Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2020-0158

Vol. **31**, No. 1, pp. 131-153
January 2022

A Deep Spatio-Temporal Forecasting Model for Multi-Site Weather Prediction Post-Processing

Wenjia Kong[1], Haochen Li[3], Chen Yu[2], Jiangjiang Xia[4], Yanyan Kang[5] and Pingwen Zhang[2,*]

[1] *Academy for Advanced Interdisciplinary Studies, Peking University, Beijing 100871, China.*
[2] *School of Mathematical Sciences, Peking University, Beijing 100871, China.*
[3] *School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China.*
[4] *Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100029, China.*
[5] *Beijing Weather Forecast Center, Beijing 100089, China.*

Received 10 August 2020; Accepted (in revised version) 3 August 2021

Abstract. In this paper, we propose a deep spatio-temporal forecasting model (DeepSTF) for multi-site weather prediction post-processing by using both temporal and spatial information. In our proposed framework, the spatio-temporal information is modeled by a CNN (convolutional neural network) module and an encoder-decoder structure with the attention mechanism. The novelty of our work lies in that our model takes full account of temporal and spatial characteristics and obtain forecasts of multiple meteorological stations simultaneously by using the same framework. We apply the DeepSTF model to short-term weather prediction at 226 meteorological stations in Beijing. It significantly improves the short-term forecasts compared to other widely-used benchmark models including the Model Output Statistics method. In order to evaluate the uncertainty of the model parameters, we estimate the confidence intervals by bootstrapping. The results show that the prediction accuracy of the DeepSTF model has strong stability. Finally, we evaluate the impact of seasonal changes and topographical differences on the accuracy of the model predictions. The results indicate that our proposed model has high prediction accuracy.

AMS subject classifications: 62P12, 86A10, 93B15, 97M10

Key words: Weather forecasting, post-processing, spatio-temporal modeling, deep learning.

*Corresponding author. *Email addresses:* pzhang@pku.edu.cn (P. Zhang), wjkong@pku.edu.cn (W. Kong), lihaochen_bjut@sina.com (H. Li), yuchen1995@pku.edu.cn (C. Yu), xiajj@tea.ac.cn (J. Xia), yanyan051022@163.com (Y. Kang)

1 Introduction

Weather forecasting has always been a matter of general concern. Accurate weather forecasts can reduce the adverse effects caused by extreme weather, reduce economic losses, and have an important impact on various industries such as tourism and transportation. Nowadays, as supercomputers gradually enter a period of rapid development, numerical weather prediction (NWP) has become a major technical method and research direction in the field of weather forecasting. The idea of numerical weather prediction was first proposed by Bjerknes [3] in the early 20th century and achieved rapid development afterwards [6, 32]. Nevertheless, NWP forecasts often carry significant systematic bias. Hence, post-processing has become standard practice since at least Glahn et al. (1972) [19], in which Glahn et al. demonstrated a version of model output statistics (MOS) that improves the raw NWP forecast accuracy. The MOS method has been widely used since it was proposed [1, 10, 18, 21, 39, 44]. Besides MOS, other statistical algorithms are commonly adopted in post-processing of weather prediction, such as Kalman filtering [13, 14, 28, 43], the analog ensemble [12], anomaly numerical-correction with observations [29] and Markov Chain models [5, 37].

In addition to traditional statistical methods, machine learning and artificial neural networks [25] are gradually being widely used in weather forecasting [4, 7, 20, 31, 40, 41, 45, 46]. Haochen Li et al. [26] proposed a model output machine learning (MOML) method for grid temperature forecasting. The results showed a better performance than the ECMWF (European Centre for Medium-range Weather Forecasts) model without post-processing and the traditional post-processing methods MOS, especially for winter. Huan Zheng et al. [48] used k-means algorithm to divide the samples into several categories based on the similarity of weather in historical days and proposed a new extreme gradient boosting (XGBoost) model for short-term wind power forecasting. Rasp et al. [30] established a fully-connected neural network to predict the 2-m temperature in Germany. The results showed that the neural network approach significantly outperforms traditional statistical methods. Zaytar et al. [47] established the seq2seq model based on LSTM (Long Short-Term Memory), and made end-to-end [25] predictions on the temperature, humidity, and wind speed of 9 cities in Morocco. The experimental results were better than traditional statistical methods. In the prediction of extreme weather forecasts, Ashesh Chattopadhyay et al. [8] proposed CapsNets to predict the geographic area of extreme surface temperature in North America. The results show that the multivariate data-driven framework is expected to achieve accurate extreme weather predictions.

In the field of weather forecasting, it is often necessary to consider the impact of both temporal and spatial characteristics. Xingjian Shi et al. [33] proposed the convolutional LSTM (ConvLSTM) and used it to build an end-to-end trainable model for the precipitation nowcasting problem. Experiments showed that the ConvLSTM network can capture spatio-temporal correlations. Due to the shortcoming of ConvLSTM to model the dynamic changes of clouds, Xingjian Shi introduced a new precipitation prediction model named TrajGRU (Trajectory Gate Recurrent Unit) that can actively learn the location-

variant structure for recurrent connections [34]. Ghaderi et al. [17] took the wind speed prediction of 57 stations in the northeastern United States as an example and built an LSTM-based spatio-temporal forecast model, which can obtain the wind speed prediction results of all stations at the same time. Karevan et al. [24] established a 2-layer stacked LSTM model based on spatio-temporal modeling to predict the temperature. In the 2-layer stacked LSTM model, the hidden state of the first LSTM layer is used as the input of the second LSTM. Experiments show that the predictability of the stacked LSTM model is improved compared to the traditional LSTM model. More references to previous work on spatio-temporal forecasting can be found in [35].

Traditional post-processing methods usually establish a prediction model for each site separately, which do not accurately reproduce or optimally exploit the spatial correlation between nearby sites. Although the individual model of each site can capture the characteristics of the site, the model of one site may not be suitable for the prediction of other sites. When the number of sites is huge, this method needs to build a great number of models, which is very inefficient. Therefore, it is necessary to establish a multi-site prediction model. Motivated by this observation, we present a new prediction model called Deep Spatio-Temporal Forecasting Model (DeepSTF). This model is concerned with short-term multi-site forecasting using both temporal data as well as spatial information. In the extraction of temporal features, a BiGRU-based Encoder-Decoder model with attention mechanisms is established. To extract spatial features, firstly, a CNN model is constructed to obtain the spatial information of the numerical forecasts around each station; Secondly, the latitude, longitude and altitude of the station are encoded with a fully-connected network to reflect the differences between stations. The main contributions of our work are the fact that our model takes full account of temporal and spatial characteristics and obtain forecasts of all meteorological stations at the same time by using one framework. We apply the DeepSTF model to the short-term weather prediction of 226 stations in Beijing to forecast the changes of 4 meteorological factors in the next 3 days: temperature, relative humidity, average wind speed, and gust wind speed. The results show that our framework significantly improves the short-term forecasts compared to a set of widely-used benchmarks models.

The remainder of the paper is organized as follows. In Section 2, we describe the problem and data in this study. In Section 3, we elaborate on the proposed architecture of the deep spatio-temporal forecasting model. Section 4 verifies our model on the short-term weather forecast in Beijing and analyzes the adaptability of the model under different months and stations. The conclusions and future work are finally drawn in Section 5.

2 Problem and data description

2.1 Problem description

In this paper, We focus on the short-term weather prediction in Beijing. In this problem, there are 226 stations that need to be predicted. They are distributed in all municipal

districts of Beijing and involve different terrains. If we use the traditional method to establish a prediction model for each station separately, there will be more repetitive work. In practical applications, it will face the problems of storage inconvenience and difficulty in maintenance. Hence, for 226 meteorological observatories in Beijing, we conduct a multi-site prediction, forecasting the changes in temperature, relative humidity, average wind speed, and gust wind speed in the next 72 hours. In this case study, our model makes 8 times of forecast every day (i.e. 2:00, 5:00, $\cdots$, 23:00). At each forecasting time, the model will predict the weather conditions every 3 hours for the next 3 coming days (namely 24 predictions in total).

2.2 Data description

In the post-processing of weather forecast, there are two types of data commonly used: meteorological observational data and numerical weather prediction data.

Historical observational data. The observational data is measured by various meteorological stations at time intervals, including some meteorological factors such as temperature and relative humidity. We record $S=(s_1,s_2,\cdots,s_K)$ as the set of stations that need to be forecasted. For each meteorological observation station s_k, the meteorological observational data is recorded as $\boldsymbol{X}_k$.

$$\boldsymbol{X}_k=\{x_{l,m}\}_{l=1,2,\cdots,L,\ m=1,2,\cdots,M}, \tag{2.1}$$

where L is the length of the time series of observations, its time interval is usually 1 hour. M is the number of meteorological factors.

In this problem, the historical observational data covers 226 meteorological stations. These stations' distribution is shown in Fig. 1 , covering all the municipal jurisdictions in Beijing. The time range of the observational data is January 1, 2015 to November 30, 2017 and the interval is 1 hour. There are 8 meteorological observation factors including temperature, air pressure, relative humidity, precipitation, average wind speed, average wind direction, gust wind speed and gust wind direction. Several stations have missing data, which are completed by using linear interpolation.

Numerical weather prediction data. The NWP data contains the grid forecast results for a region in the next few days. Compared with the observational data, it contains more meteorological factors, such as the total cloud cover, snow depth, and mean sea level pressure. The NWP data divides the area into a uniform grid, and usually cannot accurately correspond to a certain weather observation station. We assume that the length of the prediction time series of NWP data in each prediction period is T. For example, if a model has maximum lead time of 240 hours and makes a forecast every 3 hours, this means that during the prediction period (0-240h), the models forecast time series length

Figure 1: Distribution of meteorological stations in Beijing. The green dots represent the weather stations.

is $T=8$. Then the NWP data for each prediction period can be written as

$$\boldsymbol{P}=\left\{p^t_{z,g_1,g_2}\right\}^{t=1,2,\cdots,T}_{z=1,2,\cdots,Z,\ g_1=1,2,\cdots,G_1,\ g_2=1,2,\cdots,G_2}, \tag{2.2}$$

where Z is the number of meteorological predictors and the size of the NWP mesh is $G_1\times G_2$.

For NWP data, we use the data from the ECMWF-IFS (the European Centre for Medium-Range Weather Forecast Integrated Forecasting System global model). The time range is from January 15, 2015 to November 30, 2017 and the spatial range is 35°N $\sim$ 45°N, 110°E $\sim$ 120°E. The data are initialized at UTC 00:00 and UTC 12:00 each day, forecasting the weather changes in the next 240 hours. The forecast interval for the first 72 hours is 3 hours, while the remaining is 6 hours. The spatial resolution on the ground is $0.125^\circ\times0.125^\circ$ $(\approx 13km\times13km)$, while at high altitudes the resolution is $0.25^\circ\times0.25^\circ$ $(\approx 26km\times26km)$. After deleting some unnecessary predictors (e.g. land-sea mask), the NWP data include 44 predictors such as 2-meter temperature, 10 meter V wind component, and mean sea level pressure, among which there are 24 high altitude variables (listed in Table 1). To ensure data consistency, firstly, the time interval is unified to 3 hours by using linear interpolation; secondly, the spatial resolution is unified to $0.125^\circ\times0.125^\circ$ by using bi-linear interpolation.

Table 1: Information of the predictors taken from ECMWF-IFS.

Predictors		
100 meter U wind component	Low cloud cover	Sea surface temperature
100 meter V wind component	Large-scale precipitation	Temperature [500 hPa]
10 meter U wind component	Mean sea level pressure	Temperature [850 hPa]
10 meter V wind component	Potential vorticity [1000 hPa]	Total cloud cover
2 meter dewpoint temperature	Potential vorticity [500 hPa]	Total column water
2 meter temperature	Potential vorticity [850 hPa]	Total column water vapour
Convective available potential energy	Specific humidity [1000 hPa]	Total precipitation
Divergence [1000 hPa]	Specific humidity [500 hPa]	U wind component [500 hPa]
Divergence [500 hPa]	Specific humidity [850 hPa]	U wind component [850 hPa]
Divergence [850 hPa]	Relative humidity [1000 hPa]	V wind component [500 hPa]
Zero Degree Level	Relative humidity [500 hPa]	V wind component [850 hPa]
Forecast albedo	Relative humidity [850 hPa]	Vertical velocity [1000 hPa]
Geopotential height [1000 hPa]	Snow depth	Vertical velocity[500 hPa]
Geopotential height [500 hPa]	Snowfall	Vertical velocity [850 hPa]
Geopotential height [850 hPa]	Skin temperature	

3 The DeepSTF model

3.1 Network architecture

In this section, we elaborate on the proposed architecture of the Deep Spatio-Temporal Forecasting Model (DeepSTF). Our framework implements end-to-end short-term prediction of meteorological factors. We establish a model based on BiGRU (Bidirectional Gated Recurrent Unit) to extract temporal features, and a model based on a CNN (convolutional neural network) and a fully-connected network to extract spatial features, which fully reflects the relationship of meteorological factors between time and space. Because the model incorporates the regional information of NWP data and the geographic coordinates of the stations, the model can be trained with data from multiple weather stations simultaneously, reducing the disadvantages of traditional single-site prediction algorithms. It is suitable for multi-site forecasting of various conventional meteorological factors such as temperature, relative humidity, etc.

The core composition of DeepSTF is an Encoder-Decoder structure, which is usually used to solve sequence-to-sequence generation problems, such as machine translation in natural language processing. Therefore, the Encoder-Decoder model is also called Seq2seq model [11, 38]. In the encoding stage, the model can convert the input sequence into a fixed-length vector. In the decoding stage, the previously generated fixed vector will be converted into an output sequence. The weather forecast problem is a time series prediction, and the input historical weather data and the output future weather conditions are all sequences, so it is suitable to adopt the Seq2seq model. The Encoder-Decoder model is a model framework. In specific implementation, we can choose different deep learning models to combine, in this paper, we use the BiGRU [11] model as the encoder

and decoder. GRU is a variant of the LSTM (Long Short-Term Memory, [22]) model, which simplifies the network structure based on the forget gate and update gate mechanism. The BiGRU model contains two GRUs, one taking the input in a forward direction, and the other in a backwards direction, which can reflect the impact of the before and after time series on the current moment. In addition, in response to the problem of the disappearance of gradients in the Seq2seq model, which is the information loss caused by the conversion of the input sequence to the fixed-length vector, we introduced the attention mechanism [2]. The attention mechanism assigns corresponding weights to the data at different moments of the input sequence. In the weather forecast, it can pay more attention to the information of historical weather conditions and improve the long-term prediction accuracy.

As shown in Fig. 2, the model outputs predict values of each meteorological factor at n time points in the future, which is denoted as $\hat{\boldsymbol{Y}}=(\hat{y}_1,\hat{y}_2,\cdots,\hat{y}_n)$, while the true values is recorded as $\boldsymbol{Y}$. The input data consist of three parts, among which the time and geographic coordinates are defined as $\boldsymbol{W}$. It is encoded by a fully-connected neural network to obtain the expression $\boldsymbol{D}=[\boldsymbol{d_1},\boldsymbol{d_2},\cdots,\boldsymbol{d_n}]$, where $\boldsymbol{d_j}=(d_{1,j},d_{2,j},\cdots,d_{h,j})^T$ is an h-dimensional column vector at time j $(j=1,\cdots,n)$. The historical observational data is recorded as $\boldsymbol{X}$ and the NWP data is recorded as $\boldsymbol{P}=[\boldsymbol{P_1},\boldsymbol{P_2},\cdots,\boldsymbol{P_n}]$, where $\boldsymbol{P_j}$ is the grid data of the NWP data at time j. They are processed through the Encoder-Decoder model to get $\boldsymbol{Q}=[\boldsymbol{q_1},\boldsymbol{q_2},\cdots,\boldsymbol{q_n}]$, where $\boldsymbol{q_j}=(q_{1,j},q_{2,j},\cdots,q_{h,j})^T$ is an h-dimensional column vector at time j. Then we concatenate $\boldsymbol{Q}$ and $\boldsymbol{D}$ to obtain the prediction sequence through a fully-connected network. We use the Mean Square Error (MSE) as the loss function. The DeepSTF algorithm is illustrated by the pseudo-code in Algorithm 1. We refer to the existing experience for parameter settings of deep learning network like VGGNet [36] to set the hyperparameters (the number of neurons, layers, and the convolution kernel size, etc.) of our neural network structure, and the details of each sub-module are described as follows.

Algorithm 1 The calculation procedure of the DeepSTF model

Input:
$\boldsymbol{\epsilon}$ – the loss of model convergence.
$\boldsymbol{W}$ – time and geographic coordinates.
$\boldsymbol{X}$ – historical observational data.
$\boldsymbol{P}$ – the NWP data.
Output: $\boldsymbol{Y}$ – the true value of weather factors.
repeat
1: $\boldsymbol{D}=\text{Fully-Connected}(\boldsymbol{W})$
2: $\boldsymbol{Q}=\text{Encoder-Decoder}(\boldsymbol{X},\boldsymbol{P})$
3: Concatenate $\boldsymbol{Q}$ and $\boldsymbol{D}$ to get $\boldsymbol{V}=(\boldsymbol{D};\boldsymbol{Q})$
4: $\hat{\boldsymbol{Y}}=\text{Fully-Connected}(\boldsymbol{V})$
5: $Loss=\text{MSE}(\boldsymbol{Y},\hat{\boldsymbol{Y}})$
until $Loss\leq\boldsymbol{\epsilon}$

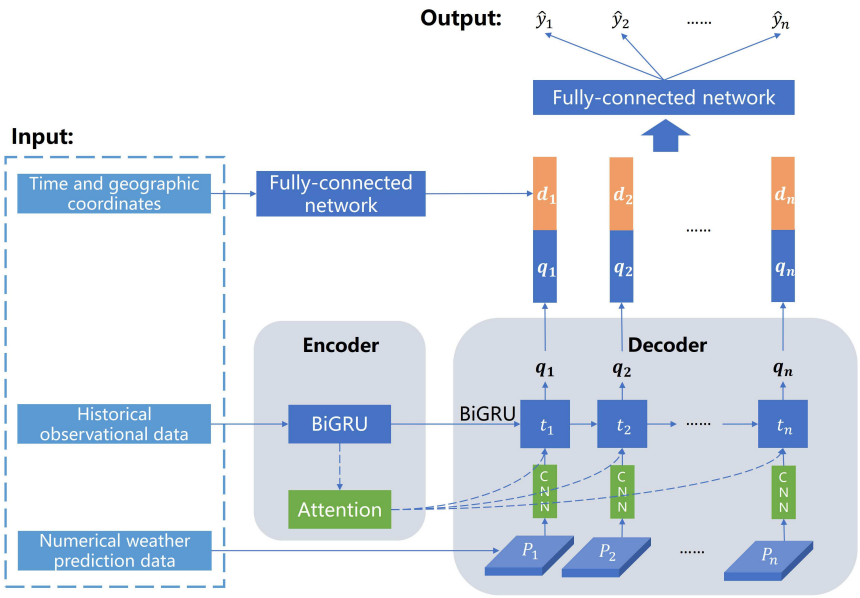


Figure 2: The Architecture of Deep Spatio-Temporal Forecasting Model (DeepSTF). The blue dotted frame on the left is the input part of the model, and the gray box is the Encoder-Decoder module of the model.

3.2 Model inputs

Based on the known data, we select the appropriate input for feature construction. Each part of the input is described below in detail.

1) **Time and geographic coordinates.** In the multi-site forecast, the forecast lead time, the day of the year, the month, and the hour are used as time features. The spatial features include the longitude, latitude, and altitude of the station. We also considered the difference between the site and the nearest grid point by calculating the latitude and longitude difference and distance between them. Through the above analysis, time and geographic coordinates features have a total of 10 dimensions.

2) **Historical observational data.** We use the hourly observational data of the past 48 hours before the start forecast time, including 8 meteorological factors: temperature, pressure, relative humidity, wind direction, wind speed, gust wind direction, gust wind speed and precipitation, which fully reflect the changes in historical weather.

3) **Numerical weather prediction data.** The NWP grid data contains 44 forecast predictors as shown in Table 1. For each station, we need to extract its corresponding numerical forecast results from the NWP data. In the traditional weather forecast-

ing algorithm, the interpolation method is usually used to obtain the numerical forecast results of the corresponding stations [26]. In this paper, We establish a CNN model for the NWP data to extract grid forecast information around the station, which fully reflects the spatial characteristics of the numerical prediction.

3.3 Encoding for the historical observational data

In the encoder part, the model mainly processes the historical real-time monitoring data of meteorological stations. To extract the information of historical weather more comprehensively, a 3-layer BiGRU model is adopted. The number of neurons in each layer is 256, and the input dimension is (48×8), which represents 48-hour historical time series and 8 meteorological predictors. At the time t, the hidden layer vector $\boldsymbol{h_t}$ is related to the historical weather conditions $\boldsymbol{x_t}$, the previous hidden layer vector, and the next hidden layer vector, which can be written as:

$$\boldsymbol{h_t}=f(\boldsymbol{h_{t-1}},\boldsymbol{x_t},\boldsymbol{h_{t+1}}),\quad t=1,2,\cdots,48. \tag{3.1}$$

3.4 Decoding for the forecast period

In the decoder stage, a 3-layer BiGRU model with attention mechanism is used, and the number of neurons in each layer is 256. First, the decoder receives the last hidden layer vector of the encoder for initialization. Then, at each prediction time step i in the decoder stage, the hidden layer vector can be written as

$$\boldsymbol{h_i}=g(\boldsymbol{h_{i-1}},\hat{\boldsymbol{p}}_i,\boldsymbol{c_i}),\quad i=1,2,\cdots,24, \tag{3.2}$$

where $\boldsymbol{h_{i-1}}$ is the hidden layer vector at the previous moment, $\hat{\boldsymbol{p}}_i$ is the feature vector of the numerical forecast at this moment, which is obtained from the NWP result $\boldsymbol{P_i}$ through CNN encoding. $\boldsymbol{c_i}$ is the attention vector related to the output of the encoder.

In the traditional weather post-processing algorithm, the nearest neighbor interpolation is usually used for the grid data to obtain the numerical prediction results of the station, which lacks the extraction of spatial features. To obtain the spatial connection of meteorological factors, we establish a CNN model on the processing of numerical weather forecast results. Specifically, we take the nearest grid point of the station as the center and extract the surrounding 9×9 grid data, which is nearly $120km\times 120km$, so it is enough to reflect the weather conditions around the weather station. Fig. 3 shows the extraction process. After extraction, a CNN model is used on the extracted grid data to obtain the spatial features. The input of the CNN model is 9×9 grid data of 44 prediction variables. The 3×3 convolution kernel whit 64 output channels (Conv3-64) is used for spatial feature extraction. Finally, the results are integrated into the decoder through a fully-connected layer with 256 neurons (FC-256). The structure of the CNN is shown in Table 2.

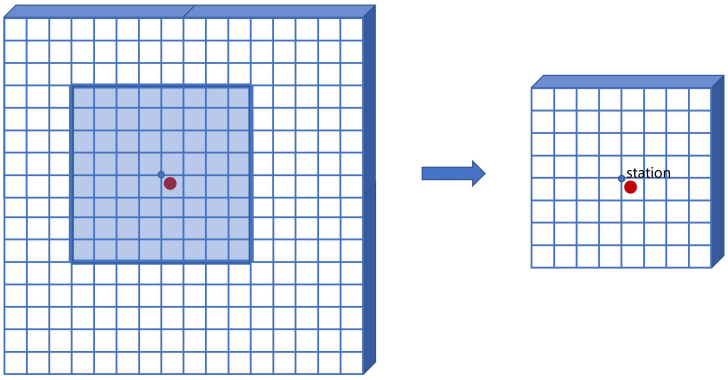


Figure 3: The extraction process of the NWP data. The red dot represents the meteorological observation station, and the surrounding 9×9 grid area is extracted with the station as the center.

Table 2: The structure of the CNN model.

Input $(44\times9\times9)$
Conv3-64
Conv3-64
Maxpool(2×2)
FC-256
Decoder

3.5 Feature modeling for time and geographic coordinates

For time and geographic coordinates, we combine the time information corresponding to the predicted time, such as year, month, and hour, and the geographic coordinates corresponding to the station. Then establish a 3-layer fully-connected model, where the number of neurons in each layer is 256. By modeling the characteristics of time and geographic coordinates, the model can fully reflect the characteristics of different stations and different prediction times.

3.6 Model output

In the final prediction output stage, we first combine the features of time and geographic coordinates with the output of the Encoder-Decoder stage, and then establish a fully-connected model to get the prediction sequence. As shown in the Fig. 2, at each prediction time step i, we will concatenate $\boldsymbol{d}_i$ and $\boldsymbol{q}_i$ to obtain v_i, and then input it into the fully-connected layer. The fully-connected model has 3 layers, and the number of neurons in each layer is 1024, 512, 256 respectively. Through the above analysis, we can see that the entire model fully considers the characteristics of space and time, reflecting the relationship of meteorological factors in the time and space dimension.

4 Numerical experiments

4.1 Experimental settings

Data set division. According to the description of historical observational data and ECMWF-IFS data, we select 500,000 samples from 2015-01-15 to 2016-09-30 randomly as the training set. The data from 2016-10-01 to 2016-10-31 is used as the validation set, and the data from 2016-11-01 to 2017-10-31 is used as the test set.

Model training. We choose AdamW as the optimizer. AdamW is an improvement of Adam [27], which can improve the generalization ability. When the neural network needs regular terms, replacing Adam with AdamW will get better performance. For the activation function, ReLU is used in our model. ReLU can effectively reduce the problem of gradient disappearance and is widely used in deep learning [25]. The initial learning rate is $3\times e^{-4}$, the batch size is 64 and the loss function is MSE. To accelerate the convergence speed in the experiment, the BatchNorm structure is added, which reduces the adverse effects caused by the overfitting problem at the same time [23]. In the second half of the training process, convergence tends to be slow, so we reduce the learning rate for further converge the model. The specific approach is to reduce the learning rate by 10% for every 5 steps of training.

Evaluation criteria. To measure and evaluate the performance of different methods, Root Mean Squared Errors (RMSE) and Accuracy (ACC) are adopted. The calculation method of ACC is different depending on the meteorological predictors. For temperature and relative humidity, we need to judge whether the error between the predicted value and the true value is within a certain threshold. The calculation formula is

$$ACC_1=\frac{|I|}{N},\quad I=\{i\ \big|\ |y_i-\hat{y}_i|\leq\theta\},\tag{4.1}$$

where y_i,$\hat{y}_i$ represent the true value and the predicted value, N is the total number of samples, θ is the threshold. For temperature $\theta=2$ Degrees Celsius, and for relative humidity $\theta=10\%$. To calculate the accuracy of wind speed, we first need to divide the wind speed into corresponding levels as shown in Table 3, and then determine the forecast score according to the difference between the predicted level and the real level [16]. The calculation can be written as

$$ACC_2=\frac{\sum_{i=1}^{N}s_i}{N},\tag{4.2}$$

where s_i represents the forecast score of the i-th sample, which ranges from 0 to 1. Note that d_i,$\hat{d}_i$ are the actual and predicted levels of wind speed respectively, then the forecast

Table 3: Wind speed and levels (m/s).

Wind Speed	Wind Levels
<0.3	0
<1.6	1
<3.4	2
<5.5	3
<8.0	4
<10.8	5
<13.9	6
<17.2	7
<20.8	8
<24.5	9
<28.5	10
<32.7	11
≥ 32.7	12

score s_i can be calculated as

$$s_i = \begin{cases} 1, & |d_i - \hat{d}_i| = 0, \\ 0.6, & |d_i - \hat{d}_i| = 1, \\ 0.4, & |d_i - \hat{d}_i| = 2, \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

4.2 Comparison of different model structures

To verify the rationality of the model structure, we first use temperature prediction as an example and compare a variety of model structures in the experiment. In these model structures, we use the XGBoost [9] algorithm as the baseline and compare the impact of the BiGRU model, Seq2seq framework, attention mechanism, and CNN module on the prediction results. We test these models on the validation set, and then make predictions on the test set. Table 4 shows the results of each model in the temperature prediction on the test set. It can be seen that the prediction results of all models exceed the results of ECMWF-IFS. It shows that the post-processing algorithm of weather forecasting has a good correction effect on the numerical prediction results.

Compared with ECMWF-IFS, the prediction accuracy of XGBoost has significantly improved. The BiGRU model uses only one bidirectional GRU without a seq2seq structure, which has no obvious advantage over XGBoost. The Seq2seq model uses the classic seq2seq model without introducing an attention mechanism, and in the processing of NWP data, the nearest neighbor interpolation is used to approximate the station's numerical prediction results without considering the spatial characteristics. This model is improved by about 3% compared to BiGRU, which shows that the seq2seq model can

Table 4: Comparison of different models in temperature forecasting.

Model	RMSE	ACC
ECMWF-IFS	2.94	56.72%
XGBoost	2.65	63.37%
BiGRU	2.69	63.15%
Seq2seq	2.57	66.08%
AttnSeq2seq	2.54	67.45%
CNNseq2seq	2.50	68.41%
DeepSTF	**2.41**	**70.03%**

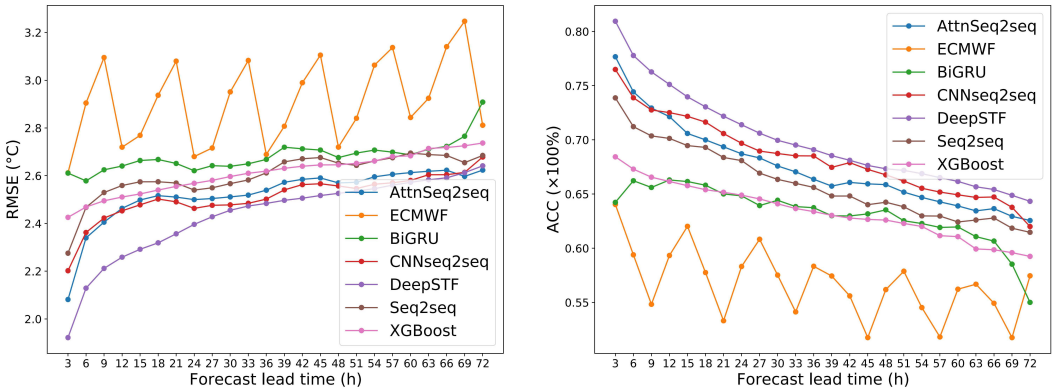
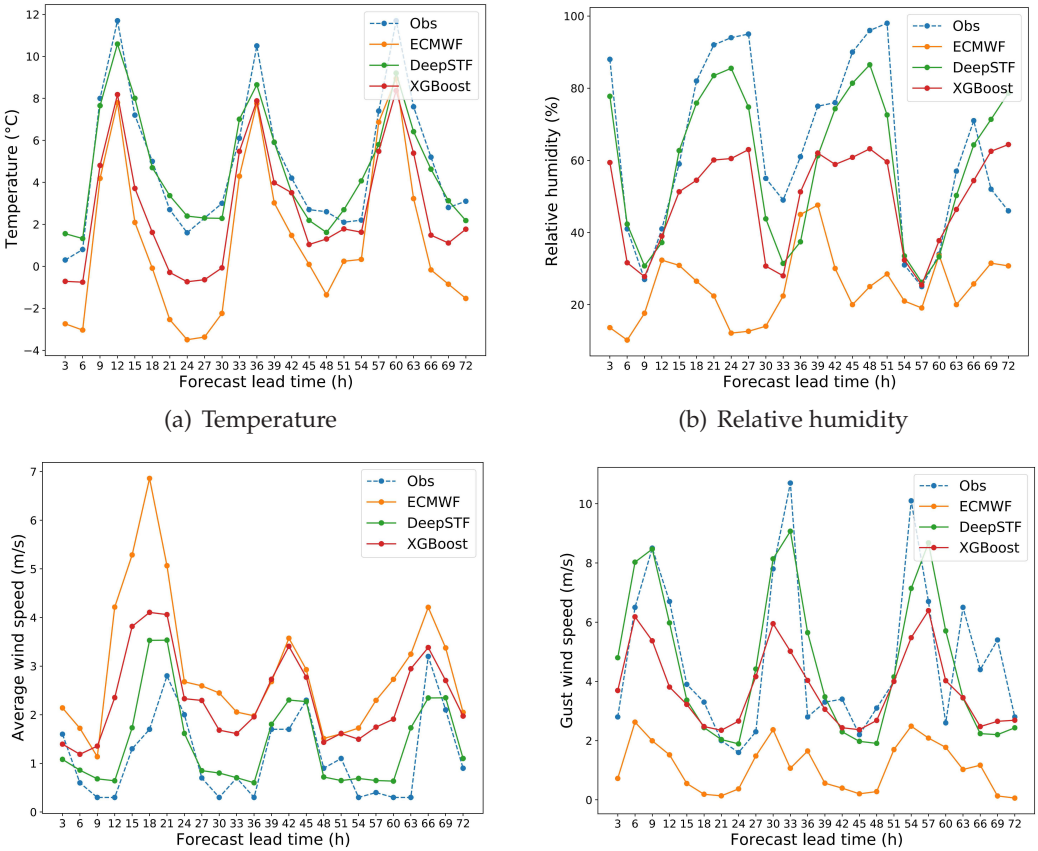


Figure 4: RMSE (left) and ACC (right) of different models in temperature forecasting. The prediction results of the DeepSTF model are better than others, which has the lowest RMSE and highest accuracy.

combine historical observational data with numerical forecast data better, and can obtain more accurate prediction results than a single BiGRU. The AttnSeq2seq model introduces an attention mechanism on basic seq2seq, and the prediction accuracy is improved. This shows that the attention mechanism can pay more attention to the information of historical observational data, and at the same time will reduce the vanishing gradient problem. The CNNseq2seq model establishes a CNN model on the NWP data during the decoder stage of seq2seq to extract the spatial features of the numerical forecast grid data. Compared with the seq2seq model, the accuracy is improved by about 2%, which shows that the model can better reflect the spatial connection of meteorological predictors. The DeepSTF model proposed in this paper integrates the above models, adopts the seq2seq structure, introduces an attention mechanism, and establishes a CNN model to extract spatial features. The prediction accuracy of DeepSTF has reached 70% on the temperature prediction problem, which is about 13% higher than ECMWF-IFS, which is the best performance among all models.

To fully evaluate the results, Fig. 4 shows the variation trend of RMSE and ACC of 0-72 h temperature forecasting. It can be clearly seen from the figure that the accu-



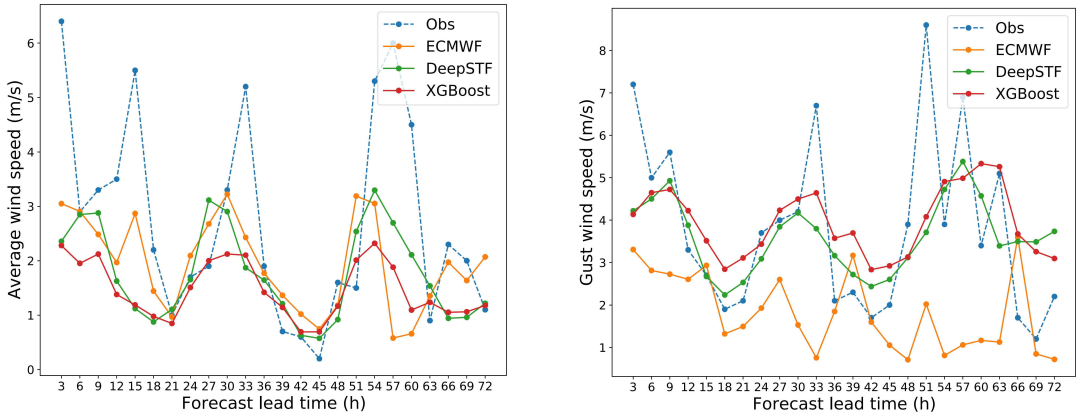
(a) Temperature (b) Relative humidity

(c) Average wind speed (d) Gust wind speed

Figure 5: Comparison of the predicted value and the observation value. The prediction results of the DeepSTF model are closer to the true values in each meteorological factor than other models.

racy of all models gradually decreases with the increase of the lead time. Except this, the post-processing algorithms tested in this article have made significant corrections to the ECMWF-IFS prediction results. Especially the AttnSeq2seq model with the attention mechanism and the CNNseq2seq model with the spatial feature extraction, both have significantly improved the predictive ability, which achieves 80% accuracy in the 3-hour forecast. After combining the advantages of these two models, the DeepSTF model has achieved better prediction results than others. The accuracy of the 3-hour forecast exceeds 80%, even in the 72-hour forecast, the accuracy rate exceeds 65%. Compared with other post-processing algorithms, the DeepSTF model significantly improves forecast accuracy.

The above comparative analysis of different model structures illustrates the rationality of the DeepSTF model structure. Then we use DeepSTF for the prediction of other meteorological factors on the test set. Fig. 5 shows the variation trend between the real



(a) Average wind speed (b) Gust wind speed

Figure 6: Cases with a poor predict performance. At the point where the wind speed changes suddenly, the model prediction result is not accurate.

value and predicted value of each meteorological factor in the forecast period. It can be seen that both the DeepSTF model and the XGBoost model can effectively correct the ECMWF-IFS data, and DeepSTF is closer to the true value than XGBoost. This indicates that the deep neural network with a more complicated structure can extract the temporal and spatial characteristics better than the traditional machine learning model, and make more accurate predictions. Furthermore, we also analyze some cases where the prediction effect is unsatisfying. We find that when the observed value changes rapidly, the model cannot achieve an accurate prediction effect. As shown in Fig. 6, this situation is more obvious in the prediction of average wind speed and gust wind speed. The reason is that the training data does not contain many extremely rapid changing values, which makes our model more likely to predict the meteorological factors in a relatively smooth trend.

4.3 Comparison of multi-site and single-site forecast

Since there are 226 stations that need to be forecasted in Beijing, there are many drawbacks to establishing a prediction model for each station, such as more repetitive work and inconvenient storage. Therefore, in view of the shortcomings of the traditional single-site forecasting model, a new multi-site forecasting model named DeepSTF is established in this paper. For each meteorological factor, the number of models is reduced from the original hundreds to only one, which greatly reduces the cost of training and storage. To analyze the advantages of the multi-site forecasting model, we compare the prediction results of the multi-site model with the single-site model. For the single-site prediction model, we selected the univariate MOS algorithm [15] and the MOML model [26, 42] based on machine learning, both of which have achieved good results in

Table 5: Comparison of accuracy between multi-site and single-site forecasting.

Predictors	ECMWF-IFS	Single-site		Multi-site	
		MOS	MOML	XGBoost	DeepSTF
Temperature	56.72%	67.06%	69.52%	63.37%	**70.03%**
Relative humidity	47.46%	60.41%	68.87%	60.81%	**70.34%**
Average wind speed	73.77%	83.14%	83.80%	83.31%	**84.44%**
Gust wind speed	64.46%	74.03%	76.18%	75.17%	**77.05%**

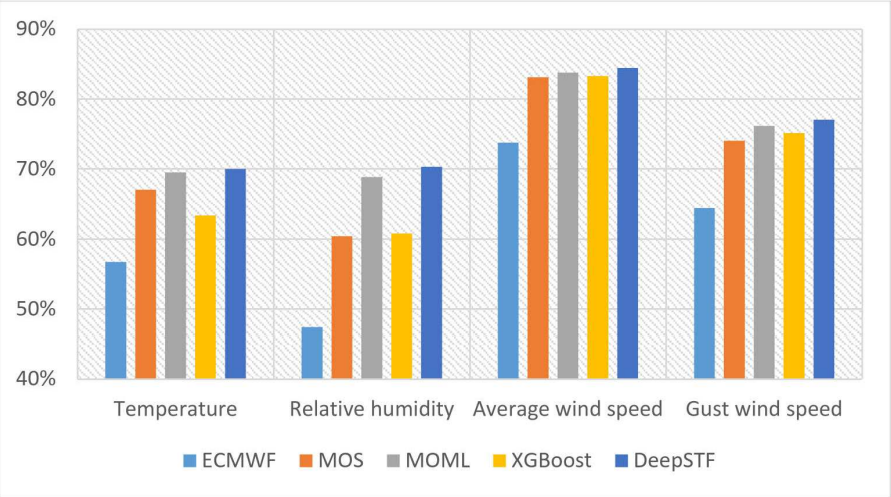


Figure 7: Comparison of prediction accuracy between different models. Among the four predicted meteorological factors, the DeepSTF model has achieved the best accuracy.

weather forecasting. We establish MOS and MOML models for each station separately. The inputs include the historical observational data of each station and the NWP data of the surrounding grid points, and the outputs are the predicted values of the meteorological factors of the station for the next 72 hours.

The comparison of prediction results between multi-site and single-site forecasting on the test set is shown in Table 5 and Fig. 7. It can be seen that the XGBoost model for multi-site forecasting has a lower temperature accuracy than MOS, while the accuracy of other variables has been slightly improved. However, compared with the MOML model, the prediction accuracy of XGBoost is worse. This indicates that multi-site forecasting requires higher model complexity due to the integration of meteorological conditions of different terrains, and it is necessary to build a deeper model to extract spatial features better.

The multi-site forecasting model DeepSTF based on deep learning spatio-temporal modeling has deeply extracted the temporal and spatial features, and the accuracy of each factor exceeds the single-site forecasting model MOML. Therefore, the deep neural

network model that combines time and space features has a greater advantage in extracting space features than the single-site model, which fully reflects the relationship between meteorological factors in the time dimension and space dimension. It reduces the number of models and improves the accuracy of prediction at the same time.

4.4 Confidence interval estimation of the forecast accuracy

In order to evaluate the uncertainty of the model parameters, we use the bootstrap method to construct random samples on the test set, yielding the distribution of the prediction accuracy and obtaining the confidence intervals. Specifically, we sample the test data with replacement, obtaining 20,000 pieces of data for each re-sampling. Then we calculate the prediction accuracy of the re-sampling. After repeating this process 1000 times, we obtain the distribution of prediction accuracy.

Tables 6-9 show the accuracy distribution and 95% confidence interval estimation of temperature, relative humidity, average wind speed and gust wind speed respectively. We can see that the multi-site post-processing algorithm proposed in this paper has strong prediction stability and robustness. Especially the DeepSTF model, which can significantly improve the prediction accuracy compared to ECMWF-IFS and the baseline model XGBoost.

Table 6: Accuracy distribution of the temperature forecasting.

Model	Average	Standard deviation	95% confidence interval
ECMWF-IFS	56.05%	0.0014	[55.79%,57.36%]
XGBoost	63.40%	0.0012	[63.19%,63.65%]
DeepSTF	70.01%	0.0012	[69.71%,71.20%]

Table 7: Accuracy distribution of the relative humidity forecasting.

Model	Average	Standard deviation	95% confidence interval
ECMWF-IFS	47.25%	0.0014	[47.04%,47.60%]
XGBoost	60.80%	0.0013	[60.55%,61.10%]
DeepSTF	70.32%	0.0010	[70.13%,70.57%]

Table 8: Accuracy distribution of the average wind speed forecasting.

Model	Average	Standard deviation	95% confidence interval
ECMWF-IFS	73.82%	0.0007	[73.68%,73.97%]
XGBoost	83.30%	0.0005	[83.20%,83.42%]
DeepSTF	84.44%	0.0005	[84.33%,84.55%]

Table 9: Accuracy distribution of the gust wind speed forecasting.

Model	Average	Standard deviation	95% confidence interval
ECMWF-IFS	64.24%	0.0010	[64.05%,64.49%]
XGBoost	75.17%	0.0006	[75.06%,75.29%]
DeepSTF	77.08%	0.0005	[76.93%,77.14%]

4.5 Model evaluation on different months and stations

Since the weather forecast has differences and continuity in time and space, in this section we compare the forecast accuracy at different months and different stations to evaluate the adaptability of the model, and we use temperature prediction as an example for detailed analysis.

Fig. 8 shows the accuracy of temperature forecasts for different months. We can see that XGBoost and DeepSTF both exceed the ECMWF-IFS in every month, and the DeepSTF model has the best predicted performance. From the perspective of different months, the prediction accuracy of the DeepSTF model is relatively stable than ECMWF-IFS. Especially from 2016-11 to 2017-03, the accuracy has been maintained above 70%, which indicates that the model has good adaptability and is less affected by seasonal changes. Fig. 9 shows the heat map of the forecast accuracy of each station in Beijing. It can be seen that ECMWF-IFS has a higher forecast accuracy in the southeast of Beijing, while the mountainous area of the northwest is seriously affected by the terrain, resulting in low accuracy. Both the XGBoost model and the DeepSTF model have a significant im-

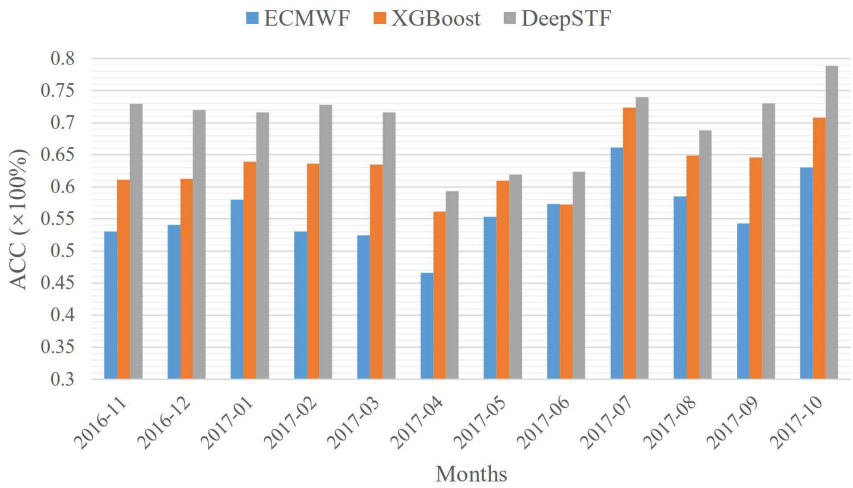


Figure 8: Accuracy of temperature forecast in different months. The DeepSTF model outperforms ECMWF-IFS and XGBoost in each month and can maintain high accuracy under seasonal changes.

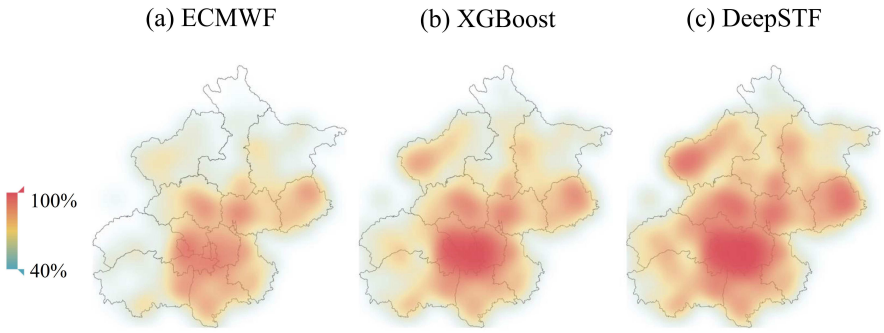


Figure 9: Accuracy distribution of temperature forecast in Beijing. The darker the color, the higher the forecast accuracy of the station. The DeepSTF model improves the forecasts compared to ECMWF-IFS and XGBoost at different stations, especially in the mountains stations.

provement over the ECMWF-IFS, especially the DeepSTF model, which can still maintain high accuracy in mountainous areas. Therefore, the multi-site prediction model DeepSTF proposed in this paper has better adaptability in time and space than ECMWF-IFS and maintains a higher accuracy rate at the same time.

5 Conclusions and future work

In this paper, a deep neural network DeepSTF based on spatio-temporal modeling is constructed for multi-site weather forecasting. In our proposed framework, we model the spatio-temporal information by an encoder-decoder structure with an attention mechanism and a CNN module. The novelty of our work mainly includes three points: first, we realize the forecasting of multiple stations by using one framework. This framework can better integrate spatial features and has good robustness, which is suitable for the overall prediction of a large number of stations. Second, the DeepSTF model combines the seq2seq structure, the attention mechanism, and the CNN module, and adds the geographic coordinates of the station, which fully reflects the relationship of meteorological factors between time and space. Third, compared with ECMWF-IFS, our model achieves a more refined short-term forecast, increasing the 2 forecast periods per day of ECMWF-IFS to 8 per day, which is more meaningful for the practical application of weather forecasting.

To verify the prediction ability of the model, we make short-term forecasts for the next three days at 226 stations in Beijing. The meteorological factors include temperature, relative humidity, average wind speed, and gust wind speed. We first use temperature prediction as an example to conduct a full comparison experiment on different model structures, to explore the influence of the seq2seq structure, attention mechanism, and the CNN module on the prediction results. Experimental results show that a network that contains all these modules outperforms a network that does not, which illustrates the ra-

tionality of the structure of the DeepSTF model. In the comparison experiment with the conventional single-site forecast, the DeepSTF model shows performance improvements compared to other baselines, including the MOML based on machine learning and the univariate MOS. It can be seen that the DeepSTF model has a greater advantage in the extraction of the spatial and temporal features of meteorological predictors. Compared with the traditional single-site forecast model, it not only greatly reduces the number of models, but also improves the accuracy of the prediction. Finally, we evaluate the impact of seasonal changes and topographical differences on the accuracy of the model predictions. The results show that the accuracy of the DeepSTF model is relatively more stable between different months, and it can maintain high accuracy in mountainous regions with complex terrain. This indicates that our proposed model is more accurate and robust even under seasonal changes and terrain differences.

In summary, our framework outperforms existing baselines in the post-processing of numerical weather prediction. In the practical application of weather forecasting, it is more suitable for the aggregate prediction of a large number of stations. In the future, we will further optimize the network structure to extract the spatio-temporal features more deeply. Moreover, we will apply the model to predict more meteorological factors, especially extreme weather such as typhoons and rainstorms, which will be more challenging in forecasting.

Acknowledgments

The authors would like to express sincere gratitude to Lizhi WANG, Lve WU and Xiao LOU for unpublished data; and Hanqiuzi WEN, Chongping JI and Yingxin ZHANG for professional guidance. This work is supported by the National Key Research and Development Program of China (Grant Nos. 2017YFC0209804 and 2018YFF0300104), Beijing Academy of Artificial Intelligence (BAAI), the National Natural Science Foundation of China (Grant No. 11421101) and the Open Research Fund of Shenzhen Research Institute of Big Data (Grant No. 2019ORF01001).

References

[1] K. Ahmed, S. Shahid, N. Nawaz, and N. Khan. Modeling climate change impacts on precipitation in arid regions of Pakistan: A non-local model output statistics downscaling approach. *Theoretical and Applied Climatology*, 137(1-2):1347–1364, 2019.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] V. Bjerknes. Das problem der wettervorhers-age, betrachtet vom standpunkte der mechanik und der physik. *Meteor. Z.*, 21:1–7, 1904.

[4] J. Booz, W. Yu, G. Xu, D. Griffith, and N. Golmie. A deep learning-based weather forecast system for data volume and recency analysis. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pages 697–701. IEEE, 2019.

[5] A. Carpinone, R. Langella, A. Testa, and M. Giorgio. Very short-term probabilistic wind power forecasting based on Markov chain models. In *2010 IEEE 11th International Conference on Probabilistic Methods Applied to Power Systems*, pages 107–112. IEEE, 2010.
[6] J. G. Charney, R. Fjörtoft, and J. Von Neumann. Numerical integration of the barotropic vorticity equation. In *The Atmosphere–A Challenge*, pages 267–284. Springer, 1990.
[7] A. Chattopadhyay, P. Hassanzadeh, and D. Subramanian. Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: Reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Processes in Geophysics*, 27(3):373–389, 2020.
[8] A. Chattopadhyay, E. Nabizadeh, and P. Hassanzadeh. Analog forecasting of extreme-causing weather patterns using deep learning. *Journal of Advances in Modeling Earth Systems*, 12(2):e2019MS001958, 2020.
[9] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.
[10] W. Y. Y. Cheng and W. J. Steenburgh. Strengths and weaknesses of MOS, running-mean bias removal, and Kalman filter techniques for improving model forecasts over the western United States. *Weather and Forecasting*, 22(6):1304–1318, 2007.
[11] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
[12] L. Delle Monache, F. A. Eckel, D. L. Rife, B. Nagarajan, and K. Searight. Probabilistic weather prediction with an analog ensemble. *Monthly Weather Review*, 141(10):3498–3516, 2013.
[13] L. Delle Monache, T. Nipen, Y. Liu, G. Roux, and R. Stull. Kalman filter and analog schemes to postprocess numerical weather predictions. *Monthly Weather Review*, 139(11):3554–3570, 2011.
[14] M. Diagne, M. David, J. Boland, N. Schmutz, and P. Lauret. Post-processing of solar irradiance forecasts from WRF model at Reunion Island. *Solar Energy*, 105:99–108, 2014.
[15] P. Friederichs and H. Paeth. Seasonal prediction of African precipitation with ECHAM4–T42 ensemble simulations using a multivariate MOS re-calibration scheme. *Climate Dynamics*, 27(7-8):761–786, 2006.
[16] C. Gao and J. Zeng. Validation of wind forecast based on MOS method at the Ningde coastal region. *Marine Forecasts*, (04):17–24, 2018.
[17] A. Ghaderi, B. M. Sanandaji, and F. Ghaderi. Deep forecast: Deep learning-based spatio-temporal forecasting. *arXiv preprint arXiv:1707.08110*, 2017.
[18] B. Glahn, M. Peroutka, J. Wiedenfeld, J. Wagner, G. Zylstra, B. Schuknecht, and B. Jackson. MOS uncertainty estimates in an ensemble framework. *Monthly Weather Review*, 137(1):246–268, 2009.
[19] H. R. Glahn and D. A. Lowry. The use of model output statistics (MOS) in objective weather forecasting. *Journal of Applied Meteorology*, 11(8):1203–1211, 1972.
[20] P. Grönquist, C. Yao, T. Ben-Nun, N. Dryden, P. Dueben, S. Li, and T. Hoefler. Deep learning for post-processing ensemble weather forecasts. *arXiv preprint arXiv:2005.08748*, 2020.
[21] K. A. Hart, W. J. Steenburgh, D. J. Onton, and A. J. Siffert. An evaluation of mesoscale-model-based model output statistics (MOS) during the 2002 Olympic and Paralympic winter games. *Weather and Forecasting*, 19(2):200–218, 2004.
[22] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[23] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015.
[24] Z. Karevan and J. A. K. Suykens. Spatio-temporal stacked LSTM for temperature prediction in weather forecasting. *arXiv preprint arXiv:1811.06341*, 2018.
[25] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
[26] H. Li, C. Yu, J. Xia, Y. Wang, J. Zhu, and P. Zhang. A model output machine learning method for grid temperature forecasts in the Beijing area. *Advances in Atmospheric Sciences*, 36(10):1156–1170, 2019.
[27] I. Loshchilov and F. Hutter. Fixing weight decay regularization in Adam. 2018.
[28] A. Pelosi, H. Medina, J. Van den Bergh, S. Vannitsem, and G. B. Chirico. Adaptive Kalman filtering for postprocessing ensemble numerical weather predictions. *Monthly Weather Review*, 145(12):4837–4854, 2017.
[29] X. Peng, Y. Che, and J. Chang. A novel approach to improve numerical weather prediction skills by using anomaly integration and historical data. *Journal of Geophysical Research: Atmospheres*, 118(16):8814–8826, 2013.
[30] S. Rasp and S. Lerch. Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, 146(11):3885–3900, 2018.
[31] X. Ren, L. Li, Y. Yu, Z. Xiong, S. Yang, W. Du, and M. Ren. A simplified climate change model and extreme weather model based on a machine learning method. *Symmetry*, 12(1):139, 2020.
[32] L. F. Richardson. *Weather Prediction by Numerical Process*. Cambridge University Press, 2007.
[33] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810, 2015.
[34] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *Advances in Neural Information Processing Systems*, pages 5617–5627, 2017.
[35] X. Shi and D.-Y. Yeung. Machine learning for spatiotemporal sequence forecasting: A survey. *arXiv preprint arXiv:1808.06865*, 2018.
[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[37] Z. Song, Y. Jiang, and Z. Zhang. Short-term wind speed forecasting with Markov-switching model. *Applied Energy*, 130:103–112, 2014.
[38] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.
[39] B. A. Veenhuis. Spread calibration of ensemble MOS forecasts. *Monthly Weather Review*, 141(7):2467–2482, 2013.
[40] B. Wang, J. Lu, Z. Yan, H. Luo, T. Li, Y. Zheng, and G. Zhang. Deep uncertainty quantification: A machine learning approach for weather forecasting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2087–2095, 2019.
[41] J. A. Weyn, D. R. Durran, and R. Caruana. Can machines learn to predict weather? Using deep learning to predict gridded 500-hPa geopotential height from historical weather data. *Journal of Advances in Modeling Earth Systems*, 11(8):2680–2693, 2019.
[42] J. Xia, H. Li, Y. Kang, C. Yu, L. Ji, L. Wu, X. Lou, G. Zhu, Z. Wang, Z. Yan et al. Machine learning-based weather support for the 2022 Winter Olympics, 2020.
[43] D. Yang. On post-processing day-ahead NWP forecasts using Kalman filtering. *Solar Energy*,

182:179–181, 2019.

[44] D. Yang. Ensemble model output statistics as a probabilistic site-adaptation tool for satellite-derived and reanalysis solar irradiance. *Journal of Renewable and Sustainable Energy*, 12(1):016102, 2020.

[45] K. Yonekura, H. Hattori, and T. Suzuki. Short-term local weather forecast using dense weather station by deep neural network. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1683–1690. IEEE, 2018.

[46] C. Yu. A data-driven random subfeature ensemble learning algorithm for weather forecasting. *Communications in Computational Physics*, 28(4):1305–1320, 2020.

[47] M. A. Zaytar and C. El Amrani. Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. *International Journal of Computer Applications*, 143(11):7–11, 2016.

[48] H. Zheng and Y. Wu. A XGBoost model with weather similarity analysis and feature engineering for short-term wind power forecasting. *Applied Sciences*, 9(15):3019, 2019.