of the Adam model are shown in Figure 5 and the results of the SGD model accuracy are shown in Figure 6.

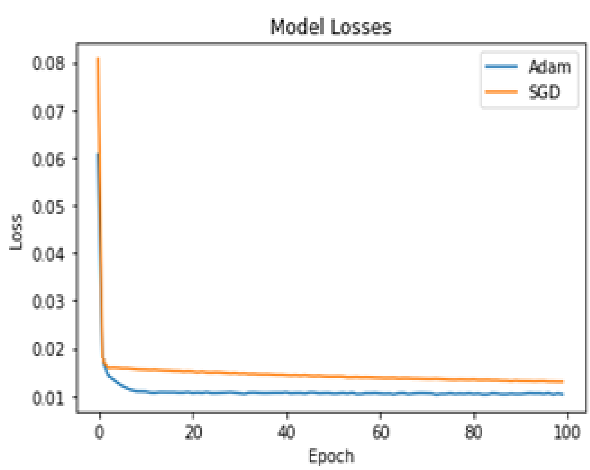

Figure 4. Loss of SGD and Adam Model

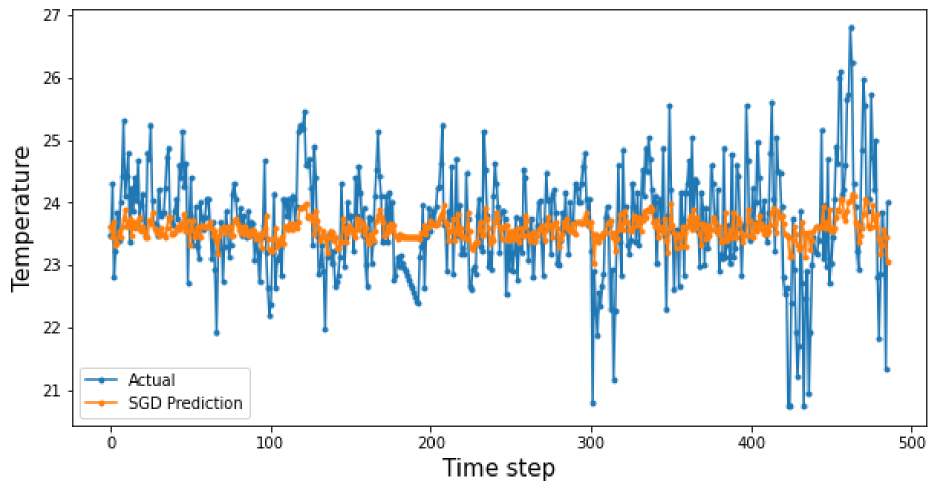

Figure 5. Accuracy of SGD Model

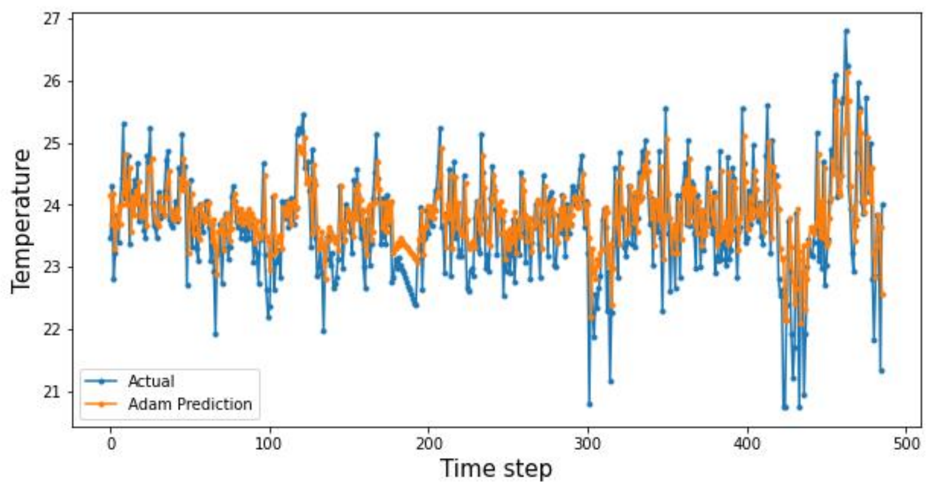

Figure 6. Accuracy of Adam Model

4.1. The Influence of the Amount of Data Set

This study predicts temperatures every 3 days with the amount of data from the past 20 years. The training process is carried out using Adam's optimization model with an epoch of 100. A comparison of the amount of data tested is twenty years, twelve years, and the last four years. The results obtained from the trial amount of data can be seen in Table 3.

Table 3. Accuracy of RNN Model Configuration

No	Dataset	Training Data		Testing Data	
		Loss	Accuracy (%)	Loss	Accuracy (%)
1.	20 years	0.01041	90.92	0.01079	80.36
2.	12 years	0.01127	89.34	0.01292	78.26
3.	4 years	0.01525	83.33	0.01325	76.54

Table 6 shows the differences in loss and accuracy values based on the number of datasets. The reduction in the number of datasets used causes higher loss results and lower accuracy results. Testing using a 20-year dataset produces an accuracy of 90.92% for training data and 80.36% for test data. Whereas using the 4-year dataset produces an accuracy of 83.33% for training data and 76.54% for test data. The conclusion resulting from this test is the number of data sets used can affect the accuracy results obtained.

4.1. The Influence of Data Sharing

Data sharing testing is performed on training data and test data that aims to determine the effect of comparison of the amount of training data and test data on learning outcomes. test data sharing 80:20, 70:30, 60:40 as shown in Table 4.

Table 4. Accuracy of Data Sharing Testing Results

Training (%)	**Testing** (%)	**Training Data**		**Testing Data**	
		Loss	Acc (%)	Loss	Acc (%)
80	20	0.01041	90.92	0.01079	80.36
70	30	0.01266	86.94	0.01166	79.14
60	40	0.01375	85.26	0.01471	75.34

This study uses the Adaptive Moment Estimation (Adam) optimization model with 100 epochs as a comparison of the three data shares. This test produces an accuracy of 90.92% by using data sharing 80% of training data and 20% of test data. And the more the distribution of small training data, the accuracy will decrease.

5. Conclusion

This research has predicted air temperature using Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM). The temperature prediction model consists of three processes. The first process is a pre-process consisting of interpolation, feature extraction, data normalization and segmentation. The second process is the training process using RNN and LSTM. The third process is testing.

The results of this study indicate that the use of RNN and LSTM can be used for daily temperature prediction. The test was carried out using two optimization models namely SGD and Adam. The results obtained using Adam's optimization model with 100 epochs in the training data yielded an accuracy of 90.92%, and for the test data, the accuracy produced 80.36%. Predictions using data from the past twenty years have gotten better results than predictions using data from the last four years. Sharing data with 80:20 also gets better accuracy. Therefore, it can be concluded that the optimization model, the amount of data, and data sharing can affect the results obtained.

References

A. Ganju and Piyush Joshi. 2012. “Maximum and Minimum Temperature Prediction over Western Himalaya Using Artificial Neural Network.” *Formerly Indian Journal of Meteorology, Hydrology & Geophysics* 2(April):283–90.

Abdel-Aal, R. E. 2004. “Hourly Temperature Forecasting Using Abductive Networks.” *Engineering Applications of Artificial Intelligence* 17(5):543–56.

Abdel-Nasser, Mohamed and Karar Mahmoud. 2017. “Accurate Photovoltaic Power Forecasting Models Using Deep LSTM-RNN.” *Neural Computing and Applications* 1–14.

Afzali, Mahboubeh, Afsaneh Afzali, and Gholamreza Zahedi. 2012. “The Potential of Artificial Neural Network Technique in Daily and Monthly Ambient Air Temperature Prediction.” *International Journal of Environmental Science and Development* 3(1):33–38.

Alhagry, Salma, Aly Aly, and Reda A. 2017. “Emotion Recognition Based on EEG Using LSTM Recurrent Neural Network.” *International Journal of Advanced Computer Science and Applications* 8(10):8–11.

Alvares, Clayton Alcarde, José Luiz Stape, Paulo Cesar Sentelhas, and José Leonardo de Moraes Gonçalves. 2013.

"Modeling Monthly Mean Air Temperature for Brazil." *Theoretical and Applied Climatology* 113(3–4):407–27.

Ben, Mustapha, El Houari, Omar Zegaoui, and Abdelaziz Abdallaoui. 2015. "Prediction of Air Temperature Using Multi-Layer Perceptrons with Levenberg-Marquardt Training Algorithm." *International Research Journal of Engineering and Technology* 02(08):2395–56.

Chen, Shyi-ming, Senior Member, and Jeng-ren Hwang. 2000. "Temperature Prediction Using Fuzzy Time Series." *IEEE Transactions on Systems, Man and Cybernetics* 30(2):263–75.

Ding-cheng, Wang and Wang Chun-xiu. 2010. "Air Temperature Prediction Based on EMD and LS-SVM." *Fourth International Conference on Genetic and Evolutionary Computing* 177–80.

Feng, Weijiang, Naiyang Guan, Yuan Li, Xiang Zhang, and Zhigang Luo. 2017. "Audio Visual Speech Recognition with Multimodal Recurrent Neural Networks." *International Joint Conference on Neural Networks (IJCNN)* 681–88.

Fente, Dires Negash and Dheeraj Kumar Singh. 2018. "Weather Forecasting Using Artificial Neural Network." *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018* 1757–61.

He, Y., F. Wetterhall, H. L. Cloke, F. Pappenberger, M. Wilson, and J. Freer. 2009. "Forecast of Daily Mean, Maximum and Minimum Temperature Time Series by Three Artificial Neural Network Methods." *Meteorological Applications* 101(February):91–101.

Ibrahim, Amr M. and Noha H. El-Amary. 2018. "Particle Swarm Optimization Trained Recurrent Neural Network for Voltage Instability Prediction." *Journal of Electrical Systems and Information Technology* 5(2):216–28.

Kök, Ibrahim, Mehmet Ulvi Şimşek, and Suat Özdemir. 2018. "A Deep Learning Model for Air Quality Prediction in Smart Cities." *International Conference on Big Data (BIGDATA) A* 1983–90.

Kong, Weicong, Zhao Yang Dong, Youwei Jia, David J. Hill, Yan Xu, and Yuan Zhang. 2019. "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network." *IEEE Transactions on Smart Grid* 10(1):841–51.

Kumar, Pankaj. 2012. "Minimum Weekly Temperature Forecasting Using ANFIS." *Computer Engineering and Intelligent Systems* 3(5):1–6.

Kwon, Juyoun and Jeongwha Choi. 2012. "The Relationship Between Environmental Temperature and Clothing Insulation Across a Year." *International Journal Biometeorol* 56:887–93.

Ningsih, Fitriana R., Esmeralda C. Djamal, and Asep Najmurrakhman. 2019. "Wind Speed Forecasting Using Recurrent Neural Networks and Long Short Term Memory." *Proceedings of the 2019 6th International Conference on Instrumentation, Control, and Automation, ICA 2019* (August):137–41.

Paniagua-Tineo, A., S. Salcedo-Sanz, C. Casanova-Mateo, E. G. Ortiz-García, M. A. Cony, and E. Hernández-Martín. 2011. "Prediction of Daily Maximum Temperature Using a Support Vector Regression Algorithm." *Renewable Energy* 36(11):3054–60.

Rather, Akhter Mohiuddin, Arun Agarwal, and V. N. Sastry. 2015. "Recurrent Neural Network and a Hybrid Model for Prediction of Stock Returns." *Expert Systems with Applications* 42(6):3234–41.

Saha, Moumita and Pabitra Mitra. 2016. "Recurrent Neural Network Based Prediction of Indian Summer Monsoon Using Global Climatic Predictors." *Proceedings of the International Joint Conference on Neural Networks* 1523–29.

Singh, Shaminder, Pankaj Bhambri, and Jasmeen Gill. 2011. "Time Series Based Temperature Prediction Using Back Propagation with Genetic Algorithm Technique." *International Journal of Computer Science Issues* 8(5):28–32.

Tsai, Yi Ting, Yu Ren Zeng, and Yue Shan Chang. 2018. "Air Pollution Forecasting Using RNN with LSTM." *Proceedings - IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, IEEE 16th International Conference on Pervasive Intelligence and Computing, IEEE 4th International Conference on Big Data Intelligence and Computing and IEEE 3* 1068–73.

Yao, Wanye, Pu Huang, and Zhaoxin Jia. 2018. "Multidimensional LSTM Networks to Predict Wind Speed." *Chinese Control Conference, CCC* 2018-July:7493–97.

Yogatama, Dani, Chris Dyer, Wang Ling, and Phil Blunsom. 2017. "Generative and Discriminative Text Classification with Recurrent Neural Networks." *Thirty-Fourth International Conference on Machine Learning (ICML 2017).*

Zaytar, Mohamed Akram and Chaker El Amrani. 2016. "Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks." *International Journal of Computer Applications* 143(11):975–8887.

Biographies

Ike Sri Rahayu is a graduate student in informatics from Universitas Jenderal Achmad Yani.

Esmeralda Contessa Djamal received a Bachelor's degree in Engineering Physics from Institut Teknologi Bandung in 1994, a Master's degree in Instrument and Control from Institut Teknologi Bandung in 1998. She finished a Ph.D. degree in Engineering Physics from Institut Teknologi Bandung in 2005. Her doctoral thesis focused on signal processing and pattern recognition. Since 1998 until now, she has published about 40 papers in an international journal or proceeding primarily in pattern recognition, machine learning, and signal processing. Currently, she is a lecturer of the Informatics Department, Universitas Jenderal Achmad Yani.

Ridwan Ilyas received a Bachelor's degree in Computer Science from Universitas Jenderal Achmad Yani in 2014, a Master's degree in Artificial Intelligent from Institut Teknologi Bandung in 2017. Now he is a student in the doctoral program from Institut Teknologi Bandung and lecturer of Informatics Department Universitas Jenderal Achmad Yani.

Abdul Talib Bon is a professor of Production and Operations Management in the Faculty of Technology Management and Business at the Universiti Tun Hussein Onn Malaysia since 1999. He has a Ph.D. in Computer Science, which he obtained from the Universite de La Rochelle, France, in the year 2008. His doctoral thesis was on topic Process Quality Improvement on Beltline Moulding Manufacturing. He studied Business Administration in the Universiti Kebangsaan Malaysia for which he was awarded the MBA in the year 1998. He's a bachelor's degree and a diploma in Mechanical Engineering which he obtained from the Universiti Teknologi Malaysia. He received his postgraduate certificate in Mechatronics and Robotics from Carlisle, United Kingdom in 1997. He had published more 150 International Proceedings and International Journals and 8 books. He is a member of MSORSM, IIF, IEOM, IIE, INFORMS, TAM, and MIM.