

## Research Article



## Time Series Prediction of Near-Surface Temperature Using an LSTM Neural Network

Sin-Gil Kim\*, Un-Song Ryu

*Department of Information Science, University of Sciences, Unjong District, Pyongyang, Democratic People's Republic of Korea*

\*Correspondence: [ksg1957@star-co.net.kp](mailto:ksg1957@star-co.net.kp)

Received: 13 February 2025 / Accepted: 15 September 2025 / Published: 14 October 2025

**Abstract:** Predicting air and soil temperature is an active research topic in response to adverse changes in the global ecological environment, such as global warming and biodiversity loss. To improve prediction performance, various machine learning models and methods have been applied, some of which are costly to implement because they use spatiotemporal data such as global numerical weather prediction data. This study aims to establish a simple method for constructing a predictive model with relatively high accuracy. In this paper, we propose a method to construct a prediction model using a long short-term memory (LSTM) neural network with only time series data of measured temperatures at certain time intervals at stations in the region. The proposed method was applied to predict daily minimum and maximum temperatures for up to 10 days using only time series of near-surface temperatures observed at meteorological stations in Pyongyang and in coastal and inland areas of the Democratic People's Republic of Korea during 2019–2023. The results show that the proposed model achieved relatively high and stable prediction performance using only observed data. The mean accuracy within  $\pm 2$  °C (ACC) was 73.70% for 4-day and 87.97% for 10-day maximum temperature predictions.

**Keywords:** earth science, air temperature, multi-step forecasting, meteorological stations, machine learning

## INTRODUCTION

In earth science, the prediction of air temperature or soil temperature is of great importance to understand and predict the changes in the environment around the earth and its surroundings. Air temperature is an important factor for agriculture, industry, energy, environment, tourism, human health, energy consumption prediction, and natural disaster prediction. Also, soil temperature depends on factors such as topography, seasonality, solar radiation, air temperature, soil moisture, vegetation cover, etc., which have important effects on the geochemical cycle of organisms, plant physiology, and the ecological environment. Currently, there is a growing body of research on the prediction of air temperature or soil temperature in response to the negative changes in the global ecological environment, such as global warming and biodiversity loss (Cifuentes et al., 2020; Penghui et al., 2020). We limit our discussion to the prediction of air temperature only. Air temperature is measured near the surface at certain time intervals at local meteorological stations, and it is possible to predict future air temperature if these observed air temperature data are available. However, it is not easy to increase its prediction accuracy because the time series of air temperature has considerable temporal and spatial variability and exhibits nonlinear behavior. Hence, there has been a growing body of research on air temperature prediction using artificial intelligence, especially machine learning (Chevalier et al., 2011; Kong et al., 2022; Smith et al., 2006).

In general, in the prediction problem, using past input-output data to reflect the causal relationship between causal factors and the resulting factors, better prediction results can be obtained. However, if the data of causal factors are insufficient or difficult to use, the influence of several causal factors on the time series data of the resulting factors can be considered to be reflected, and predictions using only those time series data at much lower cost can be made. Typical machine learning models used in time series prediction include recurrent neural networks (RNN), fuzzy neural networks, support vector regression (SVR) models, convolutional neural networks (CNN), long short-term memory (LSTM) neural networks (simply, LSTM networks), and transformer neural networks. In particular, RNNs are one of the most popular models that have been widely used for time series prediction (Bengio et al., 1994; Elman, 1990; Hewamalage et al., 2021; Torres et al., 2021). The RNN has a feedback loop on each hidden layer unit, so that the input information is continuously connected. However, RNNs have difficulty capturing

long-term dependencies due to gradient vanishing and explosion (Hewamalage et al., 2021). To improve the performance of RNN, an LSTM model has been proposed, which can learn long-term time series relationships without memory loss by adjusting the parameters of memory cells and gates (i.e., input gates, forget gates, and output gates) (Hochreiter & Schmidhuber, 1997; Karevan & Suykens, 2020; Yu et al., 2021).

The main objective of this study is to develop a model that can predict daily minimum and maximum temperatures relatively accurately for several days using only the observed near-surface temperature time series data, using the LSTM network, and to verify its good performance, thus proposing a method to solve the problem of prediction of air temperature or soil temperature with relatively high prediction performance using only observed data. This paper is organized as follows. First, we describe the methodology of time series prediction using LSTM network. Next, we discuss the performance evaluation and results for the time series prediction problem of near-surface temperature. Finally, the conclusions are presented.

## METHOD

### Principle of LSTM

The LSTM network was developed to overcome the vanishing-gradient problem in standard RNNs by improving the gradient flow of the network. The LSTM operation is to overcome the limitations of the RNN to realize not only short-term memory but also long-term memory. The original idea of LSTM was developed by Hochreiter and Schmidhuber (1997), and since then different approaches (Cho et al., 2014; Gers & Schmidhuber, 2000; Graves & Schmidhuber, 2005; Schmidhuber et al., 2007) have been proposed to improve the performance of LSTM. LSTM uses gate control mechanisms to control the information that must be maintained over time, its duration, and the reading time of that information through the memory cell, and to resolve gradient vanishing and explosion problems (Graves et al., 2009). The principles of LSTM are described in the framework of the popular approach in many studies, including Yu et al. (2021).

The main idea of LSTM is the state of the memory cell, and the addition or removal of information from the state of the memory cell is achieved by the input gate, output gate, and forget gate (Figure 1).

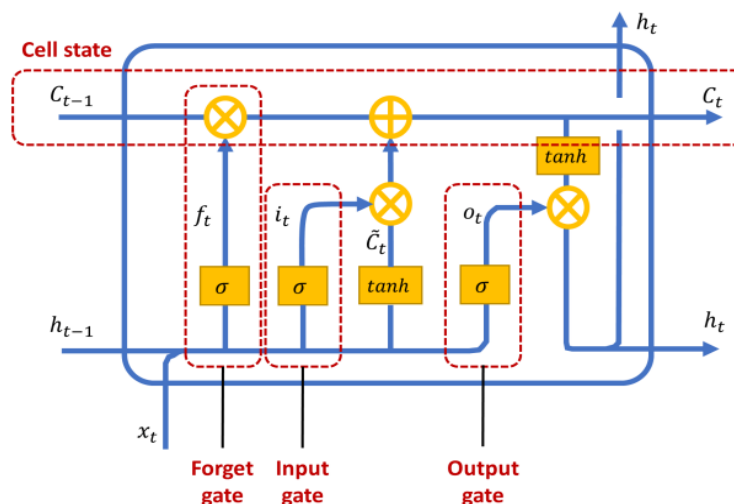


Figure 1. LSTM memory cell (Alerskans et al., 2022).

The LSTM memory cell behavior at time step  $t$  is described as follows. In Figure 1  $i_t$  is the variable used to update the cell state as the output of the input gate,  $f_t$  is the output of the forget gate, which is the variable that determines which information should be kept without forgetting in the cell state, and  $o_t$  is the variable to update the hidden state as the output of the output gate.  $\tilde{C}_t$  is a variable that represents potentially new important information about the memory cell state, and  $C_t$  is a variable that represents a new state of the memory cell.  $h_t$  is also a variable that represents the updated hidden state at time step  $t$ , computed by picking important information from the cell state.

These variables are mathematically calculated as follows.

Now, let the hidden state vector at time step  $t-1$  be  $\mathbf{h}_{t-1}$ , and the input vector at time step  $t$  be  $\mathbf{x}_t$ , then the output vector  $\mathbf{i}_t$  of input gate, the output vector  $\mathbf{f}_t$  of forget gate, and the output vector  $\mathbf{o}_t$  of output gate respectively are obtained as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{xi}\mathbf{x}_t + \mathbf{b}_i) \quad (1)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{xf}\mathbf{x}_t + \mathbf{b}_f) \quad (2)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{xo}\mathbf{x}_t + \mathbf{b}_o) \quad (3)$$

where  $\mathbf{W}_{hi}$ ,  $\mathbf{W}_{hf}$ ,  $\mathbf{W}_{ho}$ ,  $\mathbf{W}_{xi}$ ,  $\mathbf{W}_{xf}$  and  $\mathbf{W}_{xo}$  are weight matrices, and  $\mathbf{b}_i$ ,  $\mathbf{b}_f$ ,  $\mathbf{b}_o$  are bias vectors. Also,  $\sigma(u) = 1/(1 + e^{-u})$  is applied to the elements of a vector, whose values lie between 0 and 1.

The new state vectors  $\tilde{\mathbf{C}}_t$  of the memory cell are generated using  $\mathbf{x}_t$  and  $\mathbf{h}_{t-1}$  as follows:

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{W}_{xc}\mathbf{x}_t + \mathbf{b}_c) \quad (4)$$

where  $\mathbf{W}_{hc}$  and  $\mathbf{W}_{xc}$  are the weighting matrix and  $\mathbf{b}_c$  is the bias vector. Also,  $\tanh(u)$  is applied to the elements of the vector, whose values lie between  $-1$  and  $1$ .

The final state  $\mathbf{C}_t$  of the memory cell is generated as follows using the information  $\mathbf{f}_t$  to control how much the cell state  $\mathbf{C}_{t-1}$  of the previous time step is kept and the information  $\mathbf{i}_t$  to control how much new cell state is used.

$$\mathbf{C}_t = \mathbf{f}_t * \mathbf{C}_{t-1} + \mathbf{i}_t * \tilde{\mathbf{C}}_t \quad (5)$$

where symbol  $*$  means the multiplication of two vectors by elements.

If  $\mathbf{f}_t$  is always about 1 and  $\mathbf{i}_t$  is always about 0, then  $\mathbf{C}_{t-1}$  will be saved over time. This action captures the long-term dependence of time series information and also helps to resolve the gradient vanishing and explosion problems.

The hidden state vector  $\mathbf{h}_t$  at time step  $t$  is generated by using the information  $\mathbf{o}_t$  to estimate which part of the cell state  $\mathbf{C}_t$  must be reflected in that  $\mathbf{h}_t$  as follows:

$$\mathbf{h}_t = \mathbf{o}_t * \tanh(\mathbf{C}_t) \quad (6)$$

where, too,  $\tanh(\cdot)$  is applied to the elements of a vector, whose values lie between  $-1$  and  $1$ . If  $\mathbf{o}_t$  is close to 1, it effectively transfers all the memory information to the predictor, and if  $\mathbf{o}_t$  is close to 0, it keeps all the information only in the memory cell and no longer processes.

All the above weight matrices and bias vectors of LSTM converge to the appropriate values by correcting from certain initial values, according to the error backpropagation learning algorithm and the stochastic gradient descent method, when training is performed.

### An Architecture of LSTM Network for Time Series Prediction

Now, we set up the problem of constructing a prediction model for the physical quantity  $X$ , which takes as input the L-dimensional time series observation vector  $\mathbf{x}_t^{in} = (x_t, x_{t-1}, \Lambda, x_{t-(L-1)})$  from time  $t$  to that time, and then outputs the M(<L) dimensional time series vector  $\mathbf{x}_t^{out} = (x_{t+1}, x_{t+2}, \Lambda, x_{t+M})$  after that time point using the LSTM network consisting of input layer, output layer, and K LSTM layers (Figure 2).

Here, each element of the L-dimensional time series observation vector corresponds to each neuron of the input layer, and they are outputted separately to the corresponding LSTM blocks of the LSTM layer that lie behind the input layer.

The LSTM layer is generally multi-layer. This is because the multi-layer approach has better prediction performance than the single-layer approach. The LSTM layer number is usually determined experimentally to suit the characteristics of the problem. All LSTM blocks have H hidden neurons equally, and they work in temporal association, acting as adding or removing information from their memory cell state.

The output layer is connected to the hidden neurons of the LSTM block of the last time step of the last LSTM layer, outputting the prediction information as long as the time series to be output. The number of output layer neurons is equal to the length of the time series to be output. As the activation function of

the output layer neurons,  $\tanh(\cdot)$  is used.

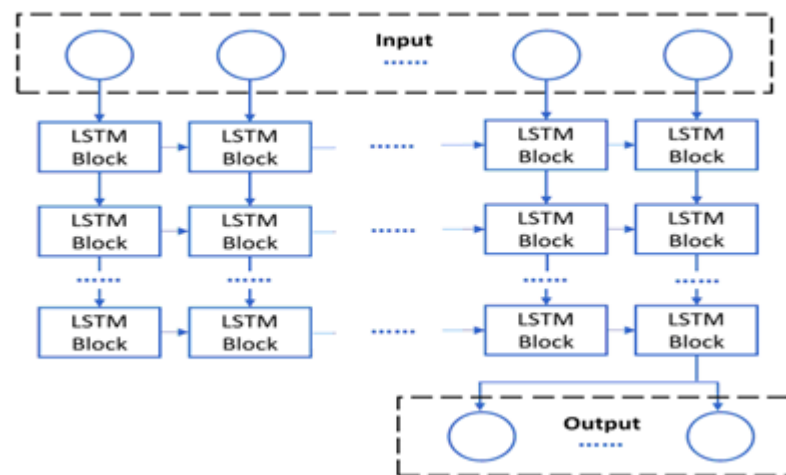


Figure 2. An architecture of LSTM network for the time series prediction.

### A Constructive Method of LSTM Network for Time Series Prediction of Near-Surface Temperature

The input data to the LSTM network for time series prediction of near-surface temperature is the observed time series data of 2 m air temperature. The hyper-parameters for obtaining a good prediction model are the number of LSTM layers and the number of hidden neurons in the LSTM block. For the training of the LSTM network, we use the mean square error (MSE) loss function and Adam optimizer (Kingma & Ba, 2015). The MSE is the mean square of the error between the predicted and the actual measured values for  $N$  samples, defined as follows:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (7)$$

The learning rate is initially set to 0.01, for example, and adaptively reduced by using ReduceLROnPlateau, the learning rate scheduler provided in the PyTorch library (Paszke et al., 2019), according to the progress of training. In other words, if the validation error during training is not improved again until 10 epochs have elapsed since the new improvement, then the learning rate is multiplied by 0.9.

To suppress the effect of especially large or small data value, the standardization of the data is done by subtracting the mean from each data value and dividing it by the standard deviation, and the standardized data is used as input and target of the LSTM network. And for prediction at real scale, we use the value of the inverse standardization of the output of the LSTM network.

### Experimental Tasks and Data

The LSTM neural network for the prediction of the time series of near-surface temperature performs the task of finding the predicted output of the daily minimum temperature (or maximum temperature) for several days immediately after the forecast time, using the time series data of the daily minimum temperature (or maximum temperature) for several days just before the forecast time for the corresponding observation point. The time series prediction model using LSTM network is constructed separately for minimum and maximum temperature prediction. Here, the input sequence length  $L$  and the output sequence length  $M$  can be set considering the requirements of a given problem or the preparation of the sample data.

The experimental task is firstly a performance comparison with the structure determination of LSTM neural network by the proposed method. Here, the accuracy of time series prediction is evaluated by varying the number of LSTM layers and the number of hidden neurons of LSTM blocks, which are the hyperparameters for structural optimization of LSTM neural networks for a given sample data. In addition, the accuracy of time series predictions for the case of placing one fully connected layer behind the last LSTM layer and without placing it are compared.

The experimental task is secondly to evaluate the time series prediction performance of the LSTM neural network by the proposed method. Here, first of all, the performance evaluation of the prediction model is carried out for the case where the input sequence length  $L$  and output sequence length  $M$  are set

to 8 and 4, respectively. In this case, performance comparison of the proposed method with some other methods is also carried out. In addition, a performance comparison is performed on the sampled data from the capital city Pyongyang, some coastal areas and inland areas to show the regional predictive performance of different meteorological and climatic conditions. In particular, the performance of the time series prediction of the LSTM neural network with the proposed method is evaluated by varying the input sequence length to 40 and output sequence length to 10.

For the experiments, near-surface temperature observations collected from several stations in the capital city Pyongyang, some coastal areas and inland areas with four seasons, spring, summer, autumn and winter, are used. The data are the minimum and maximum temperatures of the daily near-surface temperatures (2 m air temperature) observed at the local meteorological stations in 2019–2023. To ensure that each step of training, verification, and testing is executed on independent data, the data set in the order of the annual day is divided into three subsets in the order of training, validation, and testing data. The length of the sample of the time series data for the experiment is taken as the sum of the input sequence length and the output sequence length. The sample data for the experiment is constructed so that there is no gap among the subsequent observations. In practice, if one observation is blank, it can be treated by filling the average of the observations adjacent to that blank. With respect to anomalies in the data, the possibility of random inclusion of errors in the observed data cannot be excluded, but, even if the data are used as a raw, if standardization of data is performed, such as subtracting the mean from each data value and dividing it by the standard deviation, the effect of anomalies, i.e., the effect of a particular large or small data value, can be suppressed.

### Evaluation Measures

As an evaluation measure for the performance of time series predictions, the commonly used mean absolute error (MAE), root mean square error (RMSE), and accuracy (ACC) are applied.

We now denote the predicted and observed values for the  $i$  sample by  $\hat{y}_i$  and  $y_i$ , respectively. Then, MAE is defined as the average of the absolute errors between the predicted and observed values for  $N$  samples, which is defined as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (8)$$

RMSE is defined as the square root of the mean square of the errors between the predicted and observed values for  $N$  samples as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (9)$$

This metric is more sensitive to large prediction errors (Cifuentes et al., 2020).

ACC is the percentage of the number of samples that the error between the predicted and observed values does not exceed the tolerance threshold  $\theta$  divided by the total number of samples  $N$ , which is defined as follows:

$$ACC = \frac{|I|}{N} \times 100 (\%), \quad I = \{i \mid |y_i - \hat{y}_i| \leq \theta\} \quad (10)$$

where  $\theta$  is taken as 2 °C as in Kong et al. (2022).

As an evaluation measure for time series prediction performance, we also apply  $ACC_i$  defined as follows.

$$ACC_i = \frac{|I|}{N} \times 100 (\%), \quad I = \{i \mid |Y_i - \hat{Y}_i| \leq \theta\} \quad (11)$$

where  $\hat{Y}_i$  and  $Y_i$  are the rounded quantities of  $\hat{y}_i$  and  $y_i$  to integers, respectively, and  $\theta$  is taken as 2 °C as in Eq. (10).

### Execution Environment

The construction and performance evaluation experiments of the time series prediction model were carried out using the programming language Python and the library PyTorch 2.0.1 used for deep learning.

## RESULTS & DISCUSSION

### Experimental Results and Discussion for Structural Optimization of LSTM Network

In general, the structural optimization problem of neural networks is very difficult to solve theoretically, as the history of the study of artificial neural networks shows. Therefore, we have performed the following experiments for structural optimization of LSTM neural network. We set the input sequence length  $L$  and output sequence length  $M$  to 8 and 4, respectively, and prepared 22,386 time series of daily minimum and maximum temperatures of near-surface temperatures in the order of 12 days of the annual order from 2019 to 2023 observed at 10 stations in Pyongyang area as the experimental sample data. Here, we first used 75% (16,789) of the data in the front of the data as training data, followed by 10% (2,239) as validation data, and the last 15% (3,358) as testing data.

To optimize the structure of the LSTM network for the time series prediction of near-surface temperature, we first experimentally determined the number of LSTM layers and the number of hidden neurons of the LSTM block as the hyperparameters of the LSTM network. Considering the computational complexity, we have observed the generalization performance of the LSTM network by varying the two hyperparameters at appropriate intervals. Table 1 shows MAE (4-day average value) for the validation data of the 4-day daily maximum temperature predictions that varied with the number of LSTM layers and the number of hidden neurons in the LSTM block. In Table 1, the lower MAE compared to the number of LSTM layers or the number of hidden neurons in the LSTM block are indicated in bold and underlined.

Table 1. MAE (4-day average value) for the validation data of the 4-day daily maximum temperature predictions according to the number of LSTM layers and the number of hidden neurons in LSTM block (°C)

| Number of LSTM layers | Number of hidden neurons in LSTM block |      |             |      |             |             |      |             |      |      |      |
|-----------------------|----------------------------------------|------|-------------|------|-------------|-------------|------|-------------|------|------|------|
|                       | 100                                    | 110  | 120         | 130  | 140         | 150         | 160  | 170         | 180  | 190  | 200  |
| 1                     | 1.59                                   | 1.68 | 1.60        | 1.60 | 1.55        | 1.52        | 1.56 | 1.53        | 1.49 | 1.48 | 1.51 |
| 2                     | 1.41                                   | 1.35 | 1.40        | 1.40 | 1.30        | 1.35        | 1.40 | 1.35        | 1.34 | 1.37 | 1.32 |
| 3                     | 1.30                                   | 1.31 | 1.31        | 1.25 | 1.25        | 1.27        | 1.25 | 1.23        | 1.24 | 1.30 | 1.22 |
| 4                     | 1.29                                   | 1.24 | 1.24        | 1.26 | 1.19        | 1.19        | 1.20 | <b>1.15</b> | 1.19 | 1.17 | 1.18 |
| 5                     | 1.22                                   | 1.20 | 1.17        | 1.16 | <b>1.15</b> | <b>1.12</b> | 1.15 | <b>1.14</b> | 1.19 | 1.18 | 1.20 |
| 6                     | 1.23                                   | 1.17 | <b>1.11</b> | 1.14 | 1.16        | 1.23        | 1.25 | 1.26        | 1.19 | 1.18 | 1.17 |
| 7                     | 1.17                                   | 1.19 | 1.19        | 1.20 | 1.21        | 1.22        | 1.21 | 1.20        | 1.19 | 1.19 | 1.25 |
| 8                     | 1.21                                   | 1.23 | 1.25        | 1.27 | 1.25        | 1.15        | 1.20 | 1.20        | 1.20 | 1.20 | 1.30 |

From Table 1, it can be seen that MAE for validation data decreases gradually and then increases again with the increase of LSTM layer number under fixed number of hidden neurons in LSTM block, while MAE for validation data is the smallest at 1.11 °C when the LSTM layer number is 6 and the number of hidden neurons in LSTM block is 120.

Next, we evaluated MAE and RMSE for the five cases selected in Table 1 (corresponding to  $1.11\text{ °C} \leq \text{MAE} \leq 1.15\text{ °C}$  in Table 1) on the test data (not used as training data or validation data) of the 4-day daily maximum temperature prediction according to the number of LSTM layers and the number of hidden neurons in the LSTM block, and the results are shown in Table 2. The validation MAE in Table 2 is the MAE (4-day average value) in Table 1.

From Table 2, it can be seen that the MAE for the validation data is 1.11 °C for the number of LSTM layers 6 and the number of hidden neurons 120, and the MAE and RMSE for the test data are 1.56 °C and 2.21 °C, respectively, which show the best prediction performance.

Summarizing the above experimental results, we can characterize the following two trends. One is that the MAE for the validation data tends to decrease gradually and then increase again with the increase of the LSTM layer number under fixed number of hidden neurons in LSTM block. The other is that in the case of small MAE for the verification data, the MAE and RMSE for the test data also tend to be relatively small.

Table 2. MAE and RMSE for the testing data of the 4-day daily maximum temperature predictions according to the number of LSTM layers and the number of hidden neurons in LSTM block (°C)

| Number of LSTM layers | Number of hidden neurons in LSTM block | Validation MAE | Testing MAE |       |       |       |             | Testing RMSE |       |       |       |             |
|-----------------------|----------------------------------------|----------------|-------------|-------|-------|-------|-------------|--------------|-------|-------|-------|-------------|
|                       |                                        |                | 1-day       | 2-day | 3-day | 4-day | mean        | 1-day        | 2-day | 3-day | 4-day | mean        |
| 6                     | 120                                    | 1.11           | 1.42        | 1.53  | 1.59  | 1.69  | <b>1.56</b> | 1.97         | 2.18  | 2.28  | 2.42  | <b>2.21</b> |
| 5                     | 150                                    | 1.12           | 1.48        | 1.61  | 1.69  | 1.73  | 1.63        | 2.04         | 2.36  | 2.42  | 2.49  | 2.33        |
| 5                     | 170                                    | 1.14           | 1.48        | 1.71  | 1.76  | 1.83  | 1.69        | 2.04         | 2.41  | 2.46  | 2.53  | 2.36        |
| 5                     | 140                                    | 1.15           | 1.50        | 1.67  | 1.68  | 1.72  | 1.64        | 2.05         | 2.34  | 2.38  | 2.42  | 2.30        |
| 4                     | 170                                    | 1.15           | 1.52        | 1.67  | 1.75  | 1.80  | 1.69        | 2.07         | 2.39  | 2.48  | 2.56  | 2.38        |

The above experiments for the determination of the hyperparameters are essentially based on a grid search method, but to reduce the search space, we first made the evaluation results macroscopically by changing the hyperparameters at appropriate intervals, then narrowing the range by referring to them and obtaining the evaluation results in more detail. The above experimental results are for the maximum temperature, while the minimum temperature has a similar trend.

To optimize the structure of the LSTM network for the time series prediction of near-surface temperature, we next examined the necessity of placing a fully connected layer between the last LSTM layer and the output layer. To do so, we performed a comparison experiment with and without a fully connected layer behind the last LSTM layer. Table 3 shows the 4-day average of MAE for the test data of 4-day daily maximum temperature prediction for LSTM network with 6 LSTM layers and 120 hidden neurons in LSTM block, when it does not contain a fully connected layer and when it takes different number of neurons including one fully connected layer.

Table 3. Performance comparison results for the presence or absence of a fully connected layer and its neuron number (°C)

| Cases for the fully connected layer |             | Testing MAE | Testing RMSE |
|-------------------------------------|-------------|-------------|--------------|
| no fully connected layer            |             | <b>1.56</b> | <b>2.21</b>  |
| A fully connected layer with        | 10 neurons  | 1.95        | 2.78         |
|                                     | 20 neurons  | 1.89        | 2.69         |
|                                     | 30 neurons  | 1.79        | 2.55         |
|                                     | 40 neurons  | 1.83        | 2.61         |
|                                     | 50 neurons  | 1.89        | 2.68         |
|                                     | 60 neurons  | 1.90        | 2.71         |
|                                     | 80 neurons  | 1.90        | 2.71         |
|                                     | 100 neurons | 1.91        | 2.73         |
|                                     | 120 neurons | 1.92        | 2.76         |

From Table 3, it can be seen that the prediction performance for the test data is significantly higher for the LSTM network with six LSTM layers than for the case with one fully connected layer. This indicates that for an LSTM network with several (e.g., six) LSTM layers, placing a fully connected layer behind the last LSTM layer will degrade the generalization performance of the prediction model by causing overfitting due to the rather excessive nonlinearity.

The above experimental results confirm that if the LSTM network is used as a prediction model, it is necessary to conduct experiments to determine the structure of the LSTM network, given the sample data, because the structural optimization of the LSTM network is difficult to solve theoretically.

### Experimental Results and Discussion on Predictive Performance Evaluation

First, we conduct a performance evaluation of the testing data of the prediction model by the proposed method. As a performance of the prediction model by the proposed method, ACC and ACC<sub>i</sub> corresponding to MAE and RMSE were evaluated on the test data of the 4-day daily maximum temperature prediction shown in Table 2. Table 4 shows the results of the evaluation.

As shown in Table 4, the performance of the prediction model by the proposed method is that the average accuracy for the daily maximum temperature for 4 days is 73.70% for ACC and 80.96% for ACC<sub>i</sub>. It can also be seen that using the criterion ACC<sub>i</sub>, the accuracy of the 1-day, 2-day, 3-day, and 4-day

predictions and their average is about 7 percentage points higher than that using ACC. This means that ACC<sub>i</sub> is better than ACC in practical terms.

Table 4. Performance Evaluation Results on the test data of the 4-day daily maximum temperature prediction

|               | MAE(°C) | RMSE(°C) | ACC (%) | ACC <sub>i</sub> (%) |
|---------------|---------|----------|---------|----------------------|
| 24h (1-day)   | 1.42    | 1.97     | 76.38   | 83.56                |
| 48h (2-day)   | 1.53    | 2.18     | 74.51   | 81.39                |
| 72h (3-day)   | 1.59    | 2.28     | 73.26   | 80.85                |
| 96h (4-day)   | 1.69    | 2.42     | 70.64   | 78.02                |
| average value | 1.56    | 2.21     | 73.70   | 80.96                |

In fact, the use of ACC<sub>i</sub> significantly increases the maximum tolerance range compared to the use of ACC. For example, if the observed temperature is 9.5 °C and the predicted temperature is 12.49 °C, the actual error is 2.99 °C, which far exceeds the tolerance threshold of Eq. (10), but rounding them does not exceed the tolerance threshold of Eq. (11) as 10 °C and 12 °C, respectively. For another example, if the observed temperature is 11.5 °C and the predicted temperature is 9.49 °C, the actual error is 2.01 °C, but the rounded temperature is 12 °C and 9 °C, which exceeds the tolerance threshold of Eq. (11). This illustrates that, according to the case, even if the actual error is 2.99 °C, it may not exceed the tolerance threshold, and even if the actual error is 2.01 °C, it may exceed the tolerance threshold. If, on the contrary, we cannot find an example where the maximum tolerance range is larger than that of ACC<sub>i</sub> using ACC, it is proved that ACC<sub>i</sub> is larger than ACC. In fact, such an example does not exist. The analysis of average for 4 days of ACC and ACC<sub>i</sub> for the testing data of the daily maximum temperature prediction for 4 days is shown in Table 5 and Table 6.

Table 5. Analysis of the average value for 4 days of ACC on the testing data of daily maximum temperature prediction for 4 days

| $ y_i - \hat{y}_i $ (°C) | [0, 0.5] | (0.5, 1] | (1, 1.5] | (1.5, 2] | (2, 2.5] | (2.5, 3] | (3, 3.5] | (3.5, 4] | (4, 4.5] | >4.5 | sum  |
|--------------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|------|------|
| number of samples        | 453      | 595      | 674      | 753      | 311      | 208      | 179      | 76       | 62       | 47   | 3358 |
| percentage (%)           | 13.49    | 17.72    | 20.07    | 22.42    | 9.26     | 6.19     | 5.33     | 2.26     | 1.84     | 1.42 | 100  |

Table 6. Analysis of the average value for 4 days of ACC<sub>i</sub> on the testing data of daily maximum temperature prediction for 4 days

| $ Y_i - \hat{Y}_i $ (°C) | 0    | 1     | 2     | 3     | 4    | ≥5   | sum  |
|--------------------------|------|-------|-------|-------|------|------|------|
| number of samples        | 329  | 832   | 1558  | 341   | 203  | 95   | 3358 |
| percentage (%)           | 9.80 | 24.77 | 46.39 | 10.15 | 6.05 | 2.84 | 100  |

ACC is the percentage of those whose  $|y_i - \hat{y}_i|$  is not greater than 2 °C, which is found to be 73.70% as shown in Table 5. Also, ACC<sub>i</sub> is the percentage of those whose  $|Y_i - \hat{Y}_i|$  is not greater than 2 °C, which is found to be 80.96% as shown in Table 6. The above analysis illustrates the reason why the average accuracy using ACC<sub>i</sub> is significantly higher than that using ACC.

Next, we compare the performance of the prediction models by the proposed method and other methods. As the weather and climate conditions differ from country or region, it is difficult to directly compare the predicted results of near-surface temperatures with those of other countries. In view of this, we have compared the performance of Support Vector Regression (SVR) model, which has been widely used for time series prediction problems, with the model combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) neural network, another method we have proposed. Here, the SVR model consists of four models corresponding to the daily prediction for 4 days. The prediction model by combining CNN and LSTM (CNN + LSTM model) consists of input module, convolution module, long short-term memory module and output module. The input module takes as input the L-dimensional (e.g.,  $L = 8$ ) time-series vector in time steps and outputs it to the convolution module. The convolution module performs spatiotemporal feature extraction for input received in time steps and

outputs the results to the long short-term memory module. The operations performed in the convolution module are convolution, batch normalization, and rectified linear operation. During the convolution operation, zero interpolation (padding) is performed so that the size of the feature map does not change, and the convolution step size is  $1 \times 1$ . The convolution stage is tripled, each time the size of the convolution kernel is  $1 \times 2$  and the number of convolution channels is kept at 16. There is a pooling layer behind the last convolution stage. Here, the maximum pooling operation with a pooling kernel of size  $1 \times 2$  is performed by convolution channels, so that the feature maps reduced to size  $1 \times L/2$  are obtained. Then, by performing a linear combination of each element of these feature maps, an  $L/(2D)$  feature vector is obtained, which is the output of the convolution module. The Long short-term memory module has, for example, 100 hidden neurons and receives the output of the convolution module in time steps to perform a prediction operation on the temporal history. The output module has output neurons equal to the output sequence length, and they are connected to the hidden neurons of the long short-term memory module, finding the  $M$ -dimensional (e.g.,  $M = 4$ ) output vector in time steps and outputting it to the outside. The activation function of output neurons is a logistic function, the squared error function is taken as the loss function, the initial learning rate is set to 0.01, the batch size is set to 32, and the deep learning algorithm uses the commonly used Adam algorithm. The results of the performance comparison with other methods on the testing data for the daily maximum temperature prediction for 4 days are shown in Table 7.

Table 7. Performance comparison with other methods on the testing data for daily maximum temperature prediction for 4 days

|                           | MAE(°C) | RMSE(°C) | ACC (%) | ACC <sub>i</sub> (%) |
|---------------------------|---------|----------|---------|----------------------|
| SVR model                 | 1.89    | 2.95     | 62.63   | 68.59                |
| CNN+LSTM model            | 1.60    | 2.27     | 72.81   | 79.72                |
| Proposed prediction model | 1.56    | 2.21     | 73.70   | 80.96                |

The values in Table 7 are the average values of the daily maximum temperature prediction for 4 days. As shown in Table 7, the prediction model by proposed method has higher average accuracy than the CNN + LSTM model as well as the SVR model.

Next, the performance comparison of the prediction model by proposed method is carried out for the Pyongyang, coastal and inland regions. As before, the input sequence length was taken as 8 and the output sequence length as 4. For the performance comparison, the time series of daily minimum and maximum temperatures from 2019 to 2023 were also prepared, observed at 10 stations in the coastal zone and 10 stations in the inland zone with four seasonal characteristics in spring, summer, autumn and winter.

Table 8. Comparison of ACC<sub>i</sub> on the testing data from different areas of daily maximum temperature prediction for 4 days

|               | ACC <sub>i</sub> (%) |             |           |
|---------------|----------------------|-------------|-----------|
|               | coastal area         | inland area | Pyongyang |
| 24h (1-day)   | 83.21                | 79.63       | 83.56     |
| 48h (2-day)   | 82.43                | 78.74       | 81.39     |
| 72h (3-day)   | 81.37                | 77.52       | 80.85     |
| 96h (4-day)   | 78.76                | 75.41       | 78.02     |
| average value | 81.44                | 77.82       | 80.96     |

The amount of training data, verification data and testing data and the preprocessing method were the same as in the preparation of experimental data for Pyongyang. The performance comparison of the proposed prediction model on the testing data in the Pyongyang, coastal area and inland area is shown in Table 8. As shown in Table 8, the performance of the prediction model for the inland area is lower than that of the Pyongyang and coastal areas.

Finally, performance comparison with different time periods of the proposed prediction model is carried out. Here, for the prediction model by the proposed method, we performed a performance comparison experiment with the input sequence length fixed at 8 and the output sequence length varied.

The sample data used here are also data collected from 10 stations in Pyongyang. Here, the total number of training, validation, and testing data was 22,014, and 75% (16,510) of the data that was also placed in the front of the data were used as training data, followed by 10% (2,202) as validation data, and the last 15% (3,302) as testing data. Table 9 shows the comparison of  $ACC_i$  on testing data when the input sequence length is fixed to 8 and the output sequence length is altered to 2, 3, 4, 5, and 6. The experimental results in Table 9 are only those when the LSTM layer number is taken as 5 and the number of hidden neurons in LSTM block is taken as 120.

Table 9. Comparison of  $ACC_i$  with different output sequence lengths on testing data of daily maximum temperature prediction

|              | $ACC_i$ for different output sequence lengths (%) |       |       |       |       |
|--------------|---------------------------------------------------|-------|-------|-------|-------|
|              | 2                                                 | 3     | 4     | 5     | 6     |
| 24h (1-day)  | 82.23                                             | 81.71 | 81.21 | 81.97 | 81.18 |
| 48h (2-day)  | 78.63                                             | 79.39 | 78.20 | 77.08 | 78.91 |
| 72h (3-day)  |                                                   | 77.54 | 78.02 | 76.21 | 76.67 |
| 96h (4-day)  |                                                   |       | 76.06 | 76.96 | 77.77 |
| 120h (5-day) |                                                   |       |       | 75.56 | 76.28 |
| 144h (6-day) |                                                   |       |       |       | 74.31 |

As shown in Table 9, it can be seen that the performance of time series prediction by the proposed method is not very low even with increasing output sequence length while keeping the input sequence length fixed.

For the prediction model by the proposed method, we also performed a performance comparison experiment with the output sequence length fixed at 4 and the input sequence length varied. The sample data used here are also data collected from 10 stations in Pyongyang. Here, the total number of training, validation, and testing data was 16,436, and 75% (12,326) of the data that was also placed in the front of the data were used as training data, followed by 10% (1,644) as validation data, and the last 15% (2,466) as testing data. Table 10 shows the comparison of  $ACC_i$  on testing data when the output sequence length is fixed to 4 and the input sequence length is altered to 5, 10, 20, 30, and 40. The experimental results in Table 10 are also only those when the LSTM layer number is taken as 5 and the number of hidden neurons in LSTM block is taken as 120.

Table 10. Comparison of  $ACC_i$  with different input sequence lengths on testing data of daily maximum temperature prediction

|               | $ACC_i$ for different input sequence lengths (%) |       |       |       |       |
|---------------|--------------------------------------------------|-------|-------|-------|-------|
|               | 5                                                | 10    | 20    | 30    | 40    |
| 24h (1-day)   | 71.32                                            | 85.29 | 90.30 | 92.45 | 89.83 |
| 48h (2-day)   | 65.32                                            | 83.17 | 90.48 | 91.72 | 89.37 |
| 72h (3-day)   | 62.77                                            | 82.70 | 90.66 | 91.78 | 89.24 |
| 96h (4-day)   | 61.46                                            | 81.44 | 89.45 | 91.69 | 88.75 |
| average value | 65.22                                            | 83.15 | 90.22 | 91.91 | 89.30 |

It is seen from Table 10 that the performance of time series prediction is better when the input sequence length is longer ( $ACC_i$  of 88.75–92.45% for input lengths of 20–40 days). For the input sequence length of 30, the  $ACC_i$  of the daily maximum temperature predictions for 4 days is over 91.69% and their average is 91.91%, which is 10.95 percentage points higher than the average daily  $ACC_i$  of 80.96% in Table 4. Based on this trend, we carried out the experiments on daily maximum temperature predictions for 10 days.

Table 11 shows the  $ACC$  and  $ACC_i$  on the testing data of the daily maximum temperature predictions for the input sequence length of 35. The experimental results in Table 11 are only those when the LSTM layer number is 5 and the number of hidden neurons in the LSTM block is 130. Here, the total number of training, validation, and testing data was 16,250, and 75% (12,187) of the data that was also

placed in the front of the data were used as training data, followed by 10% (1,625) as validation data, and the last 15% (2,438) as testing data.

Table 11. ACC and ACC<sub>i</sub> on testing data of 10-day daily maximum temperature prediction with input sequence length 35

|                      | 1-day | 2-day | 3-day | 4-day | 5-day | 6-day | 7-day | 8-day | 9-day | 10-day | average |
|----------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|---------|
| ACC (%)              | 89.88 | 89.23 | 89.26 | 89.11 | 88.43 | 88.00 | 87.73 | 86.62 | 86.53 | 84.87  | 87.97   |
| ACC <sub>i</sub> (%) | 94.19 | 94.06 | 94.00 | 93.73 | 93.69 | 93.42 | 93.17 | 92.43 | 91.85 | 90.90  | 93.14   |

As shown in Table 11, the prediction model by the proposed method has ACC<sub>i</sub> more than 90.90% and ACC more than 84.87%, while the difference between ACC<sub>i</sub> of 1-day prediction and ACC<sub>i</sub> of 10-day prediction is 3.29 percentage points, and ACC<sub>i</sub> is monotonically decreasing with the increasing forecast time. Also, the average prediction accuracy for 10 days was ACC = 87.97%, ACC<sub>i</sub> = 93.14%, which were 14.27 and 12.18 percentage points higher than the average prediction accuracy for 4 days, ACC = 73.70% and ACC<sub>i</sub> = 80.96% shown in Table 4, respectively. This illustrates that the time series prediction model by the proposed method can achieve relatively high and stable prediction performance if the input sequence length is somewhat longer. Therefore, if this prediction model is applied in practice, it is expected that good performance can be achieved by properly setting the input sequence length and output sequence length to suit the given problem and sample data preparation.

The experimental results shown so far are for daily maximum temperature, and the daily minimum temperature has similar characteristics.

From the above results and discussion, it can be seen that the time series prediction model by the proposed method can achieve relatively high and stable prediction performance using only the observed data as the sample data.

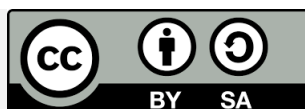
## CONCLUSION

In this paper, we propose a method to construct a time series prediction model for the prediction of air temperature or soil temperature, which is an important issue for understanding the changes of the Earth and its surrounding environment, using the LSTM neural network. The main points of this discussion are as follows. The LSTM network for time series prediction consists of an input layer, an output layer, and several LSTM layers, which, in accordance with the requirements of a given problem, takes the number of neurons in the input layer and output layer as the input sequence length and output sequence length, respectively, and the number of LSTM layers and the number of hidden neurons in LSTM blocks are difficult to obtain theoretically, and it is necessary to determine from experiments given the sample data, and if several LSTM layers are used, placing a fully connected layer between the last LSTM layer and the output layer results in a rather poor prediction performance. In this study, several experiments were carried out to evaluate the performance of the proposed prediction model, considering the practical application of time series prediction for air temperature prediction, with the task of predicting daily minimum or maximum temperatures for several days, and setting the multi-step prediction to be more difficult than one-step prediction. The experimental results showed that the time series prediction model by the proposed method can achieve relatively high prediction performance even if only the temperature observation time series data are used as input without spatiotemporal data such as numerical forecast data, especially if the input sequence length is somewhat longer, it achieved high and stable prediction performance (ACC<sub>i</sub> ≥ 90.90%) even for an output sequence length of 10. The significance of this study is that it can be effectively solved by a relatively simple method, such as the prediction of air temperature or soil temperature. In this paper, due to the difficulty of data preparation, we did not analyze the effects of additional factors, humidity or wind speed, and extreme weather events, heat waves or cold waves, which are the subject of further research.

## REFERENCES

- Alerskans, E., Nyborg, J., Birk, M., & Kaas, E. (2022). A transformer neural network for predicting near-surface temperature. *Meteorological Applications*, 29(5), Article e2098. <https://doi.org/10.1002/met.2098>
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166. <https://doi.org/10.1109/72.279181>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). Association for Computing

- Machinery. <https://doi.org/10.1145/2939672.2939785>
- Chevalier, R. F., Hoogenboom, G., McClendon, R. W., & Paz, J. A. (2011). Support vector regression with reduced training sets for air temperature prediction: A comparison with artificial neural networks. *Neural Computing and Applications*, 20(1), 151–159. <https://doi.org/10.1007/s00521-010-0363-y>
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734). Association for Computational Linguistics. <https://doi.org/10.3115/v1/d14-1179>
- Cifuentes, J., Marulanda, G., Bello, A., & Reneses, J. (2020). Air temperature forecasting using machine learning techniques: A review. *Energies*, 13(16), Article 4215. <https://doi.org/10.3390/en13164215>
- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179–211. [https://doi.org/10.1207/s15516709cog1402\\_1](https://doi.org/10.1207/s15516709cog1402_1)
- Gers, F. A., & Schmidhuber, J. (2000). Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000)* (Vol. 3, pp. 189–194). IEEE. <https://doi.org/10.1109/ijcnn.2000.861302>
- Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009). A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5), 855–868. <https://doi.org/10.1109/tpami.2008.137>
- Graves, A., & Schmidhuber, J. (2005). Frameworkwise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5–6), 602–610. <https://doi.org/10.1016/j.neunet.2005.06.042>
- Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1), 388–427. <https://doi.org/10.1016/j.ijforecast.2020.06.008>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Karevan, Z., & Suykens, J. A. K. (2020). Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Networks*, 125, 1–9. <https://doi.org/10.1016/j.neunet.2019.12.030>
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. <https://doi.org/10.48550/arXiv.1412.6980>
- Kong, W., Li, H., Yu, C., Xia, J., Kang, Y., & Zhang, P. (2022). A deep spatio-temporal forecasting model for multi-site weather prediction post-processing. *Communications in Computational Physics*, 31(1), 131–153. <https://doi.org/10.4208/cicp.aa-2020-0158>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Penghui, L., Ewees, A. A., Beyaztas, B. H., Qi, C., Salih, S. Q., Al-Ansari, N., Bhagat, S. K., Yaseen, Z. M., & Singh, V. P. (2020). Metaheuristic optimization algorithms hybridized with artificial intelligence model for soil temperature prediction: Novel model. *IEEE Access*, 8, 51884–51904. <https://doi.org/10.1109/access.2020.2979822>
- Schmidhuber, J., Wierstra, D., Gagliolo, M., & Gomez, F. (2007). Training recurrent networks by Evolino. *Neural Computation*, 19(3), 757–779. <https://doi.org/10.1162/neco.2007.19.3.757>
- Smith, B. A., McClendon, R. W., & Hoogenboom, G. (2006). Improving air temperature prediction with artificial neural networks. *International Journal of Computational Intelligence*, 3, 179–186.
- Torres, J. F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., & Troncoso, A. (2021). Deep learning for time series forecasting: A survey. *Big Data*, 9(1), 3–21. <https://doi.org/10.1089/big.2020.0159>
- Yu, M., Xu, F., Hu, W., Sun, J., & Cervone, G. (2021). Using long short-term memory (LSTM) and Internet of Things (IoT) for localized surface temperature forecasting in an urban environment. *IEEE Access*, 9, 137406–137418. <https://doi.org/10.1109/access.2021.3116809>



Copyright (c) 2025 by Sin-Gil Kim and Un-Song Ryu. This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).