Spatiotemporal Data Prediction Model Based on a Multi-Layer Attention Mechanism

Man Jiang, Harbin Engineering University, China
Qilong Han, Harbin Engineering University, China
https://orcid.org/0000-0002-5185-8387
Haitao Zhang, Harbin Engineering University*
Hexiang Liu, Harbin Engineering University, China

ABSTRACT

Spatiotemporal data prediction is of great significance in the fields of smart cities and smart manufacturing. Current spatiotemporal data prediction models heavily rely on traditional spatial views or single temporal granularity, which suffer from missing knowledge, including dynamic spatial correlations, periodicity, and mutability. This paper addresses these challenges by proposing a multi-layer attention-based predictive model. The key idea of this paper is to use a multi-layer attention mechanism to model the dynamic spatial correlation of different features. Then, multi-granularity historical features are fused to predict future spatiotemporal data. Experiments on real-world data show that the proposed model outperforms six state-of-the-art benchmark methods.

KEYWORDS

Attention Mechanism, Data Mining, Dynamic Spatial Relationships, Encoder-Decoder, LSTM, Multiple Temporal Relationships, Smart Cities, Spatiotemporal Data Prediction

INTRODUCTION

With the development of the mobile Internet, the requirements for data processing efficiency and mining depth are increasing rapidly (Zheng et al., 2014). As an important research field of spatiotemporal data processing, it is of great significance in accelerating the process of smart city construction (Bai et al., 2019) and manufacturing (Ge et al., 2011) in China and has been widely applied in a number of scenarios, such as air quality prediction (Wang et al., 2021; Pan et al., 2019), traffic flow prediction (Pan et al., 2019; Gong et al., 2019; Pan et al., 2022), medical risk prediction (Ye et al., 2020) and industrial production prediction (Cho, S., et al., 1997).

Current spatiotemporal data prediction models make some achievements. ARIMA (George & Gwilym, 1976) enables the extraction of the linear relationships between data while ignoring the complex nonlinear relationships, leading to low accuracy. ANN (Hopfield, 1982) represents complex nonlinear functions with an integrated structure of linear threshold units and partly discovers medium- and long-term patterns of spatiotemporal data (Martin et al., 2017). However, the results
are sensitive to the initial random weights and thresholds, which would be quite challenging in the case of real industrial production owing to the high demand for reliability. Support vector machines (SVMs) (Bernhard et al., 1992) are designed to map the input vector to a high-dimensional space and analyze the nonlinear characteristics of the sample using a linear algorithm to improve the accuracy (Sapankevych, N., & Sankar, R., 2009). Unfortunately, it is difficult to build a common prediction model because of the prior domain knowledge and the sensitivity of parameters and kernel functions. Random forests (Emmanouil, A., et al., 2019) train multiple decision trees for joint prediction. These methods are effective in extracting complex nonlinear relationships between high-dimensional data, and they are prone to overfitting when data noise exists. Convolutional neural networks (Yann, L., et al., 1998) model the spatial correlations of data through operations such as convolution and pooling, but they are highly dynamic and easily affected by multiple features (Liu et al., 2016; Shao et al., 2022; Zhong et al., 2022). Recurrent neural networks (Tomáš et al., 2011; Yang et al., 2022) can be used to extract historical patterns in spatiotemporal data by updating the model parameters by a backpropagation algorithm. Most existing studies use recurrent neural networks to capture a single temporal correlation (Yao et al., 2019; Wang, Y., & Liu, 2022; Hou et al., 2021) without considering the compounding effects of periodicity and abrupt variability on the data, making insufficient use of historical data.

Data in real scenarios such as smart cities and smart manufacturing have the following characteristics. In terms of sampling, the data quality is poor and locally sparse owing to the real-world environment and limited cost. For the spatial dimension, there exist correlations between both features and spatiotemporal objects. For the temporal dimension, the spatiotemporal data have both periodicity and mutability, while historical data of different granularities have different effects on the prediction results. These characteristics make it difficult for existing models to accurately model the evolution process of complex spatiotemporal data.

To solve the these problems, the authors propose a multilayer attention prediction model (spatiotemporal data prediction based on multilayer attention, STDPMA), which analyzes the interaction between different features and captures the dynamic spatial correlations of data. They then incorporate the cyclical and sudden change characteristics to fuse the historical features of multiple granularities.

This paper’s contributions are summarized as follows:

• To solve the problem of local sparsity of data, the authors leveraged spatial interpolation based on the spatial correlations between regions. The experimental results show that this spatial interpolation method will not destroy the temporal relationship of the original data. It can be applied to similar applications.
• To model the dynamic spatial correlations, features were divided according to their nature and extract dynamic spatial correlations between spatiotemporal objects. Historical features were then extracted by hour without introducing noise.
• A multigranularity encoder-decoder-based fusion network was designed to effectively fuse the patterns of the three granularities according to the periodicity and abruptness. Specifically, a multilayer attention mechanism was introduced to measure the influence of three granularity embeddings from the encoder module and leverage the decoder module to predict future spatiotemporal data.

**RELATED WORK**

Spatiotemporal data are used in the analysis when data are collected across both space and time. Prediction models must capture both spatial and temporal dependencies.

To model spatial correlations, more recent studies propose to capture high-order spatial correlations (from both static and dynamic views) using multiple convolutional neural networks.
and then introduce an aggregation module to fuse them together. For example, Zhang et al. (2016) processed spatiotemporal data into multichannel images and introduced a superimposed convolutional module to learn the spatial correlations from different sets of features. Zheng et al. (2019) proposed capturing the complex spatiotemporal correlations by different area functions. A local convolution model was also introduced to model the local spatial correlations (Yao, H., et al., 2018). Abdullaiah et al. (2020) used a weighted adjacency matrix to represent the spatiotemporal data and capture the spatiotemporal correlations with spatiotemporal convolution operations. Amir et al. (2019) used different neural networks to learn the static and dynamic spatial correlations separately and then fused them for prediction. Zheng et al. (2013) used artificial neural networks and linear chain conditional random fields to model static and dynamic spatial correlations of spatiotemporal data, respectively. Zhang et al. (2019) grouped multisource features based on linear correlations between pairs of features and used convolutional neural networks to capture the spatial correlations of different feature groups without considering the complex interactions between multiple features. Clearly, in a real-world scenario, spatiotemporal data will be influenced by multiple features from different domains (Lee et al., 2022). Simply integrating the correlations learned from static and dynamic features fails to accurately describe the dynamic spatial correlations of spatiotemporal data.

Regarding temporal correlation, recurrent neural networks have been a widely used choice, where the inputs are usually at a fixed granularity. For example, Feng et al. (2020) sliced the spatiotemporal data by hour and used neural networks to capture the temporal correlations. Lin et al. (2020) extracted features by day and proposed an attention mechanism to learn the contributions of daily data features to future data. Limited studies also consider multiple temporal granularities; for example, Chen et al. (2019) used RNN and LSTM to extract short- and long-term features of data, respectively, and then performed a simple aggregation operation. Ye et al. (2020) modeled short- and long-term temporal correlations using transformer and convolutional layers, respectively. Zhang et al. (2017) assumed that the spatiotemporal data are periodic and considered the three different temporal correlations from one week, one day, and three hours before the target time to complete the prediction task. Yao et al. (2019) showed that the periodicity of spatiotemporal data varies slightly with the actual conditions of various factors, but they only extend fixed historical time into very short time periods. In general, the continuous spatiotemporal data are in stable condition, but there are a few abrupt changes in rare cases. As periodicity and abruptness have a dynamic influence on future data, fusing fixed granularity data for multiple temporal correlation modeling would be unseemly. Some studies address external environmental factors that can also have an impact on spatiotemporal data prediction. For example, Zhang et al. (2017) used meteorological conditions and unexpected events to predict crowd flow. Liang et al. (2018) addressed the prediction by considering meteorological conditions and land use.

In this paper, a comprehensive experimental study was conducted to demonstrate that the authors’ model outperforms classical spatiotemporal data prediction models based on real datasets.

**SPATIOTEMPORAL DATA PREDICTION MODEL BASED ON A MULTILAYER ATTENTION MECHANISM**

**Analysis of Dynamic Spatial Relationships**

Air quality is influenced by meteorological conditions, local pollutants, adjacent pollutants, and multiple factors. As factors change dynamically with time, the spatial relationship is dynamic. A multifeature extraction module is designed to obtain the historical feature data of each node. It effectively extracts the dynamic spatial relationships between spatiotemporal data and establishes multifeature relationships.

The set of all nodes in the spatial region is \( S = \{s_1, s_2, \ldots, s_N\} \), where \( N \) is the number of nodes. \( La = (la_1, la_2, \ldots, la_N) \in \mathbb{R}^N \), which represents the latitude of the node. \( Lo = (lo_1, lo_2, \ldots, lo_N) \in \mathbb{R}^N \), which represents the longitude of the node, respectively. Given a historical time window
\[ T = \{ t_1, t_2, \ldots, t_H \} , \text{H is the length of time. All features } X = (x_1, x_2, \ldots, x_H) \in R^{P \times N \times H} , \text{P is the number of features, and } N \text{ is the number of nodes.} \]

**Problem:** Given H and \( X = (x_1, x_2, \ldots, x_H) \in R^{P \times N \times H} \), predict the target spatiotemporal data \( Y = (y_{H+1}, y_{H+2}, \ldots, y_{H+F}) \in R^{F \times N} \) for the next F steps.

The data quality in real scenarios is poor owing to practical environmental or cost constraints, so the missing data are first filled using linear interpolation based on distance weighting, and then all features of the input are classified as direct, indirect, and cross-domain factors, according to whether spread occurs between regions.

**Direct factors:** \( I = (i_1, i_2, \ldots, i_H) \in R^{G \times N \times H} , \text{where } G \text{ is the number of direct factors.} \)

**Indirect factors:** \( O = (o_1, o_2, \ldots, o_H) \in R^{K \times N \times H} , \text{where } K \text{ is the number of indirect factors.} \)

**Cross-domain factors:** \( V = (v_1, v_2, \ldots, v_H) \in R^{D \times N \times H} , \text{where } D \text{ is the number of cross-domain factors.} \)

Multiple features are fused to extract historical data using \( La, Lo \), and \( V \) as input.

Assume that target is \( s_i \), historical time \( t \), and judge whether there is spatial correlation between \( s_j \) and \( s_i \) based on the geographical location relationship between nodes and cross-domain factors.

If the state of \( s_j \) at \( t \) will affect \( s_i \), the value of \( r_{j,i}^{t} \) is 1; otherwise, it is 0.

Calculate the impact of all nodes on the target node at \( t \) by equation (1):

\[
e^{t}_{j,i} = \begin{cases} 
\sqrt{\alpha \cdot \left( \frac{v_j}{2} \right)^2} 
- \sqrt{\left( v_j - v_i \right)^2 + \left( v_j - v_i \right)^2}, & r_{j,i}^{t} = 1 \\
0, & r_{j,i}^{t} = 0 
\end{cases}
\]

In the equation (1), \( \alpha \) is the trainable parameter.

With the cross-domain factor \( V \) as input, the dynamic correlation of nodes is extracted hour by hour using convolutional neural networks:

\[
e_{j,i}^{t} = f \left( w_{i}^{t} \cdot e_{j,i}^{t} \cdot v_{i}^{t} + b \right)
\]

In equation (2), \( v_{i}^{t} \in R^{D \times 1} \) is the cross-domain factor of node \( s_i \), \( w_{i}^{t} \in R^{1 \times D} \) and \( b \in R^{1 \times 1} \) are trainable parameters, \( f \) is the activation function, and \( e_{j,i}^{t} \) is the spanning influence of node \( s_i \) on neighborhood nodes at \( t \). If you assume the target is \( s_i \), the degree of indirect of the neighborhood node \( s_j \) on the target node is \( e_{j,i}^{t} \).

To measure the influence on \( s_i \), the similarity between the target and neighborhood node is calculated based on the node characteristics matrix, and the influence of \( s_j \) on \( s_i \) is calculated separately using the vector and then normalized to ensure that the sum of influence weights is 1 as shown in equation (3):
When you use two training parameters to adaptively deregulate the weight of the influence of your own and cross-domain factors, indirect historical characteristics are obtained and calculated as shown in equation (4):

$$Q_i' = \sigma \left[ (W_1 \cdot O_i') + \left( W_2 \cdot \sum_{j=1}^{N} e^j_{j,i} \cdot O_j' \right) \right]$$  \hspace{1cm} (4)$$

In equation (4), $\sigma$ is the RELU function. $W_1$ and $W_2$ are trainable parameters.

Considering the influence of indirect and direct features on the target, splice the direct influence of $s_i$ at $t$ with the indirect historical features to obtain the data of $s_i$:

$$Z_i' = [Q_i'; I_i']$$  \hspace{1cm} (5)$$

The above calculations are performed each time to obtain historical data features $Z = (z_1, z_2, \cdots, z_H) \in \mathbb{R}^{(G+K) \times N \times H}$. 

**Analysis of Multitemporal Relationships**

In practical applications, spatiotemporal data follow a certain periodicity, but also produce abrupt changes. Periodicity and abruptness are dynamically changing. To extract multiple temporal relationships, the comprehensive impact of different granularities of historical features should be taken into account, and the historical time should be divided into week, day, and hour. The LSTM is used to dynamically capture the features. Accurate predictions are made using a multilayer attention mechanism that takes into account the different effects of long, medium-term, and short-term abrupt changes, calculated as follows.

First, with the historical data characteristics $Z = (z_1, z_2, \cdots, z_H) \in \mathbb{R}^{(G+K) \times N \times H}$ obtained and assuming a target time of $f$, the long-term development pattern of the target data for the week prior to $f$ is first extracted, and the extraction steps are as follows.

Historical features for each day of the week preceding the target time $f$ were input into LSTM, and the corresponding spatiotemporal data development was extracted using the encoder structure:

$$D^d, (h^d, c^d) = LSTM \left( \|_{k=d}^{e_d+24} Z', (h_{i0}, c_{i0}) \right)$$  \hspace{1cm} (6)$$

In equation (6), $D^d$ is the spatiotemporal data evolution pattern for $f$ before day $d$, and $\|$ denotes a tandem operation on the channel dimension.

According to the daily data development in a week, the attention mechanism is used in the decoder structure to quantitatively calculate the influence of daily sequence features on the long term and to accurately extract the long-term development $L'$ by considering the actual daily situation. The calculation process is shown in equation (7):
\[ L^f = \sum_{d=1}^{7} [D^d \cdot \text{softmax}(h^{f-1} \cdot W_d \cdot D^d)] \]  

(7)

In equation (7), \( h^{f-1} \) is the hidden state of the decoder at \( f - 1 \) and \( W_d \) is a trainable parameter.

Second, the medium-term data were extracted as follows. The 24-hour features of the target before \( f \) are input into LSTM to obtain the hidden state as shown in equation (8):

\[ H^t, (h^t, c^t) = \text{LSTM}(Z^t, (h_{t_0}, c_{t_0})) \]  

(8)

\( H^t \) is the feature at \( t \), and \( h^t \) and \( c^t \) are the hidden states and cell states of the LSTM at the last time step.

In the decoder, \( M^f \) is extracted by quantifying the impact of the data features on the medium term using the attention mechanism, which is calculated as shown in equation (9):

\[ M^f = \sum_{t=\Omega}^{f-1} [H^t \cdot \text{softmax}(h^{f-1} \cdot W_t \cdot H^t)] \]  

(9)

\( h^{f-1} \) is the hidden state at \( f - 1 \), and \( W_t \) is the trainable parameter.

Third, short-term mutation information is extracted. In the whole sequence, if target step \( f \) is the first, \( Z^{f-1} \) inputs LSTM to obtain the hidden state as the short-term mutation \( S^f \). If \( f \) is not, the decoder’s prediction of \( Y^{f-1} \) is used as \( S^f \).

Fourth, calculate the weight of long-, medium-, and short-term mutations on future data as shown in equations 10-12:

\[ a_L^f = \text{softmax}(h^{f-1} \cdot W_L \cdot L^f) \]  

(10)

\[ a_M^f = \text{softmax}(h^{f-1} \cdot W_M \cdot M^f) \]  

(11)

\[ a_S^f = \text{softmax}(h^{f-1} \cdot W_S \cdot S^f) \]  

(12)

In the above equations, \( a_L^f \), \( a_M^f \), and \( a_S^f \) are the weights of long-, medium-, and short-term mutations on step \( f \), respectively. \( h^{f-1} \) is the hidden state of the decoder at step \( f - 1 \), and \( W_L \), \( W_M \), and \( W_S \) are parameters.

Finally, the historical data in the three patterns are fused to predict the target values for step \( f \), which is calculated as shown in equations (13) and (14):

\[ C^f = a_L^f \cdot L^f + a_M^f \cdot M^f + a_S^f \cdot S^f \]  

(13)
\[ Y^f = LSTM(C^f) \] (14)

\( L^f, M^f \) and \( S^f \) are the long-term, medium-term and short-term mutations of the spatiotemporal data extracted by the model, respectively. \( C^f \) is the input of the decoder at step \( f \), and \( Y^f \) is the output of the model.

**Spatiotemporal Data Prediction Model Based on a Multilayer Attention Mechanism**

The proposed STDPMA model is shown in Figure 1. The DSR (dynamic spatial relationships) module analyses the dynamic spatial relationships of data using convolutional neural networks to extract the cross-domain effects. The MTR (Multiple Timing relationships) module extracts multiple temporal relationships. The output of the DSR is input to the MTR submodule to extract long-term medium-term and short-term abrupt data by two encoders and a heterogeneous gate. The decoder module completes the spatiotemporal data prediction, uses the attention mechanism to consider the impact of multiple temporal relationships on future spatiotemporal data, and combines historical data of three temporal granularities for prediction.

In the model, convolutional neural networks are used to extract the interactions between features. Data are divided into “week”, “day” and “hour”. The LSTM is used to capture historical features of the time granularity. Multiple encoders are used to capture the different patterns of spatiotemporal data. The attention mechanism is used to accurately predict spatiotemporal data at the ‘hourly’ granularity.

*Figure 1. Architecture of the STDPMA*
EXPERIMENTS AND ANALYSIS

Dataset and Relevant Parameters

The proposed method is validated using commonly used datasets and parameters, where the dataset used for the experiments consists of four mutually independent subdatasets: air quality data, meteorological data, weather forecast data, POI distribution, and road network distribution data.

- **Air quality data**: Air Quality Index for a total of 35 nodes in urban and suburban areas of Beijing, China, from January 1, 2016, to January 31, 2018, provided by the website of the China Environmental Monitoring Center, with a 1-hour interval for monitoring data, including concentrations of PM2.5, PM10, O3, NO2, CO, SO2, and real-time air quality monitoring values.

- **Meteorological data**: Grid-based meteorological data from the Nation Center for Atmospheric Research cover all regions of Beijing from January 1, 2016 to January 31, 2018. The spatial resolution of the grid data is 0.25°, the time interval of the data is 1 hour, and selected attributes of the meteorological data include temperature, humidity, barometric pressure, and wind speed data.

  Meteorological data are grid-based data, and historical air quality data are discrete. To fuse these two kinds of data, the linear distance between each grid and air quality monitoring station first needs to be calculated, and the closest meteorological grid for each air quality monitoring station needs to be found. The meteorological data of the corresponding grid then needs to be considered as the meteorological data of that air quality monitoring station, and finally, the two kinds of data need to be fused together.

- **Weather forecast data**: Weather forecasts from Beijing from January 1, 2016 to January 31, 2018 are provided by the China Meteorological Administration. Historical data are not included in the weather forecast data.

- **POI and road network data**: POI distribution information and road network data were collected by OpenStreetMap for Beijing.

- **Hyperparameters**: The samples were divided into training, validation, and test data at a ratio of 8:1:1, with no overlap. The optimization algorithm is the Adam optimizer, the learning rate is initialized to 0.0003, the batch size is set to 64, and the loss function is the mean square error (MSE).

- **Evaluation metrics**: The output of the model is evaluated according to several criteria, including the mean absolute error (MAE) and root mean square error (RMSE). The MAE and RMSE are calculated as shown in equations (15) and (16):

\[
MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| 
\]  

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} 
\]

In these equations, \( y_i \) is the target value and \( \hat{y}_i \) is the output of the model.
To assess the predictive effectiveness of the models, the constructed models were compared with six baselines: RNN, LSTM, seq2seq (Ilya, S. et al., 2014), ST-ResNet, DA-RNN (Qin, Y., et al., 2017), and GeoMAN. Each model was made to show the best performance by experimenting with different parameters. All experiments were run on a computer with two NVIDIA GTX 1080 Ti GPUs.

**Experimental Results and Analysis**

First, the model was compared with six baselines by predicting more than 24 steps, as shown in Table 1. The STDPMA model showed the best performance in forecasting from 1 to 24 hours and an improvement of 0.52% to 2.27% in MAE and 0.54% to 3.26% in RMSE compared with GeoMAN.

### Table 1. Comparison of Different Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>1-6 h MAE</th>
<th>1-6 h RMSE</th>
<th>1-12 h MAE</th>
<th>1-12 h RMSE</th>
<th>1-18 h MAE</th>
<th>1-18 h RMSE</th>
<th>1-24 h MAE</th>
<th>1-24 h RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNN</td>
<td>37.45</td>
<td>45.23</td>
<td>38.72</td>
<td>46.68</td>
<td>38.82</td>
<td>47.13</td>
<td>38.89</td>
<td>49.68</td>
</tr>
<tr>
<td>LSTM</td>
<td>31.36</td>
<td>40.02</td>
<td>32.39</td>
<td>41.40</td>
<td>33.52</td>
<td>43.78</td>
<td>34.20</td>
<td>48.86</td>
</tr>
<tr>
<td>seq2seq</td>
<td>30.44</td>
<td>38.92</td>
<td>31.84</td>
<td>40.26</td>
<td>31.93</td>
<td>41.42</td>
<td>32.03</td>
<td>45.04</td>
</tr>
<tr>
<td>ST-ResNet</td>
<td>18.87</td>
<td>23.47</td>
<td>23.96</td>
<td>29.27</td>
<td>27.32</td>
<td>37.02</td>
<td>30.56</td>
<td>43.74</td>
</tr>
<tr>
<td>DA-RNN</td>
<td>15.40</td>
<td>21.39</td>
<td>21.15</td>
<td>27.39</td>
<td>25.50</td>
<td>34.89</td>
<td>29.98</td>
<td>41.51</td>
</tr>
<tr>
<td>GeoMAN</td>
<td>14.38</td>
<td>19.42</td>
<td>20.60</td>
<td>26.15</td>
<td>24.54</td>
<td>33.58</td>
<td>28.56</td>
<td>39.67</td>
</tr>
<tr>
<td>STDPMA</td>
<td><strong>13.86</strong></td>
<td><strong>18.88</strong></td>
<td><strong>18.55</strong></td>
<td><strong>25.08</strong></td>
<td><strong>23.25</strong></td>
<td><strong>31.93</strong></td>
<td><strong>26.29</strong></td>
<td><strong>36.41</strong></td>
</tr>
</tbody>
</table>

Among the six models, the RNN and LSTM model cannot be operated in parallel, and the computation time is too long. If the time span is large and the network is deep, the computation will be large, and gradient disappearance inevitably occurs after the sequence exceeds a certain limit. The LSTM model solves the problem of long-distance dependence in prediction to a certain extent by combining short- and long-term memory. Seq2Seq uses the encoder-decoder to unfreeze the length of input and output in RNN and LSTM, making historical information more available. In the Seq2Seq model, the encoder encodes all historical information into fixed-length contextual vectors, but longer historical data are difficult to represent in a single vector.

A two-stage attention recurrent neural network is in DA-RNN, which uses an input attention mechanism to adaptively extract features and take full advantage of historical data. GeoMAN uses a recurrent neural network based on a multistage attention mechanism. It combines multisensor readings and external factors. Methods improve the performance, but lack the dynamic constraint of multisource data and the complexity of multitemporal relationships. The STDPMA model can extract dynamic spatial correlation between spatiotemporal objects, extract historical features by hour, dynamically fuse features, and calculate the correlation between complex factors and the attention mechanism so that the weight of multitemporal relationships can be calculated simultaneously. The model outperforms all baselines.

The spatiotemporal models outperform the traditional models, illustrating the importance of spatial relationships for prediction and proving the effectiveness of the attention mechanism in spatiotemporal prediction. The MAE and RMSE of each model’s prediction results are presented in Figure 2.

To assess different modules in STDPMA on the prediction accuracy, ablation experiments were conducted, and the results are shown in Table 2:
For STDP-a, delete all attention modules and use the splicing historical information.
For STDP-c, features in the channel are spliced without spatial relationships.
For STDP-l, medium- and short-term data are used.
For STDP-m, use long- and short-term abrupt information.
For STDP-s, use long- and medium-term data.

The results for each ablation experiment are presented in Figure 3. As shown, STDP-a is the worst, indicating that the multilayer attention mechanism can significantly improve the prediction accuracy. STDP-c performs second, indicating that fusing multiple features can improve the prediction accuracy.
STDP-I, STDP-m and STDP-s are similar, and STDP-m is the worst, indicating that the spatiotemporal data have a more stable pattern and a greater impact on future data. The experimental results show that fusing modules can improve the accuracy.

To verify the model validity of prediction, STDPMA for short-, medium- and long-term spatiotemporal data prediction was compared. The results are shown in Table 3. The number after each variant indicates the predicted time length of the variant. STDPMA captures the dynamic characteristics of the data in real time, which relieves the error propagation and improves the prediction accuracy.

### Table 3. STDPMA predictions for spatiotemporal data of different lengths

<table>
<thead>
<tr>
<th>Variants</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>STDPMA-24</td>
<td>26.29</td>
<td>36.41</td>
</tr>
<tr>
<td>STDPMA-18</td>
<td>23.25</td>
<td>31.93</td>
</tr>
<tr>
<td>STDPMA-12</td>
<td>18.55</td>
<td>25.08</td>
</tr>
<tr>
<td>STDPMA-6</td>
<td>13.86</td>
<td>18.88</td>
</tr>
</tbody>
</table>
To evaluate the effect of hyperparameters on model training, the performance of the model at learning rates of $1e^{-4}$, $3e^{-4}$, $5e^{-4}$, and $1e^{-3}$ are recorded in Table 4. The experimental results show that the convergence speed of the model decreases when the learning rate is too small, the model tends not to converge when the learning rate is too large, and the model performs best when the initial learning rate is $3e^{-4}$. The performance of the model at batch sizes of 32, 64, 128, and 256 is recorded in Table 5. The experimental results show that a small batch size will make the model training time too long, the model will easily fall into local minima when the batch size is too large, and the model performs best at a batch size of 64.

**Table 4. Performance of STDPMA Trained at Different Learning Rates**

<table>
<thead>
<tr>
<th>Learning rates</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1e^{-4}$</td>
<td>14.57</td>
<td>20.16</td>
</tr>
<tr>
<td>$5e^{-4}$</td>
<td>14.40</td>
<td>19.93</td>
</tr>
<tr>
<td>$1e^{-3}$</td>
<td>14.09</td>
<td>19.31</td>
</tr>
<tr>
<td>$3e^{-4}$</td>
<td>13.86</td>
<td>18.88</td>
</tr>
</tbody>
</table>

**Table 5. Performance of STDPMA Trained at Different Learning Rates**

<table>
<thead>
<tr>
<th>Batch size</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>14.48</td>
<td>20.09</td>
</tr>
<tr>
<td>128</td>
<td>13.97</td>
<td>19.14</td>
</tr>
<tr>
<td>256</td>
<td>14.81</td>
<td>20.74</td>
</tr>
<tr>
<td>64</td>
<td>13.86</td>
<td>18.88</td>
</tr>
</tbody>
</table>

**CONCLUSION**

A spatiotemporal data prediction model based on a multilayer attention mechanism is proposed. This model can be used to predict common spatiotemporal data in intelligent cities, intelligent manufacturing, and other fields, providing a theoretical basis for the construction of smart cities and the development of intelligent manufacturing. First, it analyzes the dynamic spatial relationships of data, fuses multiple features to extract historical features, and then uses the LSTM encoder to extract long- and medium-term patterns as well as short-term mutation information. The attention mechanism is used to dynamically calculate the influence weights of multiple spatiotemporal relationships, and finally the LSTM decoder is used to fully fuse historical features for prediction. Experimental results show that the model outperforms existing models in spatiotemporal data prediction. The main work in the future is to fuse with other features, such as traffic flow conditions, enterprise product processes, or scheduling plans, to improve the model accuracy and generalizability.

**ACKNOWLEDGMENT**

This work was supported by the National Key R&D Program of China under Grant No. 2020YFB1710200.
REFERENCES


Jiang Man, born in 1997, obtained a master's degree in software engineering from Harbin Engineering University in 2022. Her main research interests are data mining, big data analysis, and machine learning.


Zhang Haitao, born in 1972, Ph.D., was hired as an associate professor at the School of Computer Science, Harbin Engineering University in 2005. Her main research interests are data mining, big data analysis and trusted computing.

Liu Hexiang, born in 1991, is currently pursuing a Ph.D. at Harbin Engineering University. His main research interests are data mining and machine learning.