A Series of Models based on Long Short Time Memory for Temperature Prediction

Tingxi Chen *

School of Electronic Information and Electronical Engineering, Shanghai Jiao Tong University, Shanghai, China

* Corresponding author email: ctx.ximi17@sjtu.edu.cn

Abstract. Given the increasingly serious climate problems that have been appearing around the world in recent years, as part of the climate research, finding ways of smart, accurate machine learning to carry out the temperature prediction is of great significance both for human activities and the earth's ecosystem. In this study, 3 kinds of the Long Short Time Memory (LSTM) models, which differs on the way how the data is input and output and the network structure, respectively Directly-Multiple-Output (DMO), Single-Step-Scrolling (SSS), and Convolutional Neural Networks Plus Long Short Time Memory (CNN+LSTM) are built and trained by Pytorch on the Jena Climate dataset to compare their predicted performance. The training loss of these 3 models are 0.0053, 0.1568 and 0. 0079; testing loss are 0.0048, 0.1764 and 0.0096; Mean Absolute Percentage Error (MAPE) are 3.42%,6.58% and 3.30%. The result turns out that CNN+LSTM is the best model in comprehensive consideration with the least convergence time, low loss and MAPE, followed by the DMO model with the least loss and low MAPE but longer convergence time. SSS performs worst with increasing high loss. In general, CNN+LSTM is suitable for temperature prediction while DMO is also good when short convergence time is not needed. SSS is not recommended for temperature prediction.

Keywords: Temperature Prediction; LSTM; CNN; SSS; DMO.

1. Introduction

Temperature prediction refers to the use of excessive period of temperature and some environmental factors data to predict temperature data in the future period through specific methods e.g., empirical knowledge statistical knowledge and computer algorithms. Climate change, particularly temperature change, affects health in many ways, including the increasing frequency of extreme weather events (such as heat waves, storms and floods), disruption of food systems, zoonosis and foodborne diseases, and dramatic impacts on human life and production. However, the use of empirical knowledge and statistical knowledge to predict temperature has the problem of lack of universality and inapplicability to large-scale data, so it is very necessary to find optimized computer algorithms for temperature prediction.

As early as ancient times, people predicted temperature by observing weather phenomena, animal behavior, etc. through experience. With the development of statistics, people can use statistical mathematical methods to build simple models like linear model to predict temperature, which was used in [1]. In the research [2], an analog approach was used to do monthly and seasonal predictions but no use of machine learning algorithm. In recent years, various machine learning models have also been used for temperature prediction, which has the advantages of being smarter, more accurate and less laborious than previous methods. Traditional models like ARIMA, Artificial Neural Network (ANN), Backpropagation (BP) have already been used. For instance, C. Narendra Babu et al. used variants of ARIMA models to analyze and predict the Average Global Temperature time series data [3]. In another article [4], ANN was used to predict dewpoint temperature from 1 to 12 h ahead using prior weather data. Some new model like Long Short Time Memory (LSTM) have also been tried. Jia X et al. simply used basic LSTM model to predict the sea surface temperature [5]. Stajkowski S et al. combined LSTM with Genetic algorithm to improve performance. However, instead of simple math method or simple traditional network, this study would like to use an advanced recurrent network like LSTM which can remember the important information from the past and ignore the unimportant ones to make predictions on time series temperature data, explore the internal

mechanisms of its data input and output to find whether the way of inputting the data will influence the result. More significantly, try to combine it with a feature extraction Convolution Neural Network, in order to optimize performance.

Therefore, this study used basic LSTM model built by Pytorch for improvement and optimization, and perform temperature prediction of certain hours for the 2009-2016 Jena temperature dataset. This study tried to chart and observe the characteristic relationship between 14 variables, find the periodic regularity over months and hours, select some representative characteristics and add some variables to represent the periodic regularity, to predict the air temperature of the next 2 hours using the data from the past 24 hours. The final result is that CNN+LSTM is the best model, which is accurate with small losses becomes stable with the least training epochs. And he single-step scrolling model performs the worst, which is not recommended to be used in prediction.

2. Method

2.1 Dataset Description and Preprocessing

Jena Climate is a data set of weather time series recorded by the weather station at the Max Planck Institute for BioGeochemistry in Jena, Germany. The dataset consists of 14 separate measurements from across several years that were taken every 10 minutes, including air temperature, atmospheric pressure, humidity, and wind direction. This dataset covers data from January 1st 2009 to December 31st 2016 with 420,451 data [6].

Consider that the temperature always has periodic regularity over months in the year and over hours in a day, this study maps and rules for temperature trends within 24 hours a day of each month in the Figure 1 below:

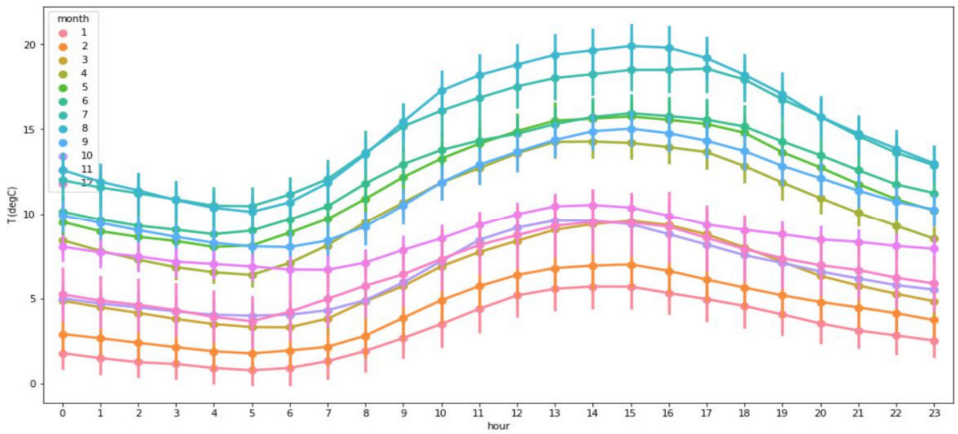

Fig 1. Temperature trends within 24 hours a day of each month

Since the daily temperature is obviously having the same changing trend within 24 hours and fixed relative highs and lows exist between different months of temperature, relevant variables month, cos (h) and sin (h) are added into the dataset as 3 of the input influencing factors. Trigonometric function of hours is chosen in order to guarantee the same regularity repeats itself every 24 hours.

This study finally uses the data from 2014 to 2016, selecting 'T (degC)', 'p (mbar)', 'rh (%)', 'H2OC (mmol/mol)' from 14 original quantities and adding 3 time-related factors, mentioned below, totally 157,824 pieces of data with 7 quantities, to do the prediction. Since the value distribution of different characteristic factors is different, and some factors have both positive and negative values, standardizing the data is considered. The data were divided into three parts: training set, validation set, and testing set, with a ratio of 6:2:2.

2.2 Machine Learning Models

2.2.1 LSTM

LSTM is a special recurrent neural network that has a more complex internal processing structure and input and output than ordinary recurrent neural networks. As shown in the Figure 2, X_t,X_{t-1},X_{t+1} represent the input of the current moment, previous moment, and next moment, respectively, and h_t,h_{t-1},h_{t+1} represent the hidden state of the current moment, previous moment, and next moment, respectively. The final result integrates all h outputs of the previous layer into the target variable through the fully connected layer. In addition to processing the input at this moment and the output of the previous moment, each structural unit of the output LSTM also processes and transmits some hidden states retained at the previous moment, which is called the unit state. The current unit state is denoted as C_t, which summarizes all the key information before the next moment arrives.

$$C_t=f_t\cdot C_{t-1}+\widetilde{C}_t\cdot i_t \tag{1}$$

For the output h_{t-1} at the previous moment, LSTM passes it through four function gates with three processing:

1) Forget gate: Each time a new input is input, the LSTM will first integrate into a separate vector according to the new input and the output of the previous moment, and then act on the unit state through the sigmoid neural layer. If a component of the integrated vector becomes 0 after passing through the sigmoid layer, the component information is "forgotten".

$$f_t=\sigma(W_f\cdot[h_{t-1},x_t]+b_f)\in(0,1) \tag{2}$$

2) Select memory gate: Control whether data at the current time is merged into a control cell in the cell state. First, the tanh function layer is used to extract the valid information in the current vector, and then the sigmoid function is used to control how the valid information is stored in the cell state.

$$\widetilde{C}_t=\tanh(W_c\cdot[h_{t-1},x_t]+b_c) \tag{3}$$

$$i_t=\sigma(W_i\cdot[h_{t-1},x_t]+b_i)\in(0,1) \tag{4}$$

3) Output gate: After integrating the current input value with the output value of the previous moment, use the sigmoid function to extract the information in it, and compress and map the current unit state through the tanh function.

$$o_t=\sigma(W_o\cdot[h_{t-1},x_t]+b_o)\in(0,1) \tag{5}$$

$$h_t=o_t\cdot\tanh(C_t) \tag{6}$$

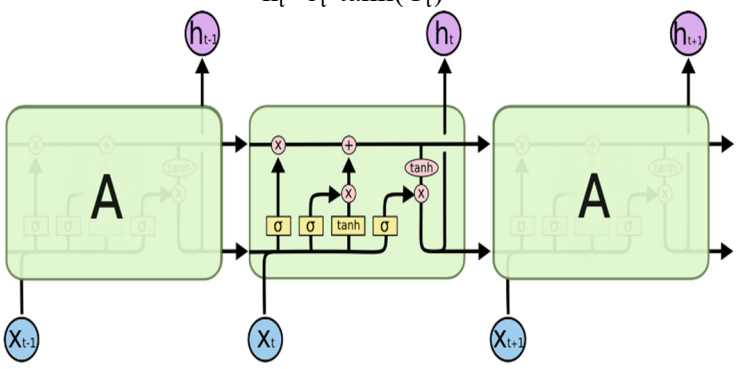

Fig 2. The structure of the LSTM model

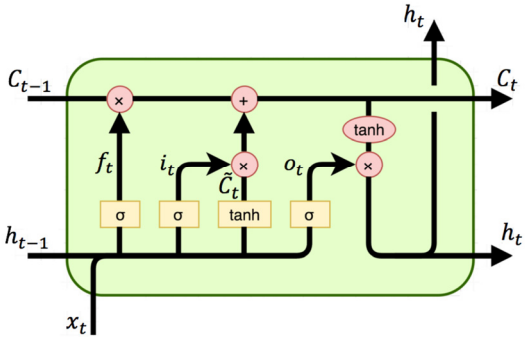

Fig 3. A structural unit of LSTM

In this study, the simplest unidirectional, single-layer LSTM network is used, and all the variable values at each moment before the target moment are arranged into vectors as the input X of each unit, and 64 output h is set for each unit, and finally the target output of the number of targets is output through the fully connected layer. Since the data of the first 24 hours is used to predict the temperature of the next 2 hours, the input vector is 24*6 pieces of 1*7 vectors, and the final output is 2*6 individual values.

2.2.2 CNN+LSTM

Convolutional neural networks are used to extract feature quantities and consist of convolutional, filling, pooling, and fully connected layers. The convolutional layer slides the convolution kernel in turn, multiplies the original data with the corresponding position data of the convolution kernel, and finally sums, and obtains the dimensionality reduction data after sliding all the original data. The pooling layer uses the average or maximum method to find the corresponding value in each area, further reduce the dimensionality, and achieve feature extraction. The fully connected layer combines the outputs of the pooled layer into target output. The fully connected layer combines the outputs of the pooled layer into target output. In this study, CNN + LSTM is used. CNN will firstly extract high-level features from the dataset, then the extracted features will be further passed into the LSTM for time-series prediction.

3. Implementation Details

3.1 LSTM

This study compares 2 ways of multiple output mission based on LSTM [7, 8], the first is Directly-Multiple-Output (DMO) and the next one is Single-Step-Scrolling (SSS). In order to just compare the influence of the way dealing with the input data every time before next prediction, a same basic simple LSTM model is built by Pytorch. There are the following parameters: the input size of the model is the number of expected features in the input x, which in this study is expected to be 7; the hidden size is the number of features in the hidden state h, this study set it to be 32; the number of layers is the number of recurrent layers, which is simply set to be 1; the sequence length means how many groups of previous data have been used to do the prediction, which in this study is 24*6.

The directly-multiple-output way refers to predicting the desired output size predicted value directly with the provided sequence length data, that is, directly passing the input data through a linear layer, turning a single output into multiple outputs. The advantage of this method is that it is relatively simple. One can change the number of data output by the full connection layer to the number of desired predicted values.

The single-step-scrolling prediction refers to the direct prediction of many times in order to obtain multiple prediction outputs, and at each turn, the prediction value of the previous one replaces the original corresponding true value as the input for the next prediction. Each predicted value is obtained and brought in to the next training session is called a scroll.

As for the other hyperparameters setting, the training epochs is 15, the batch size for the first way is 64, for the second way is 128, Adam is chosen for the optimizer, the regularization parameters are set to be 0.0001; the learning rate for the first way is 0.01, for the second way is 0.008. And every 10 epochs let the learning rate become 0.1 times of the original; mean squared error is used for the loss function.

3.2 CNN+LSTM

The combined models are also built by Pytorch. For the CNN part, a one-dimensional convolutional network with 7 input channels ,32 output channels and kernel size set to be 3 is built before the linear rectification activation function, which is then followed by the maximum pooling with kernel size is 3 and stride is 1. The output of CNN is sent to basic LSTM part which has the same structure as the model in part 3.1 [9, 10].

As for the other hyperparameters setting, the training epochs is 15, the batch size is 64, Adam is chosen for the optimizer, the regularization parameters are set to be 0.0001; the learning rate is 0.01. And every 10 epochs let the learning rate become half of the original; mean squared error is used for the loss function.

4. Result and Discussion

After training, this study got the loss curve of 3 different models. As the training epochs increase, the trend of loss of training and the loss of validation are showed in the following Figure 4, Figure 5 and Figure 6.

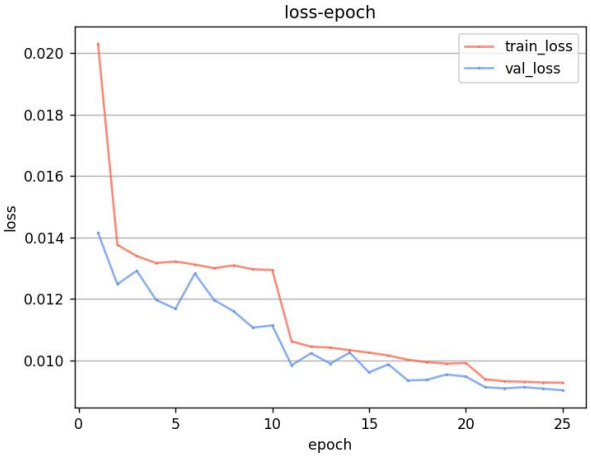

Fig 4. Loss function of DMO

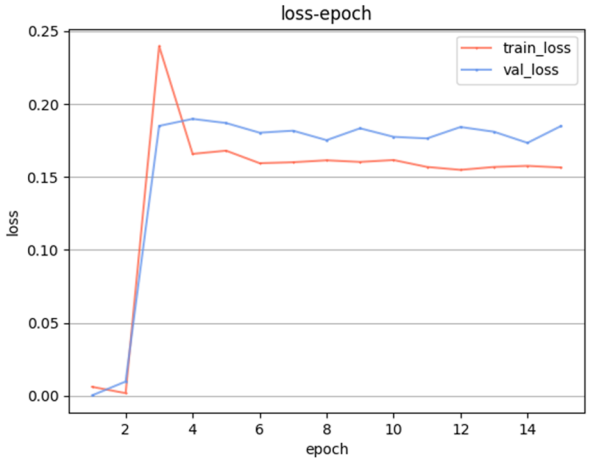

Fig 5. Loss function of SSS

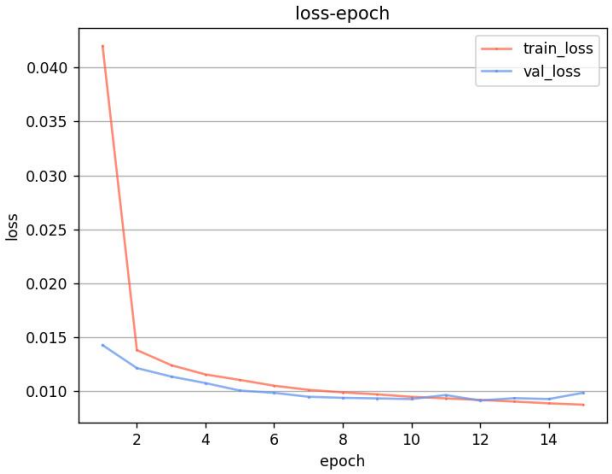

Fig 6. Loss function of CNN+LSTM

As for the loss trend, the DMO model and the CNN+LSTM have the ideal decrease trend with the epochs increase, while the SSS model has a bad performance on the loss trend that the loss increases and is much higher than the starting value. To achieve convergence, DMO takes approximately 25 epochs while the CNN+LSTM takes approximately 15 epochs, which is much less.

Take a further look into the specific value of final loss and the mean absolute percentage error (MAPE) of the 3 models, which is showed in the following Table 1:

Table 1. Loss and MAPE performance

Models	Training loss	Testing loss	MAPE
DMO	0.0053	0.0048	3.42%
SSS	0.1568	0.1764	6.58%
CNN+LSTM	0. 0079	0.0096	3.30%

DMO has the least training loss and validation loss, followed by the CNN+LSTM model, and the SSS has the highest loss.

Then compare the true value and the predicted value of the well-performed 2 models, DMO and CNN+LSTM models, which are showed in the following Figure 7 and Figure 8:

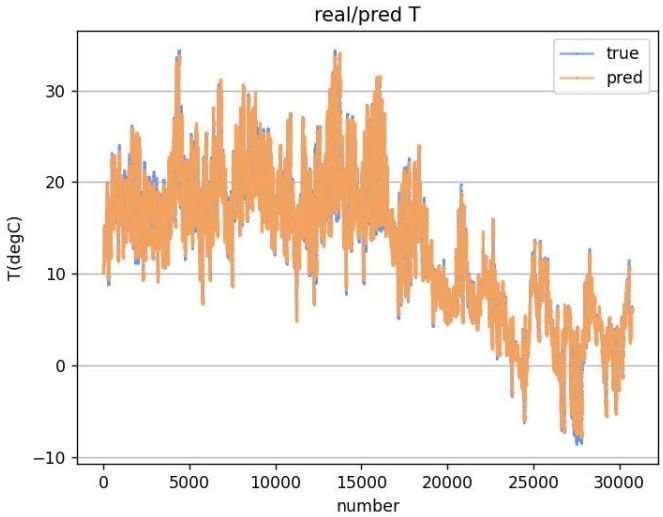

Fig 7. Predictions of DMO

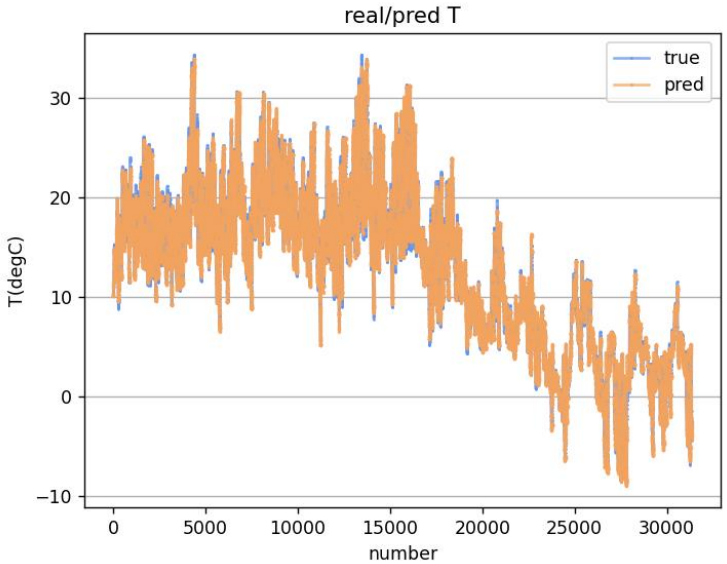

Fig 8. Predictions of CNN+LSTM

To see more clearly, focus on part of the last few predictions of these 2 models shown in Figure 9 and Figure 10.

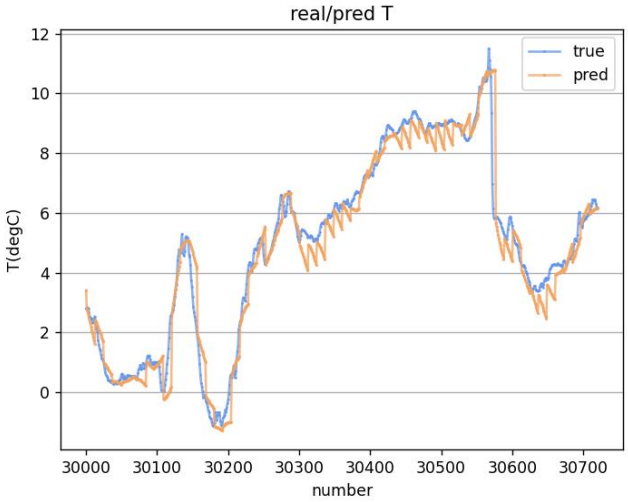

Fig 9. Predictions after number 30000 of DMO

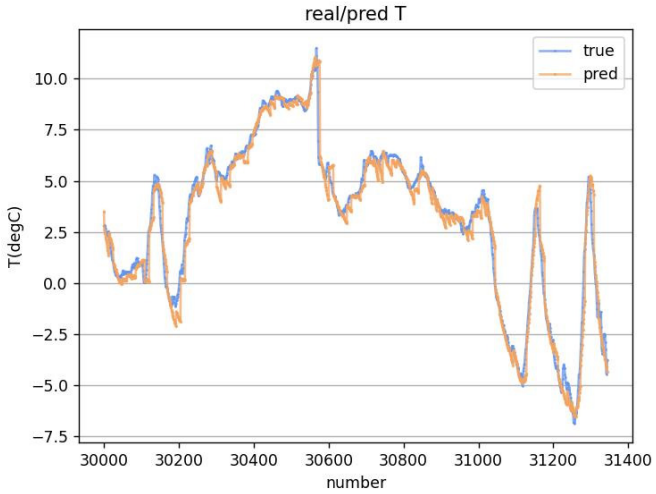

Fig 10. Predictions after number 30000 of CNN+LSTM

As for the predicted performance, both DMO and CNN+LSTM can successfully predict the temperature within acceptable levels.

Based on the intrinsic mechanisms of each model, all can finish prediction task smoothly. The DMO simply represents the performance of the original LSTM, which has a good performance. However, if a CNN is set before the LSTM, since the CNN has a strong ability to abstract useful features that are then directly sent to the LSTM, the LSTM can quickly do processing with these features rather than the huge, cluttered original data. Also, the DMO model may not consider the correlation between the time series, but directly turn it into a nonlinear conversion problem. As a result, the CNN+LSTM model can achieve convergence with less time. When it comes to the SSS model, because the input used in each prediction is the predicted value of the first several variables rather than the real value, once there is an error, the error will accumulate with the number of training times, resulting in the error becoming larger and larger, finally performs the worst. Comprehensively considered, the CNN+LSTM is the best.

5. Conclusion

In order to do the temperature predictions and compare the predicted performance of 3 different models of LSTM, the study first chooses the last 3 years of data from 2009-2016 Jena temperature dataset and processes them by ways like adding time-related variables and carrying out the normalization. Then the study trains the 3 models, namely DMO, SSS and CNN+LSTM, in basically the same way. This study compares the train loss, the validation loss, the time needed to achieve convergence and the MAPE of these 3 models. To conclude, this study comprehensively considers both the time cost and the predicted performance, CNN+LSTM is highly recommended for temperature prediction with the obviously least convergence time and low loss and MAPE. DMO is also recommended for its least loss if there are no high requirements for convergence time. SSS is not suitable for the temperature prediction in this case since it can cause a huge error. In summary, the CNN-LSTM model performs the best, followed by the direct multi-output model, and the single-step rolling model is the worst. Comparing more ways of data processing when LSTM inputs and outputs the data, and trying other network combinations are the interests of future study. Applying these models on various dataset to get more general laws is also considered to be done.

References

[1] S. P. Menon, et al. Prediction of temperature using linear regression, 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2017, pp. 1-6, Doi: 10.1109/ICEECCOT.2017.8284588.

[2] Bergen, R. E, et al. Long-Range Temperature Prediction Using a Simple Analog Approach, Monthly Weather Review, 110(8), 1982, 1083-1099.

[3] C. Narendra Babu et al. Predictive data mining on Average Global Temperature using variants of ARIMA models, IEEE-International Conference on Advances in Engineering, Science and Management (ICAESM -2012), 2012, pp. 256-260.

[4] Shank, Daniel B., G. Hoogenboom, and R. W. McClendon. Dewpoint temperature prediction using artificial neural networks. Journal of applied meteorology and climatology 47.6, 2008, 1757-1769.

[5] Jia X et al. Prediction of Sea Surface Temperature in the East China Sea Based on LSTM Neural Network. Remote Sensing, 2022, 14(14):3300.

[6] Kaggle. Jena-climate. 2020. https://www.kaggle.com/datasets/mnassrib/jena-climate.

[7] Sundermeyer, M, R Schlüter, and H. Ney. LSTM Neural Networks for Language Modeling. Interspeech 2012.

[8] Gf, A., J. Schmidhuber, and F. Cummins. Learning to Forget: Continual Prediction with LSTM. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, 1999.

[9] Shi, X., et al. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. MIT Press, 2015.

[10] Jin, W., et al. Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model. Meeting of the Association for Computational Linguistics 2016.