IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 August 2023, accepted 28 August 2023, date of publication 31 August 2023, date of current version 12 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3310538

RESEARCH ARTICLE

CloudpredNet: An Ultra-Short-Term Movement Prediction Model for Ground-Based Cloud Image

LIANG WEI[1], TINGTING ZHU[1], YIREN GUO[1], CHAO NI[1], AND QINGYUAN ZHENG[2]
[1]College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China
[2]School of Physics and Information Engineering, Jiangsu Second Normal University, Nanjing 210013, China

Corresponding author: Tingting Zhu (tingtingzhu@njfu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62006120, and in part by the Graduate Research Practice Innovation Plan of Jiangsu in 2021 under Grant KYCX21-0878.

ABSTRACT Ground-based cloud images can provide information on weather and cloud conditions, which are important for cloud monitoring and PV power generation forecasting. Prediction of short-time cloud movement from images is a major means of intra-hourly irradiation forecasting for solar power generation and is also important for precipitation forecasting. However, there is a lack of advanced and complete methods for cloud movement prediction from ground-based cloud images, and traditional techniques based on image processing and motion vector calculations have difficulty in predicting cloud morphological changes, which makes accurate prediction of cloud motion (especially nonlinear motion) very challenging. Therefore, this paper proposes CloudpredNet, a ground-based cloud ultra-short-term movement prediction model based on an ''encoder-generator'' architecture. This paper also proposes a loss function dedicated to the time series prediction of ground-based cloud images and combines the attention mechanism to train the model. The model is validated on a publicly available dataset, and it is demonstrated that it has good performance in all metrics of cloud image generation for the next 10 minutes.

INDEX TERMS Spatio–temporal prediction, ultra-short-term prediction, transformer, ground-based cloud images.

I. INTRODUCTION

A time series contains a series of data points that are indexed in order. This arrangement is essentially the time at which they occurred or were recorded, with the data points recorded at equal spatial intervals at discrete times. Time series prediction is a way of making predictions about behavior by using past data points. Prediction is useful to understand the trend of the data. Time series prediction is different from regular supervised learning. In supervised learning, it is assumed that all observations are independent of each other, and the recorded data can be used in any order and there are no criteria to follow. This is not the case with time series, in which the sequence of data points is meaningful. Time series modeling is a dynamic research area that has attracted the attention of the researcher community in recent decades [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Felix Albu.

The main traditional time series forecasting methods commonly used are statistical methods, machine learning methods and deep learning methods. For statistical methods, Box et al. [2] proposed differential integrated moving average autoregression (ARIMA) in 1970. After the time series is smoothed, one can turn to the ARIMA model, which includes both moving average and autoregressive processes. For time series that contain seasonal patterns, Williams et al. [3] proposed the seasonal differential integrated moving average autoregressive SARIMA model. Although the ARIMA model and its variants are widely used in various fields, it has the obvious limitation that the assumptions made by the model are linear, which makes the model not perform well in some scenarios. For machine learning methods, Hearst et al. [4] proposed Support Vector Machines (SVM), which has a learning strategy of maximizing the classification interval and solves the problem of high-dimensional linear indistinguishability using kernel methods. Awad et al. [5] further

improved SVM by using it for regression prediction tasks. For deep learning methods, the recurrent neural network (RNN) can make full use of historical information to predict future information. After RNN was proposed, some variants, such as LSTM [6] and GRU [7], added ''gating'' mechanisms to select the information flow and further improve the model expression capability. However, there are problems such as insufficient information flow into each layer, gradient disappearance, and gradient explosion. In recent years, models of Transformer [8] architecture have greatly outperformed RNN-like models in temporal sequence prediction by virtue of attention mechanism, and some variants such as Reformer [9], Informer [10], Autoformer [11], and TimesNet [12] further explore the implicit relationships within sequences. Compared with traditional methods, these models expand the number of parameters and the amount of training data, and perform well in both long-term and short-term prediction tasks.

For the task of time-series prediction of ground-based cloud images, the essence belongs to spatio-temporal series prediction. Since the various types of time-series models mentioned above can only deal with one-dimensional temporal data and cannot include the spatial features of images, which is an indispensable part for the analysis and processing of cloud images in the field of weather prediction. Therefore, many researchers have improved traditional time-series prediction models, especially those based on deep learning, to enable them to handle continuous-time cloud images. Shi et al. [13] combined the convolution operator and LSTM, and proposed the Convolution Long Short Term Memory Network (ConvLSTM) model so that the model can learn spatial information. ConvLSTM can be used as a landmark model in the field of image frame prediction. Many improved models of ConvLSTM have been generated subsequently. In 2017, Shi et al. [14] proposed the Trajectory Gated Recurrent Unit (TrajGRU) and used it for precipitation prediction to obtain higher prediction accuracy. Lin et al. [15] that the images generated by ConvLSTM in some details are blurred, proposed Self-Attention ConvLSTM (SA-ConvLSTM), which introduces an attention mechanism to supplement the detail information in the image, and experiments showed that the model can generate higher quality images. Wang et al. [16] improved on ConvLSTM by proposing PredRNN, which used spatio-temporal long and short term memory (ST-LSTM) module to enhance the efficiency of information utilization in both horizontal and vertical directions. In 2018 and 2022, Wang et al. proposed PredRNN++ [17] and PredRNNV2 [18]. PredRNN++ proposes a cascaded causal LSTM that allows the model to model longer time dimensions, and PredRNNV2 decouples the memory storage unit and treats it as a modular structure that can ensure the model can perform long-term learning dynamically. In addition to the above recurrent cell-based models, Jia et al. [19] proposed Dynamic Filter Networks (DFN), which can learn a variety of filtering operations, including local spatial transformations, selective blurring or adaptive feature extraction. and content dimensions, and then implemented pixel-level prediction based on an ''encoder-decoder'' architecture. Although some progress has been made in the field of spatio-temporal sequence prediction based on deep learning framework, there are still problems such as blurred prediction images and inability to predict high-resolution images and short prediction time.

Researchers often want to conduct timely observational studies of forecast data for the next few moments to perform some targeted operations. For example, for PV power systems [20], cloud motion and solar shading cause PV output ramp events that require manual intervention by the PV plant to compensate. Predicting the occurrence of such events is beneficial to improve the efficiency of PV plant operation and management. Su et al. [21] proposed a model to predict cloud image motion. The model is based on gated recurrent unit (GRU) and recurrent convolutional network (RCN), which has convolutional structures to deal with spatio-temporal features. Due to the RNN structure included in the model, its parallelism is poor and the training time is long. Crisosto et al. [22] have also conducted cloud image movement prediction research. This method uses the pre-trained VGG16 model for feature extraction, and then compares the similarity between the generated image and the real value. In this work, the ability of VGG16 to extract features is insufficient, resulting in a large room for improvement in the final result.

In this paper, we propose a novel approach called CloudpredNet for ground-based cloud ultra-short-term movement prediction. Since only the first 60 minutes of images are used, it is called ''Ultra-Short-Term''. The proposed model aims to predict the ground-based cloud image of the next 10 minutes through the cloud image and meteorological time series data of the first 60 minutes. The main contributions of this paper are as follows:

- A novel multimodal cloud prediction model, CloudpredNet, is built based on an ''encoder-generator'' architecture, which encodes cloud and weather data and then process them using transpose convolution. Unlike most time-series image generation methods, the model generates cloud images of $3 \times 256 \times 256$ size for the next 10 minutes instead of small size images of $3 \times 128 \times 128$.
- Combining $SmoothL_1Loss$, Structural Similarity Index Loss Function (*SSIM Loss*) and Learning Perceptual Image Block Similarity Loss Function (*LPIPS Loss*), we propose a loss function dedicated to cloud image movement prediction.

II. RELATED BACKGROUND THEORIES

In this section, we introduce two models used in the subsequent experiments, including TimeSformer and TimesNet.

A. TIMESFORMER

In the traditional video understanding field (similar to the task of time-series image modeling in this article), most of

the related scholars use networks based on CNN architecture, such as Two-Stream Net [23] and 3D CNN [24], etc. The design ideas of these networks are basically borrowed from 2D CNN in the image field, and adding one dimension (temporal dimension) to 2D convolution can process video data smoothly. However, this structure has some obvious drawbacks: 1) If a deep enough network is to be designed, then the computation becomes very large; 2) As with 2D convolutional neural networks, the problem that they contain too much prior knowledge and inductive bias is not addressed in the three-dimensional convolutional networks, so this can greatly limit the expressive ability of the network. Especially when the video data volume becomes larger and the time frame becomes longer, the performance degradation becomes obvious. Convolutional kernels are specifically designed to capture local spatio-temporal information, and they are not capable of modeling dependencies beyond the sensory field. Although stacking the convolutions and deepening the network will expand the receptive field, these strategies still limit the modeling of long-term dependencies by aggregating information in a very short range. In contrast, self-attention mechanisms can be used to capture both local and global dependencies over long ranges by directly comparing features at all spatio-temporal locations.

Based on this, TimeSformer [25] migrates Vision Transformer, which performs better in the image and video domain. It proposes separated spatio-temporal attention. The advantages of Transformer architecture and separated spatio-temporal attention can be a good solution to the shortcomings of other models mentioned above. Figure 1 shows the separated spatio-temporal attention used in TimeSformer and other parts of the existing attention mechanisms. It can be seen that, except for spatial attention, all other spatio-temporal attention can capture the local dependencies between neighboring image blocks and the global dependencies of distant image blocks, and the attention relation diagrams of these different attention mechanisms are shown in Figure 2. The relevant conclusions obtained by combining Figure 1 and Figure 2 are as follows.

- **Spatial attention.** Like the self-attention in Vision Transformer. It is computed only in one frame and no connection can be made with the content of the frame before and after.
- **Joint spatio-temporal attention.** The computation of attention for each video frame in all three dimensions of intra-image patch, time and space is too computationally intensive and requires resources such as graph memory that cannot be well supported.
- **Separated spatio-temporal attention.** Considering the limitations of joint spatio-temporal attention, separating spatio-temporal attention separates time and space (i.e., splitting into a one-dimensional + two-dimensional form) and performs the temporal attention calculation first and then the spatial attention calculation. This operation allows the length of each sequence to be reduced, thus reducing the computational complexity.
- **Sparse local-global attention.** Since joint spatio-temporal attention would make the sequence length too long, sparse local-global attention can split the sequence into two classes, local and global, and perform local attention computation first and then global attention computation. This operation is similar to the moving window attention computation strategy in Swin Transformer. Local attention is performed within the adjacent image patches of $H/2$ and $W/2$, and global attention is computed over the whole sequence using two image block steps.
- **Axis attention.**Axis attention can split the 3D video input data into three one-dimensional data, i.e., by temporal dimension, wide dimension, and high dimension.

TimeSformer is designed with a simple idea and relatively low computational overhead for training and inference, and its excellent performance has been verified by extensive experiments. The formula for separating spatio-temporal attention is:

$$\boldsymbol{\alpha}_{(p,t)}^{(\ell,a)} = \mathrm{SM}\left(\frac{\mathbf{q}_{(p,t)}^{(\ell,a)\top}}{\sqrt{D_h}} \cdot \begin{bmatrix} \mathbf{k}_{(0,0)}^{(\ell,a)} \left\{\mathbf{k}_{(p',t')}^{(\ell,a)}\right\} p' = 1,\ldots,N \\ t' = 1,\ldots,F \end{bmatrix}\right) \tag{1}$$

where p' is the number patch of the image; t' the number frame of the current image; ℓ is the ℓth block of the encoder; $\sqrt{D_h}$ is the scaling factor, D_h = hidden layer dimension / number of heads of multi-headed attention; a is the first heads of multi-headed attention; SM is the *softmax* function. Therefore, using TimeSformer as an encoder for ground-based cloud images, it can analyze them in both temporal and spatial dimensions and extract high-quality features.

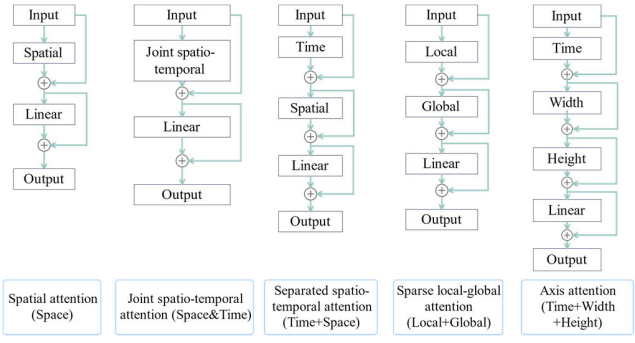

FIGURE 1. Structure of different attention mechanisms.

B. TIMESNET

In the field of time series analysis, previous approaches have attempted to accomplish this operation directly from 1D time series, which is extremely challenging due to the complex temporal patterns. TimesNet [12] decomposes the complex time variation into multiple intra-periodic and

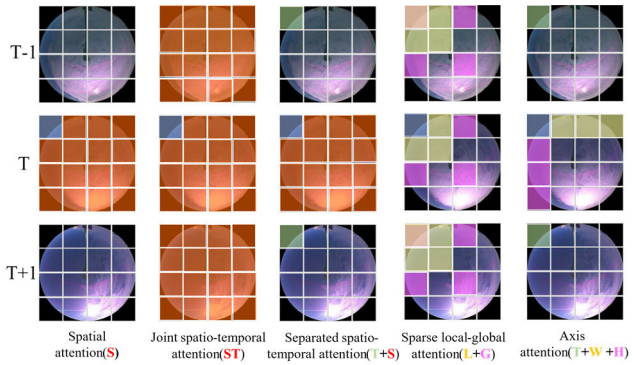

FIGURE 2. Attention map of different attention mechanisms.

inter-periodic variations based on the observation of the multi-periodicity of time series. In order to address the limitations of one-dimensional time series in terms of representational capability, what is done is to extend the analysis of time variation to a two-dimensional space by converting a one-dimensional time series into a set of two-dimensional tensor based on multiple periods. This transformation allows embedding the intra-periodic and inter-periodic variations into the columns and rows of the 2D tensor, respectively, making the 2D variations easily simulated by the 2D kernel. Technically, TimesNet with TimesBlock is proposed as a generic backbone for time series analysis. TimesBlock can adaptively discover multi-periodicity and extract complex temporal variations from the transformed 2D tensor by means of a parameter-efficient starting block.

In the original paper, time series are analyzed in a new dimension of multi-periodicity in order to address the complex temporal variation. First, it is observed that real-world time series usually exhibit multi-periodicity, such as daily and annual variations in weather observations and weekly and quarterly variations in electricity consumption. These multiple periods overlap and interact with each other, making it difficult to model the changes. Second, for each period, it is found that changes at each point in time are not only influenced by the temporal patterns in its neighboring regions, but are also highly correlated with the changes in its neighboring periods. The two types of temporal variation are named interperiod variation and intraperiod variation, respectively. The former indicates short-term temporal patterns over a period. The latter can reflect long-term trends over different consecutive periods. Intraperiod variation and interperiod variation are shown in Figure 3. It is worth noting that for time series without explicit periodicity, the variation will be dominated by intra-periodic variation, which is equivalent to a time series with infinite period length. Since different periods lead to different intra-periodic and intertemporal variations, multi-periodicity naturally allows for the derivation of a modular architecture for modeling temporal variation that can capture period-specific derived variations in a single module. Furthermore, this design allows complex temporal patterns to be unraveled, facilitating time-varying modeling.

However, it is difficult to explicitly present two different types of changes simultaneously for a one-dimensional time series. For a one-dimensional time series $\mathbf{X}_{1D} \in \mathbb{R}^{T\times C}$ of time length T and channel dimension C, its periodicity can be calculated by the fast Fourier transform (FFT) of the time dimension, as follows.

$$\begin{aligned} \mathbf{A} &= \text{Avg}\left(\text{Amp}\left(\text{FFT}\left(\mathbf{X}_{1\text{D}}\right)\right)\right) \\ f_1, \cdots, f_k &= \arg\text{Topk}(\mathbf{A}) \\ f_* &\in \left\{1, \cdots, \left\lceil \frac{T}{2} \right\rceil\right\} \\ p_1, \cdots, p_k &= \left\lceil \frac{T}{f_1} \right\rceil, \cdots, \left\lceil \frac{T}{f_k} \right\rceil \end{aligned} \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^T$ is the amplitude of each frequency component in $\mathbf{X}_{1D}$, and the k frequencies with the largest intensity $\{f_1, \cdots, f_k\}$ correspond to the most significant k cycle lengths $\{p_1, \cdots, p_k\}$, denoted as $\mathbf{A}, \{f_1, \cdots, f_k\}, \{p_1, \cdots, p_k\} = \text{Period}\left(\mathbf{X}_{1\text{D}}\right)$; Avg denotes average operation.

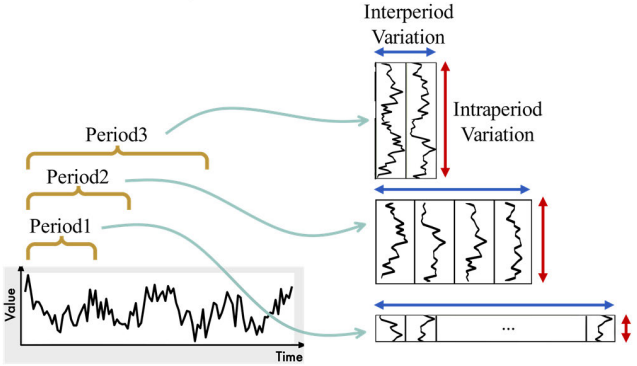

FIGURE 3. Multi-periodicity and temporal 2D-variation of time series.

As shown in Figure 4, the 1D time series is reshaped into a 2D tensor, where each column contains time points within a cycle and each row contains time points at the same stage between different cycles. The left side is to determine the top k cycles, taking 3 cycles as an example, then the 1D time series is transformed into 3 different 2D-variations (those that cannot be integrable can be padding), and the three 2D-variations are processed with 2D convolution before aggregating the results.

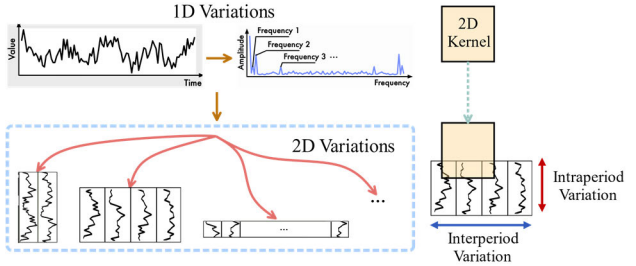

FIGURE 4. 2D structure of time series.

The network structure of TimesNet is shown as Figure 5. TimesNet consists of TimesBlock stacking, and the input data first pass through the Embedding layer to get $\mathbf{X}_{1D}^{0} \in \mathrm{R}^{T \times d_{\text{model}}}$, and TimesBlock has the following relationship for the lth layer.

$$\mathbf{X}_{1D}^{l} = \text{TimesBlock}\left(\mathbf{X}_{1D}^{l-1}\right) + \mathbf{X}_{1D}^{l-1} \tag{3}$$

where $\mathbf{X}_{1D}^{l-1}$ is the input of the layer; $\mathbf{X}_{1D}^{l}$ is the output of the layer.

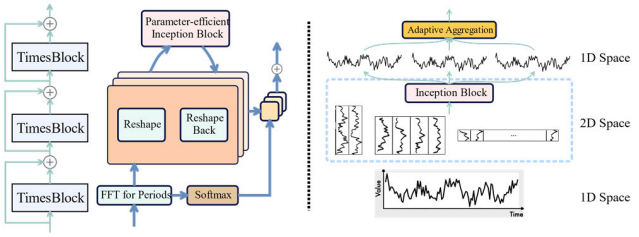

FIGURE 5. TimesNet structure diagram.

TimesBlock in TimesNet contains the following four processes.

1) 1D TO 2D CONVERSION

The period is first extracted for 1D data and then converted into a 2D tensor as follows.

$$\mathbf{X}_{2D}^{l,i} = \text{Reshape}_{p_i,f_i}\left(\text{Padding}\left(\mathbf{X}_{1D}^{l-1}\right)\right) \quad i \in \{1, \cdots, k\} \tag{4}$$

where f is the period frequency; p is the period length; and $\mathbf{X}_{1D}^{l-1}$ are the input one-dimensional time-series data.

2) EXTRACTION OF 2D TIME-SERIES VARIATION CHARACTERISTICS

For two-dimensional data, the information can be extracted using 2D convolution because of its two-dimensional localization. The InceptionV1 model is selected for feature extraction.

$$\hat{\mathbf{X}}_{2D}^{l,i} = \text{Inception}\left(\mathbf{X}_{2D}^{l,i}\right), i \in \{1, \cdots, k\} \tag{5}$$

where $\mathbf{X}_{2D}^{l,i}$ are the input time-series data; $\hat{\mathbf{X}}_{2D}^{l,i}$ are the output time-series data.

3) INFORMATION AGGREGATION, 2D TRANSFORMATION TO 1D

For the extracted time-series features, they are transformed back into a one-dimensional space for information aggregation.

$$\hat{\mathbf{X}}_{1D}^{l,i} = \mathit{Trunc}\left(\text{Reshape}_{1,(p_i \times f_i)}\left(\hat{\mathbf{X}}_{2D}^{l,i}\right)\right) \quad i \in \{1, \cdots, k\} \tag{6}$$

where *Trunc* is the operator to remove padding; $\hat{\mathbf{X}}_{1D}^{l,i}$ is the output one-dimensional time series data.

4) ADAPTIVE FUSION

The processed 1D features in step 3) are weighted and summed according to the intensity of the corresponding frequencies to obtain the final feature representation.

$$\hat{\mathbf{A}}_{f_1}^{l-1}, \cdots, \hat{\mathbf{A}}_{f_k}^{l-1} = \text{Softmax}\left(\mathbf{A}_{f_1}^{l-1}, \cdots, \mathbf{A}_{f_k}^{l-1}\right)$$
$$\mathbf{X}_{1D}^{l} = \sum_{i=1}^{k} \hat{\mathbf{A}}_{f_i}^{l-1} \times \hat{\mathbf{X}}_{1D}^{l,i} \tag{7}$$

where $\hat{\mathbf{A}}_{f_i}^{l-1}, i \in \{1, \cdots, k\}$ is the amplitude of the corresponding frequency; $\mathbf{X}_{1D}^{l}$ is the output of the Block.

III. CLOUDPREDNET METHOD

Since TimesNet can analyze the internal patterns of time series from various scales, we use TimesNet as a feature extractor for multimodal weather data in this paper. The structure of the proposed method is shown as Figure 6. The temporal image and multimodal temporal data enter TimeSformer and TimesNet to get the respective features, then converge them, and finally enter the generator to get the predicted image of the next moment and calculate the loss with the real image.

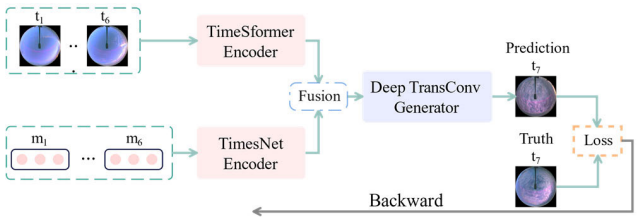

FIGURE 6. The structure of the proposed method.

CloudpredNet adopts an ''encoder-generator'' architecture. The encoder part includes cloud feature coding and weather feature coding, and the generator part is a transposed convolution-based image generator. The encoder part of TimeSformer is used for cloud feature encoding, and the output feature dimension is 256 and the generator part is based on the Deep Convolutional Generative Adversarial Network (DCGAN). DCGAN consists of Discriminator and Generator, which are trained alternately. For DCGAN, it is necessary to learn the distribution of real samples, and the Discriminator only needs to judge whether it is true or not, and clearly predict the target. In the task of this article, it is necessary to accurately predict and generate the target image, which belongs to supervised learning, so the Discriminator part of DCGAN is not required. The model in this paper draws on the Generator part of DCGAN to enlarge the image size through a series of transposed convolutions. It is worth noting that the input of the Generator in DCGAN is a random noise vector without any special information attached. The input of the Generator in this article is the feature vector extracted by the Encoder part, which belongs to the conditional generator.

Finally, the two types of features after the encoding part are subjected to a fusion operation with a dimension of 256,

and then fed into the generating part to generate the final prediction image.

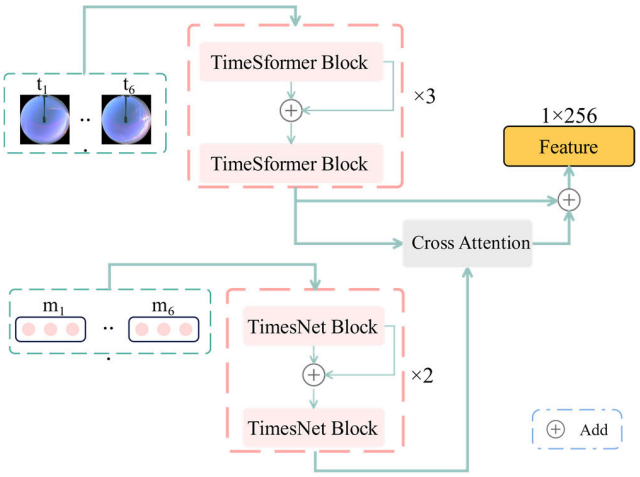

FIGURE 7. CloudpredNet encoder structure diagram.

A. IMAGE AND MULTIMODAL TIME-SERIES FEATURE EXTRACTION NETWORK

The specific structure of the CloudpredNet encoder part is shown in Figure 7. Considering the computational resources and training efficiency, the number of TimeSformer Blocks is set to 6 and the number of TimesNet Blocks is set to 4. We subject the output features of TimeSformer and TimesNet to cross-attention [26] operations. In this cross-attention, Key and Value are the features of TimesNet and Query is the feature of TimeSformer. The purpose of this operation is to update the output features of TimesNet so that they contain important information in the cloud images. Then, the features of the updated TimesNet are summed with those of TimeSformer. Through the summation operation, the historical cloud and weather information can be fully integrated, which will provide the basis for the subsequent generation of forecast images.

B. IMAGE GENERATOR BASED ON TRANSPOSED CONVLUTION

The CloudpredNet generator is based on transposed convolution with the structure shown in Table 1. Since the feature extracted by the Encoder is a one-dimensional vector, and the final predicted image's size is $3 \times 256 \times 256$, it is necessary to use a convolution kernel of a specific size to convert the one-dimensional feature vector into a multi-dimensional tensor. The 2D transpose convolution with kernel size of 4 and step size of 1 can transform the 1D data into image format. After each transpose convolution operation (except the last one), a BatchNorm layer is added to normalize the data to make the gradient smoother, and then the *ReLU* activation function is used. In the field of image generation, the final activation function of most networks is *Tanh*, and after much practice, it is known that the effect of this activation function is better than other activation functions. Therefore, the activation function of the last layer of the generator is chosen as *Tanh*.

TABLE 1. CloudpredNet generator structure diagram.

Layer	Kernel size	Stride	Pad	Channel	Output shape
Input	-	-	-	-	256*1*1
TransConv	4	1	0	1024	1024*4*4
BatchNorm /ReLU	-	-	-	-	1024*4*4
TransConv	4	2	1	512	512*8*8
BatchNorm /ReLU	-	-	-	-	512*8*8
TransConv	4	2	1	256	256*16*16
BatchNorm /ReLU	-	-	-	-	256*16*16
TransConv	4	2	1	128	128*32*32
BatchNorm /ReLU	-	-	-	-	128*32*32
TransConv	4	2	1	64	64*64*64
BatchNorm /ReLU	-	-	-	-	64*64*64
TransConv	4	2	1	32	32*128*128
BatchNorm /ReLU	-	-	-	-	32*128*128
TransConv	4	2	1	3	3*256*256
Tanh/ Output	-	-	-	-	3*256*256

Unlike traditional upsampling operations such as interpolation, transposed convolution contains trainable parameters, so it can better learn details such as texture, structure, and style in the image. In the traditional structure, the model directly predicts the target image, which prevents the network from gradually learning internal features. The Generator structure designed in this paper can fully integrate the time series feature information to generate the ground-based cloud image of the target size. In other words, the Generator gradually enlarges the small-size feature map to obtain the final image, and gradually restores the details of the real image in the process. Compared with traditional models that can only predict images from images, the architecture of our model can input other auxiliary information for feature fusion. The Generator can accept these features well and will not be affected by other factors.

C. LOSS FUNCTION OF CLOUDPREDNET

We combine $SmoothL_1Loss$, structural similarity index loss function (*SSIM Loss*) and learning perceptual image block similarity loss function (*LPIPS Loss*) to propose a loss function dedicated to cloud image timing prediction, and the expression is:

$$Loss = \alpha \times SmoothL_1Loss + \beta \times SSIM + \gamma \times LPIPS \quad (8)$$

where α, β, γ are the scale coefficients of the three. In the experiment, set $\alpha = 1, \beta = 0.1, \gamma = 0.1$.

Specifically, *SmoothL*$_1$*Loss* is less sensitive to outliers than MSE, combining the advantages of MAE and MSE. When the difference between predicted and true values is small, the loss function is smoother compared to MAE; when the difference between predicted and true values is large, the gradient is more stable and less prone to gradient explosion. The calculation formula is:

$$SmoothL_1(x, y) = \begin{cases} 0.5\,(x_i - y_i)^2 \ \ if \ \ |x_i - y_i| < 1 \\ |x_i - y_i| - 0.5 \text{ otherwise} \end{cases} \tag{9}$$

where x_i and y_i are the predicted and true values.

SSIM mainly considers three key characteristics of the image: brightness, contrast and structure. The calculation formula is:

$$SSIM(x, y) = l(x, y)^{\alpha} \cdot c(x, y)^{\beta} \cdot s(x, y)^{\gamma} \tag{10}$$

where α, β, γ are the proportional coefficients of the three characteristics, which are generally taken as 1. The value range of *SSIM* is [-1, 1] and $SSIM\ Loss = (1 - SSIM)/2$, which takes the value range of [0,1] the smaller the better.

LPIPS is used to measure the difference between two images. The metric uses a neural network to extract the features by inputting two inputs to the neural network, normalizing the output of each layer after activation as $\hat{y}^l, \hat{y}^l_0$, and then multiplying the w-layer weights and calculating the L_2 distance, and finally averaging to obtain the distance. The calculation formula is shown in (11). the lower value of *LPIPS* indicates that the two images are more similar, and vice versa, the greater the difference. In this section feature extraction is performed using AlexNet that has been pre-trained on ImageNet and the network is frozen during training.

$$d\,(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h.w} \left\| w_l \odot \left(\hat{y}^l_{hw} - \hat{y}^l_{0hw} \right) \right\|_2^2 \tag{11}$$

where l is the lth layer; H_l, W_l are the feature map height and width of the l th layer.

IV. DATA

A. GROUND-BASED CLOUD IMAGES

In this paper, we use ground-based cloud images and corresponding meteorological data collected continuously for 365 days in 2021 from the National New Energy Laboratory public database [27]. The ground-based cloud images are taken by the all-sky imager Yankee TSI-880, and the meteorological data are collected by specialized sensors.

The ground-based cloud images from January 1, 2021 to December 31, 2021 are daytime data taken during the maintenance period of the equipment, with a frequency of every 10 minutes/photo and a fixed angle of the lens, and the sun cannot be blocked because there is no shade strip during the maintenance period. 26254 ground-based cloud images are finally collated and obtained in the format of 3 × 288 × 352 images. First, for a single ground-based cloud image the black fill is performed and the original size 288 × 352 is scaled to 256 × 256. before entering the neural network, the image needs to be converted into a Tensor type readable by the deep learning framework. Considering that the activation function in the final generated image of the network is *Tanh*, which squeezes the output value to between [-1, 1] so the values of the input image need to be transformed to between [-1, 1].

B. METEOROLOGICAL DATA

The meteorological data from January 1, 2021 to December 31, 2021 is corresponding to the ground-based cloud image, but with a shorter sampling frequency of every 1 minute/strip, and finally collated to obtain 525,600 time-series data with 225 categories of meteorological features in each data such as CR3000 Zen Angle, Global LI-200, Zenith Angle, etc. These data have different units and different value ranges. For this meteorological data, feature engineering is needed first, which mainly includes normalization and variance filtering. Normalization is mainly to deflate the values to between [0, 1] which is convenient for subsequent input into the network, and the calculation formula is shown in (12). Variance filtering mainly filters out features that do not change or change very little throughout the time dimension, which are meaningless for network modeling, and the number of features filtered out by feature variance varies with the threshold value as shown in Figure 8. Finally, the variance filtering was performed with the threshold value taken as 0.005, and a total of 58 feature variables were filtered out, leaving 167, which is a suitable dimension size for the input network. The reason for not considering the calculation of correlation between features is that when the data is normalized, the distribution between features is mostly similar, and there is no clear criterion to set the threshold to filter out the features with high correlation, so better results can be obtained by keeping these data for the neural network to learn by modeling in the high-dimensional space. Excluded indicators such as CR3000 RTwr RH, Precipitation (Accumulated), etc.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{12}$$

where x is the original data; x_{min} is the minimum value in the data; x_{max} is the maximum value in the data. Further, to make the range of values of the meteorological data consistent with the cloud images, it needs to be further scaled to between [-1, 1].

C. TIMES-SERIES DATASET

For the previous processed ground-based cloud images and meteorological data, they need to be made into a data format that is convenient for model input. Firstly, the shooting frequency of the ground-based cloud images is set to the baseline sampling frequency, and the entries corresponding to the moment when the ground-based cloud images were taken are extracted from the meteorological data, 26254 entries in

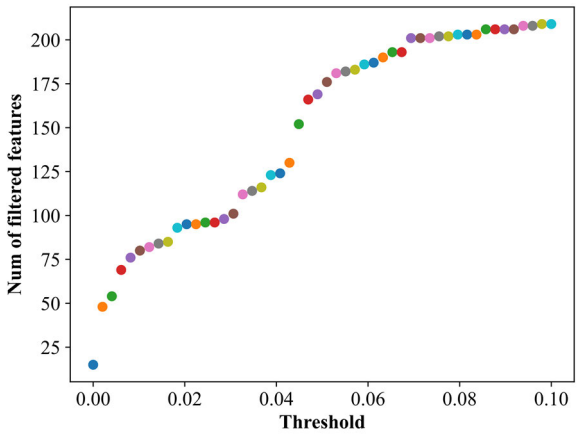

FIGURE 8. Variance filtering results at different thresholds.

total. Then, observation of the ground-based cloud images reveals that the shooting time is the daytime of each day, and the image changes continuously during the day, whereas the images across days are abruptly changed, so the days are divided as time points. Each sample of the time-series dataset is composed of ground-based cloud images + meteorological data. The ground-based cloud images are continuous images of the first 60 minutes of the day (the first 6 cloud images) and the next 10 minutes of the desired forecast (the 7th cloud image), with the interval between the start moments of each sample set to 20 minutes. The meteorological data are also collated according to this method. Finally, 7905 samples are collated, and then the training set and test set are randomly divided, with 80% of the training set and 20% of the test set. The process of dataset is shown in Figure 9.

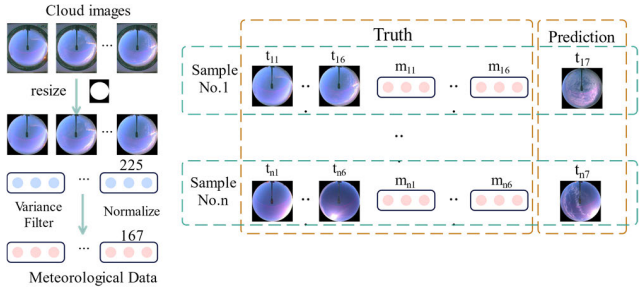

FIGURE 9. Dataset process.

V. RESULTS AND DISCUSSION

A. IMPLEMENTATION DETAILS

This section first determines the relevant settings of the experiment, and then conducts comparative experiments on CloudpredNet and other time-series image prediction models to verify the generalization performance of the proposed model.

The experimental platform comprised a server containing an Intel(R) Core(TM) i9-9900K 3.60GHz CPU, a NVIDIA GTX 2080Ti, and a memory with 32GB. The operating system is Windows 10 Professional Edition. The software environment is Python3.10 in Pytorch 1.13.

For the models, the encoder network (TimeSformer and TimesNet) and the generator network are trained from scratch without pre-training. For the parameters in the networks, we use random initialization with normal distribution. The use of Dropout between some layers prevents the model from being overfitted. In the training phase. We use AdamW optimizer and to update the parameters of the network. AdamW is an improved version of Adam, mainly adding weight decay and L_2 regularization. weight decay reduces a portion of the weights at each iteration, which can control the magnitude of the change in weight values and prevent gradient explosion, and L_2 regularization can reduce the tendency of overfitting. Set the optimization parameters are $betas = (0.8, 0.999)$ and $weight\_decay = 1 \times 10^{-4}$. The initial learning rate is set to 2×10^{-5}. limited by the experimental equipment, the total number of training iterations is set to 100, the batch size is set to 8, and the auto mixing precision (AMP) technique is used in the training phases to reduce the CUDA memory usage.

To monitor the quality of the generated images better, Peak Signal to Noise Ratio (PSNR) is also used as one of the evaluation metrics in dB, the larger the better [28]. In the calculation process of PSNR, two pictures need to be given, namely the predicted picture and the real picture, and a series of calculations are performed on the two to obtain the similarity score. The PSNR of the ground truth is the true value of the ground-based cloud image in the last 10 minutes. PSNR's calculation formula is:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{13}$$

where, MSE is the mean square error; MAX_I indicates the maximum value of the image point color.

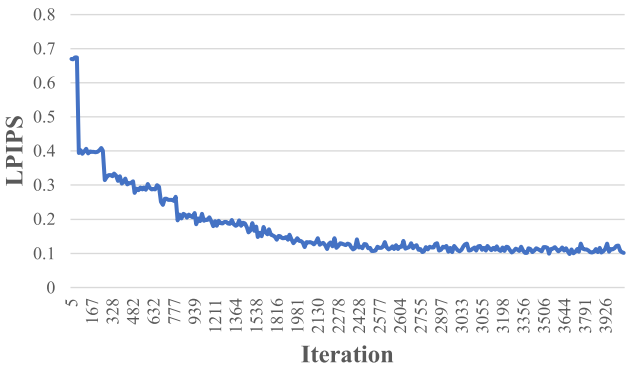

FIGURE 10. The curve of LPIPS with the number of iteration steps.

B. PERFORMANCE ANALYSIS BASED ON CLOUDPREDNET

In this section, the experiments are based on the CloudpredNet model. We use CloudpredNet to predict the next 10 minutes cloud image. Figure 10 to Figure 13 show the change curves of $LPIPS$, $SmoothL_1Loss$ and $PSNR$ with the

FIGURE 11. The curve of SSMI with the number of iteration steps.

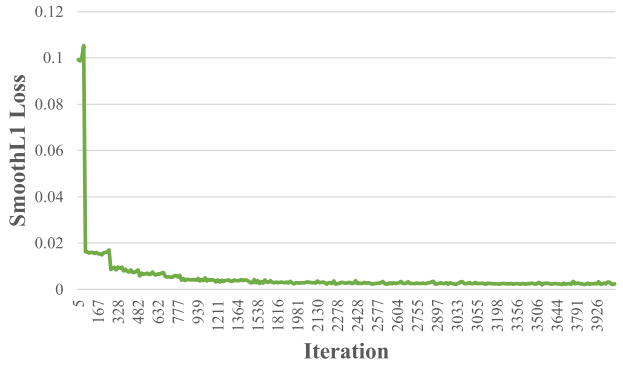

FIGURE 12. The curve of SmoothL_1 Loss with the number of iteration steps.

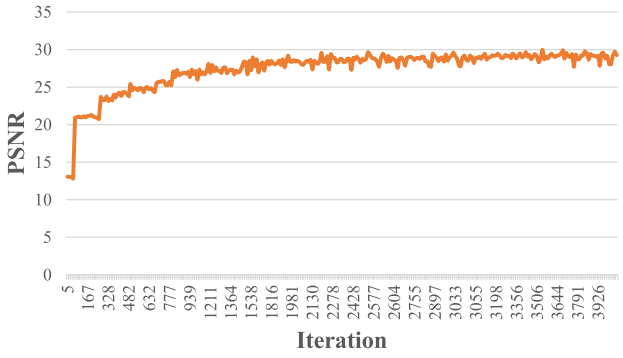

FIGURE 13. The curve of PSNR with the number of iteration steps.

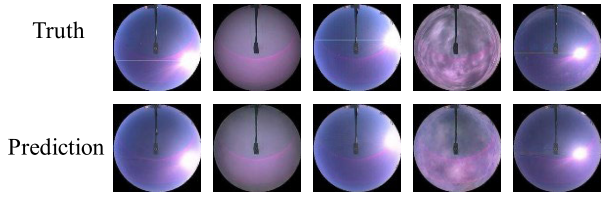

FIGURE 14. CloudpredNet prediction results and true values.

number of iteration steps, respectively. *LPIPS* finally converges to 5.86×10^{-2}, *SSIM* finally converges to 0.947, *Smooth*L_1*Loss* finally converges to 1.38×10^{-3}, and *PSNR* finally converges to 31.610. Some of the images generated

TABLE 2. Comparison of results from different models.

Models	*LPIPS*↓ ($\times 10^{-2}$)	*SSIM*↑	L_1*Loss*↓ ($\times 10^{-3}$)	*PSNR*↑
ConvLSTM	7.18	0.683	2.62	25.134
SA-ConvLSTM	6.01	0.889	1.77	28.967
PredRNN	7.22	0.725	2.22	25.438
PredRNN++	6.68	0.812	2.03	26.256
PredRNNV2	5.91	0.913	1.63	30.002
MCNet	7.01	0.764	2.11	25.978
MIM	6.38	0.854	1.99	28.013
CloudpredNet	**5.86**	**0.947**	**1.38**	**31.610**

↓ means the smaller the value, the better. ↑ means the larger the value, the better.

by CloudpredNet on the test set and the corresponding real data are shown in Figure 14. The predicted cloud images are already very similar to the real images. However, some details in the image are not well predicted, such as the reflection of the sun due to the lack of shading bands.

C. PERFORMANCE ANALYSIS BASED ON OTHER MODELS

To further validate the performance of CloudpredNet, this section experimentally compares CloudpredNet with other temporal image prediction models to predict cloud images for the next 10 minutes, including ConvLSTM [13], SA-ConvLSTM [15], PredRNN [16], PredRNN++ [17], PredRNNV2 [18], MCNet [29], and Memory-In-Memory (MIM) [30]. Most of these models are designed with the idea of "convolutional + recurrent neural network" structure, so they can be compared with the Transformer architecture-based models proposed in this section. The results of the different models are compared in Table 2, and the images predicted by the different models and the corresponding real data are shown in Figure 15.

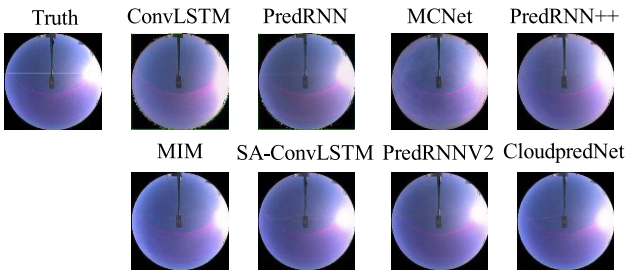

FIGURE 15. Different models prediction results.

According to Table 2 and Figure 15, ConvLSTM performs the worst, the generated images are blurrier, and the cloud images predicted by ConvLSTM and PredRNN have obvious noise, and the differences with the real cloud images are large. The subsequent improved models such as SA-ConvLSTM and PredRNNV2, which add attention mechanism, memory unit decoupling and other advanced methods to the original

ConvLSTM unit, achieve better prediction results. From the generated images, the general information of the next 10-minute cloud image can be predicted more accurately. The attention mechanism can extract the more important information from the data by operations such as matrix multiplication [31], which plays a key role in cloud image prediction. CloudpredNet performs the best and generates images with the highest quality. From the color point of view, the color predicted by CloudpredNet is closer to the real image, whereas the prediction results of other models have larger values for the red channel and the overall image color is reddish. In conventional model training, *MSE* is mostly used as the loss function, and the predicted value and the real value are expected to be as consistent as possible at each pixel, but this will lose some image style and texture information. Therefore, this paper combines *SSIM* and *LPIPS* on the basis of $SmoothL_1Loss$. These two indicators measure properties such as brightness, contrast, and structural similarity, and complement the insufficiency of $SmoothL_1Loss$. On the other hand, these detailed features are highly correlated with the content measured by *PSNR*, which can improve the *PSNR* score very well. ConvLSTM and PredRNN only get about 25 of PSNR, which means the structure of the model determines the upper limit of the model's performance and the structural defects of the traditional model make it impossible to generate images well, thus affecting the performance of indicators such as *PSNR*. Although these models have been improved the improvement of related indicators is still very limited. Since the human eye cannot judge the quality of the generated images only by the images when the indicators reach a certain level. Therefore, a combination of evaluation indicators is needed to judge.

In this ground-based cloud image movement prediction task, the parameter amount, training time and prediction time for a cloud image of each model are listed in the Table 3. Among these models, the amount of parameters of SA-ConvLSTM is the smallest, and the training time is relatively fast. Due to the extensive use of the attention mechanism in CloudpredNet, it can make full use of hardware resources for parallel operations. At the same time, the auto mixing precision (AMP) technique can further accelerate the training. Although the proposed model has a large number of parameters, its training and prediction time are acceptable combining the prediction performance shown in Table 2.

D. PERFORMANCE ANALYSIS ON LARGER TIME SCALES

To explore the prediction performance of CloudpredNet on a larger time scales, we take the output of the previous moment prediction as the input of the next moment prediction and calculate the metric with the real image including the next 20 and 30 minutes. The results are shown in Table 4. According to Table 4, the performance starts to decrease as the prediction time increases. Since we use the predicted values as inputs for the next moment, not the true values, this leads to an accumulation of errors, which reduces the prediction performance.

TABLE 3. Comparision of amount of parameters, training time and prediction time.

Models	Amount of Parameters(M)	Training Time(h)	Predicting Time(s)
ConvLSTM	16.60	1.2	0.084
SA-ConvLSTM	10.47	1.4	0.086
PredRNN	13.80	1.3	0.101
PredRNN++	13.23	1.3	0.102
PredRNNV2	23.86	1.5	0.094
MCNet	19.11	1.2	0.111
MIM	28.53	1.2	0.120
CloudpredNet	37.45	2.3	0.205

TABLE 4. Prediction results on different time scales.

Models	Time Scales	$LPIPS$ ($\times 10^{-2}$)	$SSIM$	L_1Loss ($\times 10^{-3}$)	$PSNR$
ConvLSTM	20min	8.79	0.596	4.33	21.002
	30min	10.69	0.554	4.69	19.850
SA-ConvLSTM	20min	7.99	0.800	2.43	26.788
	30min	8.53	0.767	2.55	22.961
PredRNN	20min	9.03	0.696	3.98	21.667
	30min	10.07	0.633	4.22	19.362
PredRNN++	20min	8.23	0.750	3.15	24.911
	30min	9.56	0.711	3.99	21.728
PredRNNV2	20min	7.95	0.812	2.31	**27.882**
	30min	8.82	0.778	3.41	23.484
MCNet	20min	8.10	0.774	2.80	23.424
	30min	9.95	0.721	3.69	20.643
MIM	20min	8.31	0.801	2.46	26.939
	30min	8.80	0.748	2.82	22.566
CloudpredNet	20min	**7.93**	**0.822**	**1.98**	27.383
	30min	**8.12**	**0.798**	**2.36**	**25.528**

However, CloudpredNet leads in most evaluation metrics on the 20-min and 30-min time scales.

VI. CONCLUSION

In this paper, we propose CloudpredNet, an ultra-short-term ground-based cloud movement prediction model based on the "encoder-generator" architecture, which predicts the ground-based cloud images for the next 10 minutes based on the first 60 minutes of ground-based cloud and weather data. The encoder part of the model is based on the Transformer structure, which can obtain stronger feature extraction capability, and the generator part is designed with a deep two-dimensional transposed convolutional network to fully parse the valid feature information from the encoder, thus improving the quality of the generated images. Compared with other existing image timing prediction models,

we demonstrate the excellent performance of the proposed model in the task of predicting cloud images.

However, due to the limitation of the experimental equipment, the selected series length is relatively short, only the first 60 minutes of data are used, and the predicted time scale is small, only ultra-short-term predictions are performed, and no experiments are conducted for medium-term, and long-term predictions, which is very meaningful for long-term weather prediction [32]. At the same time, the collected dataset is small and should be expanded to 10 times its size to allow the model to learn its intrinsic connections more fully.

ACKNOWLEDGMENT

The authors acknowledge the National Renewable Energy Laboratory for providing the data used in this article.

REFERENCES

[1] Y. Guo, T. Zhu, Z. Li, and C. Ni, "Auto-modal: Air-quality index forecasting with modal decomposition attention," *Sensors*, vol. 22, no. 18, p. 6953, Sep. 2022, doi: 10.3390/s22186953.

[2] G. E. P. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *J. Amer. Stat. Assoc.*, vol. 65, no. 332, pp. 1509–1526, Dec. 1970, doi: 10.1080/01621459.1970.10481180.

[3] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003, doi: 10.1061/(ASCE)0733-947X(2003)129:6(664).

[4] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul. 1998, doi: 10.1109/5254.708428.

[5] F. Zhang and L. J. O'Donnell, "Support vector regression," in *Machine Learning*, A. Mechelli and S. Vieira, Eds. New York, NY, USA: Academic Press, 2020, ch. 7, pp. 123–140, doi: 10.1016/B978-0-12-815739-8.00007-9.

[6] *Long Short-Term Memory*. Accessed: Apr. 11, 2023. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-24797-2_4

[7] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2017, pp. 1–11. Accessed: Apr. 11, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[9] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," 2020, *arXiv:2001.04451*.

[10] 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. 35th AAAI Conf. Artif. Intell.*, 2023, pp. 11106–11115. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17325

[11] 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.* Red Hook, NY, USA: Curran Associates, 2021, pp. 22419–22430, Accessed: Apr. 11, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html

[12] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-variation modeling for general time series analysis," 2022, *arXiv:2210.02186*.

[13] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2015, pp. 1–9. Accessed: Apr. 11, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/07563a3fe3bbe7e3ba84431ad9d055af-Abstract.html

[14] X. Shi, Z. Gao, L. Lausen, H. Wang, and D.-Y. Yeung, "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Proc. Adv. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2017, pp. 1–11. Accessed: Apr. 11, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/a6db4ed04f1621a119799fd3d7545d3d-Abstract.html

[15] Z. Lin, M. Li, Z. Zheng, Y. Cheng, and C. Yuan, "Self-attention ConvLSTM for spatiotemporal prediction," in *Proc. 34th AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 11531–11538, doi: 10.1609/aaai.v34i07.6819.

[16] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs," in *Proc. Adv. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2017, pp. 1–10. Accessed: Apr. 11, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/e5f6ad6ce374177eef023bf5d0c018b6-Abstract.html

[17] Y. Wang, Z. Gao, M. Long, J. Wang, and P. S. Yu, "PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning," in *Proc. 35th Int. Conf. Mach. Learn.*, Jul. 2018, pp. 5123–5132. Accessed: Apr. 11, 2023. [Online]. Available: https://proceedings.mlr.press/v80/wang18b.html

[18] Y. Wang, H. Wu, J. Zhang, Z. Gao, J. Wang, P. S. Yu, and M. Long, "PredRNN: A recurrent neural network for spatiotemporal predictive learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2208–2225, Feb. 2023, doi: 10.1109/TPAMI.2022.3165153.

[19] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Proc. Adv. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2016, pp. 1–9. Accessed: Apr. 11, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/8bf1211fd4b7b94528899de0a43b9fb3-Abstract.html

[20] P. Kumari and D. Toshniwal, "Long short term memory–convolutional neural network based deep hybrid approach for solar irradiance forecasting," *Appl. Energy*, vol. 295, Aug. 2021, Art. no. 117061, doi: 10.1016/j.apenergy.2021.117061.

[21] X. Su, T. Li, C. An, and G. Wang, "Prediction of short-time cloud motion using a deep-learning model," *Atmosphere*, vol. 11, no. 11, p. 1151, Oct. 2020, doi: 10.3390/atmos11111151.

[22] C. Crisosto, E. W. Luiz, and G. Seckmeyer, "Convolutional neural network for high-resolution cloud motion prediction from hemispheric sky images," *Energies*, vol. 14, no. 3, p. 753, Feb. 2021, doi: 10.3390/en14030753.

[23] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1933–1941. Accessed: Apr. 11, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/Feichtenhofer_Convolutional_Two-Stream_Network_CVPR_2016_paper.html

[24] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013, doi: 10.1109/TPAMI.2012.59.

[25] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2021, pp. 813–824. Accessed: Apr. 11, 2023. [Online]. Available: https://proceedings.mlr.press/v139/bertasius21a.html

[26] Z. Zheng, Y. Zhao, A. Li, and Q. Yu, "Wild terrestrial animal re-identification based on an improved locally aware transformer with a cross-attention mechanism," *Animals*, vol. 12, no. 24, p. 3503, Dec. 2022, doi: 10.3390/ani12243503.

[27] T. Stoffel and A. Andreas, "NREL Solar Radiation Research Laboratory (SRRL): Baseline Measurement System (BMS); Golden, Colorado (Data)," Nat. Renew. Energy Lab. (NREL), Golden, CO, USA, Tech. Rep. NREL/DA-5500-56488, Jul. 1981, doi: 10.7799/1052221.

[28] W. Zhao, Y. Zhao, L. Feng, and J. Tang, "Attention optimized deep generative adversarial network for removing uneven dense haze," *Symmetry*, vol. 14, no. 1, p. 1, Dec. 2021, doi: 10.3390/sym14010001.

[29] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," 2017, *arXiv:1706.08033*.

[30] Y. Wang, J. Zhang, H. Zhu, M. Long, J. Wang, and P. S. Yu, "Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9146–9154. Accessed: Apr. 11, 2023. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2019/html/Wang_Memory_in_Memory_A_Predictive_Neural_Network_for_Learning_Higher-Order_CVPR_2019_paper.html

[31] H. Zou and X. Sun, "3D face recognition based on an attention mechanism and sparse loss function," *Electronics*, vol. 10, no. 20, p. 2539, Oct. 2021, doi: 10.3390/electronics10202539.
[32] P. Kumari and D. Toshniwal, "Deep learning models for solar irradiance forecasting: A comprehensive review," *J. Cleaner Prod.*, vol. 318, Oct. 2021, Art. no. 128566, doi: 10.1016/j.jclepro.2021.128566.

LIANG WEI was born in Nanjing, Jiangsu, in April 1999. He received the B.S. degree in automation from Nanjing Forestry University, China, in 2021, where he is currently pursuing the master's degree. His research interests include ground-based cloud classification and deep learning in the field of computer vision.

TINGTING ZHU received the Ph.D. degree in pattern recognition and artificial intelligence from the School of Automation, Southeast University, in 2019. She is an Associate Professor with the College of Mechanical and Electronic Engineering, Nanjing Forestry University, China. She was a Visiting Student with the Department of Atmospheric and Oceanic Sciences, McGill University, Canada, from 2017 to 2018. She received the fellowship jointly awarded by Fonds de Recherche du Quebec—Nature et Technologies (FRQNT) and the China Scholarship Council. Her current research interests include machine learning, data processing and modeling, renewable energy generation forecast, and climate feedback.

YIREN GUO received the B.S. degree in automation from Nanjing Forestry University, China, in 2020, where he is currently pursuing the master's degree in control science and engineering. He received the funding named the Graduate Research Practice Innovation Plan of Jiangsu in 2021. His research interests include ground-based cloud classification and deep learning in the field of computer vision.

CHAO NI was born in Nanjing, Jiangsu, China, in 1979. He received the B.S. degree in automation from the Nanjing University of Science and Technology, Nanjing, in 2001, and the Ph.D. degree in control theory and control engineering from Southeast University, Nanjing, in 2008. He is currently a Professor with the Automation Department, Nanjing Forestry University, China. From October 2017 to November 2018, he was a Visiting Scholar with the University of Maryland, College Park, USA. His research interests include artificial intelligence in industrial application, data processing, and spectroscopy analysis.

QINGYUAN ZHENG received the M.S. degree in detection technology and automation from Hohai University, Nanjing, China, in July 2012. She is currently working with the School of Physics and Information Engineering, Iiangsu Second Normal University, Nanjing, China. She is also a senior engineer. Her current research focuses on the distributed control optimization and early failure warning in the smart tobacco production line.