

## Research Article

# Temperature Forecasting via Convolutional Recurrent Neural Networks Based on Time-Series Data

Zao Zhang <sup>1</sup> and Yuan Dong<sup>2</sup>

<sup>1</sup>Beijing Lucky Dragon Honest and Reputation Technology Company, Beijing, China

<sup>2</sup>School of Electrical and Information Engineering, The University of Sydney, Sydney, Australia

Correspondence should be addressed to Zao Zhang; zhangzao1003@163.com

Received 6 January 2020; Accepted 22 February 2020; Published 20 March 2020

Guest Editor: Yuan Yuan

Copyright © 2020 Zao Zhang and Yuan Dong. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Today, artificial intelligence and deep neural networks have been successfully used in many applications that have fundamentally changed people's lives in many areas. However, very limited research has been done in the meteorology area, where meteorological forecasts still rely on simulations via extensive computing resources. In this paper, we propose an approach to using the neural network to forecast the future temperature according to the past temperature values. Specifically, we design a convolutional recurrent neural network (CRNN) model that is composed of convolution neural network (CNN) portion and recurrent neural network (RNN) portion. The model can learn the time correlation and space correlation of temperature changes from historical data through neural networks. To evaluate the proposed CRNN model, we use the daily temperature data of mainland China from 1952 to 2018 as training data. The results show that our model can predict future temperature with an error around 0.907°C.

## 1. Introduction

With the rapid development of artificial intelligence in recent years, people have gained great convenience in their daily life. Image recognition, speech translation, smart recommendation, self-driving cars, and many more neural network technologies have achieved great success in their applications. However, there are still many applications that can bring great benefits to people lacking of corresponding artificial intelligence models. The meteorological forecasting application is an example that we are going to investigate in this paper.

A more accurate temperature forecasting is important in many aspects of the society. For most people, the predicted temperature helps them choose how to dress. So, in many other industries and sectors, temperature forecasting plays a key role to help people in their work. However, the current forecasting method is still based on meteorological simulations that require huge computation resources and a long time to get the accurate results.

To predict future temperature, this paper develops a new convolutional recurrent neural network (CRNN) model [1, 2], which can effectively forecast the future temperature according to the time series of the temperature data. The CRNN model developed in this paper is a multilevel neural network consisting of a convolutional neural network (CNN) portion and a recurrent neural network (RNN) portion. The CNN portion is used to process the spatial correlation in each temperature data map, and the RNN portion is used to process the time correlation in the consequent temperature data map. Through the above structure, our model can learn the time and space correlation according to past temperature data, and one dense layer is added to generate the predicted temperature values. The training data we used are the daily average temperature data from the China Meteorological Administration. The data include daily average temperature observed from about 800 temperature stations in the mainland of China from 1952 to 2018. Our experiments show that our model can successfully

predict the future temperature, and the average error is about 1.25°C.

The contribution of this paper is that we developed a reliable temperature forecasting deep learning model. Through the model, we can forecast the future temperature according to the past temperature values. Compared to traditional meteorological temperature prediction methods, our model can be used in different geographical environments and is especially useful in those environments where people are not fully aware of their meteorological models. This is because our model can learn the time and space correlation by itself according to the historical data. Therefore, our model can help people get the meteorological model of a geographical environment more easily in addition to conducting the temperature forecasting. This is a reinforcement learning process where the newly learned meteorological model will help improving the CRNN model to obtain better forecasting result.

The rest of this paper is organized as follows. In Section 2, a brief review of related work will be given, including existing temperature forecasting methods and introduction of CRNN. Then, our CRNN structure will be described in Section 3. The procedure of experiments will be shown in Section 4. In Section 5, the results of our experiments and evaluation will be given. Finally, a conclusion of our work and a discussion about some possible future research directions will be given.

## 2. Related Work

*2.1. Temperature Forecasting.* Temperature forecasting is a portion of weather forecasting; other portions include the probability of precipitation forecasting, barometric pressure forecasting, wind power forecasting, etc. One point needs to be noted; temperature forecasting models need to be adapted to different applicable environments, for example, some models are used to forecasting indoor temperature [3, 4], some models are used for large-scale temperature forecasting [5, 6], and some models are used in specific environment [7, 8]. With the rapid development of machine learning, more and more machine learning methods have been applied to weather forecasting, such as support vector machine (SVM) [9, 10], genetic algorithms [11], and neural networks [12–14]. Different methods have their own more suitable application environments.

In large-scale temperature forecasting area, there are some widely used deep learning approaches, such as operational consensus forecasts (OCFs) [15], back-propagation neural networks (BPNNs) [16], and stacked denoising autoencoders (SDAEs) [5]. Compared to original neural networks (NNs), these approaches all achieved better performance. However, the above approaches still have some weaknesses. OCF uses multiple models and integrates them for forecasting. But this method relies on critical manual selection. Original BPNN has also achieved a good result, but it leads to a high computation complexity. SDAE introduces an unsupervised pretraining architecture to initialize model weights, and it improves performance successfully [17]. However, this method

improves the risk of learning the identity function, which may lead training to useless.

In this paper, our model is used to forecast the large-scale temperature of the mainland of China, and our model will more concentrate on the spatial correlation and time correlation of temperature, so our model is also established according to those demands. The detailed introduction of our model is given in Section 3. And the forecasting result shown in Section 5 can prove our model works well in large-scale temperature forecasting area.

### *2.2. Convolutional Recurrent Neural Networks.*

Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are two widely used neural network structures. CNNs are the special neural network architectures that are especially suitable for processing two-dimensional data. Convolutional neural network architectures are usually built with the following layers: convolution layer, activation function layer, pooling layer, fully connected layer, and loss layer [18]. RNNs are developed specifically for processing sequential data with correlations among data samples. They have the nice capability of processing sequential data and can be designed to model both long- and short-term data correlations. By combining the CNN and RNN, the CRNN not only utilizes the representation power of CNN but also employs the context modeling ability of RNN. The CNN layers can learn good middle-level features and help the RNN layer to learn effective spatial dependencies between image region features. Meanwhile, the context information encoded by RNN can lead to better image representation and transmit more accurate supervisions to CNN layers during back-propagation (BP) [19].

In a single two-dimensional data, the distribution of features always relies on each other, and CRNN can work very well in this task. Because CNN can extract the embedded features and process its space correlation and RNN can process their time correlation, CRNN has been used in single-image distribution learning tasks [19]. Another task, i.e., learning the spatial dependency of the image, is more complicated. For example, if images are highly occluded, the recovery of the original image including the occluded portion is very difficult. Some researchers are still working in this area. But if the occluded images are image series with some inherent context information, this problem can be processed with the CRNN model. In the paper [20], the CRNN structure works very well and gets good performance. CRNN structure has also been applied to the text recognition problems, where CNN can be used to recognize a single character while RNN can be used to extract text dependency according to the context. Especially, if the edge feature of the text is strong, then a max-feature-map (MFM) layer can be added into the CRNN model to enhance the contrast [21]. CRNN also shows pretty good performance in music classification tasks, where CNN can be used to extract local feature and RNN can be used to extract temporal summarization of the extracted features [22].

### 3. CRNN Model for Forecasting Future Temperature

**3.1. Introduction of Training Data.** To introduce how our model works, we need to introduce our training data first. The training data are from “surface climate daily value dataset of China.” This dataset is collected by the Nation Meteorological Information Center of China. The training data include daily average temperature observed from about 800 temperature stations in the mainland of China from 1952 to 2018. The latitude and longitude of every observation station are involved. To better learn the spatial correlation of temperature values, we generate the temperature data map to fit them to our CRNN model and use convolution to learn its space correlation. The size of the generated temperature data map is  $36 \times 62$ , each row represents one degree in latitude, and each column represents one degree in longitude. To better demonstrate our experimental results, we have visualized the temperature data map according to the “Color Code for Products of Weather Forecast and Service” of China Meteorological Administration [23]. The corresponding relationship between color and temperature is shown in Figure 1.

The example of visualized temperature data map is shown in Figure 2. We will also use this kind of visualized method to show our final forecasting result in Section 5.

**3.2. CRNN Forecasting Model.** In this section, we overview the structure of the proposed CRNN model, which is illustrated in Figure 3.

As shown clearly in Figure 3, our training data are temperature data map with time-series length 4; the temperature data are daily average data observed from about 800 temperature stations in the mainland of China from 1952 to 2018. Then, we apply a CNN to process each temperature data map. The CNN portion includes convolution layer, activation function layer, pooling layer batch normalization layer, and flatten layer. After the CNN portion, there is an RNN portion with LSTM structure, which mainly consists of LSTM layer, dropout layer, and batch normalization layer. In the final, a dense layer is applied and the output of the whole model is a temperature data map series with length 4. The result will be compared with the label, which is a real temperature data map with series length 4 as well. After training, this CRNN model can be used to predict the future temperature according to past temperature data.

The imported training data of each individual CNN unit are the temperature data map, which is a two-dimensional map; the value of each pixel is temperature.

**3.3. Mapping in CRNN Model.** As shown in Figure 3, our input data are time-series temperature data map  $\mathbf{x}_{i,t}$  with size  $T \times H \times W$ , where  $i$  denotes the index number of images sequence and  $t$  denotes the time step label in time-series images sequence.  $H$  means the height of each data map, and  $W$  means the width of each data map. Input data are sent into our CNN portion and the output of CNN portion is a tensor  $\mathbf{z}_{i,t}$ , which equals to

$$\mathbf{z}_{i,t} = f(\mathbf{x}_{i,t}; \{\mathbf{w}_x\}), \quad (1)$$

where  $\mathbf{w}_x$  denotes the weighting coefficients in our CNN portion. Three CNN layers extract the space correlation in each temperature data map. Our CNN model can learn spatial dependency in each temperature data map individually. The CNN portion can map our input data  $\mathbf{x}_{i,t}$  to tensor  $\mathbf{z}_{i,t}$ , and  $\mathbf{z}_{i,t}$  is the input of the RNN portion.

In our RNN portion, the LSTM layer is the core structure to learn time dependence in time-series temperature data map sequence, and the LSTM layer maps the tensor  $\mathbf{z}_{i,t}$  to a representation series  $\mathbf{h}_{i,t}$  which equals to

$$\mathbf{h}_{i,t} = f(\mathbf{h}_{i,t-1}, \mathbf{z}_{i,t}; \{\mathbf{w}_z\}), \quad (2)$$

where  $\mathbf{w}_z$  denotes the weighting coefficients in the LSTM layer. Then, the output of the LSTM layer  $\mathbf{H}_i$  is sent to a dense layer. Through this dense layer, the prediction temperature values are generated. The size of generated data map sequence is equal to our input time-series data map sequence which is  $T \times H \times W$ . The output of the dense layer equals to

$$\hat{\mathbf{y}}_i = f(\mathbf{H}_i; \{\mathbf{w}_h\}). \quad (3)$$

Until now, our model can generate forecasting future temperature data map according to the past time-series temperature data map.

**3.4. Data Processing in CRNN Model.** In order to understand our CRNN model better, it is helpful to describe the procedure of data processing in detail, including the dimensions and values of important parameters and tensors. The values of the CRNN parameters are also selected carefully with many repeated experiments.

As shown in Figure 4, the input tensor is the past temperature data map series. The dimension of the input tensor is  $4 \times 36 \times 62$ , which means the input data are a series of temperature data map with series length 4 and the size of data map is 36 rows and 62 columns.

Then, one convolution layer is added; because the kernel size of the first convolution layer is  $3 \times 3$  and the number of filters is 64, the output of the first convolution layer is a tensor of dimension  $4 \times 34 \times 60 \times 64$ . The next activation function layer and batch normalization layer will not change the size of tensor. But the dimension of tensor is changed after one pooling layer, and the chosen pooling size is (2,2), so the dimension of data tensor becomes  $4 \times 17 \times 30 \times 64$ . Until now, one convolution process finished. Then, two similar convolution processes are used in our model; the only difference is the number of convolution filters in these two convolution layers which are 128 and 256. By the same convolution process as described in the previous paragraph, the dimension of our data tensor becomes  $4 \times 2 \times 6 \times 256$ .

Then, a flatten layer is used in order to connect the CNN with the RNN. As the layer name suggests, the function of this layer is to flatten each  $4 \times 2 \times 6 \times 256$  data tensor into a two-dimensional data array with size  $4 \times (2 \times 6 \times 256) = 4 \times 3072$ . This finishes the CNN portion of the CRNN model.

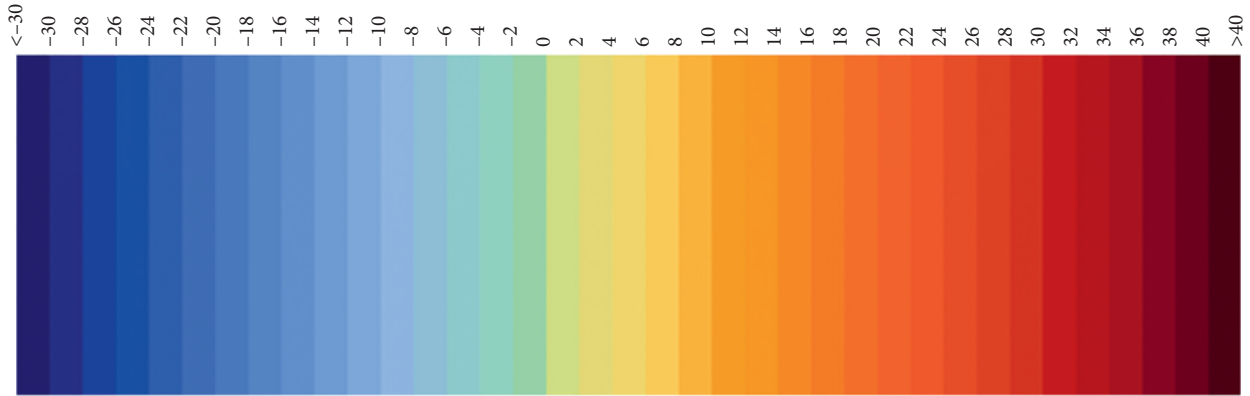


FIGURE 1: Temperature (in Celsius) and the corresponding color label.

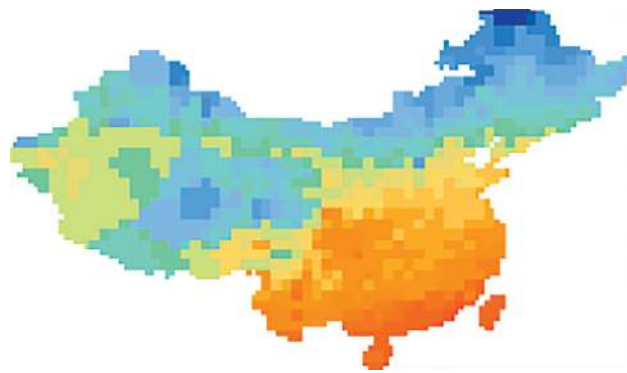


FIGURE 2: Visualized temperature map.

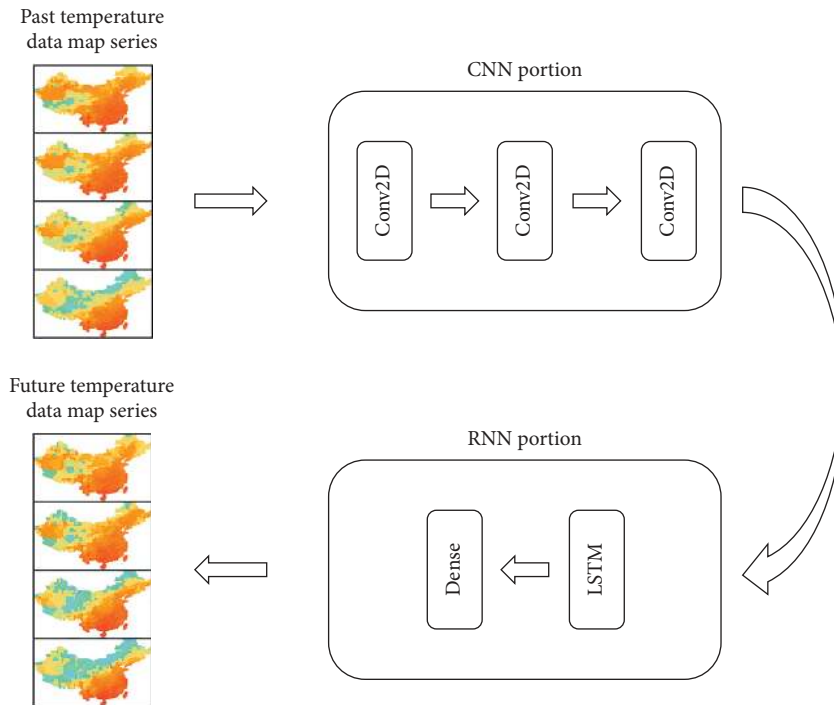


FIGURE 3: Illustration of CRNN model.

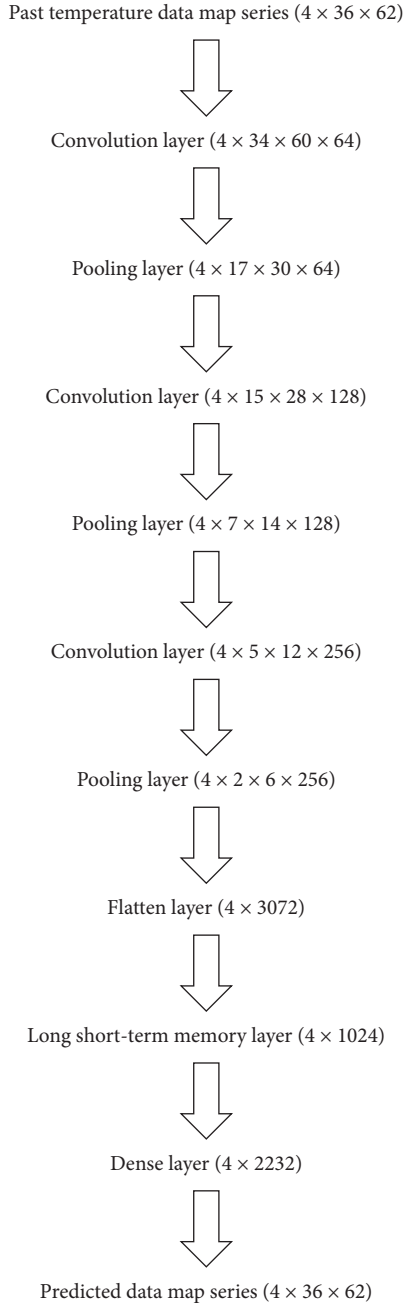


FIGURE 4: Data processing.

Note that the CNN portion processes each temperature data map individually. Next, we apply RNN to learn the information embedded in the time series. The first layer of the RNN portion is an LSTM layer. The LSTM layer has 4 time steps, which consists of 4 LSTM cells. We set the dimensions of both the LSTM states and outputs to be 1024. Therefore, the output of the LSTM layer is a data array with dimension  $4 \times 1024$ .

To generate the predicted temperature data map, we use a dense layer to generate output data tensors with the same dimension as the target data map. Specifically, the dimension is  $4 \times 2232$ . Note that 2232 equals to  $36 \times 62$ , the size of a temperature data map. We apply a reshape

step at the end to obtain 4 predicted data maps with size  $32 \times 62$ . This will be compared to the label time-series temperature data map for loss function calculation during training.

## 4. Experiment

**4.1. Data Collection and Data Preprocessing.** The training data used in this paper are the daily average temperature data provided by the China Meteorological Administration. The data label includes date, observation station number, observation station latitude, observation station longitude, and daily average temperature.

To extract the embedding space correlation and time correlation better, we put those temperature values into a two-dimensional data map according to the latitude and longitude of those observation stations. The value of each pixel is the temperature. The final size of the data map is  $36 \times 62$ , each row represents one degree in latitude, and each column represents one degree in longitude. The visualized version of the data map is shown in Figure 2.

Then, those data maps are ordered according to the time series, and the series length is 4. Because the daily temperature data are from January 1, 1952, to December 31, 2018, 24472 days in total, the number of data map series is 24469. Then, those data map series are separated by the ratio of eight to two. Eighty percent of data map series are used as training data and validation data. And twenty percent of data map series are used as testing data. All data map series are separated randomly.

The temperature values in the data map are normalized. The data are normalized according to the equation below:

$$X' = \frac{(X - X_{\min})}{(X_{\max} - X_{\min})}. \quad (4)$$

**4.2. Tuning of CRNN Model.** To get the best forecasting result, we need to tune our model to decide the hyperparameter values. We use  $k$ -fold cross validation to test the best hyperparameter values. The value of  $k$  is 10 in our experiments. The tuning result of some hyperparameters includes sequence length of temperature data map series and batch size, and the optimizer will be compared with the learning curve. And the learning curve with different hyperparameters will be shown in the following figures. And all hyperparameter values used in our CRNN model will be shown in the following table.

In Figure 5, we show the different learning curves when the input series length is different. We can see the performance is similar after the system has converged. And we finally choose to use the series length 4 to train our model because it will lead to the lowest validation loss.

Then, the difference caused by different batch sizes is shown in Figure 6. We can see we will get the best performance when using batch size 32.

The best number of LSTM neurons is also needed to be tested; according to the experiment result shown in Figure 7, we use 1024 neurons in LSTM layer.

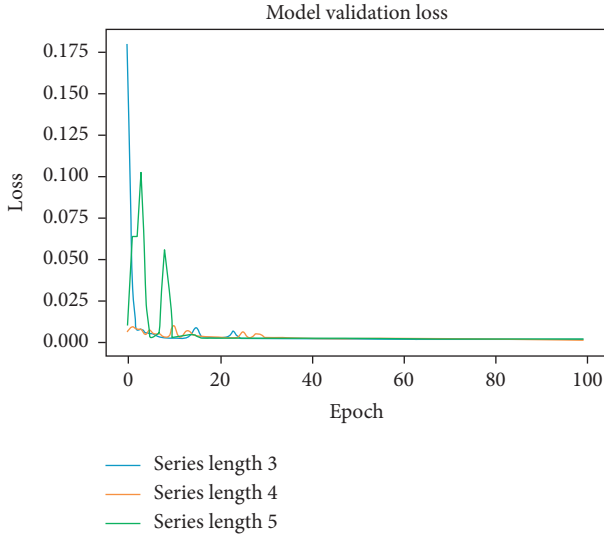


FIGURE 5: Learning curve with different lengths of temperature data map series.

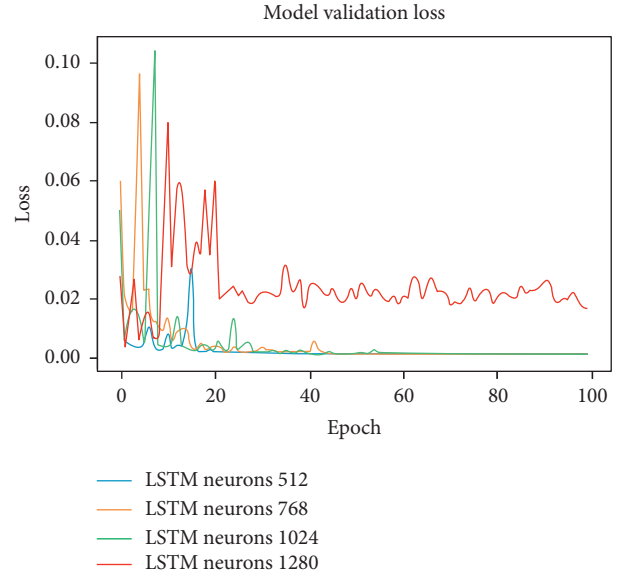


FIGURE 7: Learning curve with different numbers of LSTM neurons.

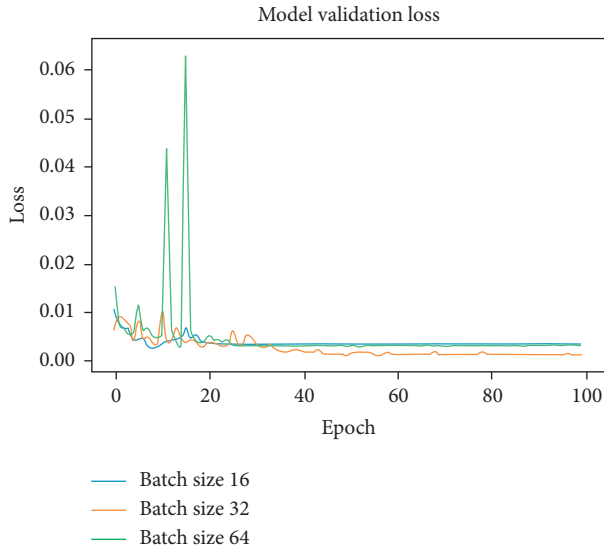


FIGURE 6: Learning curve with different batch sizes.

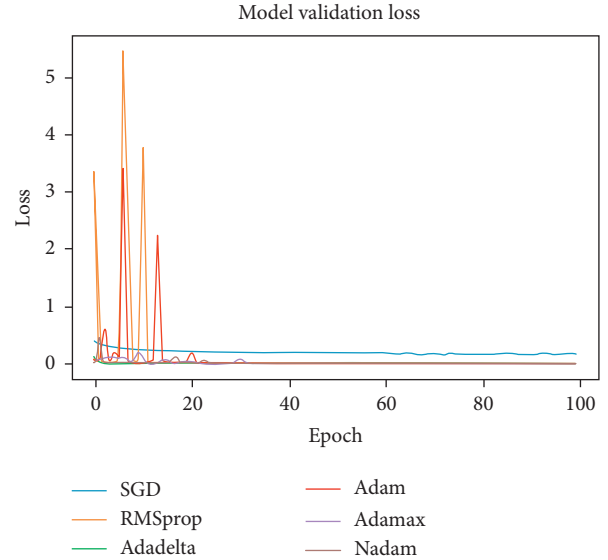


FIGURE 8: Learning curve with different optimizers.

And we also test different optimizers; except stochastic gradient descent (SGD), all other optimizers get similar results which are shown in Figure 8. Finally, we use Nesterov adaptive moment estimation (Nadam) optimization algorithm in our model training. The initial learning rate is 0.002, and the learning rate will be reduced every ten epochs if the model cannot get better performance.

Some hyperparameters which lead to smaller difference are shown in Table 1.

## 5. Result and Evaluation

Compared to the approaches demonstrated in Section 2, our CRNN has better performance; the criteria of comparison

are mean average error (MAE) and root mean squared error (RMSE). The comparison result is shown in Table 2. The equations of MAE and RMSE are shown below:

TABLE 1: Some parameter values in CRNN model.

Hyperparameters	Values
Convolutional kernel size	(3, 3)
Pooling size	(2, 2)
Number of convolutional filters	64,128,256
Recurrent dropout values in LSTM	0.2
Dropout values in LSTM	0.2

TABLE 2: Performance comparison with different approaches.

Approaches	MAE	RMSE
Operational consensus forecasts	1.253	2.174
Original backpropagation neural networks	0.934	1.952
Stacked denoising autoencoder	0.951	1.863
Convolutional recurrent neural networks	0.907	1.697

TABLE 3: Performance of CRNN model.

Criteria	Values
Mean average error	0.907
Root mean squared error	1.697
Accuracy when prediction error is smaller than 1°C	0.689
Accuracy when prediction error is smaller than 2°C	0.830
Accuracy when prediction error is smaller than 3°C	0.914

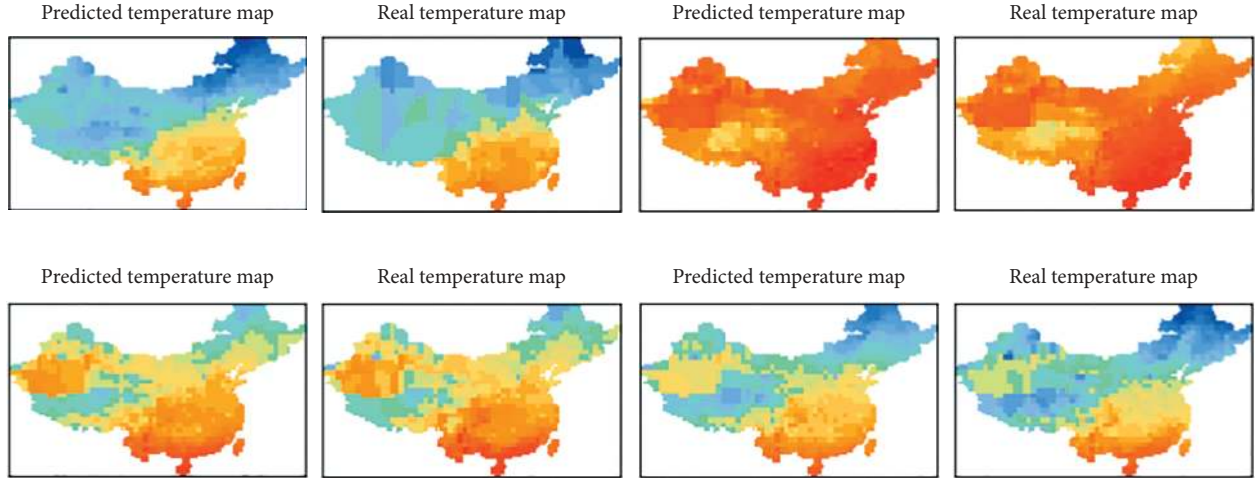


FIGURE 9: Visualized comparison figures between forecasting temperature data map and real temperature data map.

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^m |Y_{\text{forecasting}} - Y_{\text{real}}|, \quad (5)$$

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (Y_{\text{forecasting}} - Y_{\text{real}})^2}. \quad (6)$$

The performance of our CRNN for temperature prediction is listed in Table 3. The result is evaluated according to five criteria: mean average error (MAE), root mean squared error (RMSE), and the accuracy when prediction error is smaller than 1, 2, and 3°C.

All results are calculated between the forecasting data map and the real data map. Some examples of the visualized real data map and visualized forecasting data map are shown in Figure 9. As can be seen, our CRNN model can successfully predict the temperature.

## 6. Conclusion and Future Work

In this paper, we have developed a deep learning model that uses the convolutional recurrent neural network (CRNN) for temperature prediction in large-scale space. Specifically, we train the CRNN model with the daily average temperature data map set and demonstrate that this model can successfully predict the future temperature according to its past temperature data values. The predicted result of the developed CRNN is better than other benchmark methods.

There are two points that can be addressed to further improve this work. First, the shape of the mainland of China is an irregular figure, but our input temperature data map is a two-dimensional image. This means that we lack the temperature data in the pixels that are located outside the shape of China. It will bring bad influence to learn the spatial dependency in the pixels which are located near the boundary of China and cause prediction difference in those

pixels. Second, the values in temperature data maps are not fully accurate. More than 800 observation stations are still not enough to observe the temperature of every spot in China. Some lacking temperature is set according to the temperature value of the closest observation station. In actual temperature distribution, there are many factors influencing the temperature values, such as altitude, barometric pressure, humidity, and even density of population. We need to introduce a more complex meteorology-related algorithm into our CRNN model to get more accurate prediction values in the future.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] P. H. O. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *Proceedings of the 31st International Conference on Machine Learning (ICML)*, vol. 32, Beijing, China, 2014.
- [2] Z. Martin, D. Perekrestenko, and M. Tschannen, "Convolutional recurrent neural networks for electrocardiogram classification," in *Proceedings of the 2017 Computing in Cardiology (CinC)*, vol. 44, pp. 1–4, IEEE, Rennes, France, 2017.
- [3] P. Romeu, "Time-series forecasting of indoor temperature using pre-trained deep neural networks," in *Proceedings of the International Conference on Artificial Neural Networks*, Springer, Sofia, Bulgaria, pp. 451–458, September 2013.
- [4] F. Zamora-Martinez, P. Romeu, P. Botella-Rocamora, and J. Pardo, "On-line learning of indoor temperature forecasting models towards energy efficiency," *Energy and Buildings*, vol. 83, pp. 162–172, 2014.
- [5] M. Hossain, B. Rekabdar, S. J. Louis, and S. Dascalu, "Forecasting the weather of Nevada: a deep learning approach," in *Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, IEEE, Killarney, Ireland, July 2015.
- [6] H. Lin, Y. Hua, L. Ma, and L. Chen, "Application of Conv LSTM network in numerical temperature prediction interpretation," in *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*, pp. 109–113, ACM, Zhuhai, China, February 2019.
- [7] M. A. Cane, G. Eshel, and R. W. Buckland, "Forecasting Zimbabwean maize yield using eastern equatorial Pacific sea surface temperature," *Nature*, vol. 370, no. 6486, pp. 204–205, 1994.
- [8] T. Wilson, "A deep learning architecture for long-range forecasting of sea surface temperature anomalies," in *Proceedings of the AGU Fall Meeting Abstracts*, Washington, DC, USA, December 2018.
- [9] Y. Q. Ni, X. G. Hua, K. Q. Fan, and J. M. Ko, "Correlating modal properties with temperature using long-term monitoring data and support vector machine technique," *Engineering Structures*, vol. 27, no. 12, pp. 1762–1773, 2005.
- [10] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24–38, 2009.
- [11] J. P. Donate, X. Li, G. G. Sánchez, and A. S. de Miguel, "Time series forecasting by evolving artificial neural networks with genetic algorithms, differential evolution and estimation of distribution algorithm," *Neural Computing and Applications*, vol. 22, pp. 11–20, 2013.
- [12] K. L. Ho, Y.-Y. Hsu, and C.-C. Yang, "Short term load forecasting using a multilayer neural network with an adaptive learning algorithm," *IEEE Transactions on Power Systems*, vol. 7, no. 1, pp. 141–149, 1992.
- [13] I. Maqsood, M. R. Khan, and A. Abraham, "An ensemble of neural networks for weather forecasting," *Neural Computing & Applications*, vol. 13, no. 2, pp. 112–122, 2004.
- [14] K. Abhishek, M. P. Singh, S. Ghosh, and A. Anand, "Weather forecasting model using artificial neural network," *Procedia Technology*, vol. 4, pp. 311–318, 2012.
- [15] F. Woodcock and C. Engel, "Operational consensus forecasts," *Weather and Forecasting*, vol. 20, no. 1, pp. 101–111, 2005.
- [16] B. Xu, H.-C. Dan, and L. Li, "Temperature prediction model of asphalt pavement in cold regions based on an improved BP neural network," *Applied Thermal Engineering*, vol. 120, pp. 568–580, 2017.
- [17] P. Vincent, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of machine learning research*, vol. 11, pp. 3371–3408, 2010.
- [18] F. A. Gers and E. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333–1340, 2001.
- [19] Z. Zuo, B. Shuai, and G. Wang, "Convolutional recurrent neural networks: learning spatial dependencies for image representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 18–26, Boston, MA, USA, June 2015.
- [20] J. Zheng, Y. Wang, X. Zhang, and X. Li, "Classification of severely occluded image sequences via convolutional recurrent neural networks," in *Proceedings of the 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 395–399, IEEE, Anaheim, CA, USA, November 2018.
- [21] L. Chen and S. Li, "Improvement research and application of text recognition algorithm based on CRNN," in *Proceedings of the 2018 International Conference on Signal Processing and Machine Learning*, pp. 166–170, Shanghai, China, 2018.
- [22] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2392–2396, IEEE, New Orleans, LA, USA, March 2017.
- [23] China Meteorological Administration, "Color code for products of weather forecast and service," 2009, <http://www.doc88.com/p-5117300218595.html>.