Long Short-term Dynamic Graph Neural Networks: for short-term intense rainfall forecasting

Huosheng Xie
Fuzhou University
Fuzhou, Fujian, China
xiehs@fzu.edu.cn

WeiJie Wang
Fuzhou University
Fuzhou, Fujian, China
200320090@fzu.edu.cn

ABSTRACT

In practice, accurate and timely forecasting of short-term intense rainfall is critical, but the problem is extremely difficult because to its complicated spatial-temporal association. Although several spatial-temporal series forecasting methods have been used to rainfall prediction, these models continue to suffer from inadequate modeling of data's complicated intrinsic connection. We provide a new short-term intense rainfall prediction model that use two graph generators to model data correlations under distinct semantics, followed by a graph convolution module for information integration to fully extract data spatial-temporal information. Finally, a variant of recurrent neural network is employed to extract the temporal dependence. The experimental results on both datasets show that the model can model the spatial and temporal dependence across the data more effectively than the baseline model, and further improve the model's predictive performance for short-term intense rainfall.

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; *Supervised learning by regression.*

KEYWORDS

Short-term intense rainfall, spatial-temporal correlation, Graph Convolution

ACM Reference Format:
Huosheng Xie and WeiJie Wang. 2022. Long Short-term Dynamic Graph Neural Networks: for short-term intense rainfall forecasting. In *2022 5th International Conference on Machine Learning and Natural Language Processing (MLNLP 2022), December 23–25, 2022, Sanya,China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3578741.3578757

1 INTRODUCTION

Short-term intense rainfall events [1] are a kind of strong convective weather phenomenon, mainly occurring in the annual flood season. It is characterized by its suddenness and heavy rainfall, which can readily cause natural disasters such as mudslides and landslides, resulting in economic losses and casualties. Short-term rainfall forecasts within the next 3 hours are called nowcasting [10], while when rainfall is greater than or equal to 30mm in the next 3 hours, it will be recognized as a short-term intense rainfall event (Fujian Provincial Meteorological Bureau), for which accurate and timely forecasting is of great guidance for natural disaster prevention and preparation.

Numerical weather modeling (NWP), which relies on geophysical hydrodynamic theory, is a commonly used method for short-term rainfall prediction. It obtains rainfall forecasts for a future period by solving non-linear differential equations based on atmospheric physics theory [18] via a large computer. Much work has been proposed over the years to evaluate and improve [8, 9] the predictive effectiveness of NWP models, however, rainfall prediction involves complex, multi-scale interactions between observation sites and meteorological conditions, and the representation of physical processes in many NWP models is not well comprehended in existing work [14], leaving the simulation results of numerical models with varying degrees of systematic error and spatial and temporal bias.

In recent years, some researchers have started to use machine learning methods [11] for rainfall prediction with the support of a large amount of meteorological observation data. Early efforts mainly used traditional machine learning methods such as History Average (HA), Support Vector Machine (SVM) [29], Autoregressive Integrated Moving Average (ARIMA) [6], and other models to capture the correlation of rainfall processes in the time dimension, which ignore the spatial dependence of meteorological variables and their intrinsic associations. Deep learning methods can be used to automatically extract complex spatial-temporal features and physical correlations from weather data [12], which has led researchers to experiment with combining deep neural networks for rainfall prediction [17, 19, 30]. However, The application of deep learning in the short-term intense rainfall prediction field is still very challenging due to the following factors:**First**, the rainfall process changes dynamically with time and has high uncertainty.**Moreover**, there are complex correlations between meteorological observation sites in the spatial dimension.**Third**, due to the severe data distribution imbalance between rainfall data and non-rainfall data, short-term intense rainfall events only accounts for 0.11% of all events, which further increases the difficulty for deep learning models to model the spatial and temporal dependence of intense rainfall events.

To address these challenges, a Long-Short term Dynamic Graph Neural Network (LSDGNN) is proposed to dynamically capture multi-scale information on temporal, spatial and meteorological factors of rainfall data. In summary, our main contributions are as follows:

MLNLP 2022, December 23–25, 2022, Sanya, China

ACM ISBN 978-1-4503-9906-7/22/12…$15.00
https://doi.org/10.1145/3578741.3578757

- Long-Short Term Dynamic Graph Neural Networks, a spatial-temporal sequence prediction model, is proposed for short-term intense precipitation forecasting, which can effectively model the spatial-temporal dependence in the rainfall process.
- The Dynamic Time Warping (DTW) [13] algorithm is introduced to construct long-term dependency graphs of nodes, and in addition, a dynamic graph generator is proposed to improve the model's focus on recent spatial-temporal information, and simultaneous graph convolution operations using both dependency graphs can extract dynamic spatial-temporal features of meteorological data more comprehensively.
- To cope with the problem of unbalanced rainfall data distribution, we propose a weighted loss function that can significantly improve the model's focus on short-term intense rainfall events.
- Enriched experiments on two rainfall datasets were conducted, which showed that our model has significant improvements over the NWP model and other baseline models.

2 BACKGROUND

2.1 PROBLEM FORMULATION

The short-term intense rainfall problem can be translated into a spatial and temporal series problem of rainfall prediction for a future period based on historical meteorological observation data. The historical observation data of the graph structure(As shown in Fig. 1)can be expressed as $X \in R^{T \times N \times F}$, where T is the sequence length, N denotes the number of nodes, the F refers to each node's feature dimension. A frame $X_t \in R^{N \times F}$ in the data can be considered as a graph signal containing meteorological observations of each node at moment t.

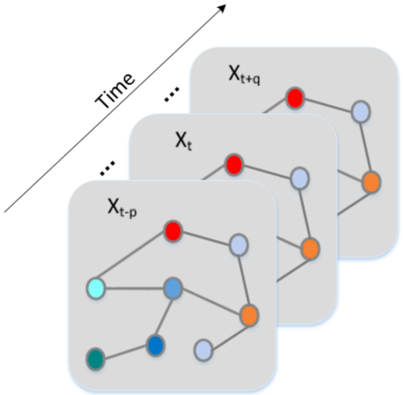

Figure 1: Spatio-temporal sequence data of the graph structure, with the connecting lines representing the spatial dependencies between the nodes

The primary goal of this task is to obtain a mapping from historical observations to rainfall forecasts for a future period by modeling the intrinsic correlation of historical observations at multiple scales, which is represented as follows:

$$\{X_{t-p}, X_{t-p+1}, \cdots X_t\} \xrightarrow{f} \{X_{t+1}, X_{t+2}, \cdots X_{t+q}\}$$

2.2 Data Sources

The original data were obtained from four datasets provided by Fujian Meteorological Bureau (see Table 1 for details). Stations is the ground measured rainfall data collected from 2170 meteorological observation stations in Fujian Province, which contains longitude, latitude, and 3h measured rainfall. In addition, the NWP products Come from Weather Research and Forecasting (WRF) method and the NWP products of the European Centre for Medium-Range Weather Forecasting (ECMWF) are used in this study. ECMWF provides two datasets with horizontal resolutions of 0.25×0.25 and 0.125×0.125 respectively, which we abbreviate as $ECMWF_{250}$ and $ECMWF_{125}$ in the following.

Table 1: Details of the data set

Dataset	Stations	ECMWF250	ECMWF125	WRF
Horizontal resolution	23.5°N-28.37°N 115.8°E-120.6°E	23.5°N-28.5°N 115.5°E-121°E		23.5°N-28.45°N 115.5°E-120.9°E
Time Range	20150201-20181231			20170101-20181231
Number of stations or Gride points	2170	23 × 21	45 × 41	62 × 56
Number of Available Features	3	113	26	24

2.3 Related Work

Spatial-temporal [16] series forecasting method, one of the basic methods for spatial-temporal data modeling, is widely used in various domains, such as weather forecasting, economic statistics, traffic forecasting, etc. How to accurately extract multi-scale information from spatial-temporal data is the key to the spatial-temporal sequence prediction problem. Traditional models based on time series forecasting, such as Long Short-Term Memory (LSTM) [5] , Gated Recurrent Unit (GRU) [2], and other Recurrent Neural Network (RNN)-based structures are used to aggregate time dimensional information. Deep Belief Network (DBN) [24], a multi-layer probabilistic generative network has also been attempted for rainfall prediction. However, these time-series-based forecasting models only consider rainfall data correlation in the time dimension and ignore rainfall data spatial dependence. To further improve time-series forecasting accuracy, many models have started to integrate the spatial dependence of data by various methods. According to the method of aggregating spatial information, we divide them into two categories in this paper: Convolutional Neural Network (CNN) and Graph Convolutional Network (GCN).

Convolutional operations capture the spatial correlation of data in CNN-based spatial-temporal sequence prediction models. ConvLSTM [15] proposes an encoder-predictor-based structure for the rainfall prediction problem by replacing the fully connected operation in LSTM with a convolutional operation to capture both temporal and spatial dimensional information. Since then, some works have been published [20, 21], that improve models' ability to capture the intrinsic correlation of spatial-temporal data by investigating improved methods for the internal structure of LSTM cells. The convolution operation was initially designed to capture the hidden state of the picture, requires the operation data to be regular Euclidean space data. Much real-world data is based on graph-structured data, which is irregular and has a variable number of neighboring nodes, making the application of convolution operations in the graph domain difficult.

In Graph convolutional networks each node in a graph continuously adjusts its state until it reaches equilibrium by aggregating information from relevant nodes [22]. In the subject of spatial-temporal sequence prediction, GCN is widely employed [27]. Spatial-Temporal Graph Convolutional Networks (STGCN) [28], by using Gated CNNs to extract information in the temporal dimension and graph neural networks for extracting patterns and features that are highly meaningful in the spatial dimension. By incorporating temporal and spatial attention modules to aid graph convolutional networks for spatial-temporal information aggregation, the Attention-based Spatial-Temporal Graph Convolutional Network (ASTGCN) [4] was created. However, since the attention mechanism requires plenty of computational resources, it can lead to difficulties in model training as the number of sites increases. Dynamic Graph Convolutional Recurrent Network(DGCRN) [7] constructs dynamically changing dependency graphs through dynamic graph neural networks, but DGCRN uses distance-based dependency graphs as a priori knowledge, which may ignore the semantic correlation of site spaces (e.g., two sites with similar spatial-temporal models, but at a spatial distance).The hierarchical dynamic graph network(HDGN) [25] was proposed to dynamically generate dependency graphs by graph pooling methods, but it is still inadequate for multi-scale spatial-temporal information extraction.

3 METHODOLOGY

3.1 Data processing

The ECMWF and WRF datasets are simulated forecast products and lack ground-truthed rainfall data. Therefore, in this paper, the raw datasets are manipulated as follows. To begin, we interpolate the actual measured rainfall data from the Stations dataset to both the ECMWF and WRF datasets using the Inverse Distance Weight (IDW) [26] interpolation method. Combining the measured data from the stations and the simulated meteorological data from NWP can enrich the diversity of the data and better evaluate the model prediction effect. For the ECMWF and WRF datasets, we used simulations from 12 to 33 hours, considering the instability of the simulations for the first 12 hours of the NWP product. Furthermore, because different meteorological features may be redundant, the Box Difference Index (BDI) [3] is utilized to feature selection. The higher the index, the better the feature for predicting short-duration intense rainfall events. The BDI value of each of these features is calculated as follows:

$$BDI = \frac{m_{30} - m_{0-30}}{\sigma_{30} + \sigma_{0-30}} \tag{1}$$

Where m_{30} and m_{0-30} denote the mean value of the feature on data with rainfall greater than 30 and rainfall in the range of 0-30, respectively. σ_{30}, σ_{0-30} denote the standard deviation of the feature on data with rainfall greater than 30 and rainfall in the range of 0-30, respectively. After the calculation, the obtained BDI values are sorted from highest to lowest for feature extraction.

Following feature selection, we convert our data into sequence data for the model, which is constructed mainly according to the following steps: (1) There are two different horizontal resolution data in the ECMWF dataset, and some of their features overlap. We fuse the two datasets into one dataset by stitching the spatially overlapping lattice points between ECMWF125 and ECMWF250. (2) The dataset has missing values due to a variety of practical causes, and we interpolate the data using linear interpolation. (3) The data is serialized as model input using a sliding window with a step size of one.

3.2 Network Architecture

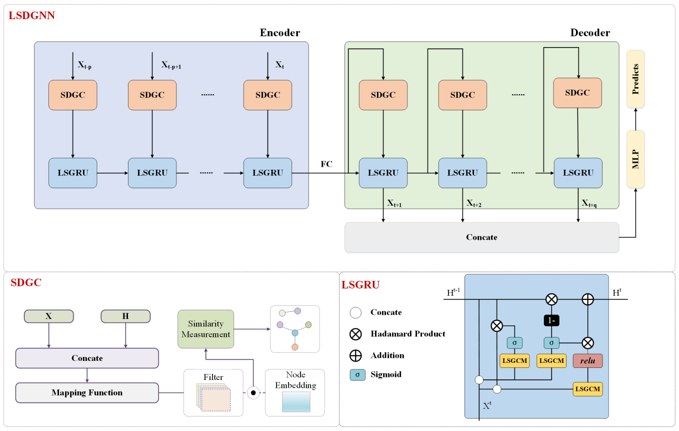

Figure 2: The encoder-decoder architecture is used in the general framework of LSDGNN. It is made up of three major parts: the Short-term dynamic graph constructor (SDGC), the LSGRU, and the Long Short-term Graph Convolution Module (LSGCM).

As shown in Fig. 2, in this work we propose an LSDGNN for short-term intense rainfall prediction based on an encoder-decoder structure.To more thoroughly reflect the spatio-temporal correlation between the data, the model used two constructors to build long-term and short-term dependency graphs. To fully apply the two dependencies, a long and short-term graph convolution module is proposed. We use LSGCM instead of the fully connected operation in the traditional GRU module for fully capturing spatial-temporal correlations.

3.2.1 *Graph Constructor.*

In graph neural networks, the definition of the neighbor matrix determines how the nodes and their neighbor nodes aggregate information during the graph convolution process. To more thoroughly reflect the spatio-temporal association between the data, the model used two constructors to create long-term and short-term dependency graphs. Following that, we will go over the construction methods for both dependency graphs separately.

Long-term dependency graph constructor: It is unreasonable to use geographical distance alone to determine the spatial correlation between nodes. Due to the geographical and atmospheric circulation movement conditions, some nodes with close geographical distance instead have completely different rainfall patterns, while some nodes with farther distance have a strong semantic correlation between them. To evaluate the spatial dependence between nodes, the DTW algorithm is introduced to calculate the time series acquaintance of different nodes.

Algorithm 1: Long-term graph construction

```
Input: Rainfall observation sequences at N
    nodes:$\{X^1_{rain}, X^2_{rain}, \ldots, X^N_{rain}\} \in R^{N\times T}$
Output: Long-term graph:$A^l$
for $i = 1; i <= N; i++$ do
    for $j = 1; j <= N; j++$ do
        $D_{i,j}$=NDC($X^i_{rain}, X^j_{rain}$);    // Defined in Alg.2
        $A_{i,j} = exp(-D_{i,j})$;
    end
    Extracting the kth largest element in $A_i$:$max_k$
    for $j = 1; j <= N; j++$ do
        if $A_{i,j} < max_k$ then
            $A_{i,j} = 0$ ;          // Filtering redundant
              associations
        end
    end
end
return $A^l = A$
```

Algorithm 2: Node distance calculation(NDC)

```
Input: Rainfall observation sequences at two nodes:
$X^m_{rain} = \{x^m_1, x^m_2, \ldots, x^m_T\}, X^n_{rain} = \{x^n_1, x^n_2, \ldots, x^n_T\} \in R^T$
Output: $dist$
for $i = 1; i <= T; i++$ do
    // Limit the search to $s$
    for $j = max(1, i - s); j <= min(T, i + s + 1); j++$ do
        $T_{i,j} = (x^m_i - x^n_j)^2$;
        $D_{i,j} = T_{i,j} + min(D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1})$ ; // Omit
        the boundary judgment
    end
end
return $dist = \sqrt{D_{T,T}}$
```

The DTW algorithm is biased to focus on the shape of two series without caring whether the two series are similar at each time point, which means that the DTW algorithm allows two series to have deviations in time. This evaluation method meets our needs for rainfall prediction. For precipitation sequence data between different nodes, we propose a long-term graph construction algorithm(Alg. 1), which calculates the semantic distance between nodes based on the DTW algorithm. In addition, a filter is used in this algorithm in order to prevent the model from aggregating too much redundant information. This filter keeps the largest value of k in each row of the dependency graph and sets the rest to zero.From this we obtain the long-term dependency graph A^l

Short-term dynamic graph constructor: The construction of a long-term dependency graph gives our model the ability to cluster long-term dependency information among nodes. However, in predicting the future rainfall of an area, the model needs to capture not only long-term dependencies among nodes, but also short-term dependencies are indispensable. In addition, a static dependence graph is difficult to characterize the dynamic changes between nodes at short periods. Therefore, we propose a short-term dynamic graph generator based on node embedding to generate short-term dependence graphs between nodes.

For each moment the meteorological observations $M_t \in R^{B\times N\times F}$ and the hidden state $H_t \in R^{B\times N\times h}$ are cascaded as inputs to the short-term graph constructor, where B is the Batch Size, and h is the feature dimension of the hidden state.

$$\mathrm{I}_t = M_t \| H_{t-1} \tag{2}$$

$\|$ is the cascade function. Two node embeddings E_1 and E_2 are applied in the network whose parameters are learnable during the network training. After that, we generate two dynamic filters by two mapping functions that model the short-term correlation between nodes.

$$F^t_1 = \phi(\mathrm{I}_t)\ ,\ F^t_2 = \psi(\mathrm{I}_t) \tag{3}$$

The concern for short-term spatial dependence of the node embedding is enhanced via a Hadamard product of two dynamic filters on the node embedding, denoted as follows:

$$E^t_1 = tanh(\alpha(F^t_1 \odot E_1))\ ,\ E^t_2 = tanh(\alpha(F^t_2 \odot E_2)) \tag{4}$$

Where $\odot$ denotes Hadamard product. Finally, by the similarity measure between nodes [23], we obtain the short-term dynamic dependence graph A^s_t at moment t.

$$A^s_t = relu(tanh(\alpha(E^t_1(E^t_2)^T - E^t_2(E^t_1)^T)))) \tag{5}$$

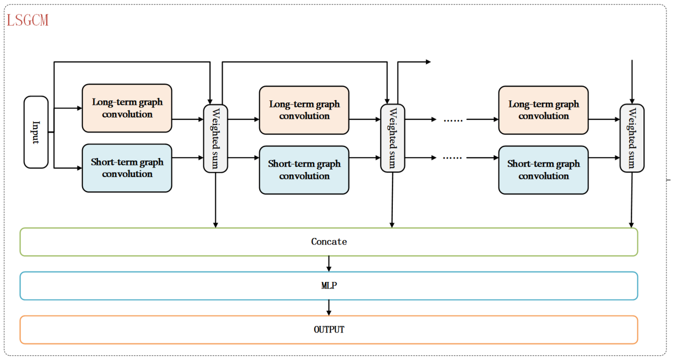
LSGCM
Input
Long-term graph convolution
Short-term graph convolution
Weighted sum
Long-term graph convolution
Short-term graph convolution
Weighted sum
......
......
Long-term graph convolution
Short-term graph convolution
Weighted sum
Concate
MLP
OUTPUT

Figure 3: Long and short-term graph convolution module, respectively, using long and short-term dependencies for graph convolution

3.2.2 *Spatial dependency modeling.*

We employ the LSGCM in this study to execute graph convolution operations on the two created dependency graphs separately and weight the results of the two modules to obtain the final graph convolution results, which are represented as follows:

$$\begin{gathered} H^k = \alpha H^{k-1} + \beta\theta_{l\star G}(H^{k-1}, A^l) + \gamma\theta_{s\star G}(H^{k-1}, A^s_t) \\ H_{out} = (H^1 || H^2 || H^3 || \ldots || H^k) W^k \end{gathered} \tag{6}$$

Where H^{k-1} is the output of the previous graph convolution layer, $H^0 = H_{in}$ is the module's input, α, β, and γ are hyperparameters used to control the original input and the weights of each component. $\theta_{l\star G}(H^{k-1}, A_l), \theta_{s\star G}(H^{k-1}, A_t^s)$ represent graph convolution operations using long-term and short-term dependency graphs, respectively. W^k denotes the learnable parameters of the kth layer used to perform the input-output dimension transformation. In addition, the graph convolution in the paper uses bi-directional graph convolution, which allows better aggregation of correlations in different directions, defined as follows:

$$\begin{gathered} \theta_{\star G}(H_{in}, A) = \tilde{A} H_{in} \theta_1 + \tilde{A}^T H_{in} \theta_2 \\ D_i = \sum_{j=1}^{N} A_{i,j}, \tilde{A} = D^{-1} A \end{gathered} \tag{7}$$

3.2.3 Temporal dependence modeling. GRU is a variant of RNN with powerful capability for modeling temporal dependence modeling, which can effectively moderate the gradient disappearance and gradient explosion problems. We obtained LSGRU by replacing the fully connected operation in GRU with the LSGCM:

$$r^t = \sigma(\text{LSGCM}(M_t \| H_{t-1})) \tag{8}$$

$$z^t = \sigma(\text{LSGCM}(M_t \| H_{t-1})) \tag{9}$$

$$c^t = relu(\text{LSGCM}(M_t \| (r^t \odot H_{t-1})) \tag{10}$$

$$H_t = (z^t \odot H_{t-1}) + (1 - z^t) \odot c^t \tag{11}$$

Where M_t and H_t are GRU's input and output at time t, respectively. σ denotes the sigmoid activation function, r^t and z^t the reset and update gates at time t, the LSGCM defined by Eq.6.

3.2.4 Loss Function. To improve the sensitivity of the model to short-term intense rainfall events, we propose a weighted loss function. This loss function can significantly improve model prediction accuracy while also mitigating the data distribution imbalance problem. $\hat{Y}$ denotes actual rainfall, Y denotes predicted value, and δ denotes rainfall events > 30, as a percentage of total rainfall events.

$$Loss = \begin{cases} \frac{1}{\delta}(Y - \hat{Y})^2, & \hat{Y} > 30, \\ (Y - \hat{Y})^2, & otherwise. \end{cases} \tag{12}$$

4 EXPERIMENT

4.1 Experiment settings

We divided the data into training and test sets by year, on the ECMWF dataset we used 2015 to 2017 data as the training set and 2018 data as the test set. On the WRF dataset, we used January 2017 to June 2018 data as the training set and June 2018 to December 2018 data as the test set. The experiments were conducted in a Python 3.6 environment, mainly based on the Pytorch framework. In this paper, 12 historical observation frames are used as model inputs to predict the future 2-frame results. The Adam optimizer is used in the model, and the initial learning rate is set to 0.001. For maximum GPU load, BatchSize is set to 48 in the ECMWF dataset and 2 in the WRF dataset. The amount of graph convolution layers is set to 2, the node embedding dimension is set to 40, and the hidden layer dimension is set to 32.

4.2 Evaluation indicators

In this paper, the following three main indicators are used to conduct the predictive assessment :(1) The critical success index (CSI), which is frequently used to evaluate rainfall prediction effects, is based on the idea that the higher the value, the greater the prediction effect. (2)An improvement on the CSI, the Equitable Threat Score (ETS) penalizes missed and false alarms, making scoring more equitable. (3) False Alarm Ratio (FAR), which is the ratio of model false alerts for rainfall events; the lower the value, the better.

According to Table 2, we define each evaluation indicator as follows:

$$\text{CSI}_i = \frac{A_i}{A_i + \sum B_i + \sum C_i} \tag{13}$$

$$\text{ETS}_i = \frac{A_i - R_i}{A_i + \sum B_i + \sum C_i}, R_i = \frac{(A_i + \sum B_i)(A_i + \sum C_i)}{A_i + \sum B_i + \sum C_i + D} \tag{14}$$

$$FAR_i = \frac{\sum B_i}{A_k + \sum B_i} \tag{15}$$

Where when i equals 1 represents 1 as the threshold, and i equals 2 represents 30 as the threshold.

Table 2: Classification of rainfall data according to two thresholds of 1 and 30

Actual	Predicted		
	0mm-1mm	1mm-30mm	>30mm
0mm-1mm	D	B_1	B_2
1mm-30mm	C_1	A_1	B_2
>30mm	C_2	C_2	A_2

4.3 Results

4.3.1 Baseline Comparison. We compare our method to six baseline methods, for all of which we use the default configuration from the original paper. All models are evaluated on two thresholds by the three evaluation metrics mentioned above. Tables 3, 4 show the performance of the baseline model and our model on the two datasets. LSDGNN achieves excellent experimental results on both datasets, especially on the prediction of short-term intense rainfall for the next 3h, our model achieves the best results. Due to the finer grid point data of the WRF dataset, we achieved better results on the WRF dataset compared to the ECMWF dataset.

Experimental results show that our model gets the best experimental results in the prediction of 3h cumulative rainfall on the majority of indicators. The results of our model in the FAR_1 index are slightly worse than those of the HA method, owing to the following two factors:(1) Our model is biased toward predicting short-term intense rainfall events, which will lead to large prediction results of the model, increasing in the false alarm rate (2) The HA model is conservative in its forecasts, so it can have a low false alarm rate but has poor or even no prediction capability for intense rainfall.

By evaluating the prediction performance of different models, we can observe that deep learning methods achieve better performance compared to traditional methods, and ConvLSTM capturing spatial

Table 3: Evaluation of the baseline method and the proposed method for future 3h and 6h predictions on the WRF DATASET

Model	Predict 3h			Predict 6h			Predict 3h			Predict 6h		
	CSI_1	ETS_1	FAR_1	CSI_1	ETS_1	FAR_1	CSI_2	ETS_2	FAR_2	CSI_2	ETS_2	FAR_2
ECMWF	0.205	0.159	0.722	/	/	/	0.0021	0.0019	0.9932	/	/	/
HA	0.103	0.087	**0.576**	0.086	0.065	0.865	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000
LSTM	0.131	0.096	0.832	0.113	0.071	0.868	0.0042	0.0020	0.9953	0.0016	0.0008	0.9965
ConvLSTM	0.174	0.140	0.814	0.163	0.131	0.816	0.0058	0.0030	0.9941	0.0032	0.0015	0.9973
DBN	0.181	0.145	0.790	0.165	**0.132**	0.806	0.0069	0.0046	0.9925	0.0036	0.0021	0.9969
ASTGCN	0.157	0.074	0.841	0.138	0.052	0.860	0.0114	0.0105	0.9787	0.0051	0.0034	0.9913
HDGN	0.116	0.027	0.875	0.116	0.014	0.872	0.0231	0.0212	0.9543	0.0082	0.0071	0.9780
LSDGNN	**0.252**	**0.190**	0.718	**0.185**	0.116	**0.798**	**0.051**	**0.0504**	**0.9262**	**0.0091**	**0.0087**	**0.9762**

Table 4: Evaluation of the baseline method and the proposed method for future 3h and 6h predictions on the ECMWF DATASET

Model	Predict 3h			Predict 6h			Predict 3h			Predict 6h		
	CSI_1	ETS_1	FAR_1	CSI_1	ETS_1	FAR_1	CSI_2	ETS_2	FAR_2	CSI_2	ETS_2	FAR_2
WRF	0.143	0.076	0.839	/	/	/	0.0047	0.0037	0.9953	/	/	/
HA	0.172	0.143	**0.523**	0.086	0.065	**0.564**	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000
LSTM	0.152	0.137	0.832	0.126	0.113	0.851	0.0059	0.0020	0.9932	0.0027	0.0011	0.9979
ConvLSTM	0.233	0.201	0.691	0.217	0.188	0.722	0.0088	0.0050	0.9847	0.0048	0.0026	0.9949
DBN	0.221	0.180	0.742	0.197	0.168	0.762	0.0089	0.0060	0.9810	0.0043	0.0030	0.9924
ASTGCN	0.145	0.051	0.853	0.126	0.032	0.870	0.0004	0.0004	0.9374	0.0003	0.0002	0.9742
HDGN	0.132	0.036	0.866	0.126	0.026	0.879	0.0353	0.0346	0.9219	**0.0139**	0.0003	0.9602
LSDGNN	**0.335**	**0.274**	0.608	**0.256**	**0.193**	0.682	**0.0757**	**0.0755**	**0.6887**	0.0098	**0.0097**	**0.8551**

dependence by CNN is significantly better than LSTM models, which shows the necessity of models capturing spatial dependence. By comparing the performance of the GCN-based model with the CNN-based model, we can find that the GCN model tends to achieve better results, which reflects the outstanding performance of the GCN modeling spatial dependence.

4.3.2 *Component Analysis.* We conduct ablation experiments on the ECMWF dataset for LSDGNN to further study the influence of different model modules on performance. First, we investigate the effects of different loss functions on the model, where MSE-Weight is the weighted loss function defined in Eq.12. From the experimental results, we can see that our weighted loss function can significantly improve the prediction ability of the model for short-duration intense rainfall. Due to the problem of the unbalanced distribution of rainfall events, the model using MAE as the loss function is biased to treat short-term intense rainfall events as noise. MSE increases the model's attention to short-term intense rainfall events to a certain extent, but the effect on indicator 2 is still poor.

Table 5: Effect of different loss functions on performance

Loss Function	Predict 3h			Predict 6h			Predict 3h			Predict 6h		
	CSI_1	ETS_1	FAR_1	CSI_1	ETS_1	FAR_1	CSI_2	ETS_2	FAR_2	CSI_2	ETS_2	FAR_2
MAE	0.340	0.311	0.311	0.181	0.158	0.440	0.000	0.0000	0.0000	0.0000	0.0000	0.0000
MSE	0.362	0.312	0.575	0.265	0.209	0.682	0.006	0.0060	0.2000	0.0000	0.0000	0.0000
MSE-weight	0.252	0.190	0.718	0.185	0.116	0.798	0.051	0.0504	0.9262	0.0091	0.0087	0.9762

To explore the impact of the use of different dependency graphs on the model performance, we set up the following experiments:

- w/o DG: indicates that we remove the SDGC module of LSDGNN and only use the long-term dependency graph
- Distance: denotes that we use a distance-based dependency graph instead of the long-term dependency graph of LS-DGNN
- w/o Filter: indicates that we remove the filter on the long-term dependency graph

Table 6 shows the superiority of the graph constructor proposed in this paper compared to the distance-based generated graphs. The distance-based graphs in this paper are obtained by calculating the distance of adjacent lattice points in eight directions and normalizing them. Using the DTW to constitute the dependency graph, the model aggregates the information from all nodes based on semantic similarity, where the redundant information leads to the degradation of the model performance. Experimentally, this paper uses a filter to actively filter the redundant information, which effectively improves the performance of the model.

Table 6: Performance impact of different dependency graph usage methods

Adjacency Matrix	Predict 3h			Predict 6h			Predict 3h			Predict 6h		
	CSI_1	ETS_1	FAR_1	CSI_1	ETS_1	FAR_1	CSI_2	ETS_2	FAR_2	CSI_2	ETS_2	FAR_2
w/o DG	0.213	0.145	0.773	0.167	0.095	0.821	0.0425	0.0419	0.9327	0.0140	0.0135	0.9629
Distance	0.218	0.150	0.769	0.175	0.103	0.814	0.0431	0.0425	0.9269	0.0037	0.0034	0.9884
w/o Filter	0.214	0.145	0.774	0.163	0.089	0.827	0.0432	0.0427	0.9189	0.0054	0.0048	0.9885
LSDGNN	0.252	0.190	0.718	0.185	0.116	0.798	0.0510	0.0504	0.9262	0.0091	0.0087	0.9762

5 CONCLUSION

In this study, we present an LSDGNN model for short-term intense rainfall forecasting. Because rainfall forecasting is based on the aggregation of historical spatial-temporal data, two graph constructors with different semantics are presented to extract spatial-temporal connections. Furthermore, the LSGCM is used to combine the spatial dependencies of the two graphs. We use the IDW interpolation method to generate ground truth rainfall data for the ECMWF and WRF datasets and test our model on both. Ablation experiments are set up to demonstrate the effectiveness of our proposed module. Our model can aggregate multi-scale spatial and temporal information more comprehensively and significantly improve the forecasting performance of short-term intense rainfall. In addition, the model can be used to solve a variety of additional sub-problems.

REFERENCES

[1] Guoxing Chen and Wei Chyung Wang. 2022. Short-Term Precipitation Prediction for Contiguous United States Using Deep Learning. *Geophysical Research Letters* 49 (2022).

[2] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[3] Bing Fu, Melinda S. Peng, Tim Li, and Duane E. Stevens. 2012. Developing versus nondeveloping disturbances for tropical cyclone formation. Part II: Western North Pacific. *Monthly Weather Review* 140, 4 (2012), 1067–1080.

[4] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan. 2019. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* (2019).

[5] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[6] Garima Jain and Bhawna Mallick. 2017. A study of time series models ARIMA and ETS. *Available at SSRN 2898968* (2017).

[7] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. 2021. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data (TKDD)* (2021).

[8] C. Liu, J. Sun, X. Yang, S. Jin, and S. Fu. 2021. Evaluation of ECMWF Precipitation Predictions in China during 2015-2018. *Weather and Forecasting* (2021).

[9] A. Manzato, A. Pucillo, and A. Cicogna. 2018. Improving ECMWF-based 6-hours maximum rain using instability indices and neural networks. *Atmospheric Research* 217, MAR. (2018), 184–197.

[10] Meteorology. 2018. Glossary of Meteorology. *Available from http://glossary.ametsoc.org/* (2018).

[11] Seung-Hyun Moon, Yong-Hyuk Kim, Yong Hee Lee, and Byung-Ro Moon. 2019. Application of machine learning to an early warning system for very short-term heavy rainfall. *Journal of Hydrology* 568 (2019), 1042–1054.

[12] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat. 2019. Deep learning and process understanding for data-driven Earth system science. *Nature* 566, 7743 (2019), 195.

[13] Stan Salvador and Philip Chan. 2007. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis* 11, 5 (2007), 561–580.

[14] J. H. Seinfeld, C. Bretherton, K. S. Carslaw, H. Coe, P. J. Demott, E. J. Dunlea, G. Feingold, S. Ghan, A. B. Guenther, and R. Kahn. 2016. Improving our fundamental understanding of the role of aerosol-cloud interactions in the climate system. *Proceedings of the National Academy of Sciences of the United States of America* 113, 21 (2016), 5781.

[15] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28 (2015).

[16] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting *(Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34).* 914–921. 01.

[17] Casper Kaae Sønderby, Lasse Espeholt, Jonathan Heek, Mostafa Dehghani, Avital Oliver, Tim Salimans, Shreya Agrawal, Jason Hickey, and Nal Kalchbrenner. 2020. Metnet: A neural weather model for precipitation forecasting. *arXiv preprint arXiv:2003.12140* (2020).

[18] M. Tolstykh and A. Frolov. 2005. Some Current Problems in Numerical Weather Prediction. *Izvestiya Atmospheric and Oceanic Physics* 41, 3 (2005), 285–295.

[19] B. Wang, J. Lu, Z. Yan, H. Luo, and G. Zhang. 2019. Deep Uncertainty Quantification: A Machine Learning Approach for Weather Forecasting *(the 25th ACM SIGKDD International Conference).*

[20] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S. Yu. 2017. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems* 30 (2017).

[21] Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S. Yu. 2019. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics *(Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition).* 9154–9162.

[22] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S. Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[23] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks *(Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining).* 753–763.

[24] Huosheng Xie, Lidong Wu, Wei Xie, Qing Lin, Ming Liu, and Yongjing Lin. 2021. Improving ECMWF short-term intensive rainfall forecasts using generative adversarial nets and deep belief networks. *Atmospheric Research* 249 (2021), 105281.

[25] Huosheng Xie, Rongyao Zheng, and Qing Lin. 2022. Short-Term Intensive Rainfall Forecasting Model Based on a Hierarchical Dynamic Graph Network. *Atmosphere* 13, 5 (2022), 703.

[26] H. Y. Xie, S. Y. Zhang, S. C. Hou, and Xin Zheng. 2018. Comparison research on rainfall interpolation methods for small sample areas. *Research of Soil and Water Conservation* 25, 3 (2018), 117–121.

[27] Junchen Ye, Leilei Sun, Bowen Du, Yanjie Fu, and Hui Xiong. 2021. Coupled layer-wise graph convolution for transportation demand prediction *(Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35).* 4617–4625. 5.

[28] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting *(Proceedings of the 27th International Joint Conference on Artificial Intelligence).* 3634–3640.

[29] Wenying Zhang, Huaguang Zhang, Jinhai Liu, Kai Li, Dongsheng Yang, and Hui Tian. 2017. Weather prediction with multiclass support vector machines in the fault detection of photovoltaic system. *IEEE/CAA Journal of Automatica Sinica* 4, 3 (2017), 520–525.

[30] K. Zhou, Y. Zheng, B. Li, W. Dong, and X. Zhang. 2019. Forecasting Different Types of Convective Weather: A Deep Learning Approach. *Journal of Meteorological Research* 33, 5 (2019), 797–809.