*Journal on Big Data*
DOI:10.32604/jbd.2021.016993
***Article***

Tech Science Press

# Survey on Research of RNN-Based Spatio-Temporal Sequence Prediction Algorithms

**Wei Fang[1,2,*], Yupeng Chen[1] and Qiongying Xue[1]**

[1]School of Computer & Software, Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science & Technology, Nanjing, 210044, China

[2]Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, 215006, China

[*]Corresponding Author: Wei Fang. Email: fangwei@nuist.edu.cn

Received: 17 January 2021; Accepted: 08 April 2021

**Abstract:** In the past few years, deep learning has developed rapidly, and many researchers try to combine their subjects with deep learning. The algorithm based on Recurrent Neural Network (RNN) has been successfully applied in the fields of weather forecasting, stock forecasting, action recognition, etc. because of its excellent performance in processing Spatio-temporal sequence data. Among them, algorithms based on LSTM and GRU have developed most rapidly because of their good design. This paper reviews the RNN-based Spatio-temporal sequence prediction algorithm, introduces the development history of RNN and the common application directions of the Spatio-temporal sequence prediction, and includes precipitation nowcasting algorithms and traffic flow forecasting algorithms. At the same time, it also compares the advantages and disadvantages, and innovations of each algorithm. The purpose of this article is to give readers a clear understanding of solutions to such problems. Finally, it prospects the future development of RNN in the Spatio-temporal sequence prediction algorithm.

**Keywords:** RNN; LSTM; GRU; spatio-temporal sequence prediction

## 1 Introduction

In the past few years, Recurrent Neural Networks (RNN) have developed rapidly and have attracted widespread attention due to their powerful modeling capabilities. Compared with traditional methods, the introduction of RNN has brought great improvements in image processing and natural language processing, such as weather forecasting, stock forecasting, and speech recognition.

However, the fully connected neural network has the problem of gradient disappearance or gradient explosion when the network has many layers [1]. This is because the previous layer forward propagation will be used to the output during the process (as shown in Fig. 1) in the backpropagation through the derivation of the chain rule. And the current output depends on the output of the previous layer. This is a calculation process of continuous multiplication.

Each layer of RNN has a time-step loop calculation. The length of the current time step depends on the length of the input sequence. If the sequence is too long, then it is equivalent to a deep continuous multiplication of forwarding propagation. The backpropagation process of RNN will produce long-term dependence with the time sequence. This is because the hidden state $S_t$ of each step propagates forward with the time series, and $S_t$ is a function of $W_x$ and $W_s$, so there will be a continuous multiplication calculation of the indirect hidden state between each time step. As long as there is continuous multiplication, there is a danger of the gradient disappearing.

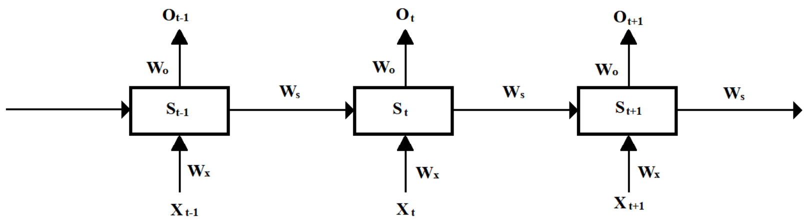


**Figure 1:** The structure of the RNN network

Therefore, in modern applications, almost no researchers use the original RNN model but use RNN-based LSTM and GRU model as the basic research objects to explore and study. According to different application scenarios, researchers will novel these two models or combine them with other methods to solve problems according to their specific needs.

Spatio-temporal sequence data are complex. When doing Spatio-temporal sequence forecasting, it is necessary to consider not only the continuity and periodicity in time, but also the spatial correlation between different regions, and these spatial correlations will change over time [2]. Therefore, it is very difficult to accurately dig out the required Spatio-temporal features from complex Spatio-temporal sequence data. The old methods are mainly based on statistical principles. These methods treat Spatio-temporal sequences as multiple time series. It is difficult to capture spatial correlations, and it is difficult to dig out nonlinear Spatio-temporal patterns.

Traditional machine learning methods can capture the nonlinear relationship in Spatio-temporal data to a limited extent, but they are not effective in long-term prediction and are difficult to adapt to large-scale Spatio-temporal sequence datasets. Whether it is in prediction accuracy or processing large-scale data sets, deep learning methods have advantages over traditional methods. In the Spatio-temporal sequence prediction using deep learning methods, the general processing method is to treat the Spatio-temporal sequence as a series of time-sequential pictures, use a convolutional network to extract the spatial features in the data, and then send it to the RNN-based model to capture time sequence characteristics, and finally get the prediction result [3]. At present, it has become a trend to use deep learning methods to analyze Spatio-temporal sequence data from many aspects [4], and then construct a Spatio-temporal sequence prediction mixed model.

Precipitation nowcasting and traffic flow forecasting have been widely concerned by researchers as typical Spatio-temporal sequence forecasting problems. In addition to research value, the practical application of these two directions is also very important. This paper will focus on the analysis of Spatio-temporal sequence prediction algorithms in these two directions, hoping to help researchers who are new to these two directions quickly establish their understanding and help experienced researchers generate new ideas.

## 2 Brief Overview of RNN

### *2.1 Developments History of RNN*

In 1986, Jordan [5] proposed the Jordan network under the theory of distributed parallel processing. Each hidden layer node of the Jordan network is connected to a state unit to realize the delay input, and the logistic function is used as the activation function. The Jordan network uses a backpropagation algorithm for learning and extracts the phonetic features of a given syllable in the test. Then in 1990, Elman [6] proposed the first fully connected RNN, the Elman network. Both Jordan network and Elman network build recursive connections from a single-layer feedforward neural network, so they are also called simple recurrent networks (SRN) [7].

In the same period when SRN appeared, the learning theory of RNN was also developed. After the backpropagation algorithm was proposed, the academic circle began to try to train the recurrent neural network under the BP framework. In 1989, Williams et al. proposed the real-time loop learning of RNN [8].

Then Werbos proposed a backpropagation algorithm over time in 1990 [9].

The idea behind RNN is to use sequential information. In traditional neural networks, we assume that all inputs (including outputs) are independent of each other. For many tasks, this is a very bad assumption. If you want to predict the next word in a sequence, you better know which word comes before it. RNN loops because it performs the same operation for each element in the series, and each operation depends on the previous calculation result. Thinking in another way, it can be considered that the RNN has memorized the information that has been calculated so far. In theory, RNN can use arbitrarily long sequence information, but due to the problem of gradient explosion and gradient disappearance, in practice, RNN can only review the previous steps.

### *2.2 Basic Concepts of LSTM*

LSTM is the earliest proposed RNN gating algorithm, and its corresponding loop unit, the LSTM unit contains 3 gates: input gate, forget gate, and output gate. Compared with the recursive calculation of the RNN's establishment of the system state, the three gates establish a self-loop on the internal state of the LSTM unit [10]. Specifically, the input gate determines the input of the current time step and the update of the internal state of the system state of the previous time step; the forgetting gate determines the update of the internal state of the previous time step to the internal state of the current time step; the output gate determines the effect of the internal state on the system. The status update is shown in Fig. 2.

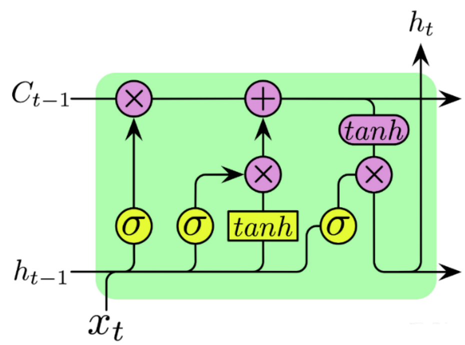


**Figure 2:** Inner structure of LSTM

The update method of the LSTM unit is shown in Eq. (1):

$$
\begin{aligned}
&h^{(t)} = g_o^{(t)} f_h(s^{(t)}) \\
&s^{(t-1)} = g_f^{(t)} s^{(t-1)} + g_i^{(t)} f_s(wh^{(t-1)} + uX^{(t)} + b) \\
&g_i^{(t)} = sigmoid(w_i h^{(t-1)} + u_i X^{(t)} + b_i) \\
&g_f^{(t)} = sigmoid(w_f h^{(t-1)} + u_f X^{(t)} + b_f) \\
&g_o^{(t)} = sigmoid(w_o h^{(t-1)} + u_o X^{(t)} + b_o)
\end{aligned}
\tag{1}
$$

where $f_h, f_s$ are the activation functions of the system state and internal state, usually hyperbolic tangent function, g is the gating updated with time step, essentially a feedforward neural network with the sigmoid function as the activation function. The reason for using the sigmoid function is that its output is in the interval [0,1], which is equivalent to a set of weights. In the formula, the subscripts $i$,$o$, $f$ represent input gate, forget gate, and output gate. In addition to the above update rules, LSTM can also further introduce internal state update gating. The algorithm using this strategy is called "peephole LSTM" [11].

The original version of LSTM did not have a forget gate and performed output in a fully connected manner. But essentially LSTM units can be introduced into other types of RNN architectures, such as LSTM autoencoders [12], stacked LSTMs [13], etc. When initializing the weights of LSTM, it is necessary to set a larger initial value for the forgetting gate. A small value will make the forgetting gate quickly forget the information of the previous time step during learning, which is not conducive to the long-distance dependence of neural network learning, and may lead to gradient disappears.

### *2.3 Basic Concepts of GRU*

Since the three gates in LSTM have different contributions to improving their learning ability, omitting the small contribution gate and its corresponding weight can simplify the neural network structure and improve its learning efficiency. GRU is an algorithm proposed based on the above concepts. Its corresponding loop unit contains only two gates: update gate and reset gate. The function of the reset gate is similar to the input gate of the LSTM unit. The update gate implements both the forget gate and the output gate.

Comparing the updated rules of LSTM and GRU, we can find that the total number of parameters of GRU is smaller, and the parameter update sequence is different from LSTM. GRU updates the state first and then updates the gating, so the state of the current time step uses the gating parameters of the previous time step, LSTM first updates the gating and uses the gating parameters of the current time step to update the state. The two gates of GRU do not form a self-loop, but directly recurse between system states,

There are many variants of LSTM and GRU, including the sharing of update gate and reset gate parameters between cyclic units, and the use of global gating for the entire chain connection. However, studies have shown that these improved versions do not show obvious advantages over standard algorithms. The possible reason is that the performance of the gating algorithm mainly depends on the forget gate, and the above variant and the standard algorithm use a similar forget gate mechanism [14].

## 3 Contrastive Research on Spatio-Temporal Sequence Prediction Algorithm

### *3.1 Algorithms Applied on Precipitation Nowcasting*

The purpose of precipitation nowcasting is to predict the future rainfall intensity in a certain area in a relatively short period. The mainstream method is to use the optical flow method [15] and other methods to extrapolate radar echo. However, because the flow estimation step size and the radar echo extrapolation step size are separated, and the determination of model parameters is challenging, the success of these optical flow-based methods is limited. Few previous studies have studied this crucial and challenging weather forecasting problem from the perspective of machine learning. However, with the development of deep learning, researchers have gradually studied this important weather forecast problem from the perspective of machine learning. Besides, the short-term precipitation forecast relies on Spatio-temporal sequence data, so more and more researches apply the convolutional neural network algorithm that best fits the Spatio-temporal sequence data to precipitation nowcasting.

#### *3.1.1 ConvLSTM*

Shi et al. [16] is the researcher who first thought of applying deep learning methods to short-term precipitation forecasting. Srivastava et al. [17] and others pointed out the importance of multi-step prediction in learning useful representations. They established an LSTM encoding-decoding predictor model, which can reconstruct the input sequence and predict the future sequence at the same time. Although Srivastava's method can also be used to solve the temporal and spatial sequence prediction problems such as short-term precipitation forecasting, the fully connected LSTM (FC-LSTM) layer used in their model does not consider spatial correlation.

Shi et al. [16] extended the idea of FC-LSTM to both the input to state and state to state transition have convLSTM with convolutional structure. By superimposing multiple convLSTM layers to form a coded forecast structure, an end-to-end trainable precipitation nowcast model can be established. For evaluation, they also created a new real-life radar echo data set that can facilitate further research, especially in the design of machine learning algorithms.

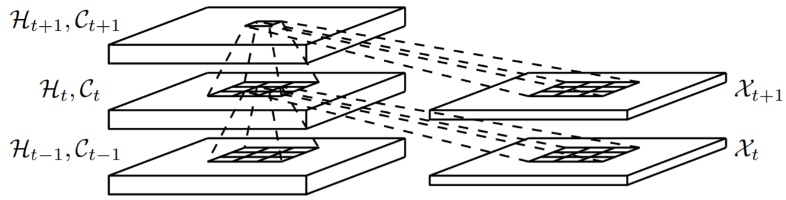


**Figure 3:** Inner structure of convLSTM

To overcome the problem that FC-LSTM uses complete connections in the input-to-state and state-to-state transitions that do not encode spatial information, a salient feature of the convLSTM design is that the last two dimensions (rows and columns) are all inputs of the spatial dimension $\mathcal{X}_1, \ldots, \mathcal{X}_t$, unit output $\mathcal{C}_1, \ldots, \mathcal{C}_t$, hidden state $\mathcal{H}_1, \ldots, \mathcal{H}_t$, and the three gates $i_t, f_t, o_t$ are all three-dimensional tensors. To better understand the inputs and states, you can think of them as vectors on a spatial grid. ConvLSTM determines the future state of a cell in the grid through its adjacent input and past state. This is easily achieved by using convolution operators in state-to-state and input-to-state transitions (as shown in Fig. 3). The key equation of convLSTM is shown in Eq. (2), where "*" represents the convolution operator and "∘" represents the Hadamard product.

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ tanh(W_{xc}\mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathscr{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o) \\
\mathcal{H}_t &= o_t \circ tanh(\mathcal{C}_t)
\end{aligned} \tag{2}
$$

In subsequent experiments, convLSTM was used to predict the same set of data together with the traditional extrapolation method based on optical flow and the method based on FC-LSTM. Experimental results show that convLSTM has achieved the best experimental results in terms of mean squared error (MAE), critical success index (CSI), correlation, etc. Shi et al. [16] claim that by superimposing multiple convLSTM layers and forming a coded forecast structure, not only a network model for precipitation nowcasting problems can be established, but also a network model for more general Spatio-temporal sequence forecasting problems can be established.

*3.1.2 TrajGRU*

Shi et al. [18] summarized some experiences in their previous research on precipitation nowcasting, found the shortcomings of the previous work, and proposed a GRU-based short-term precipitation forecast method, TrajGRU. Although the convolutional recursive structure used in ConvLSTM is superior to the fully connected recursive structure in capturing Spatio-temporal correlation, it is not optimal and there is room for improvement. For motion modes such as rotation and zooming, the local correlation structure of consecutive frames will vary depending on the spatial position and time stamp. Therefore, it is inefficient to use a position invariant filter to express the convolution of this position change relationship.

The TrajGRU model uses a subnet to output a state-to-state connection structure before state transition. TrajGRU allows states to be aggregated along some learning trajectories, so it is more flexible than convolutional GRU (ConvGRU) [19] with a fixed connection structure. TrajGRU uses the current input and previous state to generate a set of local neighborhoods for each location in each timestamp. Since the position index is discrete and non-differentiable, Shi uses a set of continuous optical flows to represent these "indexes". The advantage of this structure is that it can learn connection expansion by learning the parameters of the subnet, and the TrajGRU model can more effectively use these parameters.

The final experimental result shows that all deep learning models trained with balanced loss are better than models based on optical flow. In the deep learning model, TrajGRU has the best performance, better than traditional LSTM and ConvLSTM, which shows that a suitable network structure is a key to good performance.

### *3.1.3 GA-ConvGRU*

Based on ConvGRU, Tian et al. proposed a generated adversarial ConvGRU (GA-ConvGRU) model [20]. GA-ConvGRU consists of two opposite learning systems, namely a generator and a discriminator. Lin uses the ConvGRU model as a generator and attaches a five-layer convolutional neural network as a discriminator [21]. These two systems learn by playing the minimax game. The purpose of the generator is to fool the discriminator by generating extrapolation of the radar echo map, and the discriminator is responsible for identifying the extrapolation result from actual observations. Adding a discriminator can enhance the multi-modal distribution of details and models. Therefore, the extrapolation method becomes more realistic and accurate.

### *3.1.4 PredRNN*

Wang et al. [22] proposed a predictive recurrent neural network (PredRNN) and first applied it to precipitation nowcasting. Wang believes that Spatio-temporal prediction learning should remember the appearance of space and changes in time in a unified memory pool. Specifically, the memory state is no longer restricted to each LSTM unit. Instead, they are allowed to zigzag in two directions: vertically through the stacked RNN layers, and horizontally through all RNN states. The core of this network is a new Spatio-temporal LSTM (ST-LSTM) unit (as shown in Fig. 4), which can simultaneously extract and memorize the representation of space and time.

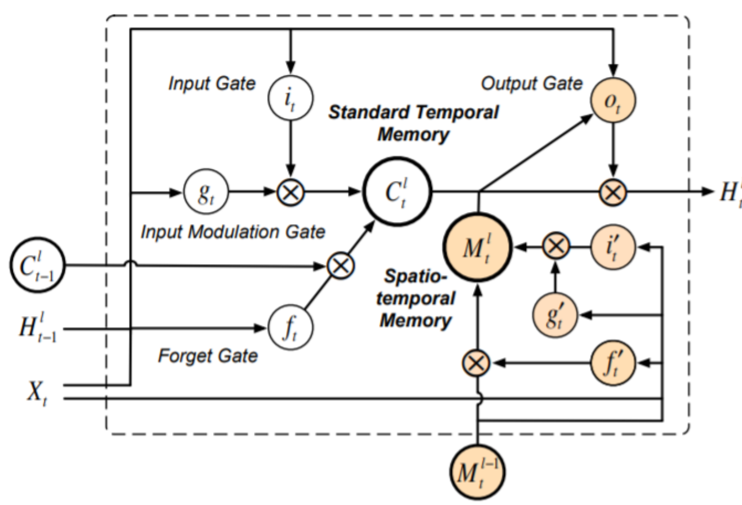


**Figure 4:** Inner structure of ST-LSTM

Wang et al. [22] found that although the stacked LSTM architecture proved to be powerful for supervised Spatio-temporal learning, this type of data needs to meet two conditions, obvious temporal features and a sufficiently simple visual structure. However, Spatio-temporal prediction learning does not satisfy these conditions. In Spatio-temporal prediction data, spatial deformation and temporal dynamics are equally important for generating future frames. Based on this, Wang proposed a new recurrent architecture called Predictive RNN (PredRNN), which allows cross-layer interaction of memory states belonging to different LSTMs (in traditional RNNs, they are independent of each other). The key component of PredRNN is the Spatio-temporal LSTM (ST-LSTM) unit, which models space and time representation in a unified storage unit, and transfers the memory vertically and horizontally in the hierarchy. It means that for each input, there is an information extraction after each layer of the network structure. The final abstract information extracted should be reserved for the next input of the first layer. The original idea was that a predictive learning system should remember the appearance of space and changes in time in a unified memory pool. By doing so, the memory state flows in the entire network in a tortuous direction, and then further studies on how to make Spatio-temporal memory interact with the original long and short-term memory.

Finally, when the PredRNN model is applied to the actual precipitation nowcasting, compared with convLSTM, although PredRNN is slightly larger than convLSTM in training time and memory usage, its accuracy is about 50% higher than convLSTM.

### *3.1.5 DeepRain*

Kim et al. [23] proposed a data-driven precipitation prediction model called DeepRain. The structure is shown in Fig. 5. This model uses convolutional LSTM (ConvLSTM) to predict rainfall using weather radar data. Shi et al. [16] uses three-dimensional single-channel data, while Kim uses ConvLSTM to process three-dimensional (width, height, and depth) and four-channel (4 different altitudes) data. Another difference is that Shi uses a many-to-many output, while Kim uses a many-to-one method. Each row of data in the data set consists of radar reflectivity and ground truth value. The radar reflectivity is 15-time intervals (interval 6 min) at 4 heights from 101 km × 101 km to the ground, a total of 101 × 101 × 4 × 15 numbers value; the ground-truth value is the rainfall in the range of 50 × 50 from the grid center for 1 to 2 h. The complete data set includes 10000 rows of data randomly selected during 2 years. The model sequentially inputs 15 three-dimensional four-channel data in time series, and the output is the predicted rainfall information. Experimental result shows that the convolution operation can effectively extract potential features from the data, and can be quickly trained. Compared with linear regression, 2-layer ConvLSTM reduces RMSE by 23.0%.

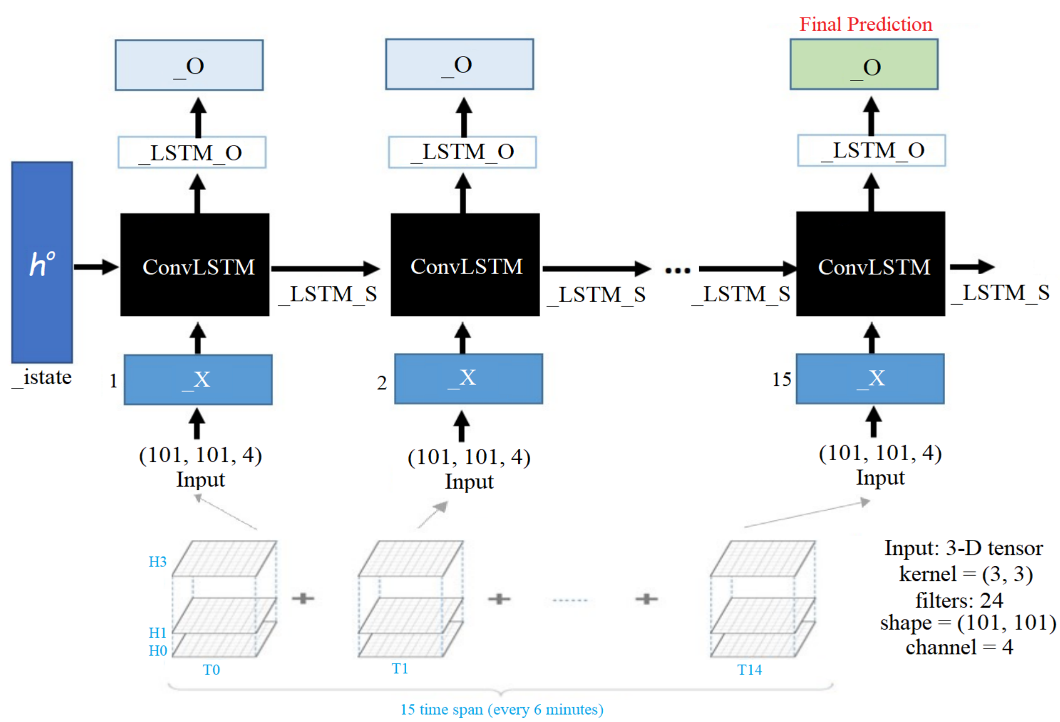


**Figure 5:** Inner structure of DeepRain

*3.1.6 Summary of Algorithms Applied in Precipitation Nowcasting*

This section makes a comprehensive comparison of the algorithms for precipitation nowcasting introduced earlier. As shown in Table 1.

**Table 1:** Innovative points, advantages, and disadvantages of algorithms applied in precipitation nowcasting

| Model | Innovative points | Advantages | Disadvantages |
|---|---|---|---|
| ConvLSTM | Combine LSTM and CNN | Combining CNN and LSTM for the first time, which is a significant improvement over traditional methods | There are deficiencies in capturing Spatio-temporal features and there is still room for improvement |
| Traj-GRU | Allow states to aggregate along some learning trajectories | The structure of changes between radar sequences can be adjusted by learning the parameters of the subnet | The model is complicated |
| GA-ConvGRU | Introduce GAN model, introduce the idea of generator and discriminator | Enhanced the multi-modal distribution of details and models to make the extrapolation results more accurate | The training of the GAN model is unstable |
| PredRNN | Proposed PredRNN and ST-LSTM | Compared with the previous model, the spatial correlation and short-term dynamic modeling capabilities are improved | The training time of the model is long, and there is a problem of disappearing gradient |
| DeepRain | Use ConvLSTM to sequentially input radar image sequences and use rainfall directly as data labels | Compared with linear regression, two-layer ConvLSTM reduces RMSE by 23.0% | The rainfall is directly used as the label training model, and the result is strongly unexplainable |

***3.2 Algorithms Applied to Traffic Flow Prediction***

Traffic flow prediction is one of the core issues of intelligent transportation systems. Accurate prediction results enable commuters to choose appropriate travel modes, travel routes, and departure times, which is of great significance to traffic management. To improve the prediction accuracy of traffic data, it is a feasible way to develop a more effective traffic data analysis method. In recent years, the emergence of a large amount of traffic data and computing power has prompted us to improve the accuracy of short-term traffic predictions through deep learning methods.

*3.2.1 Bi-ConvLSTM*

Liu et al. [24] proposed a new end-to-end deep learning architecture, which consists of two modules. According to the traffic flow data contains three main features: time feature, space feature, and periodic feature, firstly, convolution and LSTM are combined to form a ConvLSTM module that can extract the temporal and spatial information of traffic flow information. Besides, the bidirectional LSTM (Bi-LSTM) module is used to analyze the historical traffic flow data at the predicted point to obtain the periodic characteristics of the traffic flow.

This new deep learning architecture for short-term traffic flow prediction first combines the ideas of convolution and LSTM to generate a ConvLSTM module to extract the temporal and spatial characteristics of traffic data, which can fully integrate temporal and spatial features to predict traffic flow in neighboring areas point [25]. Then add the Bi-LSTM module to the deep learning model as an auxiliary module to extract the periodic characteristics of the traffic data. Using the temporal and spatial characteristics and periodic characteristics of traffic flow extracted by ConvLSTM and Bi-LSTM modules, the short-term traffic flow is predicted respectively. The ConvLSTM module is used to process the short-term traffic flow data in the neighboring area, and the temporal and spatial characteristics are extracted; the two-way LSTM is used to process the on-site historical traffic data to extract the periodic characteristics of the traffic flow data, and its performance is better than the algorithm based on LSTM.

### *3.2.2 AT-ConvLSTM*

Zheng et al. [26] proposed a deep learning model based on ConvLSTM with an attention mechanism, which uses a hybrid and multi-layer structure to automatically extract the inherent characteristics of traffic flow data. First, based on CNN and LSTM networks, an attention-based ConvLSTM module is developed to extract spatial and short-term temporal features. By automatically assigning different weights, the attention mechanism is appropriately designed to distinguish the importance of the flow sequence at different times. Secondly, to further explore the long-term temporal characteristics of traffic flow, Zheng introduced a Bi-LSTM module to extract daily and weekly periodic characteristics to obtain the variance trend of forwarding and backward traffic flows.

Finally, the experimental result shows that, compared with the existing methods, this new model combining the attention-based ConvLSTM and Bi-LSTM has better results than LSTM and DCRNN on the three evaluation criteria of MAE, RMSE, and MAPE.

### *3.2.3 CNN-rGRU*

Saravanan et al. [27] proposed an improved deep hybrid model for short-term traffic congestion prediction. The model applies the convolutional layer and GRU to residual learning, which can adaptively learn the temporal and spatial characteristics of traffic congestion. Both of these methods can effectively capture the temporal and spatial correlation of information.

The traditional RNN uses time backpropagation to reduce node weights by increasing the weights of the previous layer, resulting in the problem of gradient disappearance. Therefore, a special neural network is used to store long-term information, and LSTM's advanced model GRU is used to read arbitrarily long information sequences. However, the GRU model can make short-term predictions of traffic flow. In this method, a deep hybrid neural network based on CNN-rGRU, as shown in Fig. 6, can predict short-term traffic flow under different conditions. The deep hybrid model is composed of a one-dimensional CNN that captures spatial information and a GRU model that relies on temporal information for long-term representation learning. Finally, the hybrid model makes short-term predictions of traffic flow. CNN is used to train the traffic flow data set, the GRU model extracts deep features, and the residual learning improves the accuracy by using GRU to summarize the density of CNN.

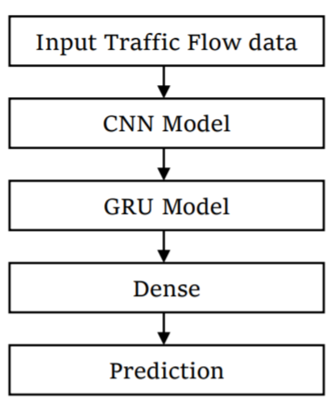


**Figure 6:** Structure of CNN-rGRU

The deep hybrid model is composed of two parts. One-dimensional CNN captures spatial information with 10 layers, and the GRU model captures long-term dependency time information to represent learning. Finally, the hybrid model makes short-term predictions of traffic flow. CNN is used to train the traffic flow data set, the GRU model extracts deep features, and the residual learning improves the accuracy by using GRU to summarize the density of CNN.

### *3.2.4 CLTFP*

Wu et al. [28] proposed a new deep architecture combining CNN and LSTM to predict future traffic flow (CLTFP). Different from the convLSTM proposed by Shi et al. [16], CLTFP consists of one CNN layer and two LSTM layers. One-dimensional CNN is used to obtain the spatial characteristics of traffic flow, and two LSTMs are used to mine the short-term variability and periodic characteristics of traffic flow. For these meaningful features, feature-level fusion is performed to realize short-term traffic flow prediction. After feature extraction, there is a full-connected layer used to output the prediction. Because CLTFP makes full use of the spatial distribution, short-term temporal variability, and long-term periodicities, experimental results show that the algorithm has obvious advantages in traffic flow prediction compared with other popular forecasting methods on the open data set.

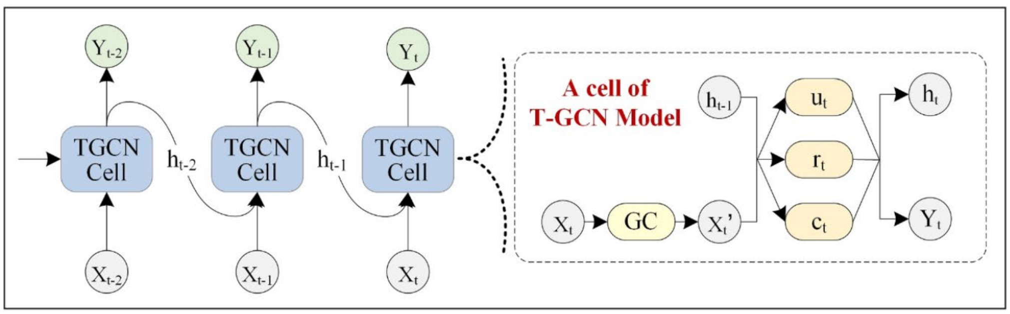


**Figure 7:** Inner structure of T-GCN

### *3.2.5 T-GCN*

Zhao et al. [29] proposed a neural network called Time Graph Convolutional Network (T-GCN) model to predict the traffic flow. To capture the temporal and spatial dependencies of traffic, T-GCN combined with the image volume proposed by Bruna et al. [30] and GRU. Among them, GCN is used to learn complex topological structures to obtain spatial dependence, and GRU is used to learn dynamic changes of traffic data to obtain temporal dependence. Then, the T-GCN model is used to predict the traffic based on the urban road network. Experiments show that the T-GCN model can obtain temporal and spatial correlations from traffic data, and its prediction effect is better than the latest baseline of real traffic data sets.

As shown in Fig. 7, the left side is the process of Spatio-temporal traffic prediction, and the right side shows the specific structure of T-GCN cells. $h_{t-1}$ represents the output at time $t-1$, GC is the process of graph convolution, $u_t, r_t$ are the update gate and reset gate at time t, and $h_t$ represents the output at time $t$.

In summary, the T-GCN model can handle complex spatial dependence and time dynamics. On the one hand, the GCN is used to obtain the topological structure of the urban road network and obtain its spatial dependence. On the other hand, the GRU is used to obtain the dynamic changes of traffic information on the road, obtain the time dependence, and finally realize the task of traffic prediction. In specific experiments, T-GCN is compared with GCN, GRU, and traditional methods. By comparing experimental results, T-GCN performs far better than traditional methods, and slightly ahead of other deep learning methods.

*3.2.6 Summary of Algorithms Applied in Traffic Flow Prediction*

This section makes a comprehensive comparison of the algorithms for traffic flow prediction introduced earlier. As shown in Table 2.

**Table 2:** Innovative points, advantages, and disadvantages of algorithms applied in traffic flow prediction

| Model | Innovative points | Advantages | Disadvantages |
|---|---|---|---|
| Bi-ConvLSTM | Use different algorithms to extract Spatio-temporal features and periodic features | Simple and easy to implement | The performance is lagging behind the new method |
| AT-ConvLSTM | Added attention mechanism to ConvLSTM | The attention mechanism can distinguish the importance of different time sequences to make training results more accurate | Due to the defects of ConvLSTM itself, it is easy to fall into gradient disappearance or gradient explosion |
| CNN-rGRU | Combine CNN and GRU | Solve the problem of gradient disappearance or gradient explosion when CNN and LSTM are combined | There are deficiencies in capturing Spatio-temporal features and there is still room for improvement |
| CLTFP | Use two LSTM layers to mine short-term variability and periodic characteristics of traffic flow | Enhance the sparsity of features, limit the weight of the regression layer, and greatly improve performance | Over-focusing on a small number of influential features can easily lead to overfitting |
| T-GCN | Combine graph convolutional neural network and GRU | Able to deal with complex spatial dependence and time dynamics | Limited performance improvement |

## 4 Future Development Outlook

Although the RNN-based Spatio-temporal sequence prediction method has made great progress, there are still many problems. In traditional Spatio-temporal sequence prediction methods, the selection of models still relies on specific data sets and personal experience, there is no guiding idea, and these traditional methods are difficult to capture the dynamic Spatio-temporal relationships in the data. In data-driven models such as Spatio-temporal support vector regression and deep learning, the generalization ability of the model is learned from historical samples, so it relies heavily on training data. The Spatio-temporal series data are often complex and are often affected by various factors. There are many missing and errors in the original data, which also affects the accuracy of Spatio-temporal series prediction [31]. Data processing and model design and selection are the keys to improve the prediction accuracy of the Spatio-temporal series [32]. Therefore, Spatio-temporal data preprocessing, especially data denoising, is particularly important; for some areas with less data, the prediction results obtained by direct Spatio-temporal series modeling are not accurate, and the knowledge learned in other areas with sufficient data needs to be transferred come. Finally, the further research directions of this article can be summarized as follows:

(1) *Spatio-temporal data denoising and transfer learning.* Spatio-temporal series data is affected by various visible and invisible factors and has the properties of high-dimensional, dynamic, multi-scale, and fuzzy. Therefore, there are usually more error samples in the data, which greatly affects the prediction accuracy of various models. At present, there is relatively little research work on the removal of noise from Spatio-temporal sequence data. You can consider designing a certain algorithm to screen and modify the training samples of the model to make the trained model more accurate. For example, the method of sliding time window is used to extract data from the original data at intermediate intervals (such as one day, one week, etc.) to form a new Spatio-temporal sequence [33], which makes the periodicity of Spatio-temporal data easier to extract. For cities lacking data, the use of migration learning

can transfer the Spatio-temporal sequence prediction models learned from one or more cities with sufficient data to solve the problem of data lack.

(2) *Time-space series online prediction and model simplification.* This article mainly discusses the Spatio-temporal sequence prediction method based on RNN, but the real-time performance of the Spatio-temporal sequence prediction is generally high in practical applications. In the follow-up work, we can consider combining online learning and Spatio-temporal sequence prediction methods, so that the model can not only update the model quickly in real-time but also quickly feedback the prediction results online. Also, Spatio-temporal sequence prediction models are often more complex, with many model parameters [34]. Compressing and simplifying the model while maintaining the prediction accuracy to meet the requirements is also a direction worth studying.

(3) *Enhance the interpretability of the Spatio-temporal sequence model.* In the Spatio-temporal sequence prediction method based on deep learning, when extracting features of other influencing factors, besides directly using CNN, RNN, fully connected feature extraction, and semantic network analysis, are there other extraction methods? Whether there is a corresponding logical explanation behind the method is a question worth considering [35]. By visualizing the model, we can deeply understand the process of extracting the characteristics of Spatio-temporal sequence data by the model, and then we can better explain the model.

## 5 Conclusion

This article starts from the development of RNN in the field of deep learning and discusses the current popular Spatio-temporal sequence prediction problems. After briefly introducing the relevant knowledge of RNN, the algorithm based on LSTM or GRU in short-term precipitation forecast and traffic flow forecast is introduced in detail. The realization principle of each algorithm is briefly described, and the advantages, disadvantages, and innovations of each algorithm are summarized and contrasted. In the process of research, it was discovered that although algorithms based on LSTM or GRU are simple and easy to use in Spatio-temporal sequence prediction, most algorithms are combined with other deep learning methods to solve problems, which to a certain extent also exposes the shortcomings of RNN-based LSTM and GRU model. In most cases, RNN-based methods must be combined with the CNN-based method. The CNN-based method is used to extract graph features and the RNN-based method is used to extract temporal features. This is also an important future research direction.

**Funding Statement:** This work was supported by the National Natural Science Foundation of China (Grant No. 42075007), the Open Project of Provincial Key Laboratory for Computer Information Processing Technology under Grant KJS1935, Soochow University, and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] J. F. Kolen and S. C. Kremer, "Gradient flow in recurrent nets: The difficulty of learning long term dependencies," In: *A Field Guide to Dynamical Recurrent Networks*, pp. 237–243, 2001.

[2] Q. Zhu, J. Chen and L. Zhu, "Wind speed prediction with Spatio–temporal correlation: A deep learning approach," *Energies*, vol. 11, no. 4, pp. 1–18, 2018.

[3] Q. Zhu, J. Chen and D. Shi, "Learning temporal and spatial correlations jointly: A unified framework for wind speed prediction," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 1, pp. 509–523, 2019.

[4] B. Yan, X. Tang, J. Wang, Y. Zhou and G. Zheng, "An improved method for the fitting and prediction of the number of COVID-19 confirmed cases based on LSTM," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1473–1490, 2020.

[5] M. I. Jordan, "Serial order: a parallel distributed processing approach," *Advances in Psychology*, vol. 121, pp. 471–495, 1997.

[6] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[7] J. L. Elman, "Distributed representations, simple recurrent networks, and grammatical structure," *Machine Learning*, vol. 7, no. 2, pp. 195–225, 1991.

[8] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.

[9] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, vol. 1, no. 2, pp. 1550–1560, 1990.

[10] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. Cambridge: MIT Press, vol. 1, no. 2, 2016.

[11] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proc. of the IEEE-INNS-ENNS Int. Joint Conf. on Neural Networks*, vol. 3, pp. 189–194, 2000.

[12] F. Zhao, J. Feng and J. Zhao, "Robust LSTM-autoencoders for face de-occlusion in the wild," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 778–790, 2017.

[13] P. Malhotra, L. Vig and G. Shroff, *Long Short-Term Memory Networks for Anomaly Detection in Time Series*. Louvain: Presses Universitaires De Louvain, vol. 89, pp. 89–94, 2015.

[14] K. Greff, R. K. Srivastava and J. Koutník, "LSTM: A search space odyssey," *IEEE Transactions on Nneural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[15] P. Cheung and H. Y. Yeung, "Application of optical-flow technique to significant convection nowcast for terminal areas in Hong Kong," in *3rd WMO Int. Sym. on Nowcasting and Very Short-Range Forecasting*, pp. 6–10, 2012.

[16] X. Shi, Z. Chen and H Wang, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Advances in Neural Information Processing Systems*, pp. 802–810, 2015.

[17] N. Srivastava, E. Mansimov and R Salakhudinov, "Unsupervised learning of video representations using LSTMs," in *Int. Conf. on Machine Learning*, pp. 843–852, 2015.

[18] X. Shi, Z. Gao and L Lausen, "Deep learning for precipitation nowcasting: A benchmark and a new model," *Advances in Neural Information Processing Systems*, pp. 5617–5627, 2017.

[19] B. Nicolas, Y. Li and P Chris, "Delving deeper into convolutional networks for learning video representations," in *The Int. Conf. on Learning Representations*, 2016.

[20] L. Tian, X. Li and Y. Ye, "A generative adversarial gated recurrent unit model for precipitation nowcasting," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 4, pp. 601–605, 2019.

[21] Y. Shen, Y. Li, J. Sun, W. Ding, X. Shi *et al.,* "Hashtag recommendation using LSTM networks with self-attention," *Computers, Materials & Continua*, vol. 61, no. 3, pp. 1261–1269, 2019.

[22] Y. Wang, M. Long and J Wang, "Predrnn: recurrent neural networks for predictive learning using Spatio-temporal LSTMs," *Advances in Neural Information Processing Systems*, pp. 879–888, 2017.

[23] S. Kim, S. Hong and M. Joh, "Deeprain: Convlstm network for precipitation prediction using multichannel radar data," arXiv:1711.02316, 2017.

[24] Y. Liu, H. Zheng and X. Feng, "Short-term traffic flow prediction with Conv-LSTM," in *9th Int. Conf. on Wireless Communications and Signal Processing*, pp. 1–6, 2017.

[25] A. H. Vo, T. Nguyen and T. Le, "Brent oil price prediction using Bi-LSTM network," *Intelligent Automation & Soft Computing*, vol. 26, no. 6, pp. 1307–1317, 2020.

[26] H. Zheng, F. Lin and X. Feng, "A hybrid deep learning model with attention-based Conv-LSTM networks for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.

[27] S. Saravanan and K. Venkatachalapathy, "A deep hybrid model for traffic flow prediction using CNN-rGRU" *The International Journal of Analytical and Experimental Modal Analysis*, pp. 1554–1560, 2019.

[28] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," arXiv:1612.01022, 2016.

[29] L. Zhao, Y. Song and C. Zhang, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[30] J. Bruna, W. Zaremba and A. Szlam, “Spectral networks and locally connected networks on graphs,” arXiv:1312.6203, 2013.

[31] W. Fang, F. Zhang, Y. Ding and J. Sheng, “A new sequential image prediction method based on LSTM and DCGAN,” *Computers, Materials & Continua*, vol. 64, no. 1, pp. 217–231, 2020.

[32] S. Deng, S. Jia and J. Chen, “Exploring spatial-temporal relations via deep convolutional neural networks for traffic flow prediction with incomplete data,” *Applied Soft Computing*, vol. 78, pp. 712–721, 2019.

[33] J. Zhang, Y. Zheng and D. Qi, “DNN-based prediction model for Spatio-temporal data,” in *Proc. of the 24th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, pp. 1–4, 2016.

[34] G. Atluri, A. Karpatne and V. Kumar, “Spatio-temporal data mining: A survey of problems and methods,” *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–41, 2018.

[35] W. Fang, L. Pang and W. N. Yi, “Survey on the application of deep reinforcement learning in image processing,” *Journal on Artificial Intelligence*, vol. 2, no. 1, pp. 39–58, 2020.