

## RESEARCH ARTICLE

# Evaluation of Different Deep Learning Methods for Meteorological Element Forecasting

RUIBO QIU<sup>1</sup>, WEN DAI<sup>1,2,3,4</sup>, GUOJIE WANG<sup>1,2,3,4</sup>, ZICONG LUO<sup>1</sup>, AND MENGQI LI<sup>5</sup><sup>1</sup>School of Geographical Sciences, Nanjing University of Information Science and Technology, Nanjing 210044, China<sup>2</sup>Key Laboratory of Meteorological Disaster, Ministry of Education (KLME), Nanjing University of Information Science and Technology, Nanjing 210044, China<sup>3</sup>Joint International Research Laboratory of Climate and Environment Change (ILCEC), Nanjing University of Information Science and Technology, Nanjing 210044, China<sup>4</sup>Collaborative Innovation Center on Forecast and Evaluation of Meteorological Disasters (CIC-FEMD), Nanjing University of Information Science and Technology, Nanjing 210044, China<sup>5</sup>Department of Geography, University of Zurich, 8057 Zürich, Switzerland

Corresponding author: Wen Dai (wen.dai@nuist.edu.cn)


This work was supported in part by the National Natural Science Foundation of China under Grant 42301478 and Grant 42275028, and in part by the Natural Science Foundation of Jiangsu Higher Education Institutions of China under Grant 22KJB170016.

**ABSTRACT** Deep Learning (DL) models can make short- and long-term predictions in just a few seconds, beyond the capabilities of traditional physical models. However, the capabilities of different DL models for meteorological element forecasting, are still yet to be comprehensively evaluated. Here, DL models were used to forecast multiple meteorological elements, including temperature (T), surface net solar radiation (SSR), soil moisture (SM), and evapotranspiration (ET). We compared the seven models in term of training performance, prediction accuracy, and the effects of parameters. We found that the training of RNN-based models (LSTM, GRU, and Bi-LSTM) was faster than others. However, with sufficient training epochs, Transformer-based models consistently achieve the lowest loss function. Among the Transformers, the Informer demonstrates the best prediction accuracy in most scenarios. Beyond the choice of DL model, the prediction performance is also influenced by the meteorological element itself. The MTGNN is comparable to Transformer-based models for T and SSR forecasting, but it does not perform as well as the Informer for SM and ET. The sliding window size and prediction time step have a slight impact on the performance differences between the models. The results can offer insights into applying DL models in meteorological element forecasting.

**INDEX TERMS** Meteorological elements forecasting, deep learning, RNN, GNN, transformer.

## I. INTRODUCTION

Weather forecasting is a fundamental part of people's lives worldwide as the global weather pattern is constantly changing, and how it behaves in the 21st century is unpredictable and complex [1], [2]. The abnormal change in the weather pattern reflects drought, floods, typhoons, and several other natural disasters worldwide, posing unprecedented damage to nature and society [3], [4]. Hence, the timely and accurate prediction of various meteorological elements, including temperature, radiation, and humidity, is essential for multiple applications that benefit human lives [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Geng-Ming Jiang .

Weather forecasting is the scientific process of predicting atmospheric conditions for specific time periods and locations. Traditional weather forecasting usually depends on numerical simulation methods. Numerical simulation methods perform parameterized mathematical modeling of nonlinear weather system processes, and then simulate complex physical processes in the weather system to forecast various meteorological elements [6]. In addition, with the development of computer technology and detection technology, weather radar and satellite data are further assimilated into numerical models to improve the accuracy of weather forecasts [7], [8], [9]. Numerical simulation methods are now extensively applied in both global and regional weather forecasting [10], [11], [12]. However, the inherent

uncertainty in modeling the parameterization of nonlinear weather systems contributes to ongoing uncertainty in local-scale predictions [13]. Recently, machine learning methods have been adapted to forecasting meteorological elements. Compared with statistical models, traditional machine learning methods such as shallow neural network [14], [15], Bayesian statistics [16], [17], and support vector machine [18], [19] are increasingly effective in dealing with issues related to non-linear relationships. However, for complex problems, the generalization of their use and abilities are restricted [20].

Deep Learning (DL) models are superior to traditional machine learning methods for complex issues and extensive data analysis [21]. DL models are usually data-driven without having prior physical knowledge. They are powerful in capturing non-linear relationships among forecasting elements and thus are being widely adopted [22]. Recurrent neural networks (RNNs) within DL frameworks are commonly employed for researching time series data predictions, including meteorological elements. For example, Qing and Niu predicted hourly day-ahead solar irradiance based on LSTM networks [23], whilst Shi et al. utilized the radar echo map to predict the probability of rainfall based on the GRU network [24]. RNNs have revolutionized environmental forecasting and resolved problems associated with considerable management complexity in various fields, including water resources, agriculture, and soil sciences [25], [26]. However, the input time series continues to increase and the model continues to iterate, the RNN models meet problems such as gradient explosion and vanishing [27]. Thus, Graph Neural Networks (GNNs) have been impressive in the task of time series prediction [28], [29]. A graph is a special data form describing the relationship between different entities. Multivariate time series prediction can be viewed from the perspective of a graph [30]. The variables in a multivariate time series can be regarded as nodes in the graph, and the edges of connecting nodes represent the relationship between variables. GNNs can capture directional relationships efficiently [31].

While the RNNs and GNNs is widely used in meteorological element forecasting over the past decade, the transformers, created in 2017, broke the monopoly of the RNN and CNN. The network structure of the transformers is composed of a self-attention module and a feed-forward layer [32]. Although the transformers were initially designed for natural language processing tasks in humans, many studies have now applied them to cross-border tasks such as time series prediction, music generation, image classification, etc., [33] and [34]. Transformers allow the creation of a long enough look-back window. When the computational force is sufficient, it can theoretically capture infinite rich sequence context, which leverages transformers in modeling long-term dependencies, realizing a more powerful large model [35], [36].

Various deep learning methods, including RNNs, GNNs, and Transformers, exist within a single framework that can

incorporate multiple models. RNN-based models, such as LSTM [37], GRU [38], Bi-LSTM [39], GNN-based models, such as MTGNN [30], GPT-GNN [40], DAG-GNN [41], and Transformer-based models, such as Deep transformer [42], Informer [43] and Autoformer [44], have been widely used in meteorological element forecasting. However, comprehensive comparisons of the capabilities of different DL models for meteorological element forecasting have yet to be further conducted. Accordingly, this study aims to comprehensively evaluate the performance of different DL models for forecasting multiple meteorological elements, including temperature, surface net solar radiation, soil moisture and evapotranspiration, across dimensions of training performance, prediction accuracy, and the effects of parameters.

## II. DATASETS

### A. STUDY AREA

To fully consider the interaction between meteorological elements, we choose a farmland area in Cangzhou City, Hebei Province, China (shown in the hashed rectangle in Fig. 1) as the study area. This farmland represents a typical transitional zone in China, with area of 250 km<sup>2</sup>, featuring both dry and wet climates [45]. The location is highly susceptible to climate fluctuations where drought and flood disasters occurs, characterized by a strong gradient of climate variables and instability [46], [47]. Due to the microclimatic influences, the meteorological elements change dramatically locally. The changes in microclimate meteorological elements are closely related to the radiation transfer and heat balance in each canopy layer [48]. In addition, human activities, such as

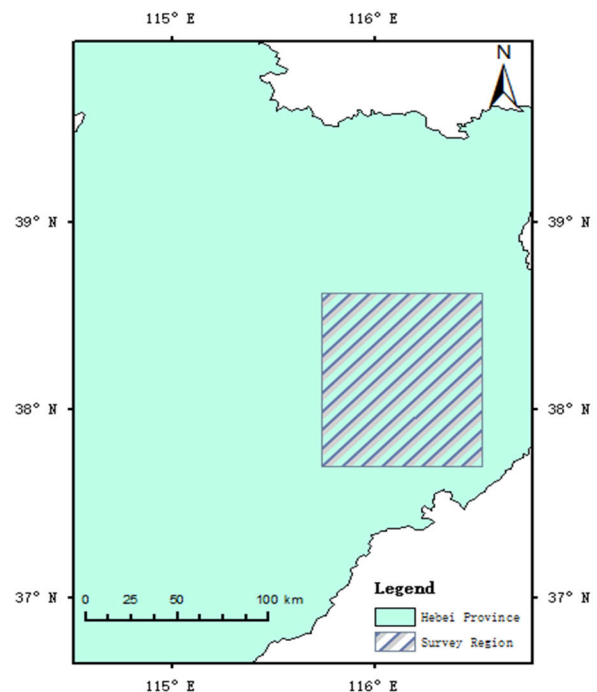


FIGURE 1. The geographical location of the study area.

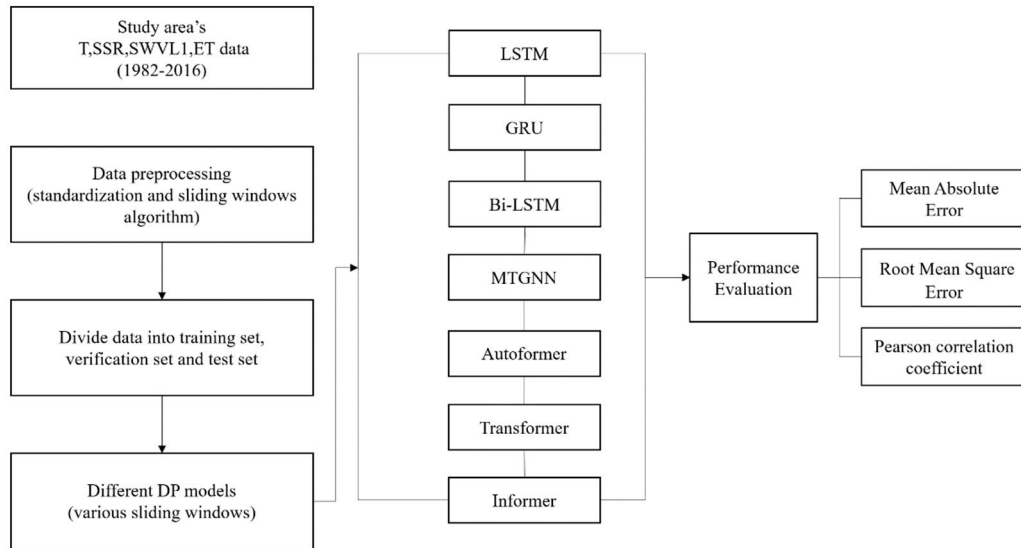


FIGURE 2. Workflow of the proposed method.

irrigation, sowing, and tillage, constantly change the state of solar radiation near the surface, soil temperature, and humidity [49]. These factors contribute to the relatively unstable meteorological changes in this region, which are crucial for assisting in model simulation and understanding the interrelationships among the meteorological elements.

### B. ERA-5 DATASET

We select the 2m air temperature (T), surface latent heat flux, surface net solar radiation (SSR), and volumetric soil water layer 1 (SM) data from the ERA5 hourly product, spanning from 1982 to 2016, at a spatial resolution of  $0.25^\circ$ . The ERA5 provides reanalyzed observations of the atmosphere, land, and ocean, which results from the best combination of observations from different measurement sources and the output of a numerical model using a Bayesian estimation process called data assimilation [50]. Note that we converted latent heat flux into evapotranspiration (ET) values using the guidelines from FAO56 documentation, as detailed in Table 1.

The main reasons for choosing the ERA5 product are: (1) it is a comprehensive set of grid data from 1950 to the near present, which can provide sufficient input data for our model; (2) ERA5 data has been used in the area of North China [51], [52], boasting both high temporal and spatial resolution. We trained each model with ERA5 data and evaluated the accuracy of each model on the test dataset with evaluation metrics described in section III-C.

## III. METHODOLOGY

### A. OVERVIEW

Fig. 2 shows the workflow of this paper. First, the T, SSR, SM, and ET data of the study area were extracted for the years 1982 to 2016. Then, these meteorological data were

TABLE 1. Meteorological weather forecast parameters used in our work.

Variables	Unit
2m temperature (T)	$^\circ\text{C}$
Surface net solar radiation (SSR)	$\text{J}/\text{m}^2$
Volumetric soil water layer 1 (SM)	$\text{m}^3/\text{m}^3$
Surface latent heat flux	$\text{J}/\text{m}^3$
Evapotranspiration (ET)	mm

pre-processed by data standardization and sliding windows algorithm. Third, different DL models were trained with various sliding windows for the meteorological elements forecasting. Finally, the performance of different DL models was compared using accuracy metrics, such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Pearson correlation coefficient (R).

### B. DATA PRE-PROCESSING

Before the training, the ERA5 data were standardized by z-score to eliminate the differences in measurement units and value ranges for each meteorological element. The maximum and minimum values of the same meteorological element may also differ by several orders of magnitude [53].

Then, the times-series dataset was generally divided into multiple samples of the same size for training using the sliding window algorithm. Previous studies [54], [55] have showed the effectiveness of the sliding window algorithm in deep learning prediction. Given a p-step historical time series data  $X = \{\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \dots, \mathbf{x}_{t+p}\}$ , where  $\mathbf{x}_t \in \mathbf{R}^N$ , are the values of N variables selected at time

step  $t$ , our goal is to predict a future time series  $Y = \{x_{t+p+1}, x_{t+p+2}, x_{t+p+3}, \dots, x_{t+p+q}\}$ , using the DL models. The sliding window size ( $p$ ) affects the sample size and, subsequently, the training of the models. Here, we set the sliding window sizes to 24, 48, and 72 h, respectively, to explore how different settings affect the training performance of various models. It should also be noted that the sliding window slides forward one step at a time, and the model will output future continuous prediction values after each sliding operation. For example, if the model output is set to 24 steps, there will be 24 continuous prediction values for each sliding. Since our model is based on historical time series, the predicted sequence's real value is known, and each prediction step's error can be calculated.

Meteorological forecasting is generally based on meteorological station, namely, point-based. Here we used the point model after averaging the meteorological element value across the study area for the time series prediction. The entire time series dataset was divided into three parts: 70% allocated for training, 10% for validation, and the remaining 20% for independent testing. The training and validation sets are used to train and update model hyper-parameters and the test set is used for the final performance evaluation of the models.

### C. THE DEEP LEARNING MODELS

This study compared seven deep learning models, encompassing three RNN-based models, one GNN-based model, and three Transformer-based models.

#### 1) THE RNN -BASED MODELS

Typical sequence models, RNN-based models, utilize a sequence-to-sequence structure to process time series data as sequences of inputs and outputs [56], [57], [58]. Known for their proficiency in handling time series problems, RNNs effectively save and update state information due to their unique looping structure, which updates and saves the context state in each iteration [59].

We select three RNN-based models, LSTM, GRU and Bi-LSTM, as baseline models. RNNs can deal with time series problems [60], [61], but they are better at feature extraction rather than saving and updating information [62]. The LSTM [63] solves these limitations by adding gradient flows of the forgetting stage, memory selection stage and output stage to the hidden layer. The forgetting stage selectively loses the information transmitted from the previously hidden layer. The memory selection stage selectively remembers the input information of the current hidden layer. The output stage determines which will be regarded as the output of the current state. Compared with LSTM, GRU can achieve significant results, and it is easier to train; this can greatly improve the training efficiency [64]. Bi-LSTM combines an LSTM moving from the beginning of the sequence and an LSTM moving from the end of the sequence to the beginning of the sequence, so that the feature obtained at time  $t$  has both past and future information.

#### 2) THE GNN-BASED MODELS

MTGNN is the first model to process multivariate time series data using a graph [30]. The MTGNN consists of three main modules: graph learning layer, time convolution module and graph convolution module. The graph learning layer adaptively learns the graph adjacency matrix as the input of the graph convolution module and is specially designed for data without an explicit graph adjacency matrix. The graph learning layer in MTGNN is designed to extract one-way relationships and only focuses on the relationship between paired variables. It is helpful to simulate the driving relationship between meteorological elements. Multiple one-dimensional convolution filters in the time convolution module allow the model to capture different periodic signals through the size of the receptive fields.

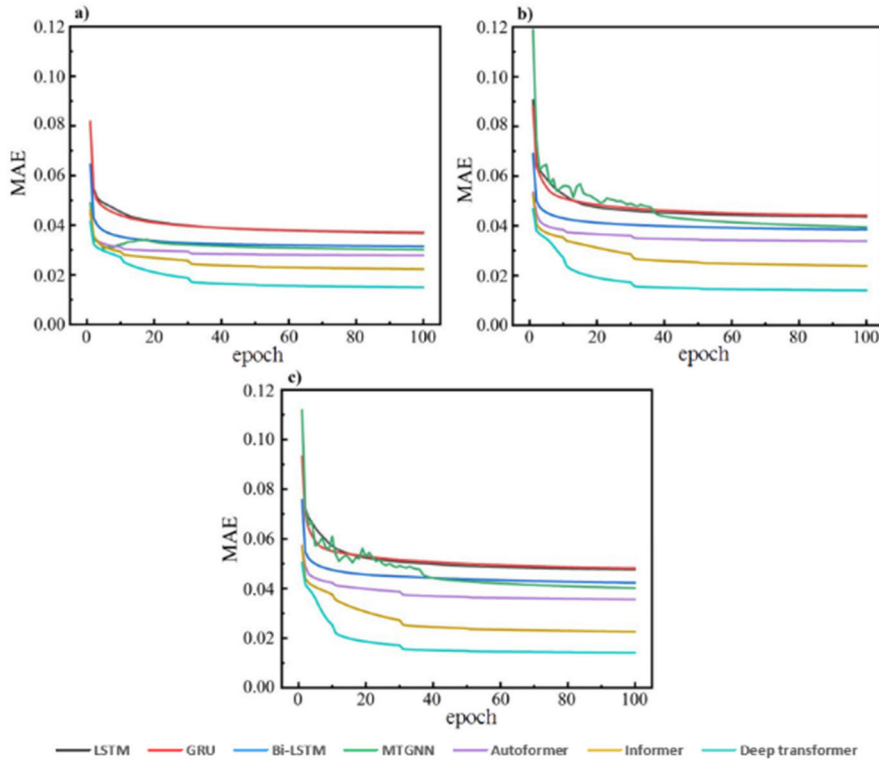
#### 3) TRANSFORMER-BASED MODELS

The Transformers are an encoder-decoder model based on the self-attention mechanism. It is suitable for parallel computing and the complexity of its own model. When the encoder processes each time point in the sequence, it determines how much the value of the time point is affected by itself or other time point values by calculating the attention score to obtain a weighting vector. The decoder gradually restores the low-dimensional representation of the target sequence obtained by the encoder to the target sequence. The Informer [43] and Autoformer [44] have been derived from the traditional Transformer. The Informer breaks the inherent limitations of conventional transformers, such as secondary time complexity, high memory utilization, and encoder-decoder architecture; it has significant performance in dealing with long-term dependence. The Autoformer can progressively decompose complex time series, focusing on seasonal pattern modeling [44]. The deep transformer, Informer, and Autoformer were selected for the comparison.

We standardized the training parameters across all models for the model training, utilizing ten grids and three sliding window settings. In the configuration, we set the epochs to 100 and the batch size to 64. The optimizer of each model is the Adam optimizer. Each experiment was conducted thrice to obtain an average value. We assess each model by the decline of the loss curve in the training period and the prediction error accumulation in the test period. All the experiments were performed on an NVIDIA Tesla T4 GPU.

### D. PERFORMANCE EVALUATION

The Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Pearson correlation coefficient (R) (significance level at  $P < 0.05$ ), are selected to evaluate model performances. The MAE indicates the average value of the distance between the model's predicted value and the true value. The RMSE measures the deviation between the predicted value and the true value, which is sensitive to large or small errors. The R is used to measure the linear correlation between the predicted and true values; a larger absolute value



**FIGURE 3.** The training loss of each model under three different sliding window settings (24h (a), 48h (b) and 72h (c)). Each curve represents the average value of ten grids. MAE is selected as the loss function in the training.

of R signifies a stronger correlation. The formulas of the metrics are shown below.

$$MAE(y, y') = \frac{1}{n} \sum_{i=1}^n |y_i - y'_i| \tag{1}$$

$$RMSE(y, y') = \sqrt{\frac{1}{n} \sum_{i=1}^n (y'_i - y_i)^2} \tag{2}$$

$$r(y, y') = \frac{Cov(y, y')}{\sqrt{Var[y] Var[y']}} \tag{3}$$

In the above equations  $y_i$  and  $y'_i$  represent the original ERA5 time series at time I and Its predicted value at the corresponding time.  $Cov(y, y')$  is the covariance between the predicted values and the true values.  $Var[y]$  and  $Var[y']$  is the variance of the true values and the predicted values. The lower the MAE and RMSE, the better; R is the inverse.

#### IV. RESULT

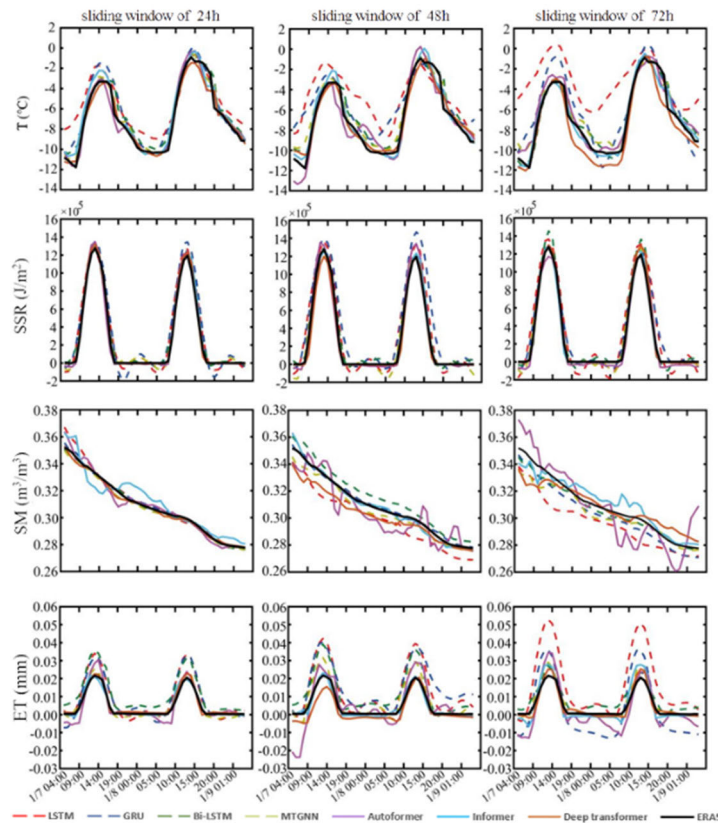
##### A. THE TRAINING PERFORMANCE

Fig. 3 shows the loss function curve of each model during training under three different sliding window settings. As the number of epochs increases, all models' MAE decreases quickly, then tends to be stable. However, the speed of stabilization varies across models. The RNN-based models always tend to be stable first, followed by the Transformers-based models and the MTGNN. However, the Transformer-based

models have the lowest MAE once they become stable. This shows that the RNN-based models have the highest efficiency in training. Yet, the Transformers-based models have the best performance in loss function in training.

The size of the sliding window has a minimal impact on the training performance of both RNN-based and Transformer-based models (Fig. 3). Regardless of whether the sliding window is set to 24, 48, or 72 hours, the RNN-based models typically stabilize after about 25 epochs. In contrast, the MTGNN usually stabilizes after 35 epochs, while the Transformers consistently stabilize after 30 epochs. Moreover, although the sizes of sliding windows are different, the transformers achieve the best training performance.

However, the training performance of the MTGNN model is notably affected by the sliding window size. The loss curve of MTGNN is different and fluctuates with different sliding window sizes (Fig. 3). This may be related to the complexity of the model. The model architecture of RNNs is relatively simple, and the loss curve decreases steadily. Compared with RNNs, the MTGNN has one more learning process of the graph adjacency matrix. The graph learning layer also updates the structure of the adjacency matrix with the introduction of data in the training stage. This leads to a slight fluctuation in the loss curve of the MTGNN at the beginning of training, but the fluctuation decreases, which is acceptable.



**FIGURE 4.** Prediction results on testing set at prediction step 1. The black line represents the real value of ERA5; lines of other colors represent prediction sequences of different models. The three columns from left to right represent the sliding window for 24h, 48h and 72h respectively.

### B. THE PREDICTION RESULTS OF DL MODEL

After model training, the best epoch of each method is saved and used to predict values of T, SSR, SM, and ET. Fig. 4 shows the prediction results of different models under different sliding window sizes. For each meteorological element, we can clearly see the impact of the change in sliding window size on the prediction. Specifically, with the increase of sliding window sizes, the prediction error of each model correspondingly increases. Among them, the prediction sequence of the Transformer-based models fits the real value more accurately (Fig. 4). Especially in the prediction of T, SSR and ET, when the predicted values of other models gradually deviate from the real values with the increase of sliding window sizes, the prediction sequence of the Transformer-based models are still consistent with the real value. Notably, in the prediction of SM, the variation in results of Transformer-based models is relatively larger, suggesting a potential dependency of model performance on the specific variable.

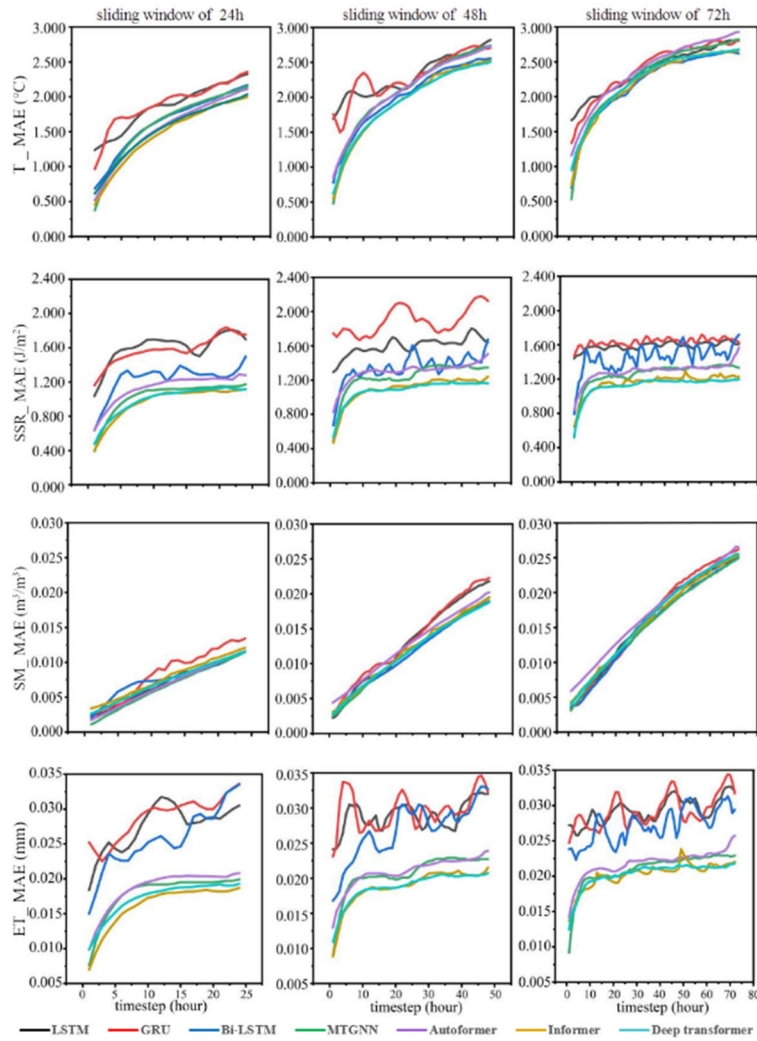
In addition, we found that the errors are mainly concentrated at the extreme values (peak and valley of the series), especially at the peak. All models are not good at predicting extreme values, which is also one of the challenges faced by the time series prediction task based on DL [65]. At the valley

values, the performance of each model is different. In the prediction of SSR and ET, RNN-based models and MTGNN perform poorly near 0, while transformers do the opposite. With the increase of sliding window, the error near 0 will increase slightly.

Overall, the Transformer-based models are better than MTGNN and RNN-based models. The MTGNN is comparable to Transformer-based models for T and SSR; the Informer and Transformer have smaller prediction errors for SM and ET, and MTGNN gradually surpasses them as the value increases. Among the three transformers, the Informer performs best overall.

### C. THE PREDICTIVE PERFORMANCE OF THE TRAINED MODEL

The test dataset is then used to evaluate the predictive performance of the trained model. In general, the error will increase gradually with the increase of the prediction time step in meteorological element forecasting. Here, the sliding window technique is employed to produce predictions for the next 24, 48, or 72 hours, corresponding to the sliding window sizes we set. Then, we evaluate the MAE of the predicted values in the different time steps.



**FIGURE 5.** Comparison of MAE changes of four variables on the test set under three different sliding window size settings (24h, 48h, 72h) of the models. The horizontal axes are the comparison of different sliding window settings under the same variable, and the vertical axes are the comparison of different variables under the same sliding window setting.

Fig. 5 shows the MAE of the predicted T, SSR, SM, and ET, with different prediction time steps and sliding windows. As expected, the MAE increases with the prediction time step in all models. Meanwhile, the prediction time step has a minor effect on the prediction performance of different models. For example, in the prediction of the T, the prediction error of the MTGNN in the first few steps is the lowest. After about 5h, the Informer and Transformer are the better performing models among all models.

The models have different performances in the four meteorological elements. Across the board, the Transformer-based models have the best performance, followed by MTGNN and the RNN-based models. This superior performance of the Transformer-based models is evident in SSR and ET prediction but becomes minor in T and SM prediction. This indicates that the effectiveness of a DL model may depend

on the specific meteorological element being forecasted, suggesting the need for tailored model selection for different elements.

The models with sliding windows of 24h have the lowest MAE, followed by the 48h and 72h windows. This trend aligns with expectations: smaller sliding windows yield better prediction performance. However, the sliding window has a minor effect on the prediction ability of the models. Although the sliding window changed from 24 to 72h, the trends of MAE are almost the same in all models. Notably, with the increase of the sliding window size, the gap between the models gradually narrowed in T and SM prediction, suggesting that the meteorological element itself could be substantial.

Given the effect of the prediction time step, short-term meteorological forecasting is more important and significant. Table 2. shows the comparison of RMSE and R of each model

TABLE 2. Metrics summary of each model in the 1st step under different sliding window settings.

Variables		T (°C)		SSR (J/m <sup>2</sup> )		SM (m <sup>3</sup> /m <sup>3</sup> )		ET (mm)	
Metric		RMSE	R	RMSE	R	RMSE	R	RMSE	R
Window_24	LSTM	1.6232	0.9919	2.3161	0.9654	0.0053	0.9973	0.0334	0.9677
	GRU	1.4135	0.9928	2.0785	0.9654	0.0060	0.9983	0.0369	0.9620
	Bi-LSTM	0.9180	0.9975	1.1017	0.9901	0.0051	0.9987	0.0307	0.9818
	MTGNN	<b>0.5582</b>	<b>0.9989</b>	0.9099	0.9928	<b>0.0036</b>	<b>0.9988</b>	0.0202	0.9926
	Autoformer	0.7722	0.9978	1.1901	0.9884	0.0047	0.9978	0.0192	0.9885
	Informer	0.6094	0.9986	<b>0.8916</b>	<b>0.9932</b>	0.0053	0.9979	<b>0.0147</b>	<b>0.9928</b>
	Transformer	0.7840	0.9984	1.0121	0.9928	0.0047	0.9980	0.0183	0.9916
Window_48	LSTM	1.9224	0.9862	2.2710	0.9537	0.0080	0.9963	0.0455	0.9453
	GRU	1.7143	0.9906	2.8200	0.9392	0.0073	0.9969	0.0510	0.9242
	Bi-LSTM	0.9618	0.9969	1.1368	0.9892	0.0064	0.9973	0.0239	0.9838
	MTGNN	<b>0.6768</b>	<b>0.9985</b>	1.0875	0.9902	0.0053	<b>0.9978</b>	0.0231	0.9910
	Autoformer	1.2505	0.9941	1.6293	0.9767	0.0087	0.9926	0.0245	0.9798
	Informer	0.7280	0.9983	<b>0.9807</b>	<b>0.9924</b>	0.0053	0.9974	<b>0.0169</b>	<b>0.9911</b>
	Transformer	0.8069	0.9976	1.1049	0.9908	<b>0.0050</b>	0.9975	0.0195	0.9897
Window_72	LSTM	2.1097	0.9838	2.5821	0.9399	0.0087	0.9938	0.0461	0.9334
	GRU	2.6719	0.9896	3.0501	0.9192	0.0060	0.9931	0.0469	0.9274
	Bi-LSTM	0.9368	0.9962	1.7008	0.9796	0.0069	0.9973	0.0309	0.9801
	MTGNN	<b>0.7220</b>	<b>0.9984</b>	<b>1.0682</b>	0.9902	<b>0.0055</b>	<b>0.9981</b>	0.0233	<b>0.9901</b>
	Autoformer	1.5073	0.9914	1.6138	0.9786	0.0101	0.9894	0.0252	0.9790
	Informer	0.9241	0.9968	1.2595	<b>0.9912</b>	0.0065	0.9958	0.0243	0.9891
	Transformer	1.1915	0.9964	1.1074	0.9906	0.0061	0.9965	<b>0.0223</b>	0.9884

in the prediction of each element under the prediction time step of 1h with different sliding window sizes. The bold font indicates the model with the lowest RMSE and the highest R under the same sliding window setting. It can be seen from the table that the MTGNN performs best in the prediction of T and SM, while Informer is better than other models in the prediction of SSR and ET. On the other hand, with the increase of sliding window, RMSE increases and R decreases. Overall, the MTGNN and transformers are better than RNN-based models, reinforcing earlier observations.

V. DISCUSSION

In recent years, data-driven models have had an important impact on science and society with their ability to minimize computing costs, increase speed, and generate large-scale integration among the meteorological elements [66]. The application of modern DL methods in the field of meteorology has achieved some exceptional successes in weather forecasting [67], [68]. One of the significant successes is the strength in the prediction ability. The DL models can make short-term predictions in the future in just a few seconds on

the trained model, which is far from being achieved by any physical model [66].

This study demonstrates the ability of different DL models for meteorological element forecasting. Transformers always perform better than RNNs and MTGNN in many scenarios. This is related to the special attention mechanism of the transformers [32]. It completely abandons the previously widely used RNN architectures and uses a specific encoding and decoding structure. When the encoder processes time points in the sequence, it confirms the extent to which the value of the time point is affected by its own or other time point values by computing the attention score entirely by the self-attention mechanism to obtain a weighted vector. Subsequently, the decoder progressively reconstructs the low-dimensional representation of the target sequence obtained from the encoder into the original target sequence. This design greatly improves the ability and efficiency of the model in processing sequence data.

We note that the loss curve of MTGNN decreases moderately compared to the other models (Fig. 3). This may be related to the complexity of the model. The model



architecture of RNNs is relatively simple, and the loss curve decreases steadily. Compared with RNNs, the MTGNN has one more learning process of the graph adjacency matrix. The graph learning layer also updates the structure of the adjacency matrix with the introduction of data in the training stage. This leads to a slight fluctuation in the loss curve of the MTGNN at the beginning of training, but the fluctuation decreases, which is acceptable. Under the three sliding window settings, the training loss of MTGNN is reduced to different lowest points, suggesting that different sliding window size settings (that is, different model input sizes) impact the MTGNN (Fig. 3). This phenomenon seems non-existent in other models inferring that the transformer has performed best so far.

There are some limitations of this study. On the one hand, this study does not consider the impact of spatial dimension on model accuracy. Adding a spatial information aggregation module to the model may further improve the performance, which may be a potential direction for future studies on high-performing weather forecasting. On the other hand, predicting extreme values remains a challenge in time series prediction. Integrating specific components to monitor and predict extreme values may be necessary.

DL models continually evolve, showing increased accuracy and predictive power, especially in larger models with hundreds of billions of parameters. In this study, the T, SSR, SM, and ET were forecasted by DL models. The atmospheric process is complex, so the four meteorological elements may have interactive effects. This means that if the computing resources are enough, more variables such as wind speed, precipitation, and so on should be considered and incorporated into the DP models to improve prediction accuracy.

## VI. CONCLUSION

This study compared the performance of seven DL models, including LSTM, GRU and Bi-LSTM, MTGNN, deep transformer, Informer, and Autoformer, on forecasting multiple meteorological elements (T, SSR, SM, and ET). The training performance, prediction accuracy, and parameters effect of all seven models were comprehensively evaluated. The results show that:

- 1) The models have different training speeds. The RNN-based models (LSTM, GRU and Bi-LSTM) always tend to be stable first, followed by the Transformers-based models and the MTGNN. If training epoch is enough, the transformers always achieve the best training performance (lowest loss function).
- 2) Transformer-based models (deep transformer, Informer, and Autoformer) generally have the better prediction accuracy in most of scenarios. Among the three transformers, the Informer performs best overall.
- 3) The predictive performance of a DL model varies according to the specific meteorological element. The MTGNN is comparable to Transformer-based models for T and SSR; the Informer and Transformer have

smallest prediction errors for SM and ET. The model selection is related to meteorological elements.

- 4) The prediction accuracy of all models is affected by the sliding window (i.e., the length of input sub-sequence) and prediction time step. The smaller sliding window and prediction time step is, the higher prediction accuracy. However, the two parameters have only a minor effect on the performance differences of different models.

The DL models continues to evolve, with each model possessing unique capabilities. The performance of the transformers is generally better than RNN based models. However, we argue that the choice of model should not only be based on the model's inherent characteristics but also on other factors, such as the specific meteorological element being forecasted.

## REFERENCES

- [1] M. B. Baker and G. H. Roe, "The shape of things to come: Why is climate change so predictable?" *J. Climate*, vol. 22, no. 17, pp. 4574–4589, Sep. 2009.
- [2] Y. Wang, A. Wang, J. Zhai, H. Tao, T. Jiang, B. Su, and T. Fischer, "Tens of thousands additional deaths annually in cities of China between 1.5 C and 2.0 C warming," *Nature Commun.*, vol. 10, no. 1, p. 3376, 2019.
- [3] L. Gu, J. Chen, J. Yin, L. J. Slater, H. Wang, Q. Guo, M. Feng, H. Qin, and T. Zhao, "Global increases in compound flood-hot extreme hazards under climate warming," *Geophys. Res. Lett.*, vol. 49, no. 8, Apr. 2022, Art. no. e2022GL097726.
- [4] T. Hasegawa, G. Sakurai, S. Fujimori, K. Takahashi, Y. Hijikawa, and T. Masui, "Extreme climate events increase risk of global food insecurity and adaptation needs," *Nature Food*, vol. 2, no. 8, pp. 587–595, Aug. 2021.
- [5] E. Gold, "Dynamic meteorology and hydrography. Part II. Kinematics. By V. Bjerknes, Th. Herselberg, and O. Devik. Washington, DC, published by the Carnegie Institute of Washington, 1911," *Quart. J. Roy. Meteorological Soc.*, vol. 39, no. 166, pp. 161–165, Apr. 1913, doi: 10.1002/qj.49703916612.
- [6] P. D. Williams, "Modelling climate change: The role of unresolved processes," *Philos. Trans. Royal Soc. A, Math., Phys. Eng. Sci.*, vol. 363, pp. 2931–2946, Dec. 2005.
- [7] R. Dambreville, P. Blanc, J. Chanussot, and D. Boldo, "Very short term forecasting of the global horizontal irradiance using a spatio-temporal autoregressive model," *Renew. Energy*, vol. 72, pp. 291–300, Dec. 2014.
- [8] Q. F. Tan, X. Wang, H. Wang, and X. H. Lei, "Comparative study of ANN, ANFIS and AR model for daily runoff time series prediction," *South-North Water Transfers Water Sci. Technol.*, vol. 14, no. 6, pp. 12–17, 2016, doi: 10.13476/j.cnki.nsbdqk.2016.06.003.
- [9] I. Dimoulkas, P. Mazidi, and L. Herre, "EEM 2017 forecast competition: Wind power generation prediction using autoregressive models," in *Proc. 14th Int. Conf. Eur. Energy Market (EEM)*, 2017, pp. 1–6.
- [10] G. T. Knofczynski and D. Mundfrom, "Sample sizes when using multiple linear regression for prediction," *Educ. Psychol. Meas.*, vol. 68, no. 3, pp. 431–442, Jun. 2008.
- [11] A. Sarraf, S. F. Vahdat, and A. Behbahaninia, "Relative humidity and mean monthly temperature forecasts in Ahwaz Station with ARIMA model in time series analysis," in *Proc. Int. Conf. Environ. Ind. Innov. (IPCBEI)*, vol. 12. Singapore: IACSIT Press, 2011.
- [12] E. De Saa and L. Ranathunga, "Comparison between ARIMA and deep learning models for temperature forecasting," 2020, *arXiv:2011.04452*.
- [13] Y. Du et al., "Adarnn: Adaptive learning and forecasting of time series," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manag.*, 2021, pp. 402–411.
- [14] C. Jeong, J.-Y. Shin, T. Kim, and J.-H. Heo, "Monthly precipitation forecasting with a neuro-fuzzy model," *Water Resour. Manag.*, vol. 26, no. 15, pp. 4467–4483, Dec. 2012.
- [15] J. Rhee and J. Im, "Meteorological drought forecasting for ungauged areas based on machine learning: Using long-range climate forecast and remote sensing data," *Agricult. Forest Meteorol.*, vols. 237–238, pp. 105–122, May 2017.

- [16] Y. Jiang, Z. Song, and A. Kusiak, "Very short-term wind speed forecasting with Bayesian structural break model," *Renew. Energy*, vol. 50, pp. 637–647, Feb. 2013.
- [17] A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski, "Using Bayesian model averaging to calibrate forecast ensembles," *Monthly Weather Rev.*, vol. 133, no. 5, pp. 1155–1174, May 2005.
- [18] Z. Ding, J. Zhang, and G. Xie, "LS-SVM forecast model of precipitation and runoff based on EMD," in *Proc. 6th Int. Conf. Natural Comput.*, 2010, pp. 1721–1725.
- [19] M. A. Nayak and S. Ghosh, "Prediction of extreme rainfall event using weather pattern recognition and support vector machine classifier," *Theor. Appl. Climatol.*, vol. 114, nos. 3–4, pp. 583–603, Nov. 2013.
- [20] X. Wang, "Facial expression recognition based on deep learning and traditional machine learning," *Appl. Sci. Technol.*, vol. 45, no. 1, pp. 65–72, 2018.
- [21] O. J. Reichman, M. B. Jones, and M. P. Schildhauer, "Challenges and opportunities of open data in ecology," *Science*, vol. 331, no. 6018, pp. 703–705, Feb. 2011.
- [22] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat, "Deep learning and process understanding for data-driven earth system science," *Nature*, vol. 566, no. 7743, pp. 195–204, Feb. 2019.
- [23] X. Qing and Y. Niu, "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM," *Energy*, vol. 148, pp. 461–468, Apr. 2018.
- [24] X. Shi, Z. Gao, L. Lausen, H. Wang, D. Y. Yeung, W. Wong, and W. Woo, "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
- [25] H. Han and R. R. Morrison, "Improved runoff forecasting performance through error predictions using a deep-learning approach," *J. Hydrol.*, vol. 608, May 2022, Art. no. 127653.
- [26] Y. Feng, N. Cui, W. Hao, L. Gao, and D. Gong, "Estimation of soil temperature from meteorological data using different machine learning models," *Geoderma*, vol. 338, pp. 67–77, Mar. 2019.
- [27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [28] S. Wang, L. Hu, Y. Wang, X. He, Q. Z. Sheng, M. Orgun, L. Cao, N. Wang, F. Ricci, and P. S. Yu, "Graph learning approaches to recommender systems: A review," 2020, *arXiv:2004.11718*.
- [29] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 890–897.
- [30] R. A. Rossi, L. K. McDowell, D. W. Aha, and J. Neville, "Transforming graph data for statistical relational learning," *J. Artif. Intell. Res.*, vol. 45, pp. 363–441, 2012.
- [31] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020.
- [32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
- [33] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 12873–12883.
- [34] M. H. Guo, J. X. Cai, Z. N. Liu, T. J. Mu, R. R. Martin, and S. M. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, pp. 187–199, Jun. 2021.
- [35] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surv.*, vol. 54, no. 10, pp. 1–41, Jan. 2022.
- [36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
- [37] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi: [10.1162/neco\\_a\\_01199](https://doi.org/10.1162/neco_a_01199).
- [38] A. Shewalkar, D. Nyavanandi, and S. A. Ludwig, "Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU," *J. Artif. Intell. Soft Comput. Res.*, vol. 9, no. 4, pp. 235–245, Oct. 2019, doi: [10.2478/jaiscr-2019-0006](https://doi.org/10.2478/jaiscr-2019-0006).
- [39] R. L. Abduljabbar, H. Dia, and P.-W. Tsai, "Development and evaluation of bidirectional LSTM freeway traffic forecasting models using simulation data," *Sci. Rep.*, vol. 11, no. 1, p. 23899, Dec. 2021, doi: [10.1038/s41598-021-03282-z](https://doi.org/10.1038/s41598-021-03282-z).
- [40] Z. Hu, Y. Dong, K. Wang, K. W. Chang, and Y. Sun, "GPT-GNN: Generative pre-training of graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 1857–1867.
- [41] Y. Yu, J. Chen, T. Gao, and M. Yu, "DAG-GNN: DAG structure learning with graph neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97, C. Kamalika and S. Ruslan, 2019, pp. 7154–7163.
- [42] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep transformer networks for anomaly detection in multivariate time series data," 2022, *arXiv:2201.07284*.
- [43] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11106–11115.
- [44] H. Wu, J. Xu, J. Wang, and M. Long, "AutoFormer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 22419–22430.
- [45] L. Wang, W. Chen, G. Huang, and G. Zeng, "Changes of the transitional climate zone in east asia: Past and future," *Climate Dyn.*, vol. 49, no. 4, pp. 1463–1477, Aug. 2017.
- [46] J. Huang, Y. Li, C. Fu, F. Chen, Q. Fu, A. Dai, M. Shinoda, Z. Ma, W. Guo, Z. Li, L. Zhang, and Y. Liu, "Dryland climate change: Recent progress and challenges," *Rev. Geophys.*, vol. 55, no. 3, pp. 719–778, Sep. 2017.
- [47] C. Fu, *Landscape Boundaries: Consequences for Biotic Diversity and Ecological Flows*. Berlin, Germany: Springer, 1992, pp. 394–402.
- [48] W. Xiao, Q. Yu, G. N. Flerchinger, and Y. Zheng, "Evaluation of SHAW model in simulating energy balance, leaf temperature, and micrometeorological variables within a maize canopy," *Agronomy J.*, vol. 98, no. 3, pp. 722–729, May 2006.
- [49] M. E. Newman, K. P. McLaren, and B. S. Wilson, "Long-term socio-economic and spatial pattern drivers of land cover change in a Caribbean tropical moist forest, The Cockpit Country, Jamaica," *Agric. Ecosyst. Environ.*, vol. 186, pp. 185–200, Mar. 2014.
- [50] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horanyi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, and A. Simmons, "The ERA5 global reanalysis," *Quart. J. Roy. Meteorological Soc.*, vol. 146, pp. 1999–2049, Jul. 2020.
- [51] J. Zhang, R. Shen, C. Shi, L. Bai, J. Liu, and S. Sun, "Evaluation and comparison of downward solar radiation from new generation atmospheric reanalysis ERA5 across Mainland China," *J. Geo-Inf. Sci.*, vol. 23, pp. 2261–2274, Dec. 2021.
- [52] Y. Song and J. Wei, "Diurnal cycle of summer precipitation over the north China plain and associated land-atmosphere interactions: Evaluation of ERA5 and MERRA-2," *Int. J. Climatol.*, vol. 41, no. 13, pp. 6031–6046, Nov. 2021.
- [53] S. Sun, H. Chen, W. Ju, G. Wang, G. Sun, J. Huang, and G. Yan, "On the coupling between precipitation and potential evapotranspiration: Contributions to decadal drought anomalies in the Southwest China," *Climate Dyn.*, vol. 48, pp. 3779–3797, 2017.
- [54] J.-S. Chou and N.-T. Ngo, "Time series analytics using sliding window metaheuristic optimization-based machine learning system for identifying building energy consumption patterns," *Appl. Energy*, vol. 177, pp. 751–770, Sep. 2016.
- [55] N. M. Norwawi, *Data Science for COVID-19*. Amsterdam, The Netherlands: Elsevier, 2021, pp. 547–564.
- [56] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecasting*, vol. 36, no. 3, pp. 1181–1191, Jul. 2020.
- [57] B. Lim, S. Zohren, and S. Roberts, "Recurrent neural filters: Learning independent Bayesian filtering steps for time series prediction," in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 1–8.
- [58] G. Gelly and J. L. Gauvain, "Optimization of RNN-based speech activity detection," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 3, pp. 646–656, 2017.
- [59] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philos. Trans. Roy. Soc. A*, vol. 379, Apr. 2021, Art. no. 20200209.
- [60] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.
- [61] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.
- [62] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9.

- [63] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [64] "R-NET: Machine reading comprehension with self-matching networks," 2017.
- [65] D. Qi and A. J. Majda, "Using machine learning to predict extreme events in complex systems," *Proc. Nat. Acad. Sci. USA*, vol. 117, no. 1, pp. 52–59, Jan. 2020.
- [66] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, P. Hassanzadeh, K. Kashinath, and A. Anandkumar, "FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators," 2022, *arXiv:2202.11214*.
- [67] Y.-G. Ham, J.-H. Kim, and J.-J. Luo, "Deep learning for multi-year ENSO forecasts," *Nature*, vol. 573, no. 7775, pp. 568–572, Sep. 2019, doi: [10.1038/s41586-019-1559-7](https://doi.org/10.1038/s41586-019-1559-7).
- [68] S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, M. Fitzsimons, M. Athanassiadou, S. Kashem, S. Madge, and R. Prudden, "Skilful precipitation nowcasting using deep generative models of radar," *Nature*, vol. 597, pp. 672–677, Sep. 2021, doi: [10.1038/s41586-021-03854-z](https://doi.org/10.1038/s41586-021-03854-z).



**RUIBO QIU** is currently pursuing the degree with Nanjing University of Information Science and Technology. His current research interests include deep learning and photogrammetry.



**WEN DAI** received the Ph.D. degree in cartography and geographic information systems from Nanjing Normal University, in 2021. He is currently a Teacher with Nanjing University of Information Science and Technology. His research interests include terrain analysis, remote sensing, deep learning, and 3D modeling.



**GUOJIE WANG** received the Ph.D. degree in philosophy (meteorology) from Vrije Universiteit Amsterdam, in 2011. He is currently a Professor with Nanjing University of Information Science and Technology. His research interests include land-atmosphere interaction, the impact of climate change, disaster risk management, and artificial intelligence remote sensing.



**ZICONG LUO** received the M.S. degree in 3S integration and meteorological application from Nanjing University of Information Science and Technology, in 2023. His research interests include the application of artificial intelligence to meteorology and artificial large models.



**MENGQI LI** received the B.S. degree in GIS from Nanjing University of Information Science and Technology in 2023. She is currently pursuing the master's degree with the University of Zurich. Her research interests include remote sensing, deep learning, and 3D modeling.

...